Names: Amanda Wang, Jennifer Jasperse, Arely Alcantara
Team Name: Push Mode People
Team Captain: Amanda Wang
Emails: afw3@illinois.edu, jbj3@illinois.edu, arely2@illinois.edu
NetIDs: afw3, jbj3, arely2

Project Proposal

What is your free topic? Please give a detailed description. What is the task? Why is it important or interesting? What is your planned approach? What tools, systems or datasets are involved? What is the expected outcome? How are you going to evaluate your work?

For our final project, we have selected the ‘free topic’ theme and we’re planning on doing a book recommendation system! Netflix, Spotify, and Tiktok exist for movie/music/social media lovers but what about a book recommender for book lovers? Can you imagine getting a personalized list of books you may like? And even having the ability to search a topic and get the perfect book to satisfy your needs? Well, we’ve got you covered! We plan on using a Goodreads book dataset to help recommend books to different users on the site. The data itself will be coming from this website, so we won’t be scraping the website for information:
https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home
We found this topic interesting because there is a lot of information about books contained in reviews. It would be great to be able to harness that knowledge and use it to build a recommender system for users.

We plan to do two main tasks in this project. First, we would like to be able to recommend books based on user input. For example, a user might input that they’re interested in reading “books with complex magic systems,” and then we would output a list of books that they might be interested in. We plan on using the content we learned relating to “web search queries” for this part of the project.

Our second main task is to build two recommender systems: one using content-based filtering, and one using collaborative filtering. Ideally, given that we have extra time, we would love to implement a hybrid model - one that combines both techniques. The books dataset currently has a property called ‘similar_books’, but we are unsure of where/how these recommendations were determined so we will ignore this portion of the data and build our own content-based approach as well as collaborative filtering approach. We could use book reviews to determine a list of topics each book covers (using the Probabilistic Topic Models), and then use this information along with book metadata and other users’ ratings to recommend books to users. We are not yet sure how we will combine the two recommendation techniques, but we will build both systems separately first, and then read up on different ways to combine them (if time allows).

The expected outcome would be one “book search engine” that can recommend books based on a simple query, and a recommender system that will recommend books to different users. We will evaluate the first task by manually labeling the output of our data, similar to MP 2.3. For evaluating our recommender systems, we will temporally segment each user's data into training and testing sets. We will evaluate our recommender systems based on how many of the books in the testing set we were able to accurately recommend based on the user's reading history in the training set.

Which programming language do you plan to use?
Python

Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

- Cleaning data, specifically removing reviews that aren't in English (2 hrs)
- Research how to implement a web search query in Python (5 hrs)
- Implement Task 1 (recommending books based on user inputted phrase) (15 hrs)
- Test, debug, and revise code for Task 1 (5 hrs)
- Evaluate Task 1 (5 hrs)
- Research how to build content-based and collaborative filtering recommender systems (10 hrs)
- Develop solution to cold-start problem for collaborative filtering (5 hrs)
- Implement Task 2 - build 2 recommender systems:
    - One content-based (10 hrs)
    - One collaborative filtering (10 hrs)
    - *Optional: one hybrid model (10 hrs)*
- Test and evaluate code for Task 2 (recommender systems) (10 hrs)
- Develop demo and final documentation (15 hrs)