Books Recommender System

System recommends books using the K-Nearest Neighbours and Cosine Similarity algorithms from a list of 10000 books with 1 million ratings

Requirements

Python version: 3.10.2
Python modules: streamlit, numpy, Pillow, pandas, urllib3, scikit-learn

Dataset

- goodbooks-10k

Dataset contains ratings (go from 1 to 5) for 10000 books. Unique number of every book

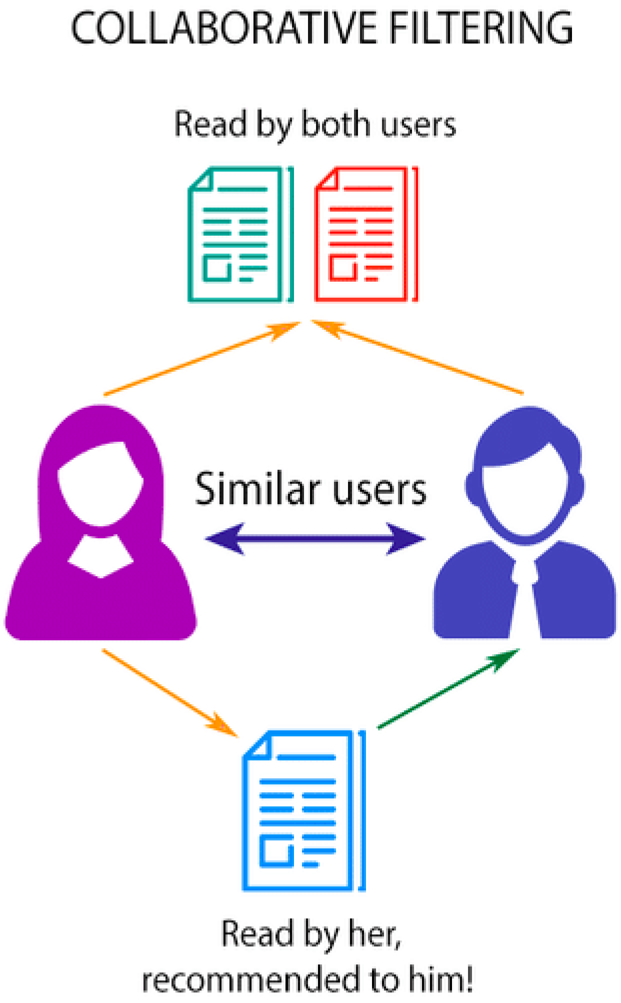
The most important file is books.csv which has metadata for each book (goodreads IDs, authors, title, average rating, etc.).

I used book IDs and user IDs in recommending process. For books, they are 1-10000, for users, 1-53424. Every user has made at least two ratings.

Motivation

Recommenders

Collaborative Filtering



Our content based engine suffers from some severe limitations. It is only capable of suggesting books which are *close* to a certain book. That is, it is not capable of capturing tastes and providing recommendations across genres.

Also, the engine that we built is not really personal in that it doesn't capture the personal tastes and biases of a user. Anyone querying our engine for recommendations based on a book will receive the same recommendations for that book, regardless of who s/he is.

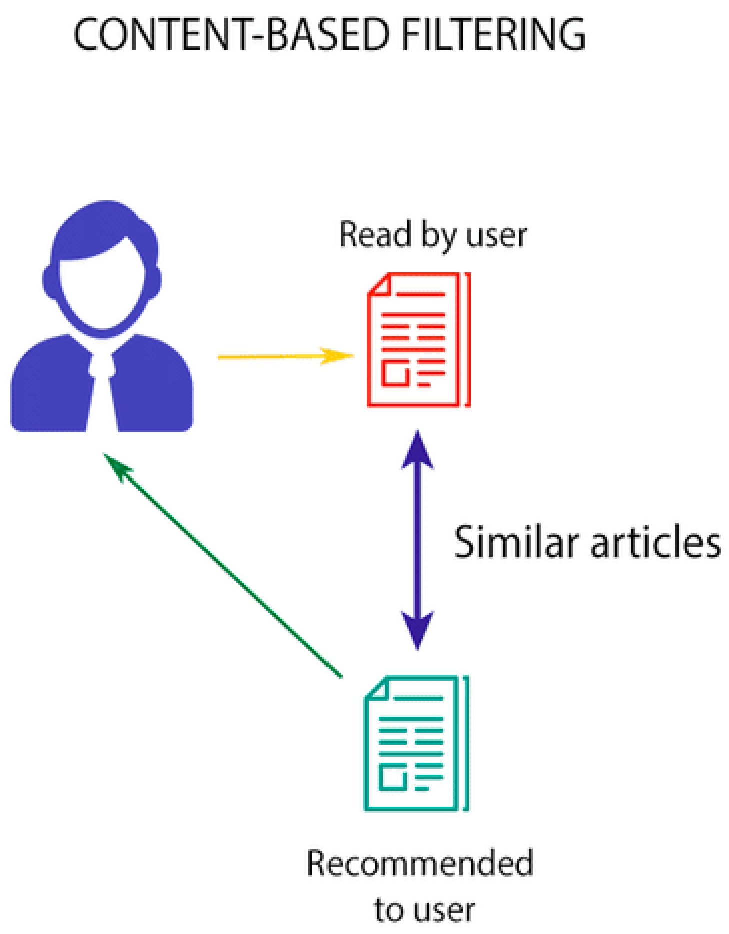
Therefore, in this section, we will use a technique called **Collaborative Filtering** to make recommendations to Book Readers. Collaborative Filtering is based on the idea that users similar to a me can be used to predict how much I will like a particular product or service those users have used/experienced but I have not.

I will not be implementing Collaborative Filtering from scratch. Instead, I will use the **Surprise** library that used extremely powerful algorithms like **Singular Value Decomposition (SVD)** to minimise RMSE (Root Mean Square Error) and give great recommendations.

KNN

A recommender system k-Nearest Neighbor (k-NN) based is a one of collaborative filtering systems that uses the ratings (from users) to recommend. k-NN calculate the similarity between each pair of items, and then system (using these values) try to estmiate how this specific user will rate a given item.

Content Based Recommender



The recommender we built in the previous section suffers some severe limitations. For one, it gives the same recommendation to everyone, regardless of the user's personal taste. If a person who loves business books (and hates fiction) were to look at our Top 15 Chart, s/he wouldn't probably like most of the books. If s/he were to go one step further and look at our charts by genre, s/he wouldn't still be getting the best recommendations.

For instance, consider a person who loves *The Fault in Our Stars*, *Twilight*. One inference we can obtain is that the person loves the romaintic books. Even if s/he were to access the romance chart, s/he wouldn't find these as the top recommendations.

To personalise our recommendations more, I am going to build an engine that computes similarity between movies based on certain metrics and suggests books that are most similar to a particular book that a user liked. Since we will be using book metadata (or content) to build this engine, this also known as **Content Based Filtering.**

I will build this recommender based on book's *Title*, *Authors* and *Genres*.

Cosine Similarity

I will be using the Cosine Similarity to calculate a numeric quantity that denotes the similarity between two books. Mathematically, it is defined as follows:

$cosine(x,y) = \frac{x. y^\intercal}{||x||.||y||} $

Files

- `app.py` the main file of streamlit web-app, have to be runned in by command 'streamlit run'
- `knn.py` python file containing a k-NN Algorithm
- `cossim.py` python file containing a Cosine Similarity Algorithm
- `Analysis of dataset.ipynb` draft jupyter notebook with some helpful charts, commands, not included in app.
- `requirements.txt` text file with names of needed python modules

Running application

- Clone repo
- Open cmd prompt in working directory
- Run command:

  ```
  pip install -r requirements.txt
  ```

- To run app, write following command in cmd prompt

  ```
  streamlit run app.py
  ```

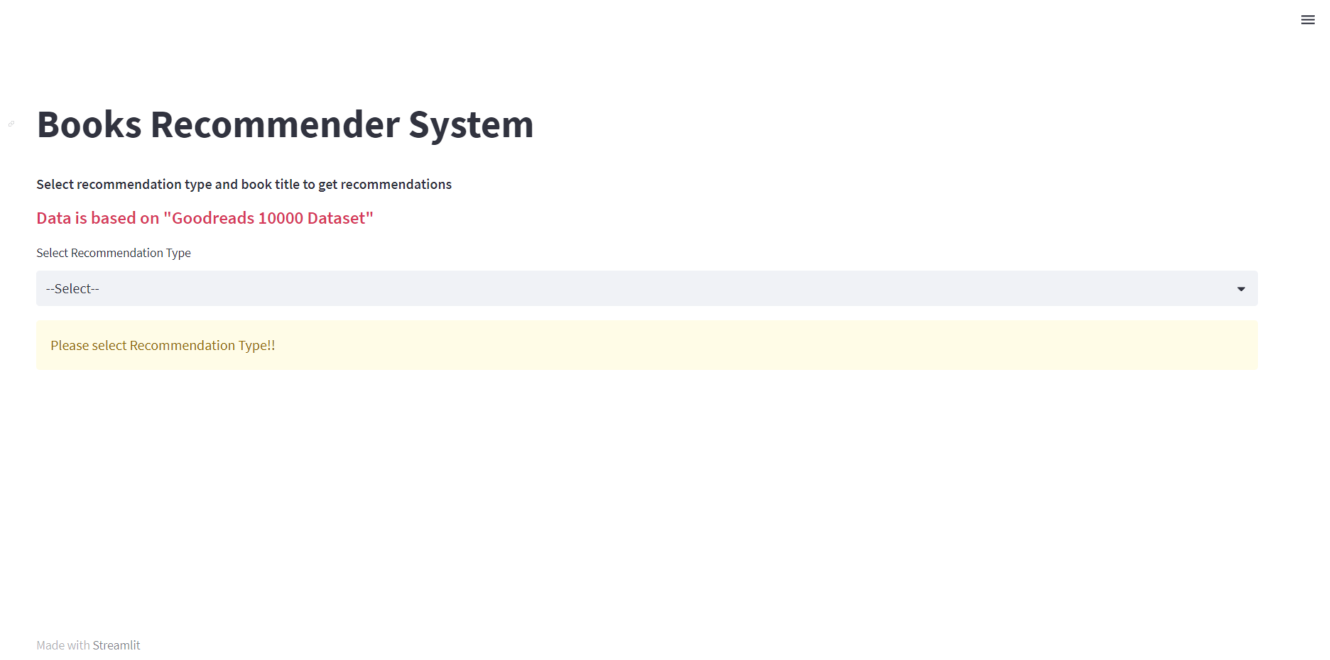
- Then wait a moment, after few seconds you shoulds following code:

```
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.1:8501
```

And web-app should open in your browser. If not just copy and paste link `Local URL` or `Network URL`

After page is loaded you shold see following view:



Books Recommender System

Select recommendation type and book title to get recommendations

Data is based on "Goodreads 10000 Dataset"

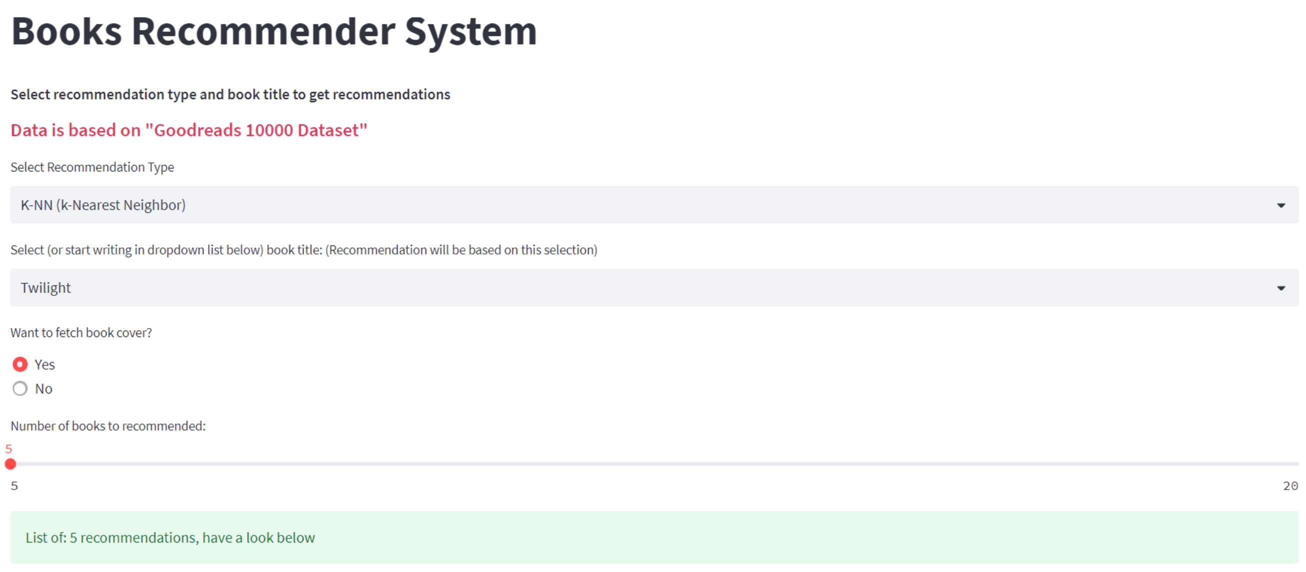
Select Recommendation Type

--Select--

Please select Recommendation Type!!

Made with Streamlit

Input page for KNN (same for Cosine Similarity)



Books Recommender System

Select recommendation type and book title to get recommendations

Data is based on "Goodreads 10000 Dataset"

Select Recommendation Type

K-NN (k-Nearest Neighbor)

Select (or start writing in dropdown list below) book title: (Recommendation will be based on this selection)

Twilight

Want to fetch book cover?

● Yes

○ No

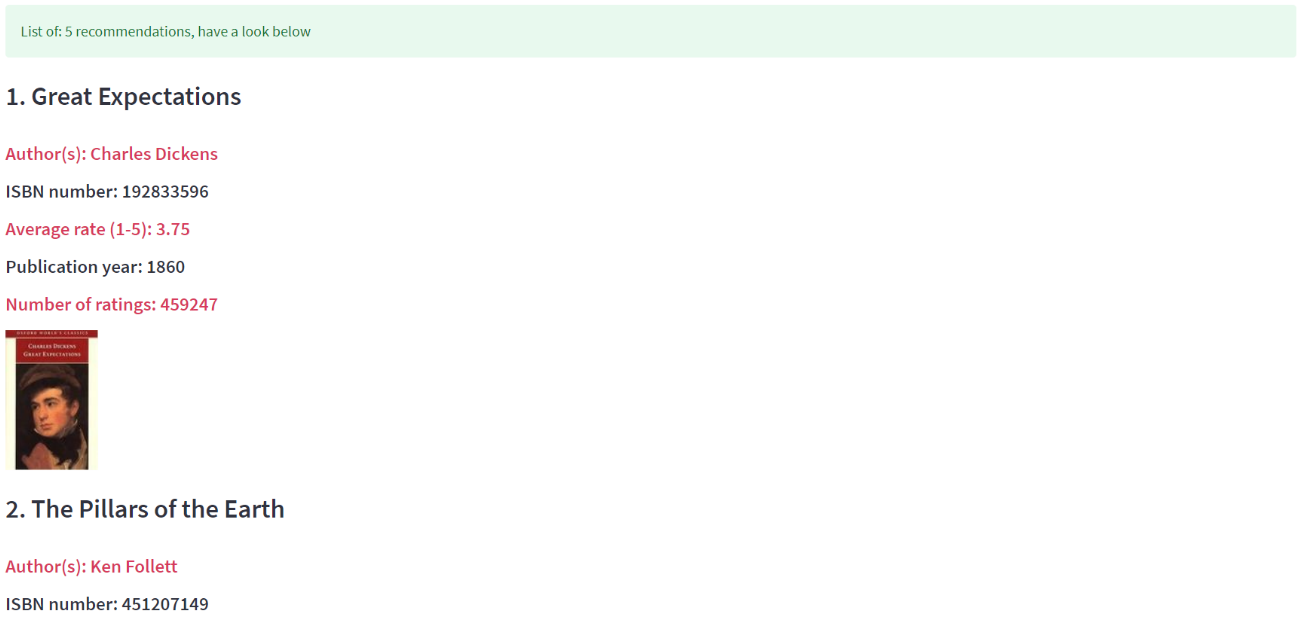
Number of books to recommended:

5

5 20

List of: 5 recommendations, have a look below

Example results of recommendation:



List of: 5 recommendations, have a look below

1. Great Expectations

Author(s): Charles Dickens

ISBN number: 192833596

Average rate (1-5): 3.75

Publication year: 1860

Number of ratings: 459247

2. The Pillars of the Earth

Author(s): Ken Follett

ISBN number: 451207149

Screenshots

KNN

Cosine similarity

Conclusion