Homework 2

Data Mining Technology for Business and Society

Deadline: **20 May 2020 23:59 (Rome Time Zone)**

Having **TWO** students per group is **RECOMMENDED**.

Having one student per group is allowed only if coordination is not possible.

The total length of the report cannot exceed 4 pages.

It is forbidden to print or store this document, you can only read this document online.

It is forbidden to submit software written with Python-Notebook.

Only ".py" software is considered as a valid solution.

The software must be commented.

Data and software are available at:

http://www.diag.uniroma1.it/~fazzone/Teaching/Data_Mining_Technology_for_Business_and_Society_2019_2020/DMT4BaS_2019_2020.html

The homework is composed of two parts: "Recommendation-System" and "Local Community Detection with PersonalizedPageRank".

Part 1

In this part of the homework, you have to improve the performance of a recommendation-system by using non-trivial algorithms and also by performing the tuning of the hyper-parameters.

Part 1.1

Using the data available in "DMT_2020__HW_2/Part_1/dataset/", you must apply all algorithms for recommendation made available by "Surprise" libraries, according to their default configuration.
For this part of the homework, and also for the next one, it is mandatory to use all CPU-cores available on your computer, by specifying the value in an explicit way with an integer number greater than 1.

Results for 1_1

You have to "copy-paste" in the final report all the "TABLES" in output from the execution of the "cross_validate" command on all algorithms: the number of folds to use is equal to 5.
Moreover, you have to rank all recommendation algorithms you tested according to the MEAN_RMSE metric value: from the best to the worst algorithm.
Finally, you have to explain, by writing exactly one sentence, how you exploited all CPU-cores available on your machine.

Part 1.2

In this part of the homework, you have to improve the quality of both KNNBaseline and SVD algorithms, by performing hyper-parameters tuning always over **five-folds**. Even for this part of the homework, it is mandatory to use all CPU-cores available on your computer, and you have to use, again, the dataset available in "/DMT_2020__HW_2/Part_1/dataset/".
Only configurations with an average RMSE over all five folds **less then 0.89** will be accepted.
In particular, you have to perform a **Random**-Search-Cross-Validation process for tuning the hyper-parameter of the KNNBaseline algorithm. Instead, for tuning the hyper parameter of the SVD algorithm, you have to use a **Grid**-Search-Cross-Validation approach.

Results for 1_2

By using **at most two** pages of the report, you must:
.) put in the report the complete "Grid-of-Parameters" you used to increase the performances for each method.
.) put in the report the best configuration you found for each method.
.) put in the report the two average-RMSE associated to the two best estimators you tuned.
.) put in the report the total time required to select the best estimators.
.) put in the report the number of CPU-cores you used.
.) put in the report, by writing exactly one line, an explanation on how you exploited all CPU-cores available on your machine.

Part 2

In this part of the homework, it is requested to discover the social communities around particular characters of the well-known series of epic fantasy novels called *"A Song of Ice and Fire"*.
Interactions among the characters of the novels are collected inside the four tsv files stored in the directory "DMT_2020__HW_2/Part_2/dataset". In particular, each file corresponds to a novel: "*A Game of Thrones*" (book_1.tsv), "*A Clash of Kings*" (book_2.tsv), and "*A Storm of Swords*" (book_3.tsv), "*A Feast for Crows*" merged with "*A Dance with Dragons*" (book_4.tsv)". Each row in the tsv files represents the fact that the names of the two characters represented in the first and second column appeared within 15 words of one another in the corresponding book.
What is requested by the homework is to discover, for each book of the series, the local communities centered in **only** the following **four** characters: "Daenerys-Targaryen", "Jon-Snow", "Samwell-Tarly" and "Tyrion-Lannister". For discovering these local communities you must create, for each provided book, an unweighted and undirected graph where nodes are characters of the book and where edges represent the interactions reported in the corresponding tsv file.
The technique to use for discovering local communities must be the one explained in the lecture "Lab 3 part 2" of the course, but with the following two changes:

.1.) Instead of using a single fixed value for the PageRank damping factor, you have to try all the following values: [0.95, 0.9, 0.85, 0.8, 0.75, 0.7, 0.65, 0.6, 0.55, 0.5, 0.45, 0.4, 0.35, 0.3, 0.25, 0.2, 0.15, 0.1, 0.05].

.2.) Instead of normalizing the Personalized-PageRank value of each node in the graph by its degree as explained during the lectures, you have to implement the following more general normalization method:

`normalized_score(v) = PPR(v)/(Degree(v)**exponent)` .

Similarly to the previous point, for the "exponent" variable you have to try all the following values: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0].

It is clear now that, for finding a local community with a good conductance value for a given character inside a particular book, you **must** run the modified local community detection method for each of the possible 19*6=114 configurations given by all combinations of the values of the following two parameters: "PageRank damping factor" and "exponent".

WARNING: Communities with a conductance value of 0 or 1 are not considered as valid communities.

It is important to remark that **it is not** requested to find a unique combination of parameters that is good for all inputs, but, what is requested, is to find a good ad-hoc combination of parameters for every single input.

Results for 2_1

By using **at most two** pages of the report, you must represent as a table in the report the content of a tsv file with the following fields/columns:

..) **book_file_name**.
..) **character_name**.
..) **Dumping_factor**_of_the_**best_configuration**.
..) **Exponent**_of_the_**best_configuration**.
..) **Conductance_value**_of_the_local_community.
..) **Number_of_Characters**_inside_the_comunity_belonging_to_the_**Baratheon**_family.
..) **Number_of_Characters**_inside_the_comunity_belonging_to_the_**Lannister**_family.
..) **Number_of_Characters**_inside_the_comunity_belonging_to_the_**Stark**_family.
..) **Number_of_Characters**_inside_the_comunity_belonging_to_the_**Targaryen**_family.
..) **Total_number_of_characters_inside_the_comunity**.

This tsv file will contain **16** records/rows and must be sorted by ascending values of the first column and then by ascending values of the second column.

Where/What To Send

At the end of the process, you have to create a **zip** file with **ONLY** the following data:

1. The software for addressing Part_1: /DMT_2020/HW_2/part_1/sw/ (.py files).
2. The software for addressing Part_2: /DMT_2020/HW_2/part_2/sw/ (.py files).
3. The Part_2 tsv output file : /DMT_2020/HW_2/part_2/output.tsv (.tsv file).
4. The final report in **PDF**: /DMT_2020/HW_2/report.pdf .
5. **PLEASE, DO NOT PUT THE INPUT DATASETS IN THE ZIP FILE.**

The name of the zip file must have this format:

DMT_2020__HW_2__StudentID_StudentName_StudentSurname_StudentID_StudentName_StudentSurname.zip

Finally you must send the ".zip" file to fazzone@diag.uniroma1.it with the following email subject:

DMT_2020__HW_2__StudentID_StudentName_StudentSurname_StudentID_StudentName_StudentSurname.