Quan Ta, Ayesha Mubashir

DSCI 351

August 29th, 2021

Project Report

1. Select a dataset for the project, provide a description of the dataset.
    - Book: https://www.kaggle.com/zygmunt/goodbooks-10k
    - This dataset consists of the metadata of 10,000 books and over 53,000 users' ratings. The metadata of each book includes its id, authors, ISBN, title, the language used in it, and the number of ratings.
2. What is the type of recommender system you would like to build?
    - We would want to build a content-based recommender system. Based on the author, the tags, and the rating of one book, we can recommend the users with books that they may like.
3. What is/are the method(s)/model(s) utilized in your system?

Models:

- We are using the vector space model and cosine distance between books' vectors to find similar books. In this case, the vectors we are using are author_vector and tags_vector.

Methods:

- Preprocessing the Data:
    - Merge the book_tags and tags dataset to generate the final book_tags dataset. This dataset contains book_id, tag_id, and count
    - Drop unnecessary columns and kept only: book_id, original_publication_year, original_title, language_code, average_rating, authors, book_count
    - Drop rows with null values
        - By doing this, we reduce the number of rows from 10,000 to 8,405
    - Generate author_vector and tag_vector columns based on the book_tags data frame and column authors in the books data frame
- Content-based Recommendation System:
    - get_rated_books():
        - Input: user_id
        - Output: the list of book

    - In this function, we are rating the book if they are greater than 4 by using user_id as an input.
- cal_distance():
    - Input: 2 book_id and data frame
    - Output: The distance
    - In this function, we are calculating the cosine distance between book1 to book2 to get the target one from the dataset based on author_vector and tags_vector.
- top_k_neighbor() :
    - Input: book_id, list of user_id's favorite book, data frame
    - Output: list of similar book_id and distance
    - We are using the top_k_neighbor function to find the nearest neighbor of the book_id we provide. K is the number of neighbors that we need to find.
- recommend_book():
    - Input: user_id, number of books the user want to have
    - Output: The name of those books that we recommend to users
    - This function takes the user_id and an integer n as inputs. The output is n books that are the most similar to those the user likes.

4. How to interpret the recommended results?

After running this function we can see the 8 books that are similar to those that are rated over 4 by the user. In this case, we test to recommend 8 books for user_id 9.

```
[29]: #test for user_id 9 that wants to have 8 new books to read
recommend_book(9,8)
```

```
And the Shofar Blew
An Echo in the Darkness (Mark of the Lion, #2)
As Sure as the Dawn (Mark of the Lion, #3)
The Atonement Child
Her Mother's Hope (Marta's Legacy, #1)
Mad About Madeline
Madeline's Rescue
The Scarlet Thread
```

The recommender system manages to recommend 8 books for the users and prints out the names of those books.

5. How to evaluate your system's performance?

- We evaluated the system based on
    - Accuracy: We use Root Mean Square Error (RMSE). The accuracy of 0.8466729436823051 is based on all users' ratings. From this accuracy, we can say this dataset is the best fit for our recommended systems.
    - Coverage: We did not cover books that are rated lower than 4, as well as books that contain null values.
    - Novelty: We are likely to recommend books that the user may not aware of their existence. It is due to the tags are taken into account of the recommender system.

6. What is the limitation and future improvement of your system?

- The running time of the system is long. It is due to the number of books we take from the users to recommend from is high.
    - To improve on this, we may narrow down the number of books we take from the user down to only five or three. We can do this by changing the score threshold to be 4.5. Other than this, we can change the fixed number of books we take from the user to be five with the highest ratings. By doing this, we can reduce the training time.
- Some tags and authors' names are not in English. It reduces the number of books we can recommend to users, as we cannot access those books' tags that are not in English.
    - To improve on this, we can apply a text classification to deal with characters that are not English. As we can read these words, we can improve our system by a huge margin. Users can have access to books that are similar in content but are not titled in English.
- We did not take many features of books into consideration. Therefore, our recommendation results may not be the best.
    - To improve on this, we would want to take the published years into account. As books from different generations tend to be different, users tend to prefer books at certain times. Therefore, the recommender system can take years into account to provide better books for users.