Applied Data Science(Stat GU4243/GR5243) - Project 5

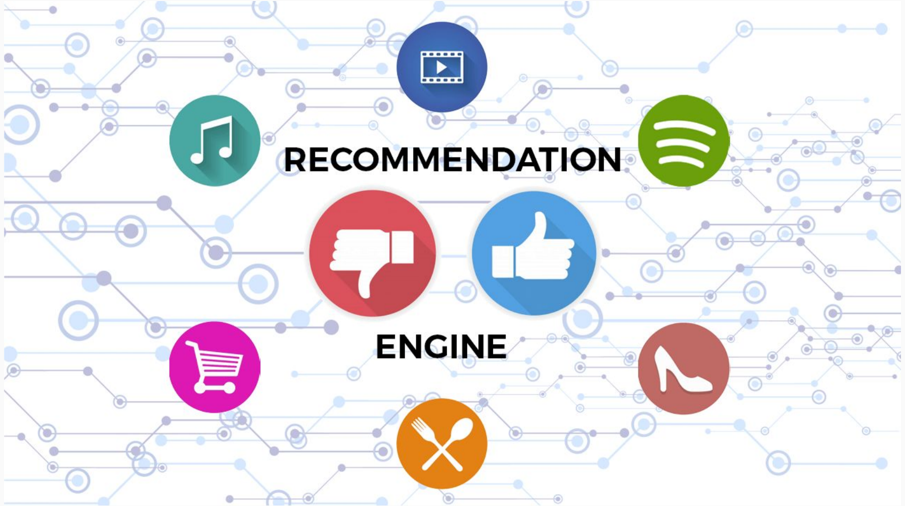
Book Recommendation System

Shefali Shrivastava
Ritika Nandi
Mansi Singh

About the project

Identify books that users are most likely to read based on their past behaviour *or* other similar books

Shiny app to dynamically generate recommendations



Source: https://www.activestate.com/blog/how-to-build-a-recommendation-engine-in-python/

Data Cleaning and Preprocessing

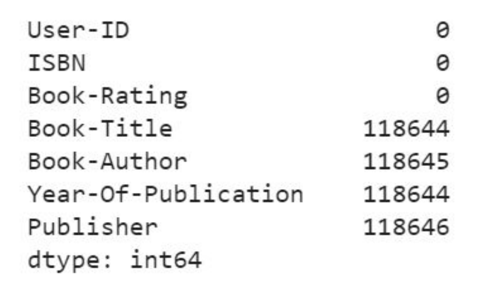
Dataset - three separate CSV files with books, users and rating information

The dataset was relatively clean; minimal errors in the values.

We first had to **merge** the data files to create a single dataframe

Next, we checked for **missing values**:

- Approx. 10% missing in last four columns
- Imputation not possible (unique values)

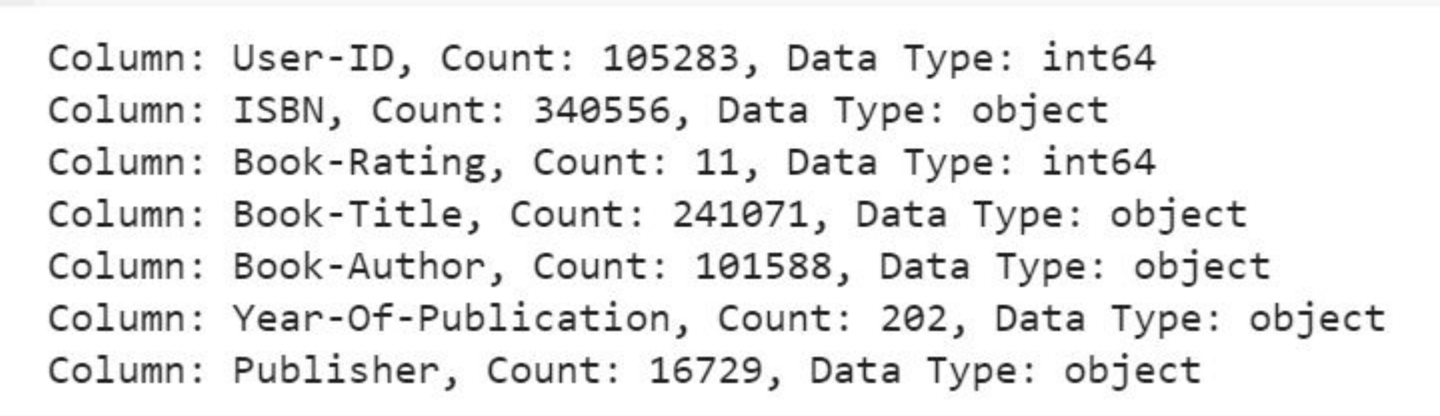


```
User-ID                   0
ISBN                      0
Book-Rating               0
Book-Title           118644
Book-Author          118645
Year-Of-Publication  118644
Publisher            118646
dtype: int64
```

Additionally, we removed duplicate values for content based filtering (further data cleaning required for similar titles)

Exploratory Data Analysis

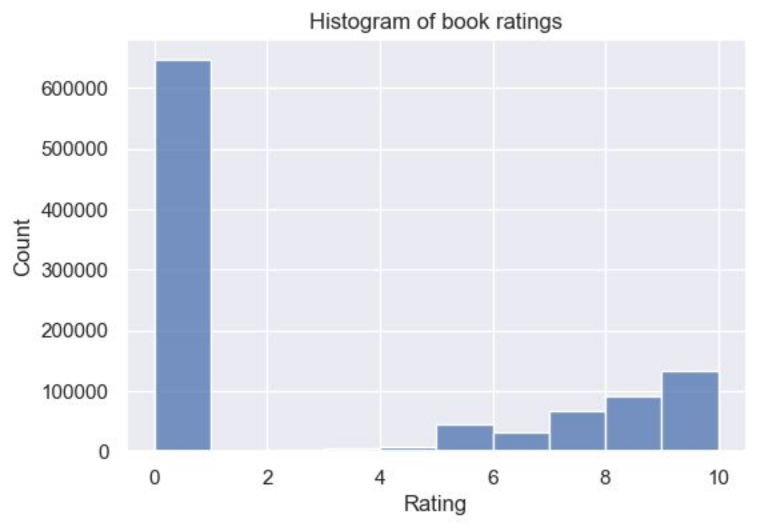
Basic descriptives



```
Column: User-ID, Count: 105283, Data Type: int64
Column: ISBN, Count: 340556, Data Type: object
Column: Book-Rating, Count: 11, Data Type: int64
Column: Book-Title, Count: 241071, Data Type: object
Column: Book-Author, Count: 101588, Data Type: object
Column: Year-Of-Publication, Count: 202, Data Type: object
Column: Publisher, Count: 16729, Data Type: object
```

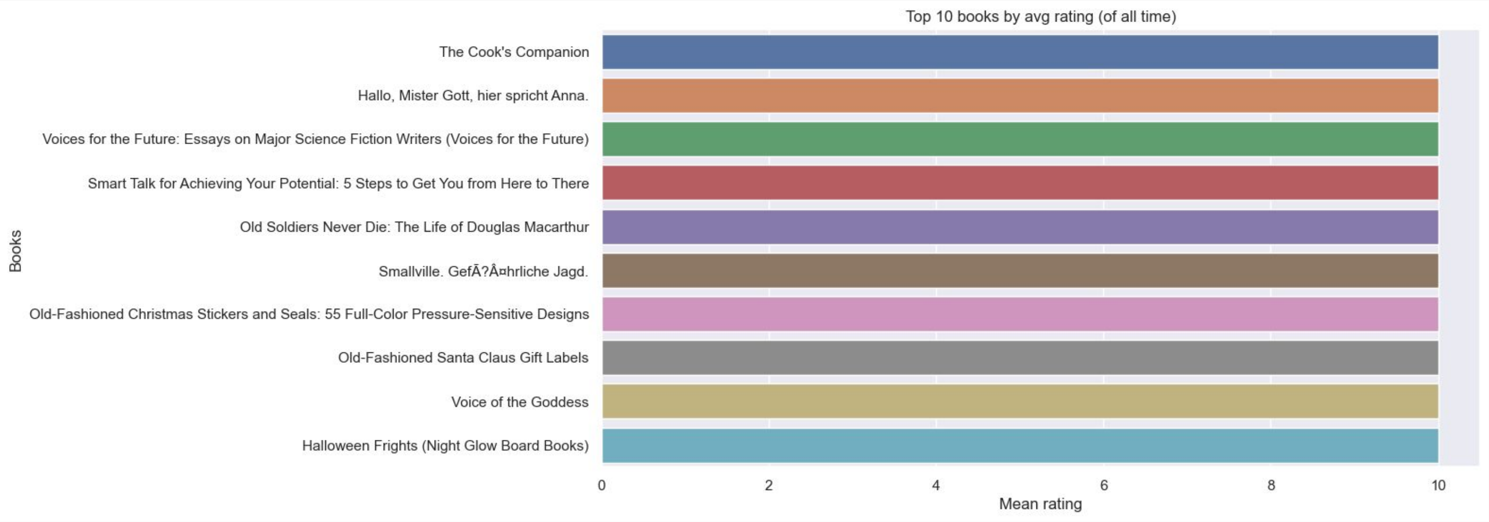
Exploratory Data Analysis

Distribution of book ratings over time



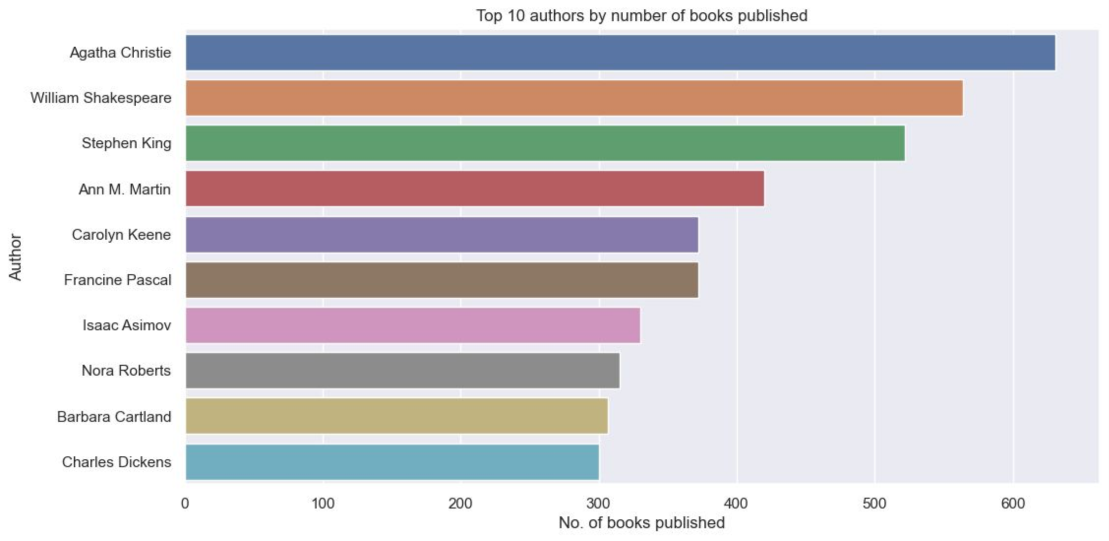
Exploratory Data Analysis

Top 10 books (by average rating) of all time



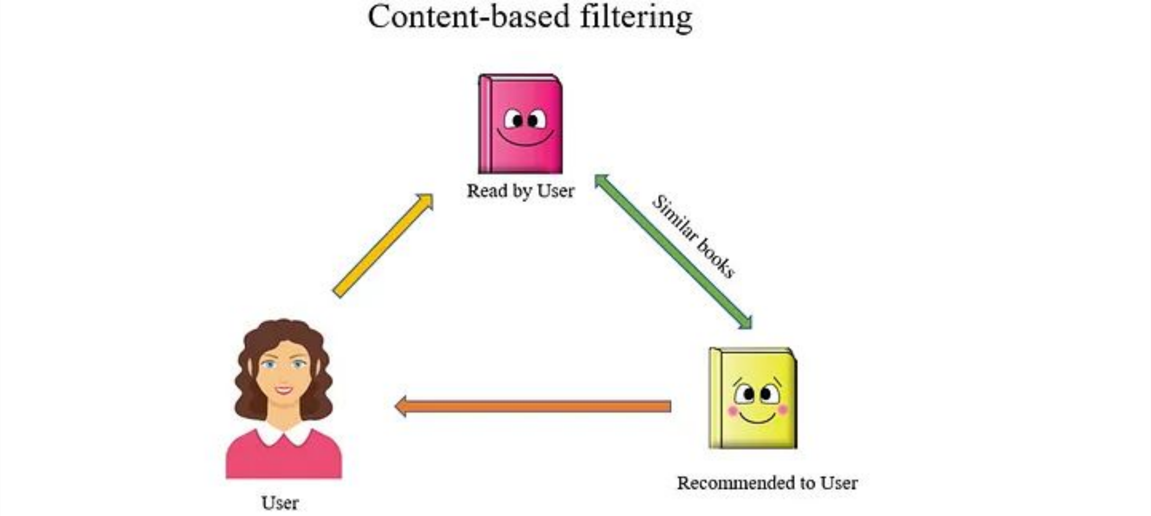
Exploratory Data Analysis

Top 10 authors (by # of books published) of all time



Method 1 - Content Based Filtering

Content-based filtering



Source: https://medium.com/@beepabose/content-based-filtering-for-book-recommendation-using-pyspark-4369c4cbe006

Goal: Recommend ‘similar’ books to users based on current selection

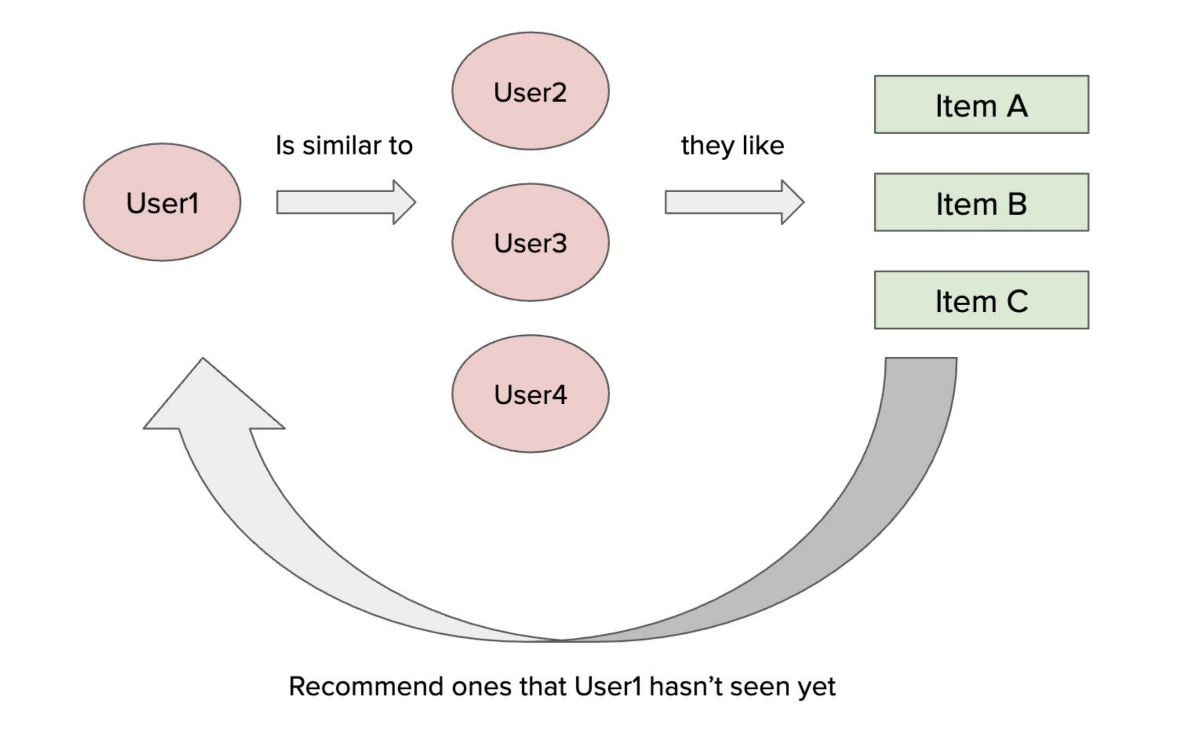
Content-based filtering

- Create feature matrix - generally consists of all variables that may affect recommendations. We used title, author and year of publication
- Vectorize ‘words’ - TF-IDF, word2vec (we used a pre-trained model)
- Calculate similarity between two items: we used cosine similarity (most commonly used metric)
- Display top 20 recommendations based on current selection (user input based)

Method 2 - Collaborative Filtering

Goal:

Recommend book titles that similar users to the input user have already interacted with

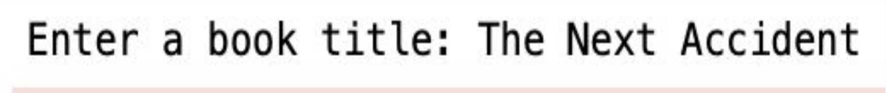


Working:

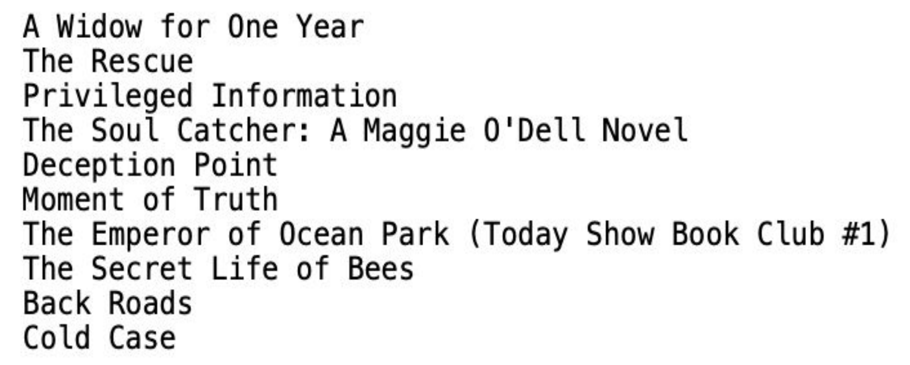
- Construct a matrix with users as rows, book titles as columns, and book ratings as values
- Calculate cosine similarities between users based on their book rating patterns.
- For a given book title, find users who have rated it.
- Aggregate these users similarities to all other users.
- Compute weighted ratings for each book, factoring in user similarities.
- Sort the books based on their weighted ratings.
- Exclude the input book title from the recommendations.
- Select the top N book titles as recommendations.

Results

Example 1:

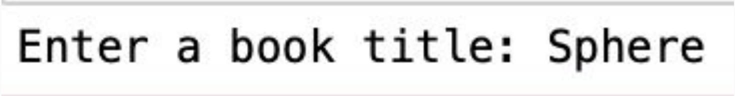


```
Enter a book title: The Next Accident
```

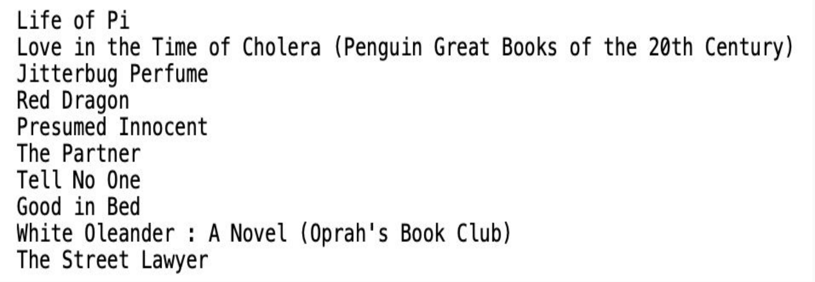


```
A Widow for One Year
The Rescue
Privileged Information
The Soul Catcher: A Maggie O'Dell Novel
Deception Point
Moment of Truth
The Emperor of Ocean Park (Today Show Book Club #1)
The Secret Life of Bees
Back Roads
Cold Case
```

Example 2:



```
Enter a book title: Sphere
```



```
Life of Pi
Love in the Time of Cholera (Penguin Great Books of the 20th Century)
Jitterbug Perfume
Red Dragon
Presumed Innocent
The Partner
Tell No One
Good in Bed
White Oleander : A Novel (Oprah's Book Club)
The Street Lawyer
```

Next steps

Next steps

Evaluation criteria - determine which model performs best

- Conduct user studies/surveys to gather feedback on the relevance, diversity, and overall satisfaction with the recommendations from both methods.
- Monitor user engagement metrics like click-through rates for the recommended books.
- Enhance current model - hyperparameter tuning to improve quality of recommendations, additional data cleaning etc.

UI