Book Recommender

Ching & Chloe

Table of Contents

Business Objectives

01

Business Objectives

Increase average order value

Showing tailored alternatives can allow customers to reveal options that fulfill his/her interest, hence likely to add purchase

Drive website engagement

Potential customers can easily dive deeply into product line without the need of doing search after search, enhance user satisfaction

Better marketing direction

Marketing team can set promotion prices directives to customer's profile, thus facilitate boosting sales

02

Data Collection

UCSD Book Graph

Home · Books · Shelves · Reviews · Github (Code Samples)

The datasets were collected in late 2017 from goodreads.com, where we only scraped users' *public* shelves, i.e. everyone can see it on web without login. User IDs and review IDs are anonymized.

We collected these datasets for academic use only. Please do not redistribute them or use for commercial purposes.

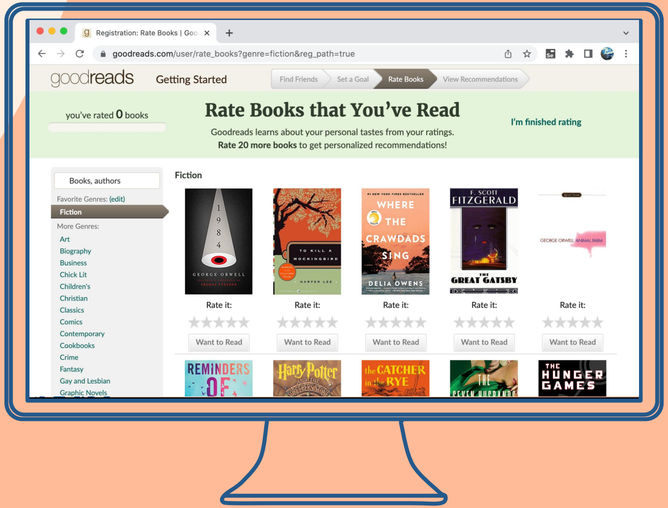
If you are using our datasets, please cite the following papers:

- Mengting Wan, Julian McAuley, "Item Recommendation on Monotonic Behavior Chains", in *RecSys'18*. [bibtex]
- Mengting Wan, Rishabh Misra, Ndapa Nakashole, Julian McAuley, "Fine-Grained Spoiler Detection from Large-Scale Review Corpora", in *ACL'19*. [bibtex]

https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home

Goodreads.com

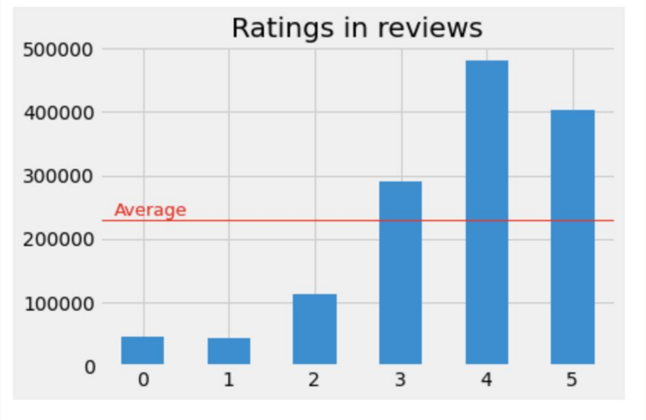
- No. of books: 2,080,190
- No. of reviews: 15,739,967
- No. of users: 465,323

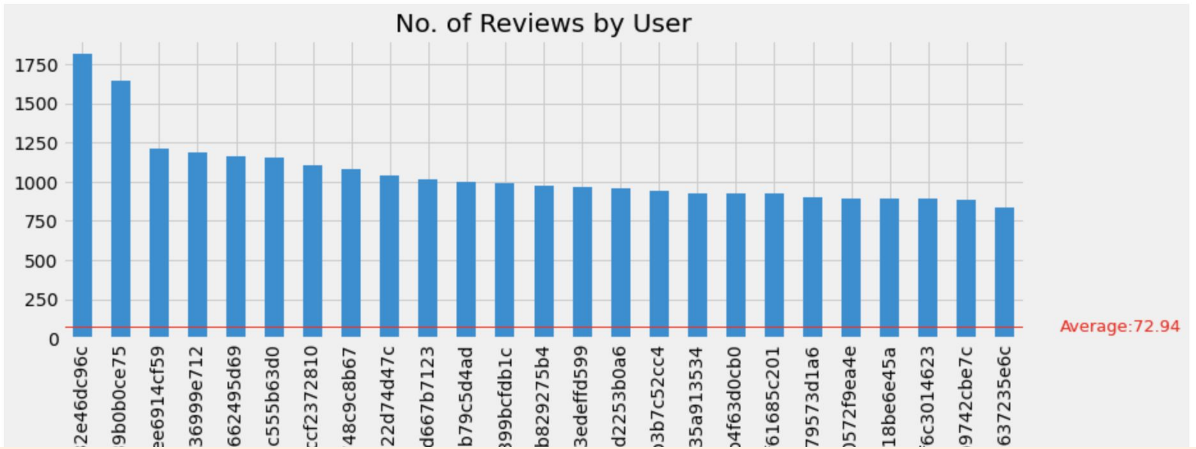


EDA & Data Preprocessing

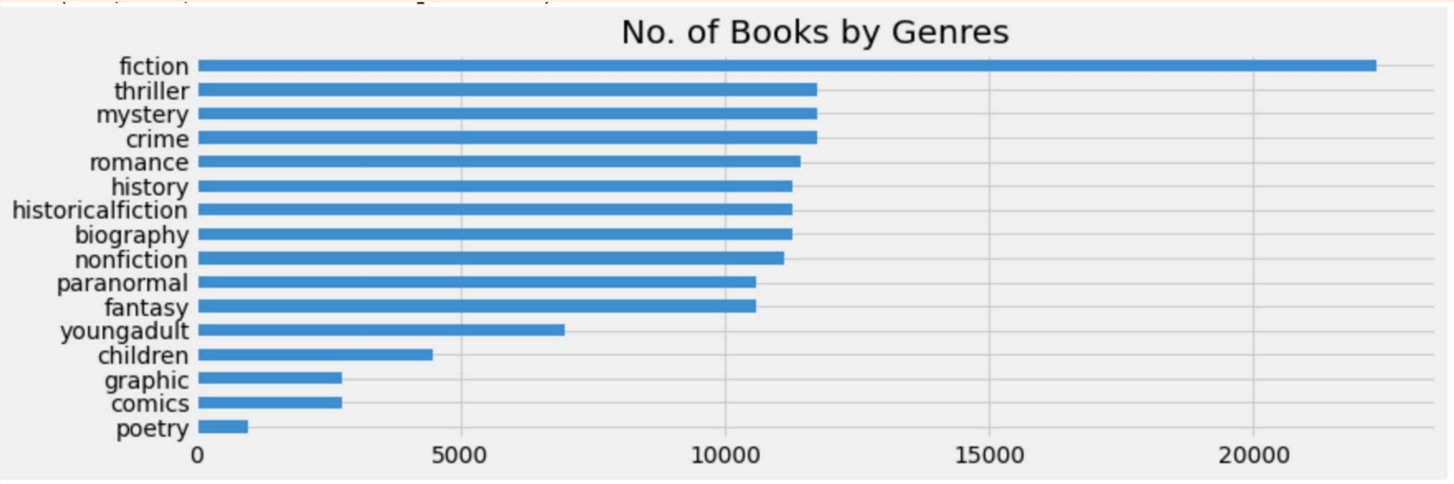
03

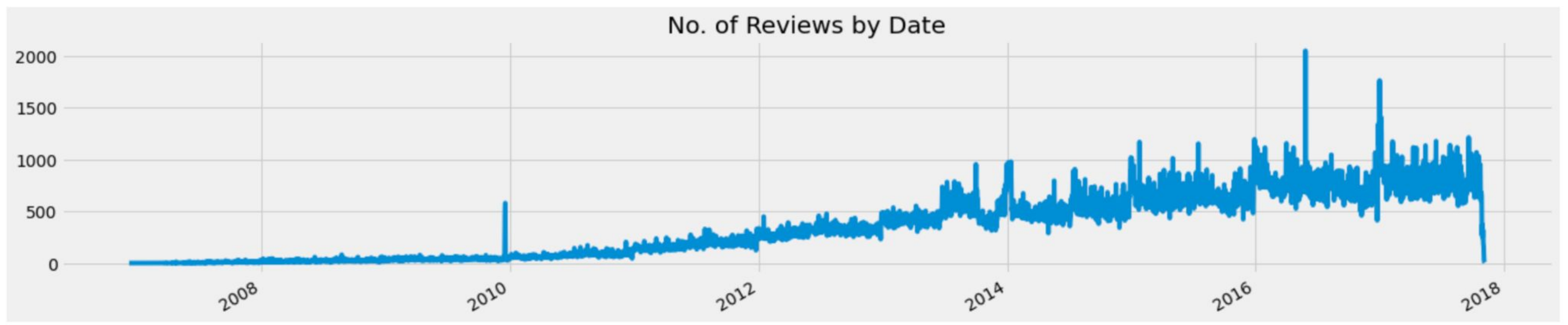
EDA


Ratings in reviews
500000
400000
300000
200000
100000
0
Average
0
1
2
3
4
5

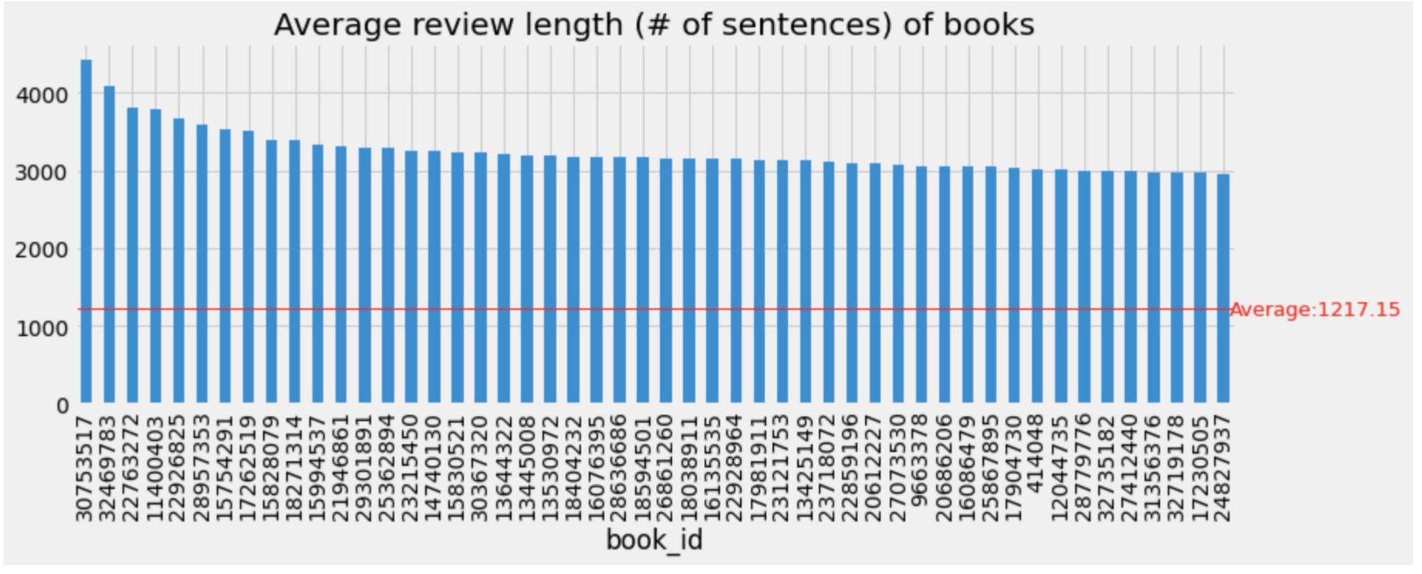

No. of Reviews by User
1750
1500
1250
1000
750
500
250
0
Average:72.94
2e46dc96c
9b0b0ce75
e6914cf59
36999e712
662495d69
c555b63d0
cf2372810
48c9c8b67
22d74d47c
d667b7123
b79c5d4ad
399bcfdb1c
b829275b4
3edeffd599
d2253b0a6
3b7c52cc4
35a913534
4f63d0cb0
61685c201
79573d1a6
572f9ea4e
18be6e45a
6c3014623
9742cbe7c
637235e6c

EDA


No. of Books by Genres
fiction
thriller
mystery
crime
romance
history
historicalfiction
biography
nonfiction
paranormal
fantasy
youngadult
children
graphic
comics
poetry
0
5000
10000
15000
20000

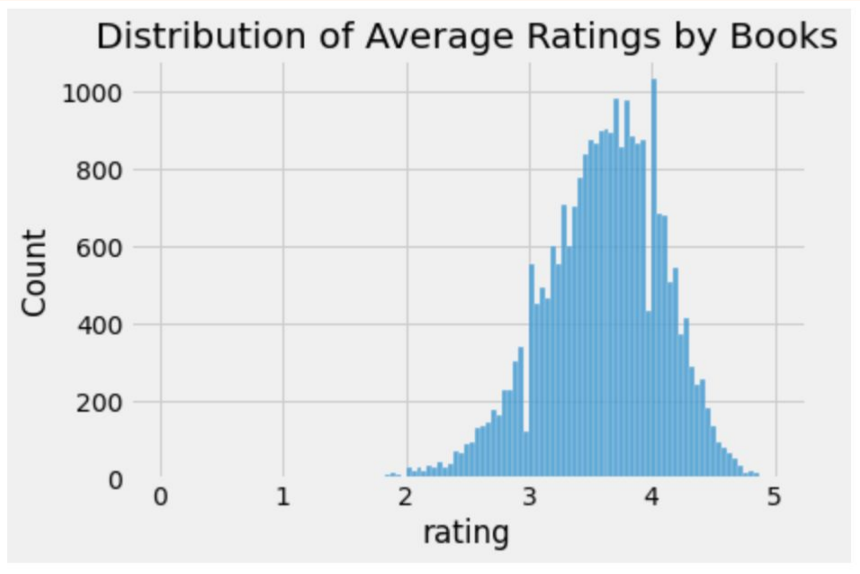

No. of Reviews by Date
2000
1500
1000
500
0
2008
2010
2012
2014
2016
2018

EDA

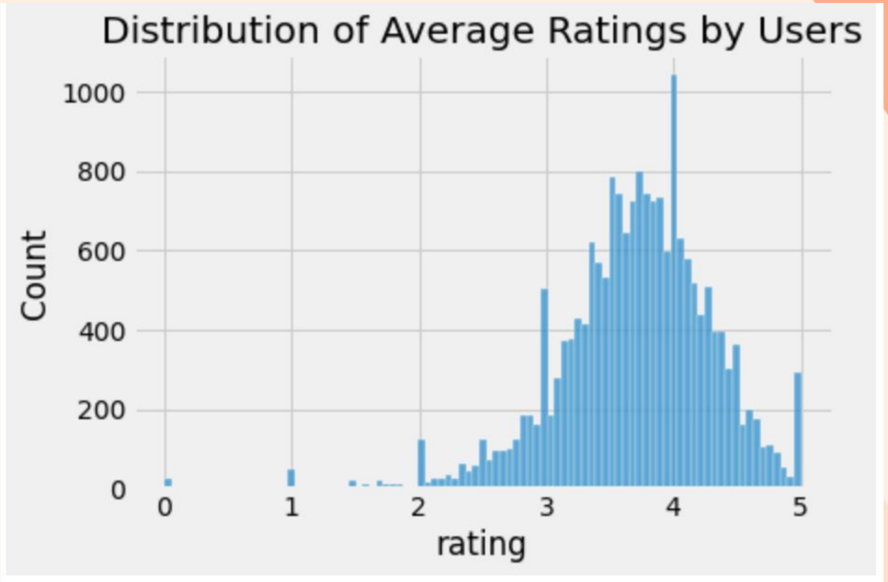


Average review length (# of sentences) of books
4000
3000
2000
1000
0
Average:1217.15
30753517
32469783
22763272
11400403
22926825
28957353
15754291
17262519
15828079
18271314
15994537
21946861
29301891
25362894
23215450
14740130
15830521
30367320
13644322
13445008
13530972
18404232
16076395
28636686
18594501
26861260
18038911
16135535
22928964
17981911
23121753
13425149
23718072
22859196
20612227
27073530
9663378
20686206
16086479
25867895
17904730
414048
12044735
28779776
32735182
27412440
31356376
32719178
17230505
24827937
book_id

EDA

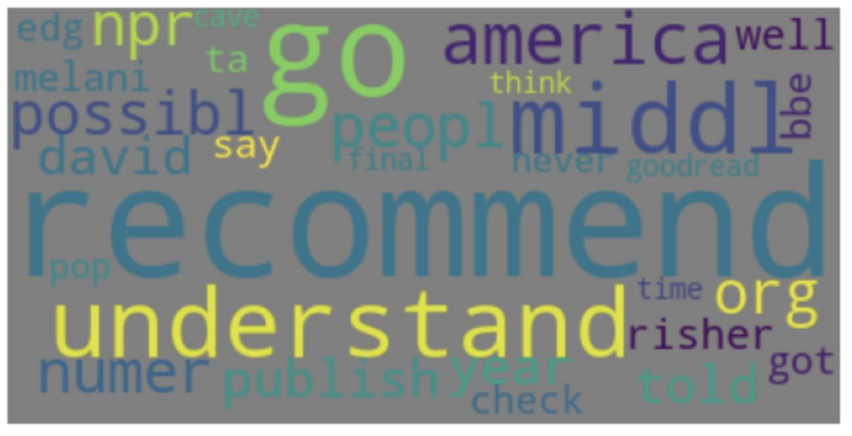


Distribution of Average Ratings by Books
1000
800
600
400
200
0
Count
0
1
2
3
4
5
rating



Distribution of Average Ratings by Users
1000
800
600
400
200
0
Count
0
1
2
3
4
5
rating

Data preprocessing



Data preprocessing

	isbn	series	language_code	average_rating	description	authors	publication_year	image_url	book_id	work_id	title
0	0312853122	[]	NaN	4.00	NaN	[{'author_id': '604031', 'role': ''}]	1984.0	https://images.gr-assets.com/books/1310220028m...	5333265	5400751.0	W.C. Fields: A Life on Film
1	0743509986	[]	NaN	3.23	Anita Diamant's international bestseller "The ...	[{'author_id': '626222', 'role': ''}]	2001.0	https://s.gr-assets.com/assets/nophoto/book/11...	1333909	1323437.0	Good Harbor

	user_id	book_id	rating	has_spoiler	review_sentences	work_id	title	language_code
0	0	18245960	5	1	[[0, 'This is a special book.'], [0, 'It start...	25696480	The Three-Body Problem (Remembrance of Earth's...	NaN
1	0	16981	3	0	[[0, 'Recommended by Don Katz.'], [0, 'Avail f...	170957	Invisible Man	NaN
2	0	28684704	3	1	[[0, 'A fun, fast paced science fiction thrill...	43161998	Dark Matter	NaN

01 List Data

Used 'eval' to convert series, authors from string to lists

02 Mapping across dataframes

Books, Reviews, Genre and Authors are all separate tables

03 Downcast to save memory

Convert the numerical data to float with less precision/ integer

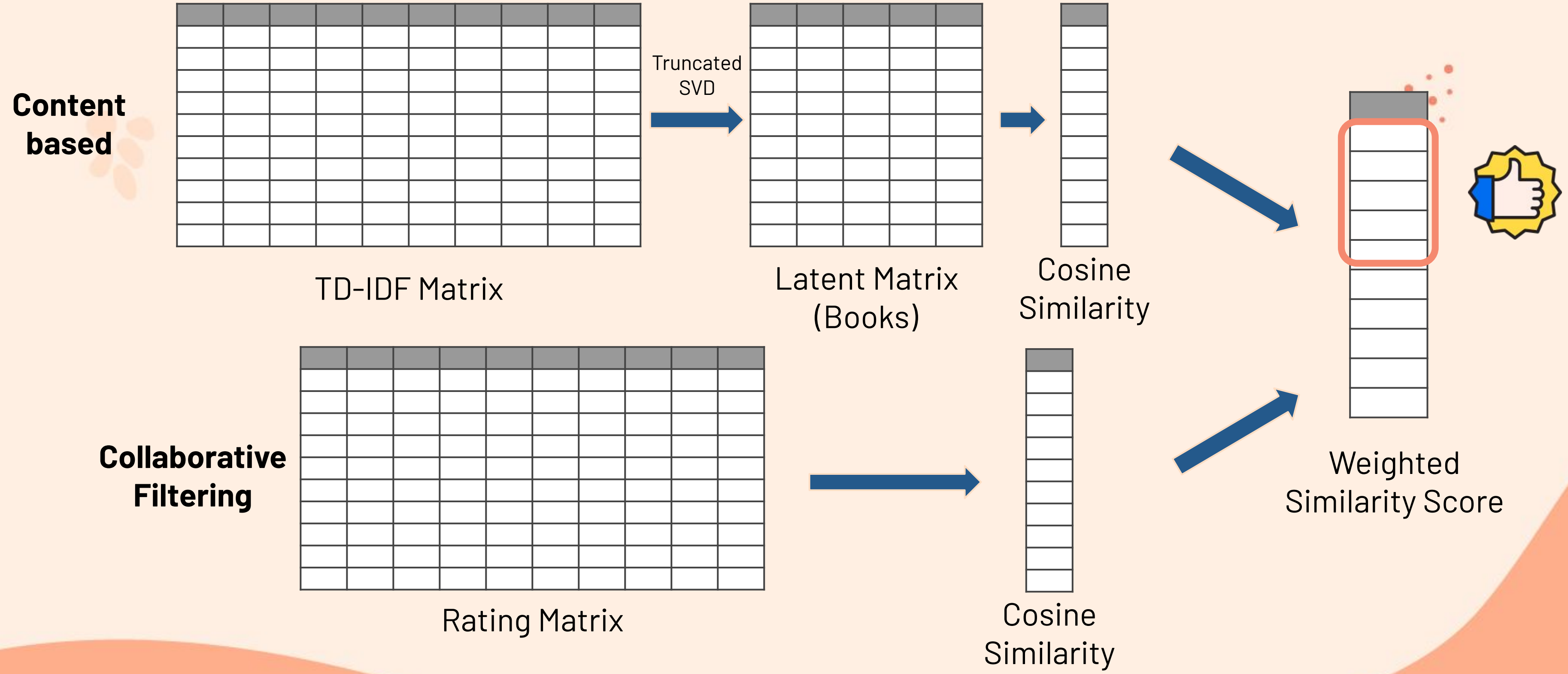
04 Text preprocessing

Remove contractions and punctuation. Tokenize and stem words.

04

Model Creation

Book-to-book Recommendation

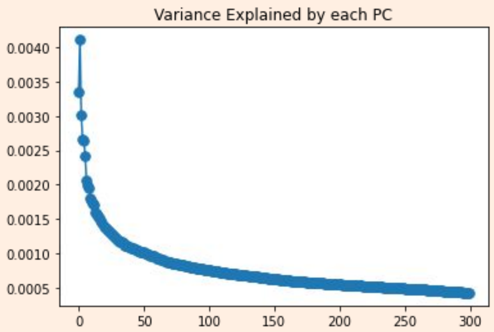


Model Creation - Latent Matrices

Latent matrix on book content

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vect = TfidfVectorizer(min_df=0.0005,max_df=0.7, stop_words='english', dtype= np.float32)
text_M = tfidf_vect.fit_transform(text)
```

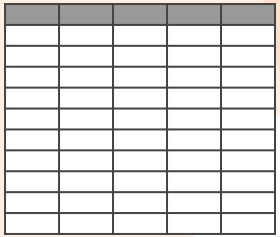
```
from sklearn.decomposition import TruncatedSVD
svd_text = TruncatedSVD(n_components=300)
svd_text.fit(text_M)
```



Latent matrix on user ratings

```
ratingmatrix = pd.pivot_table(df_reviews, values='rating',index=['work_id'], columns=['user_id'])
ratingmatrix.fillna(0, inplace=True)
ratingmatrix.shape
```

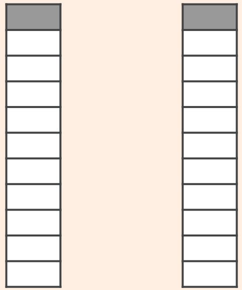
user_id	0	1	2	3	4	5	6	7	8	9	...	18882	18883	18884	18885	18886	18887	18888	18889	18890	18891
work_id																					
104	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
114	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
115	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
423	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
434	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
57767021	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
57824359	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
57888058	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
57978319	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
57997938	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0



Latent Matrix

Book-to-book (Similarity Score)

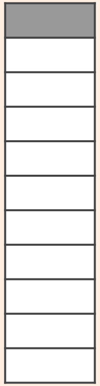
```
def book_to_books(seedbookID, latentmatrix, rec_mode):
    if rec_mode == 'collaborative':
        seed_book = np.array(latentmatrix.loc[seedbookID]).reshape(1,-1)
    if rec_mode == 'content':
        seed_book = latentmatrix[df_books.index[df_books['work_id'] == seedbookID]]

    similarities = cosine_similarity(latentmatrix, seed_book, dense_output=True)

    if rec_mode == 'collaborative':
        index = latentmatrix.index.tolist()
    if rec_mode == 'content':
        index = df_books['work_id'].tolist()

    similarities = pd.DataFrame(similarities, index = index)
    similarities.columns = ['similarity_score']
    similarities.sort_values('similarity_score', ascending=False, inplace=True)
    similarities = similarities.iloc[1:]
    similarities = similarities[similarities['similarity_score'] > 0]

    return similarities
```



Cosine Similarity   Cosine Similarity

Book-to-book (Get Recommendations)

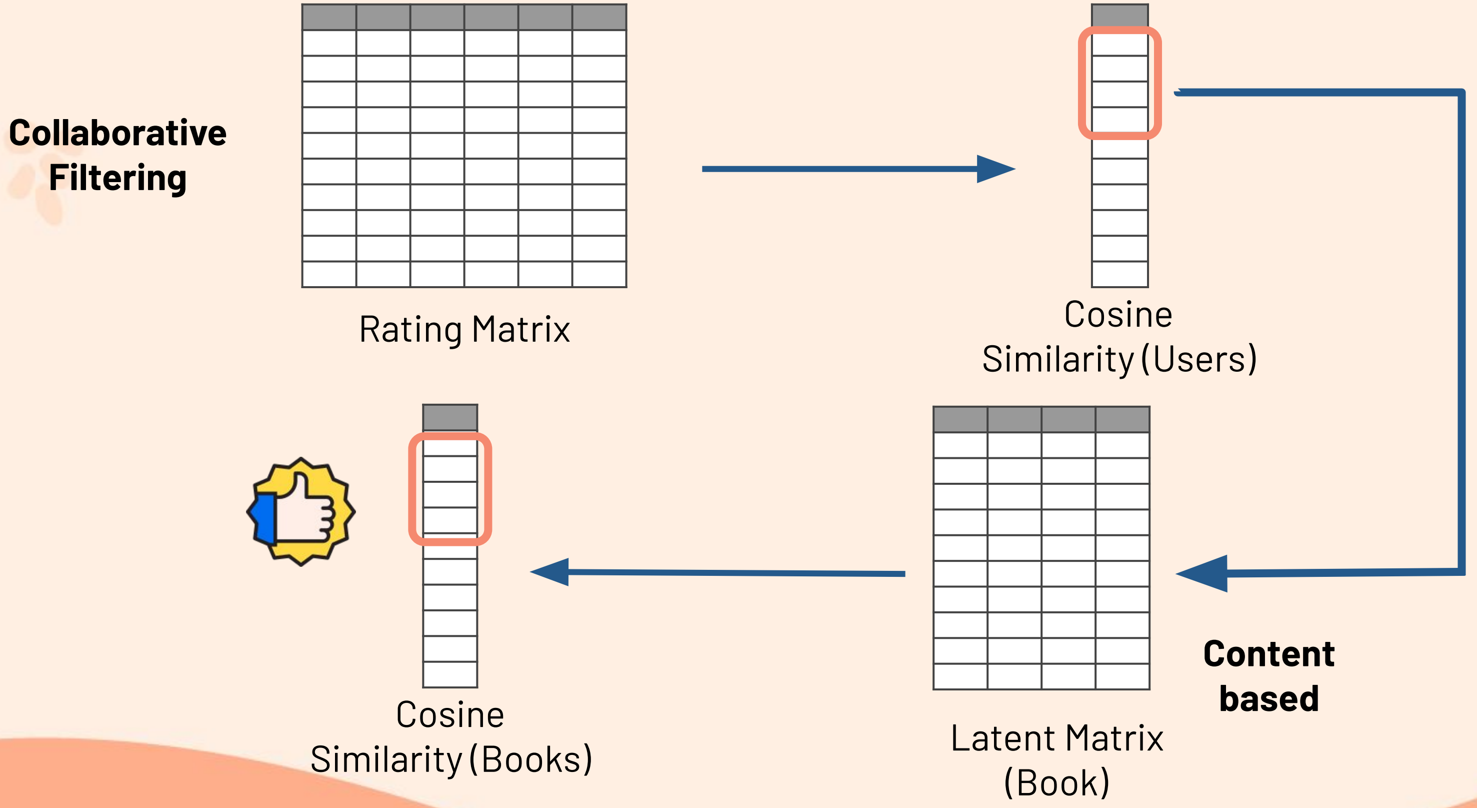
```
def  similarity_scores(collaborative_score,  content_score):
    #average  both  similarity  scores
    df_sim  =  pd.merge(collaborative_score,  pd.DataFrame(content_score['similarity_score']),  left_index=True,  right_index=True)
    df_sim['similarity_score']  =  (df_sim['similarity_score_x']*0.5  +  (df_sim['similarity_score_y'])*0.2)/2
    df_sim.drop("similarity_score_x",  axis=1,  inplace=True)
    df_sim.drop("similarity_score_y",  axis=1,  inplace=True)

    #sort  by  average  similarity  score
    df_sim.sort_values('similarity_score',  ascending=False,  inplace=True)

    #round  similarity  score
    df_sim['similarity_score']  =  df_sim['similarity_score'].round(4)

    return    df_sim.head(20)
```



Weighted
Similarity Score

```
def  get_recommendation(seed_book):
    collaborative  =  book_to_books(seed_book,  rating_latent,  'collaborative')
    content  =  book_to_books(seed_book,  books_latent,  'content')
    rec  =  similarity_scores(collaborative,  content)
    rec  =  pd.merge(df_books,  rec,  how='right',  left_on='work_id',  right_index=True).reset_index().drop(['index','similarity_score'],  axis=1)
    rec  =  rec[['title','authors','work_id','isbn','description','average_rating','image_url']]
    return  rec[:5]
```

User-to-book Recommendation



User-to-book (Get Recommendations)

1. Return users with highest similarity scores

```
def user_to_user(seeduserID):
    seed_user = np.array(latent_rating_u2u.loc[seeduserID]).reshape(1,-1)
    similarities = cosine_similarity(latent_rating_u2u, seed_user, dense_output=True)
    index = latent_rating_u2u.index.tolist()
    similarities = pd.DataFrame(similarities, index = index)
    similarities.columns = ['similarity_score']
    similarities.sort_values('similarity_score', ascending=False, inplace=True)
    similarities = similarities.iloc[1:]
    similarities = similarities[similarities['similarity_score'] > 0]

    return similarities[:min(len(similarities),30)]
```

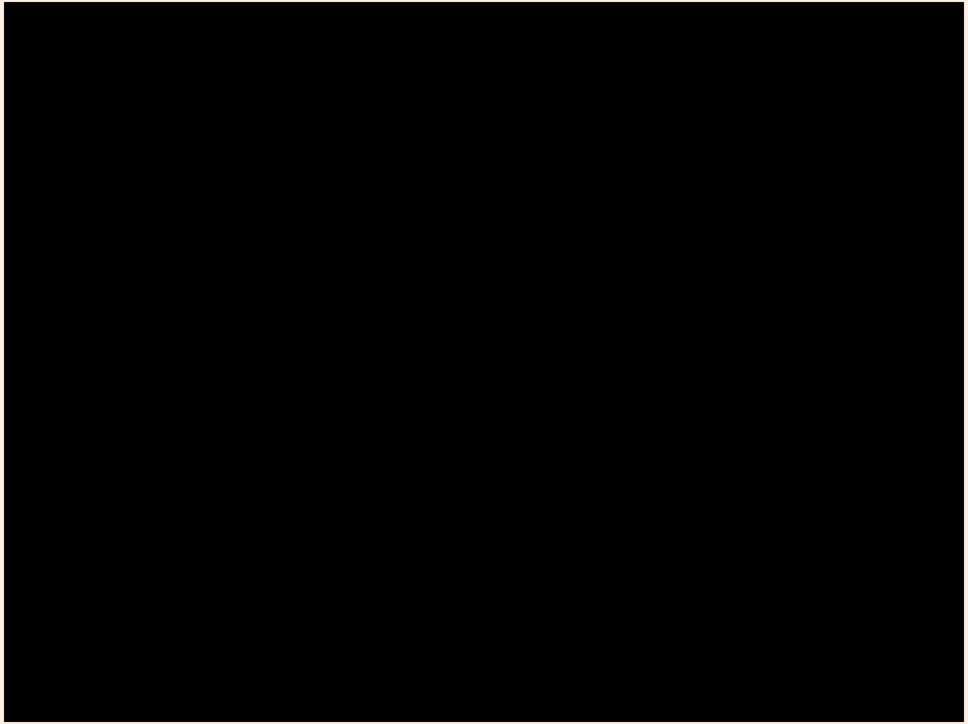
2. Get reviewed books by the similar users and filter books by content-based similarity

```
def user_to_book(userID):
    latent_rating_u2u = rating_latent.transpose()

    sim_user = user_to_user(userID)

    book_read = latent_rating_u2u.loc[userID]
    book_read = book_read.loc[(book_read > 0)]
    book_read = book_read.sort_values(ascending=False)[:min(30,len(book_read))]

    sim = latent_rating_u2u[latent_rating_u2u.index.isin(sim_user.index)]
    sim = sim.loc[:, (sim > 0).any(axis=0)].loc[~sim.index.isin(book_read.index)]
    avgrating = sim.replace(0,np.nan).apply(np.nanmean).dropna()
    avgrating = avgrating.sort_values(ascending=False).index[:min(50,len(avgrating))]

    similarities = cosine_similarity(book_latent[df_books.index[df_books['work_id'].isin(list(avgrating))]],book_latent[df_books.index[df_books['work_id'].isin(list(book_read.index))]])
    cos = pd.DataFrame(similarities, index = list(avgrating), columns = list(book_read.index))
    rec = df_books[df_books['work_id'].isin(
        cos.index[[a for (a,b) in [divmod(i, 30) for i in np.argsort(similarities, axis=None)[-10:]]][::-1]])][['title','authors','work_id','isbn','description','average_rating']]

    return rec[:5]
```

Web Application

05

Streamlit WebApp



06

Limitations & Next Steps

Limitations & Future Improvements

Matrix too large

- Used dimensionality reduction by truncated SVD
- Removed unused columns
- Due to large database, there is not enough local ram to compute

Goodreads.com could not fetch image

- Use another book cover api to fetch image
- But some books within the dataset didn’t include isbn, so those books can’t show image

Next Steps

Combine other user-book interaction

(is_read, is_reviewed, add to cart...)

Apply other hybrid techniques

(Cascade, Switching...)

Build Models of Neural Network

Thanks!