MLE P40 - Challenge

Introduction

This take-home challenge aims to assess the candidate's proficiency in problem understanding and solving, creativity, critical thinking, coding skills, business acumen, self-driven capabilities and leadership skills. The role being applied to involves addressing marketing challenges in the B2B space, requiring the candidate to develop prototypes, communicate effectively, and lead the productization phase with stakeholder feedback in mind.

Use Case Description

Our client, an online movie rental platform, seeks assistance in increasing their net profit. The platform allows authenticated users to watch, rent-to-watch or buy movies, each with specific storage and viewing constraints (see Business Context for more details on constraints). Additionally, users can engage in virtual communities to share comments and reviews. The client envisions two strategies to enhance profit: building a movie recommendation engine to increase the number of rented movies and a community recommendation engine because they believe communities can influence users to rent more movies.

Tasks & Deliverables

1. Movie Recommendation Engine

Use the data provided to build a movie recommendation engine

Deliverables:

- **Data Analysis Report:**
  - Identify factors influencing user movie rentals.
- **Modeling and Evaluation Report:**
  - Develop and evaluate the recommendation model.
- **Business Evaluation Report:**
  - Validate the profitability of the recommendation engine. See Business Context for some additional information needed here.
- **Additional Recommendations:**
  - Suggest alternative approaches to increase profit.
- **Deployment:**
  - Deployable model as a service with an endpoint for user recommendations. The endpoint should take user input as user_id and return a list of top 5 recommended movies. You are free to use any platform but we recommend using docker containerization.
    - **Bonus:** Provide factors influencing each movie recommendation.
    - **Bonus:** Prototype a Streamlit app for added presentation.

2. Community Recommendation Engine Experiment

Deliverables: No code implementation required for this but your assessment of the client's hypothesis is needed - **Experimental Design Report:** - Design an experiment to test the impact of communities on movie rentals. - **Evaluation Metrics:** - Define metrics to assess the success of the community recommendation engine.

3. Git Repository and Documentation

- **Public Git Repo:**
  - Push all your work to a public git repository and share the link with us.
- **Docker Deployment:**
  - Use Docker or Docker-compose for easy deployment.
- **Readme File:**
  - Write a clear readme for straightforward replication of the setup in a local environment.

Additional Information

Business Context

- Movie Rental Fee: $5
- Movie Purchase Fee: $12
- Monthly Membership: $20
- Cost of Storing Uncompressed Movie: $0.75/day
- Cost of Movie Recommendation: $0.01/recommended movie
- Constraints on movie renting and purchasing: Every rented movie has a rental expiration period of 72 hours. When a rented movie is started, i must be completed in the next 24 hours.These contraints are associated with the client's expenses to store rented movies in a de-compressed format. When a purchased movie is not watched in the last 15 days, it gets compressed in order to reduce storage costs.

Evaluation Criteria

- Creativity, critical thinking, and problem-solving skills.
- Coding proficiency and ability to deploy models as services.
- Business acumen and understanding of profitability factors.
- Effective communication and leadership skills.

We look forward to reviewing your comprehensive solution. Please ensure timely submission and clarity in your documentation. Good luck!

Summary about the data

This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018.

Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

The data are contained in the files `links.csv`, `movies.csv`, `ratings.csv`, `tags.csv` and `tmdb.zip`. More details about the contents and use of all these files follows.

This dataset has been modified with no violation of usage license agreement.

Content and Use of Files

Formatting and Encoding

The dataset files are written as comma-separated values files with a single header row. Columns that contain commas (`,`) are escaped using double-quotes (`"`). These files are encoded as UTF-8. If accented characters in movie titles or tag values (e.g. Misérables, Les (1995)) display incorrectly, make sure that any program reading the data, such as a text editor, terminal, or script, is configured for UTF-8.

User Ids

MovieLens users were selected at random for inclusion. Their ids have been anonymized. User ids are consistent between `ratings.csv` and `tags.csv` (i.e., the same id refers to the same user across the two files).

Movie Ids

Only movies with at least one rating or tag are included in the dataset. These movie ids are consistent with those used on the MovieLens web site (e.g., id `1` corresponds to the URL https://movielens.org/movies/1). Movie ids are consistent between `ratings.csv`, `tags.csv`, `movies.csv`, and `links.csv` (i.e., the same id refers to the same movie across these four data files).

The Movie DB Ids

These ids are found in `links.csv` under column `tmdbId`. These Ids are consistent with the Ids used on themoviedb.org API. (e.g., id `1` corresponds to https://api.themoviedb.org/3/movie/1).

Ratings Data File Structure (ratings.csv)

All ratings are contained in the file `ratings.csv`. Each line of this file after the header row represents one rating of one movie by one user, and has the following format:

```
userId,movieId,rating,timestamp
```

The lines within this file are ordered first by userId, then, within user, by movieId.

Ratings are made on a 5-star scale, with half-star increments (0.5 stars - 5.0 stars).

Timestamps represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

Tags Data File Structure (tags.csv)

All tags are contained in the file `tags.csv`. Each line of this file after the header row represents one tag applied to one movie by one user, and has the following format:

```
userId,movieId,tag,timestamp
```

The lines within this file are ordered first by userId, then, within user, by movieId.

Tags are user-generated metadata about movies. Each tag is typically a single word or short phrase. The meaning, value, and purpose of a particular tag is determined by each user.

Timestamps represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

Movies Data File Structure (movies.csv)

Movie information is contained in the file `movies.csv`. Each line of this file after the header row represents one movie, and has the following format:

```
movieId,title,genres
```

Movie titles are entered manually or imported from https://www.themoviedb.org/, and include the year of release in parentheses. Errors and inconsistencies may exist in these titles.

Genres are a pipe-separated list, and are selected from the following:

- Action
- Adventure
- Animation
- Children's

- Comedy
- Crime
- Documentary
- Drama
- Fantasy
- Film-Noir
- Horror
- Musical
- Mystery
- Romance
- Sci-Fi
- Thriller
- War
- Western
- (no genres listed)

Links Data File Structure (links.csv)

Identifiers that can be used to link to other sources of movie data are contained in the file `links.csv`. Each line of this file after the header row represents one movie, and has the following format:

`movieId,imdbId,tmdbId`

movieId is an identifier for movies used by https://movielens.org. E.g., the movie Toy Story has the link https://movielens.org/movies/1.

imdbId is an identifier for movies used by http://www.imdb.com. E.g., the movie Toy Story has the link http://www.imdb.com/title/tt0114709/.

tmdbId is an identifier for movies used by https://www.themoviedb.org. E.g., the movie Toy Story has the link https://www.themoviedb.org/movie/862.

Use of the resources listed above is subject to the terms of each provider.

To register to themoviedb go to https://www.themoviedb.org/signup

Movie metadata files from themoviedb.org (tmdb/*)

This contains metadata json files for movies in movie.csv. Each file is named after the tmdbId as .json Each json file has the following fields:

`overview,popularity,original_title,runtime,release_date,vote_average,vote_count,status,tagli`

Usage License

Neither the University of Minnesota nor any of the researchers involved can guarantee the correctness of the data, its suitability for any particular purpose,

or the validity of results based on the use of the data set. The data set may be used for any research purposes under the following conditions:

- The user may not state or imply any endorsement from the University of Minnesota or the GroupLens Research Group.
- The user must acknowledge the use of the data set in publications resulting from the use of the data set (see below for citation information).
- The user may redistribute the data set, including transformations, so long as it is distributed under these same license conditions.
- The user may not use this information for any commercial or revenue-bearing purposes without first obtaining permission from a faculty member of the GroupLens Research Project at the University of Minnesota.
- The executable software scripts are provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of them is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event shall the University of Minnesota, its affiliates or employees be liable to you for any damages arising out of the use or inability to use these programs (including but not limited to loss of data or data being rendered inaccurate).

If you have any further questions or comments, please email grouplens-info@umn.edu

Citation

To acknowledge use of the dataset in publications, please cite the following paper:

> F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872

Further Information About GroupLens

GroupLens is a research group in the Department of Computer Science and Engineering at the University of Minnesota. Since its inception in 1992, GroupLens's research projects have explored a variety of fields including:

- recommender systems
- online communities
- mobile and ubiquitious technologies
- digital libraries
- local geographic information systems

GroupLens Research operates a movie recommender based on collaborative filtering, MovieLens, which is the source of these data. We encourage you to visit http://movielens.org to try it out! If you have exciting ideas for experimental work to conduct on MovieLens, send us an email at grouplens-info@cs.umn.edu - we are always interested in working with external collaborators.