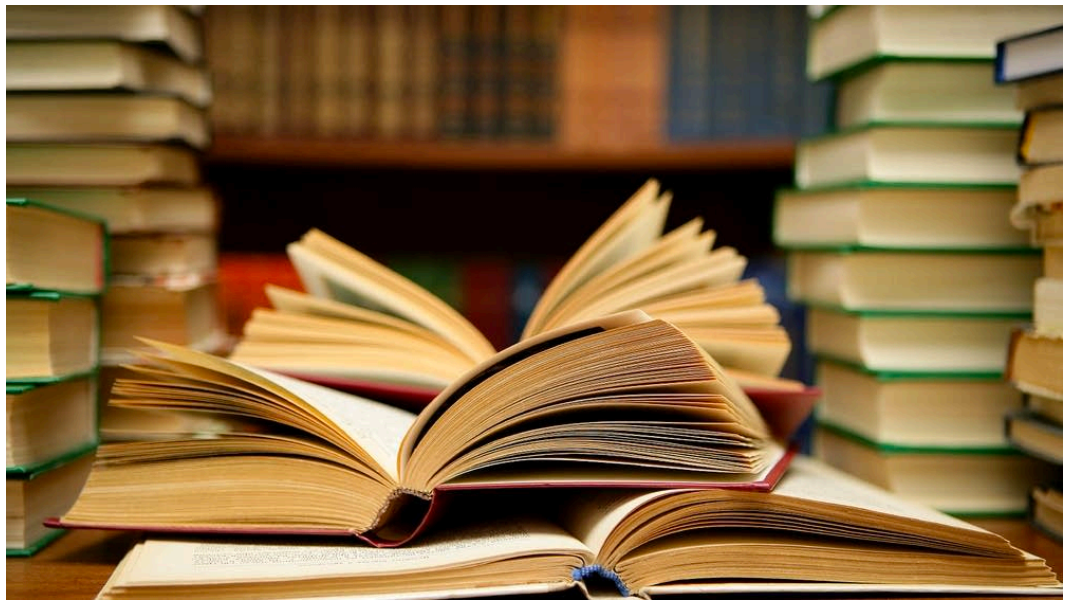
Book Recommendation System

- Motivation & Background
  - Content Based & Collaborative Based
  - Memory based & Model Based
- Data Cleaning & Exploration
- Machine Learning
  - Model Selection
  - Hyperparameter Tuning
  - Generating recommendations
- Conclusions & Next Steps



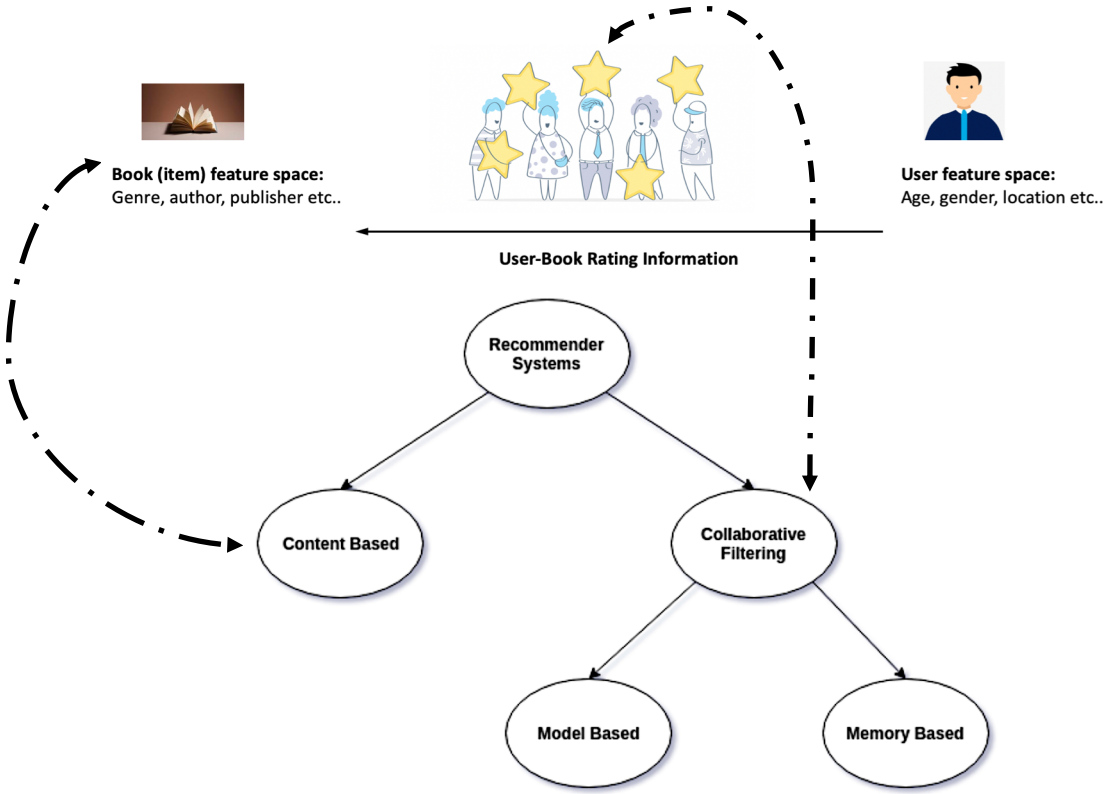
Rochita Sundar
March'22

Recommendation Systems: One of the largest application areas of ML. Enable tailoring personalized content for users, thereby generating revenue for businesses

Content based recommendations based on user past likes & dislikes – System recommends items similar to items users have liked based on item feature space

Collaborative based recommendations – more robust & widely used. Disregards item & user feature space & solely based on how different users rate different items

- **Memory based approach**
- **Model based approach**

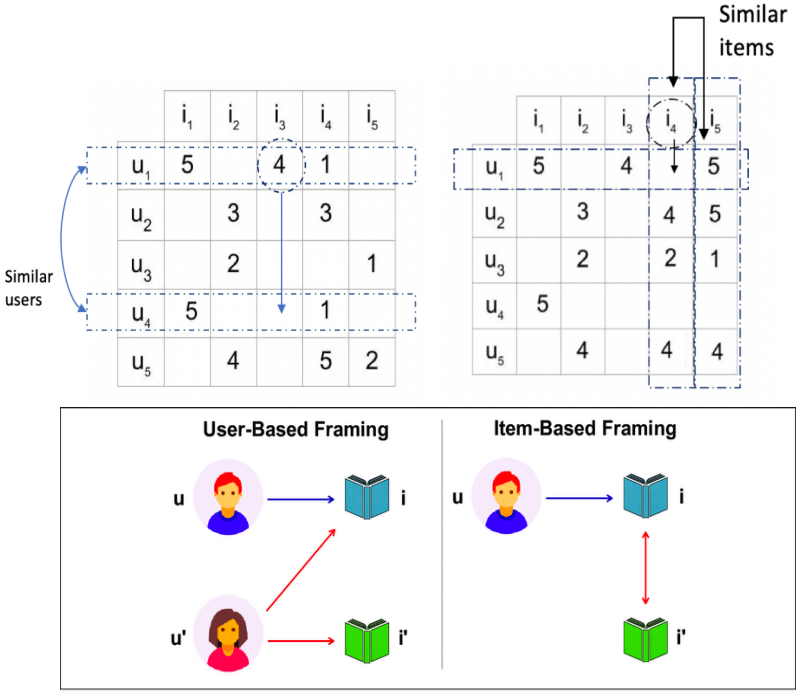


Memory based based Collaborative filtering
- e.g. KNN (K-Nearest Neighbors)

Utilizes entire user-item rating information to calculate similarity scores between users or items

In user based collaborative filtering, two user's are considered similar, if they rate items in a similar manner. An item is recommended to a user, if another user i.e., similar to the user in question has liked the item

In item based collaborative filtering, two item's are considered similar, if users rate them in a similar manner. An item is recommended to a user, that is similar to the items the user has rated in the past



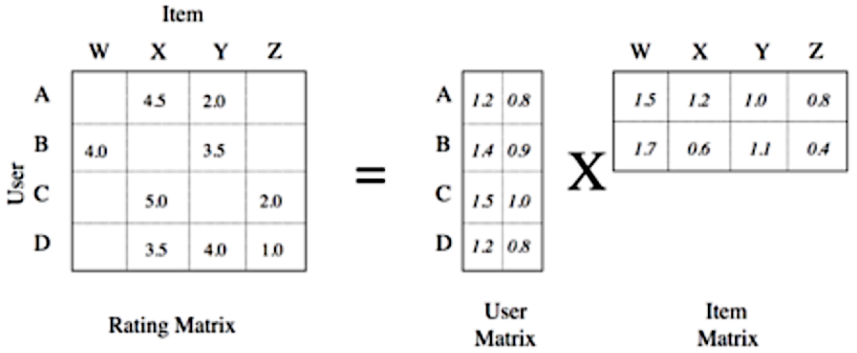
**Making Recommendations More Effective Through Framings: Impacts of User- Versus Item-Based Framings on Recommendation Click-Throughs - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Illustration-of-the-definitions-of-user-based-and-item-based-framing_fig1_335781811 [accessed 4 Mar, 2022]*
**https://realpython.com/build-recommendation-engine-collaborative-filtering/*

Model based based Collaborative filtering
- e.g. Singular Value Decomposition

Model based approach utilizes information from the dataset to build a model & the model (not the entire dataset) is thereafter used for making recommendations

For dimensionality reduction, the model optimizes certain latent features (hidden characteristics) that are able to map user item preferences

In the process, the model is able to predict an estimated rating for all user-item pair, where user has not yet rated an item



**Source: data-artisans.com*

Data exploration & cleaning

- 'Book-Crossing dataset' collected by Cai-Nicolas Ziegler (http://www2.informatik.uni-freiburg.de/~cziegler/BX/)

User-ID	ISBN	Book-Rating
277427	002542730X	10
277427	006092988X	0
277427	0060930535	0
277427	0060932139	0
277427	0060934417	0
...	...	...
275970	1400031354	0
275970	1400031362	0

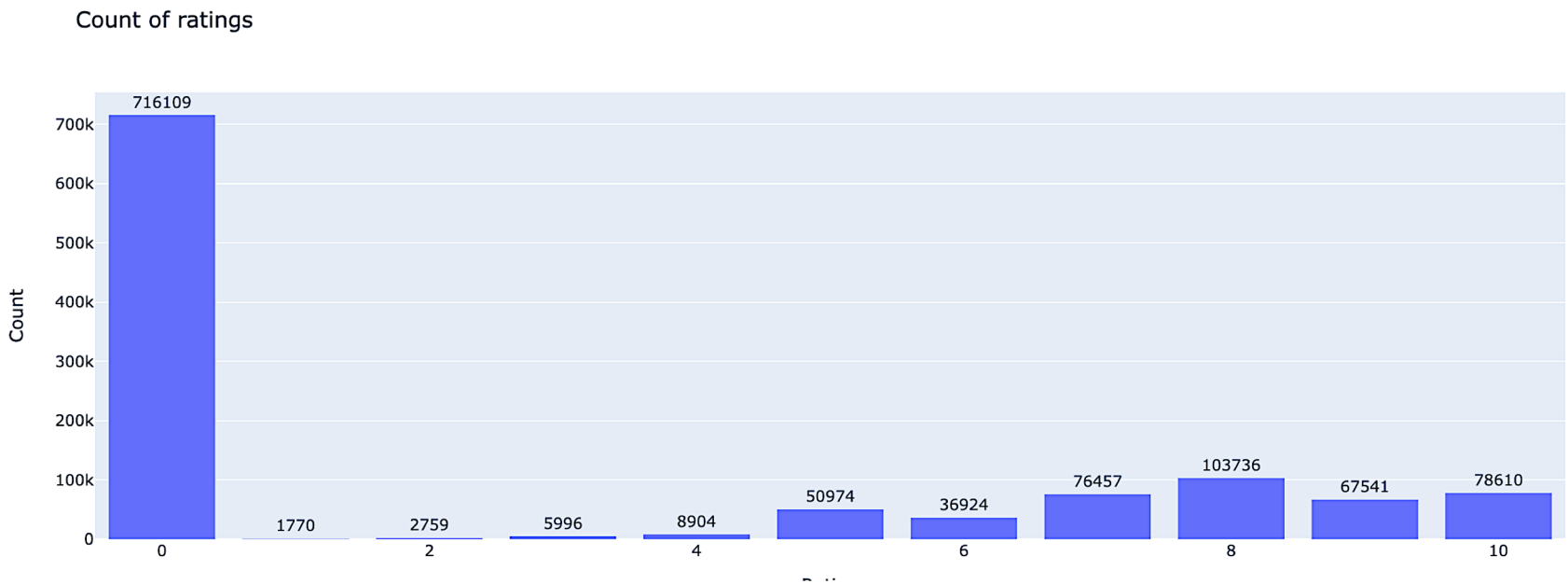
– 1,149,780 records

merge rating table with book table using 'ISBN' to include book titles instead of 'ISBN' for interpretability

User-ID	Book-Title	Book-Rating
277427	Politically Correct Bedtime Stories: Modern Ta...	10
3363	Politically Correct Bedtime Stories: Modern Ta...	0
11676	Politically Correct Bedtime Stories: Modern Ta...	6
12538	Politically Correct Bedtime Stories: Modern Ta...	10
13552	Politically Correct Bedtime Stories: Modern Ta...	0
...	...	...
234828	Ringworld	8
236283	Ringworld	0

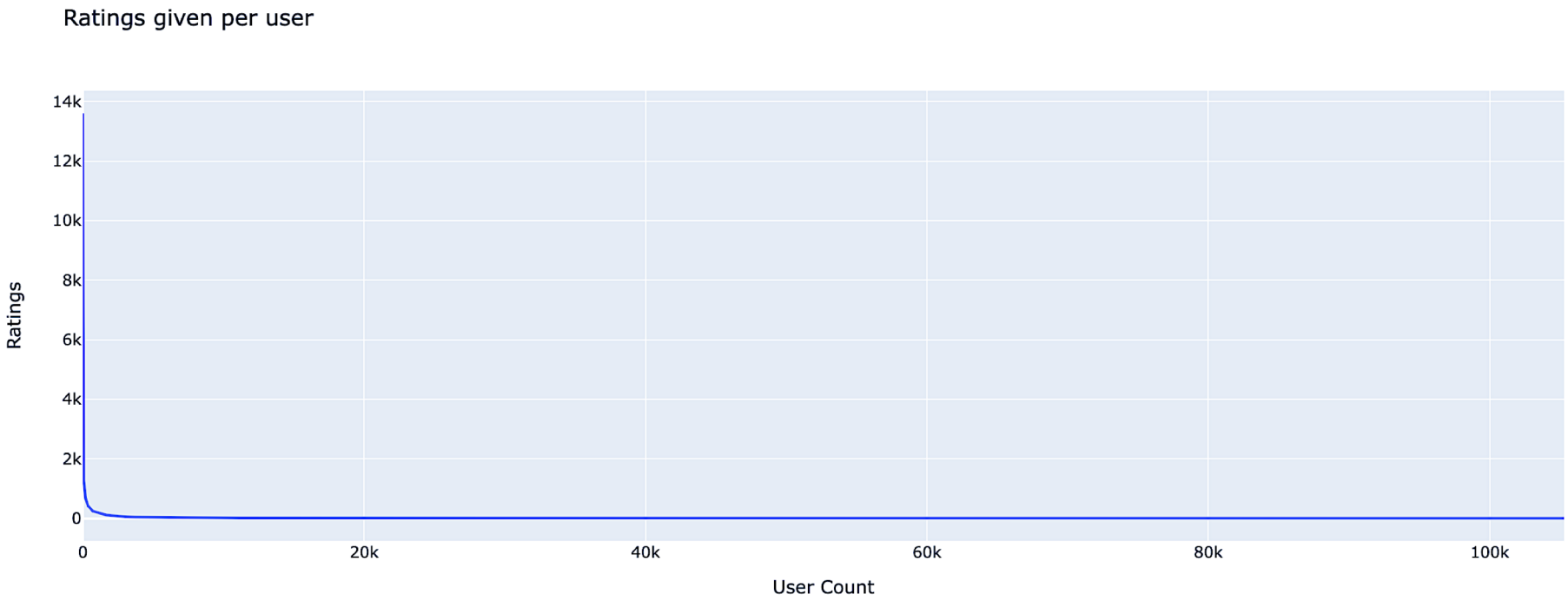
Check for missing values & duplicate entries
- Drop them from the database

Exploratory data analysis



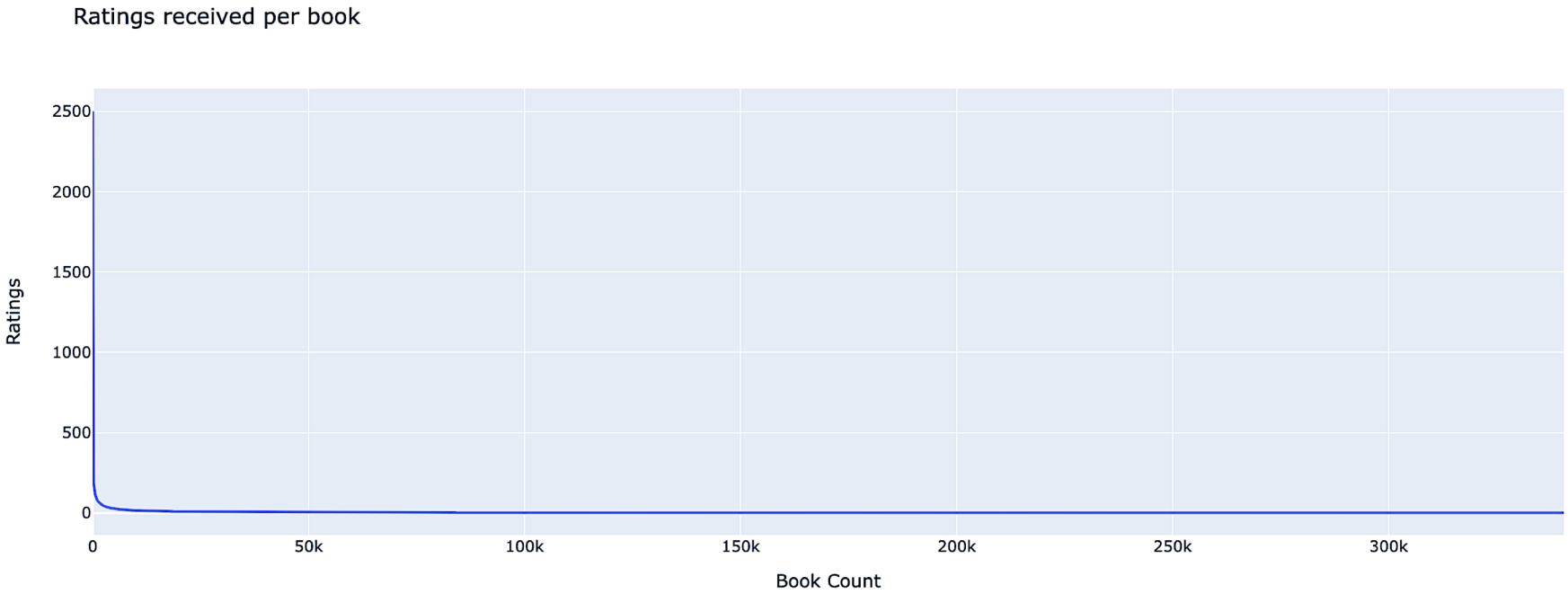
Ratings are of two types, an *implicit rating* & *explicit rating*. An implicit rating is based on tracking user interaction with an item such as a user clicking on an item '0'. An explicit rating is when a user explicitly rates an item, i.e., b/w '1-10'

- Majority of ratings are implicit i.e., rating '0'
- Rating of '8' has the highest rating count among explicit ratings '1-10'



There is inherent bias in the dataset. There are few users who rate a lot & several users that provide very few ratings. One user has provided 13K+ ratings, only ~700 users (out of 100K+ users) have provided over 250 ratings

Filter dataset to only include users that have provided > 250 ratings



A similar bias is observed. There are a few books that have received many ratings and several books that have received very few ratings. One book has received over 2500 ratings, only ~2100 books (out of 300K+ books) have received more than 50 ratings

Filter dataset to only include books that have received > 50 ratings

Machine Learning – Model Selection

- After data cleaning, there were 78,782 records left
  - 686 unique users who have rated > 250 books each
  - 1913 unique book titles that have received > 50 ratings each
- Use Python's surprise library algorithms for building recommendations

5 fold cross validation model performance on training set (with default model parameters):

	MAE	RMSE	fit_time	test_time
knns.KNNWithMeans	2.351416	3.289229	0.131446	1.228990
knns.KNNBaseline	2.356247	3.299194	0.148220	1.486081
matrix_factorization.SVD	2.392815	3.313362	5.206008	0.194471
knns.KNNWithZScore	2.336986	3.327676	0.247611	2.502699
knns.KNNBasic	2.445218	3.501194	0.201198	1.567211

- KNNWithMeans has the least RMSE (root mean square error) among KNN algorithms
- The model fit_time is the maximum for SVD but the model test_time is the least

Machine Learning – Hyperparameter tuning with GridSearchCV

KNNWithMeans Model Parameters:

- **Name:** distance measure, e.g., MSD or cosine
- **Min_support** refers to minimum number of items to consider for calculating similarity between users in user based collaborative filtering (or vice versa)
- **User_based**: True or False (item based)

MAE: {'sim_options': {'name': '**cosine**', 'min_support': **3**, 'user_based': **False**}} ; MAE: 2.35610

RMSE: {'sim_options': {'name': 'msd', 'min_support': 5, 'user_based': True}} ; RMSE: 3.29518

```
Unbiased Testing Performance:
MAE:  2.3052
RMSE: 3.2597
```

Model generalizes well!

Machine Learning – Hyperparameter tuning with GridSearchCV

SVD Model Parameters:

- **n_factors** refers to number of latent factors for dimensionality reduction
- **n_epochs, lr_all, reg_all** refers to number of iterations, learning rate & regularization rate of SGD (stochastic gradient descent) procedure utilized by SVD for matrix factorization

MAE: {'n_factors': 10, 'n_epochs': 20, 'lr_all': 0.005, 'reg_all': 0.2} ; MAE: 2.37744

RMSE: {'n_factors': **50**, 'n_epochs': **10**, 'lr_all': **0.005**, 'reg_all': **0.2**} ; RMSE: 3.15542

```
Unbiased Testing Performance
MAE:  2.3800
RMSE: 3.1341
```

Model generalizes well!

Building Recommendations

KNNWithMeans

An item similarity matrix is computed between 1913 unique books

- Using cosine distance measure
- Using at least how 3 or more users have rated the books

The similarity score is between 0 & 1

	0	1	2	3	4	5	6	7	8	9	...	1903	1904	1905	1906	1907	1908	1909	1910
0	1.000000	0.34418	0.136531	0.0	0.373682	0.442840	NaN	0.570309	0.307166	0.423457	...	NaN	NaN	0.65938	0.0	NaN	0.0	NaN	0.000000
1	0.344180	1.00000	0.596550	NaN	0.000000	0.553372	0.0	0.847998	0.784592	0.695554	...	0.0	0.0	0.00000	0.0	0.0	0.0	0.00000	0.000000
2	0.136531	0.59655	1.000000	0.0	0.372429	0.294626	NaN	NaN	0.000000	0.202110	...	0.0	0.0	0.00000	0.0	NaN	NaN	0.41804	0.000000
3	0.000000	NaN	0.000000	1.0	NaN	0.000000	0.0	0.000000	NaN	0.000000	...	0.0	0.0	0.00000	0.0	0.0	0.0	0.00000	0.000000
4	0.373682	0.00000	0.372429	NaN	1.000000	0.507470	NaN	0.342624	0.366045	0.216942	...	0.0	NaN	0.00000	1.0	NaN	0.0	NaN	0.820939
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1908	0.000000	0.00000	NaN	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	NaN	0.0	0.00000	0.0	0.0	1.0	0.00000	0.000000
1909	NaN	0.00000	0.418040	0.0	NaN	0.992278	0.0	0.624695	0.000000	0.000000	...	0.0	0.0	0.00000	0.0	0.0	0.0	1.00000	0.000000
1910	0.000000	0.00000	0.000000	0.0	0.820939	0.000000	0.0	0.000000	NaN	0.000000	...	0.0	0.0	0.00000	0.0	0.0	0.0	0.00000	1.000000
1911	NaN	0.00000	NaN	0.0	NaN	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.00000	0.0	0.0	0.0	0.00000	0.000000
1912	0.000000	0.00000	0.316228	0.0	NaN	0.000000	0.0	NaN	0.000000	0.000000	...	0.0	0.0	0.00000	0.0	0.0	0.0	0.00000	0.000000

1913 rows × 1913 columns

A custom function is written to generate as many recommendations as a user wants based on their "top K" rated books:

- User rating for top "K' rated books is multiplied with similarity scores with other books using the similarity matrix. The ratings are added & divided by sum of similarities to calculate weighted rating for a book the user has not yet read

Building Recommendations

SVD

SVD uses the available information of users rating some books to build a user matrix & item matrix with some latent features (n=50)

This process by default enables a predicting a rating for all combinations of users & books where a user has not yet rated a book

A custom function is written to recommend items with high estimated ratings (as many as the user would like)

	uid	iid	r_ui	est	details
0	277427	Robert Ludlum's The Hades Factor	1.826115	1.196323	{'was_impossible': False}
1	277427	Disclosure	1.826115	1.442137	{'was_impossible': False}
2	277427	Dark Rivers of the Heart	1.826115	1.539746	{'was_impossible': False}
3	277427	Whirlwind (Tyler, Book 1)	1.826115	1.432492	{'was_impossible': False}
4	277427	Mystic River	1.826115	1.077684	{'was_impossible': False}
...	...	...	...	...	...
1233846	41700	Reversible Errors: A Novel	1.826115	1.877264	{'was_impossible': False}
1233847	41700	One Hundred Years of Solitude (Oprah's Book Club)	1.826115	1.415529	{'was_impossible': False}
1233848	41700	Passage	1.826115	2.105304	{'was_impossible': False}
1233849	41700	Blessings : A Novel	1.826115	1.057896	{'was_impossible': False}
1233850	41700	Ringworld	1.826115	2.180349	{'was_impossible': False}

1233851 rows × 5 columns

Comparing Recommendations

The outputs (above & below) shows the top 10 recommendations for the user 13552 using KNNWithMeans & SVD

Both algorithms recommended instances of Harry Potter novels for user 13552. Additionally, the recommended books seem to be a similar genre lending confidence in interpretability of recommendations

```
['The Lake House',
 'Harry Potter and the Chamber of Secrets (Book 2)',
 'Why Girls Are Weird : A Novel',
 'SKINNY LEGS AND ALL',
 '2nd Chance',
 'Round Ireland With a Fridge',
 "Harry Potter and the Sorcerer's Stone (Book 1)",
 'Summer Pleasures',
 'And Then There Were None : A Novel',
 'This Present Darkness',
 'Vittorio the Vampire: New Tales of the Vampires']
```

```
["Sabine's Notebook: In Which the Extraordinary Correspondence of Griffin &amp;amp; Sabine Continues",
 'Griffin &amp;amp; Sabine: An Extraordinary Correspondence',
 "Harry Potter and the Sorcerer's Stone (Book 1)",
 'The Darwin Awards: Evolution in Action',
 'The Lion, the Witch, and the Wardrobe (The Chronicles of Narnia, Book 2)',
 'Harry Potter and the Prisoner of Azkaban (Book 3)',
 'To Kill a Mockingbird',
 '84 Charing Cross Road',
 "Big Cherry Holler: A Big Stone Gap Novel (Ballantine Reader's Circle)",
 'The Book of Questions']
```

Conclusions & Next Steps

- We have successfully implemented a memory based as well as method based collaborative filtering approach to make recommendations in this project

- In instances with a new user or new item where little is known of the rating preference, collaborative filtering may not be the method of choice for generating recommendations. Content based filtering methods may be more appropriate. Often, a hybrid approach is taken for building real time recommendations using multiple different approaches in industry! The project can be extended to build hybrid recommendation systems in the future