Northwestern

Book Recommendation System

Michel LeRoy

AVC Mid-Project Review

Highlights

Top accomplishments during this sprint:

- Data Wrangling
  - Downloaded the data, cleaned it, and have 15 separate dataframes, one for each genre, for my book recommendation models
- AWS setup
  - Set up S3 bucket, RDS instance, and EC2 instance

Review Progress

Completed Stories:

Epic 1: Download Data and Create Full dataset for model

- (1 point) *** Story 1: Understand how the different tables merge together, and decide what data is necessary for this project
- (2 points) *** Story 2: Perform EDA, making sure the data is clean, makes sense, and there are no outliers, etc
- (2 points) *** Story 3: Create Final clean dataset that will be used to train the model

Epic 2: Create Recommendation System Model

- (4 points) ** Story 1: Create a User-Based Collaborative Filtering Model (this story is half completed)

The other stories I have completed were not originally stories when the backlog was created; I only had rough epics in the IceBox. I have listed these epics below, along with the stories I have completed.

Epic 2: Create the Flask/html/css code to run the user app

- Story 1: Create an EC2 instance that the Flask app will run on

Epic 3: Store the final model/ratings matrix in S3

- Story 1: Upload raw Kaggle data into S3 bucket

Epic 4: Store the book images in the RDS, and call images of the user's recommendations from here

- Story 1: Create the database schema in RDS for prediction results generated offline ahead of time

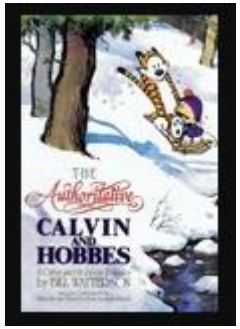
Analysis

- The first task accomplished this sprint was data wrangling.

To the right is an example of my data cleaning; removing books with negative tags, which obviously do not belong

	book_id	title	tag_id	count
922053	1935	Kindle Paperwhite User's Guide	17246	-1
922054	1935	Kindle Paperwhite User's Guide	6552	-1
922055	1935	Kindle Paperwhite User's Guide	2272	-1
959611	7803	Kindle User's Guide	9221	-1
922052	1935	Kindle Paperwhite User's Guide	21619	-1
922051	1935	Kindle Paperwhite User's Guide	10197	-1

- Next I used the book tags to create the most popular genres and categorize each book. After cleaning, I have 16 data frames of ratings, one for each genre.
- Finally, I have run a preliminary SVD model on the “humor” genre of the book ratings data



This model has a **93% precision rate** which is above the 80% metric outlined in my Project Charter Success Criteria.

One of the top books recommended for the humor category is “Calvin and Hobbes”

Lessons Learned

- This sprint was much more focused on the coding and software aspects of the project rather than the model itself, which surprised me
- I learned how to set up an RDS database, an EC2 instance, as well as an S3 bucket, and the various ways these products interact with one another
- I also learned a great deal about configurability and how to create code that another person can run, with packages such as argparse, and config files.

Recommendations

The next sprint should focus primarily on the actual model. The model needs to be finalized, and then the notebooks need to be turned into scripts that have modularized code.

Epic 1: Download Data and Create Full dataset for model

- (1 points) * Story 4: Document the code, paying special attention to any decision points

Epic 2: Create Recommendation System Model(4 points) ** Story 1: Create a User-Based Collaborative Filtering Model

- (4 points) ** Story 2: Create an Item Based Collaborative Filtering Model
- (2 points) ** Story 3: Test how long each model takes to give users new recommendations - Decide which is better in terms of precision and run time
- (2 points) * Story 4: Document the code, keeping the final model, and pay special attention to any decision points
- (4 points) * Story 5: Create Unit tests
- (1 point) * Story 6: Discuss with QA

The secondary priority for this sprint is to get the flask app up and running

Epic 2: Create the Flask/html/css code to run the user app