PROJECT GUIDELINES: MOVIE RECOMMENDATION

THE OBJECTIVES

The main objective of this project is to develop a *reliable* and *efficient* movie recommendation system that infers users' preferences for movies as accurately as possible based on their watch histories and various features of movies. More specifically, (a) **develop an algorithm** that can predict a user's preference for a movie, which can be quantified by number of stars (from 0.5 to 5.0 stars) and recommend the movie, if the predicted preference is *equal to or greater than 4.0 stars* (i.e., "like") using the two data sets provided (`data_rating.csv` and `data_movies.csv`). Then, (b) **evaluate the performance** of your algorithm. Finally, (c) **apply the algorithm** to the 1,000 user-movie pairs (`question_movie.csv`) and **determine which movie to recommend to whom** by filling up the last column with either TRUE or FALSE.

DATA

Two data sets are available (`data_rating.csv` and `data_movies.csv`), which will be *sufficient* to complete the tasks. Developers may collect additional data, which is *NOT* required, necessary, or recommendable. Below is the detailed information about the data sets

USER RATINGS ON MOVIES (`data_rating.csv`)

- 3,364,773 rows, 3 columns, 57.7 MB
- `user_id`: a unique ID assigned to each of 10,249 users (string, e.g., `u00032`)
- `movie_id`: a unique ID assigned to each of 1,144 movies (string, e.g., `m0017`)
- `rating`: the number of stars given to a movie by a user (numeric, e.g., `3.5`)

MOVIE INFORMATION (`data_movie.csv`)

- 1,144 rows, 22 columns, 412 KB
- `movie_id`: a unique ID assigned to each of 1,144 movies (string, e.g., `m0017`)
- `title`: the official title of movie (string, e.g., `The Last of the Mohicans`)
- `year`: the release year (numeric, e.g., `1992`)
- `release`: the release date in yyyy-mm-dd format. (string, e.g., `1992-09-24`)
- `trivia`: the synopsis of movie (string, e.g., `Three trappers protect...`)
- `mpaa`: US Motion Picture Association's film ratings (string, e.g., `R`)[1]
- `run_time`: the duration of movie in minutes (numeric, e.g., `112`)
- `director`: the director of movie (string, e.g., `Michael Mann`)
- `writer`: the script writer of movie (string, e.g., `James Fenimore Cooper`)
- `producer`: the producer of movie (string, e.g., `Hunt Lowry`)
- `composer`: the composer of original soundtrack (string, e.g., `Randy Edelman`)
- `main_actor_1`: the first actor in the closing credits (string, e.g., `Daniel Day-Lewis`)
- `main_actor_2`: the second actor in the closing credits (string, e.g., `Madeleine Stowe`)
- `main_actor_3`: the third actor in the closing credits (string, e.g., `Russell Means`)

[1] For more details, see https://en.wikipedia.org/wiki/Motion_Picture_Association_film_rating_system

- `main_actor_4`: the fourth actor in the closing credits (string, e.g., `Eric Schweig`)
- `budget`: the budget of movie in US Dollars (numeric, e.g., `40000000.0`)
- `domestic`: the box office revenue in the US in US Dollars (numeric, e.g., `75505856.0`)
- `worldwide`: the box office revenue worldwide in US Dollars (numeric, e.g., `75505856.0`)[2]
- `genre_1`: the genre of movie (string, e.g., `Action`)[3]
- `genre_2`: the genre of movie (string, e.g., `Adventure`)
- `genre_3`: the genre of movie (string, e.g., `Drama`)
- `genre_4`: the genre of movie (string, e.g., `Romance`)

SUBMISSION

Each team must submit (a) *Final Report*, (b) *Results*, and (c) *Python Scripts* separately. Failure to submit any of these will cause significant point deduction.

SUBMISSION DEADLINE

- 11:59 PM, 18 November (Fri.) 2022
- No later submission will be accepted

FINAL REPORT

Prepare a final report assuming that the potential readers (e.g., clients) are lay people who have limited knowledge in statistics and data analysis.

- `FORMAT`: PDF, Times New Roman (11 pts.), single-spacing, no more than 10 pages, including everything, page number on every page. DO NOT include a cover page.
- `FILE NAME`: `CS4050_REPORT.pdf`

[2] The estimates are often inaccurate. Students must decide whether and how to incorporate it into their analysis with justification.

[3] Most movies are classified into multiple genres, but the number of genres is varying across movies. Therefore, `genre_2`, `genre_3`, and `genre_4` columns contain considerable numbers of missing values (i.e., `NA`). Students must manage these missing values in a reasonable manner.

CS4050 FINAL PROJECT: MOVIE RECOMMENDATION

The final report must include the following sections. The percentages in parenthesis are suggested proportion of the section.

- TITLE (up to 15 words)
- AUTHORS
- PROBLEM STATEMENT (10%)
  - Restate the objectives and goals of the project in your own words.
  - Briefly describe your algorithm used for the analysis in plain language.
  - Outline the organization of the report.
- DATA (20%)
  - Provide sufficient description of the data sets used in your analysis, so that readers can picture what they look like without looking at them.
  - Basic statistics needs to be provided, including frequency, mean, standard deviation, median, or any others necessary and relevant.
  - Strongly encourage to visualize the data using charts and plots rather plain text or lengthy tables. Descriptive captions and proper numbers should be assigned to all figures and tables.
- METHOD (30%)
  - Describe your methods used in analysis as detailed as possible in a step-by-step manner: DO NOT simply say “Please refer to our Python script” (your clients cannot understand).
  - Your description of analytic procedure must be detailed enough for readers to replicate.
  - You must justify the choices you made for analysis (e.g., choice of features).
  - Use technical jargons correctly with explanations in plain language, where necessary.
- EVALUATION (25%)
  - The performance of your algorithm must be properly evaluated and reported.
  - Use technical jargons correctly with explanations in plain language, where necessary.
  - Various performance measures must be presented and interpreted with brief explanations of their meanings.
  - If you developed more than one algorithm, compare them in terms of their strengths and weakness and choose among them to apply for the prediction task with justification.
- RESULTS (5%)
  - Summarize the results of your prediction of 1,000 users’ preferences.
  - DO NOT paste the 1000-row table.
- DISCUSSION (10%)
  - Discuss the limitations of your algorithm and suggest strategies to further improve it.
  - Discuss possible applications of your algorithm other than content/product recommendations.
- REFERENCES (if any)
- CONTRIBUTIONS OF TEAM MEMBERS (up to 5 lines)
  - Describe the role each member played for this project.

RESULTS

Based on your prediction of users' preferences for movies, complete the third column ("recommend") of question_movie.csv.

- TRUE, if the predicted rating is equal to **4.0 or above**.
- FALSE, otherwise.

Example:

order	user_id	movie_id	recommend
1	u00005	m0328	TRUE
2	u00007	m0103	FALSE
3	u00021	m0708	TRUE
4	u00038	m0994	TRUE
...	...	...	...
998	u10230	m0931	TRUE
999	u10232	m0196	FALSE
1000	u10236	m0112	TRUE

DO NOT change the order of rows in the table or the column names.

- FILE NAME: CS4050_RESULTS.csv

PYTHON SCRIPTS

Submit all the Python scripts used for analysis. The submitted scripts will be executed on the instructor's computer to check whether they produce the same results reported and measure the computation time. Failure to reproduce the reported results, including encountering error messages, will cause significant point deduction.

If you wrote multiple scripts, add *Roman alphabets* to file names in the order that they should be executed.

- FILE NAMES:
  CS4050_SCRIPT_A.py,
  CS4050_SCRIPT_B.py,
  CS4050_SCRIPT_C.py, ...

ASSESSMENT CRITERIA

Criterion	Expectation	Percentage
Accuracy	- The results accurately predict the preferences of 1,000 users in term of F1 scores.	30%
Efficiency	- Important features were selected and combined for prediction in an insightful way, instead of using all the available features. - The computational algorithm produces the results within a reasonable amount of time (i.e., the total amount of time for running all the submitted Python scripts will be measured).	20%
Creativity	- The team develops and executes an algorithm in creative and insightful ways rather than relying on the conventional methods. - The team completes the project independently without much help from the instructor.	15%
Writing	*REPORT* - The logic is sound, rigorous, and easy to follow. - The report is structured neatly and written clearly, concisely, and coherently in plain language. - Figures and tables are informative and intuitive. - No grammatical, punctuation, or spelling errors are found.	20%
	PYTHON SCRIPTS - Objects are named economically. - Indents are properly placed. - Comments are used in a way to improve readability and provides informative descriptions and explanations.	10%
Teamwork	- The roles of members are clearly defined and well-coordinated. - Every member makes a unique contribution to the project.	5%