Book Recommendation system

SUML 11C, GROUP 4

Dominic Dąbrowski, Polina Lysenko, Oleksandr Kurchak, Andrii Mostovyi

S23328, S22821, S22831, S22435

Contents

[Introduction 2](#_Toc157451077)

[Project goal 2](#_Toc157451078)

[Selected technology 2](#_Toc157451079)

[Kedro 2](#_Toc157451080)

[Streamlit 2](#_Toc157451081)

[Pandas / NumPy / SciPy 2](#_Toc157451082)

[Method 3](#_Toc157451083)

[ML model parameters 3](#_Toc157451084)

[NUMBER\_OF\_FACTORS\_MF 3](#_Toc157451085)

[EVAL\_RANDOM\_SAMPLE\_NON\_INTERACTED\_ITEMS 3](#_Toc157451086)

[m 3](#_Toc157451087)

[C 3](#_Toc157451088)

[Description of functionality 3](#_Toc157451089)

[Pipeline 4](#_Toc157451090)

[Additional information 4](#_Toc157451091)

# Introduction

Project goal  
The primary objective of our project is to develop a complex predictive model capable of recommending books with a high degree of accuracy to a given user. The primary objectives of this project are to provide personalized book recommendations and enable improved reading experiences, discerning the reading preferences of a user to correctly and predominantly target the right books , handling vast datasets of published books to find interesting insights into a user’s reading habits, providing an alternative to traditional book discovery methods, contributing to literacy initiative by promoting diverse and fulfilling reading choices, and driving ongoing innovation in the world of literature leading to more refined user engagement.

## Selected technology

### Kedro

For our overall project structure, we have decided to use Kedro which is an open source Python toolbox to create production ready data science projects which follow best practices for reproducible, maintainable, and modular pipelines. We used it to implement the data catalog function (to create data connectors for saving and loading data, tracking parameters and formatting), templates (to easily navigate and create code that was clean to follow) and pipeline abstraction (which creates model and process workflows using Kedro nodes for high quality and readable code).

### Streamlit

For our application front end, we have decided to use Streamlit which is commonly used for fast ML app development. It turns data scripts into sharable web apps using an open-source Python framework. It is very simple to use and does not require much knowledge to learn the basic functionalities making it fast and lightweight for our project. We used it to implement the main functionalities of our application, Browse Books (listing the available books in dataset and rating them) and Recommended Books (creating personalized recommendations for a user).

### Pandas / NumPy / SciPy

To process the data from our dataset, we used Pandas, which is a library commonly used for working with datasets and allows analysis, cleaning, exploration, and manipulation of data. We used it to work with .csv and similar files, converting them into data frames for use in our ML system, performing analysis on the shape and description of columns, as well as transforming data frames (changing the shape, and dropping unnecessary columns) among other tasks.  
  
NumPy is a general purpose array processing library which provides high performance multidimensional array objects and tools for working for them. In addition, there is the SciPy library which is a scientific computation framework built on NumPy that offer additional utility functions related to optimization, statistics, and signal processing. We used them to make models from the data frames created in pandas, and as part of this process, csr\_matrix (compressed sparse row) and SVDs (singular value decompositions) were used from the linalg module.

# Method

## ML model parameters

In our project, we used two main methodologies through a recommendation system subset of information filtering which seeks to predict the rating or preference a user would give to a book. They are popularity-based (most popular among all users) and collaborative filtering (finding similar users basing on rating histories and suggesting similar items that this group would like).

The parameters used are:

NUMBER\_OF\_FACTORS\_MF: As the system used SVD for the matrices, this specifies the number of factors to factor the user-item matrix. Modifying this value can either capture more complex patterns or less sophisticated ones, which can lead to underfitting or overfitting of the given data.

EVAL\_RANDOM\_SAMPLE\_NON\_INTERACTED\_ITEMS: This parameter determines the number of non-interacted items to consider when evaluating the overall model. Modifying this value would either make the evaluation more robust and consider more negative cases (items not interacted with), or make the evaluation less robust and consider less negative cases, therefore the completion time would be impacted.

m: This parameter is used for the calculation of the weighted rating represented by the minimum votes required to be listed in the top 5% (or 95th quantile of the ratings count). Changing this value would make the ratings chart either more or less exclusive with the total number of votes for a particular book.

C: This parameter is the mean vote across the whole data frame and determines the scale of the ratings used for books. Changing this value would either shift the book ratings up or down by a constant, therefore it is important to set properly based on how the ratings should be represented and assessed.

## Description of functionality

The overall project was created in such a way to utilize a fully comprehensive workflow which would aid in the accurate classification of book recommendations for users based on the rating assigned to them. The project begins with the collection and preparation of our main dataset, using data preprocessing and analysis where necessary to optimize the data and better understand data relationships for eventual training. Next, the development of the recommendation system occurred, where our methodologies, Popularity-based and Collaborative Filtering, are chosen and executed. The dataset is divided into appropriate sets for model evaluation with training and testing, then the model evaluation can occur which observes overall performance of the recommendation system using metrics such as recall. The final stage is model deployment, where predictions on new data are made which generates a list of recommended books. This is visible not only in the back end but through our front-end website which makes the process more friendlier and suitable for use by a public facing user.

## Pipeline

A diagram of data flow

Description automatically generated

# Additional information

The link to our Trello team can be found [here](https://trello.com/b/621r7WRs/suml114) (https://trello.com/b/621r7WRs/suml114).

The link to our GitHub repository can be found [here](https://github.com/SashaKur/SUML-Group-Project) (https://github.com/SashaKur/SUML-Group-Project).