Capstone Project-4

Book Recommendation System

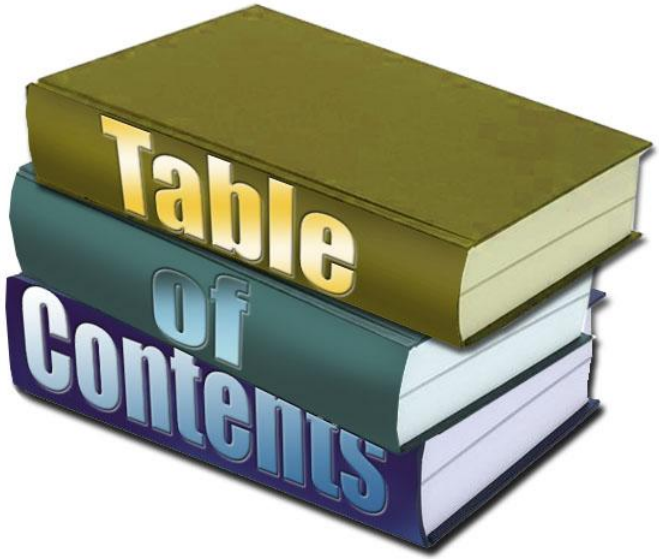
Team Members

i) Saurabh Yadav

ii) Shubham Deshmukh

Content

1. Problem Statement
2. Data Summary
3. Analysis of Data
4. Null value Imputation/ Data Cleaning
5. Data Preprocessing
6. Model Training
7. Evaluation Metrics
8. Challenges
9. Conclusion



Problem Statement:

- Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors.
- The objective of the project is to build a book recommendation system for users based on popularity and user interests.

Data Summary:

The Book-Crossing dataset comprises 3 files which Contains 278,858 users (anonymised but with demographic information) providing 1,149,780 ratings about 271,360 books.

Users_dataset.

- User-ID (unique for each user)
- Location (contains city, state and country separated by commas)
- Age

Shape of Dataset - (278858, 3)

Books_dataset.

- ISBN (unique for each book)
- Book-Title
- Book-Author
- Year-Of-Publication
- Publisher
- Image-URL-S
- Image-URL-M
- Image-URL-L
- Shape of Dataset - (271360, 8)

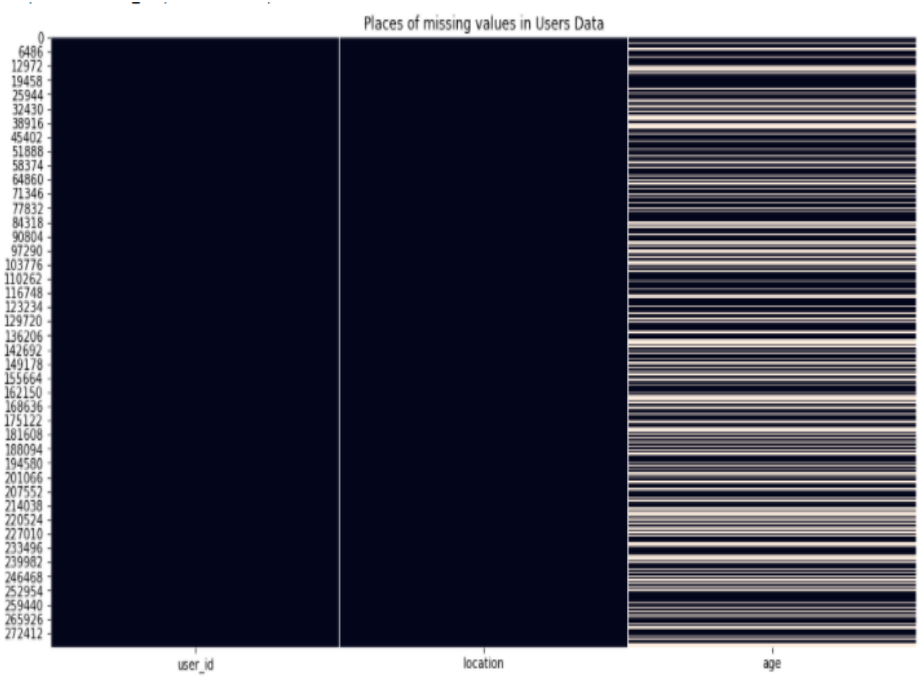
Ratings_dataset.

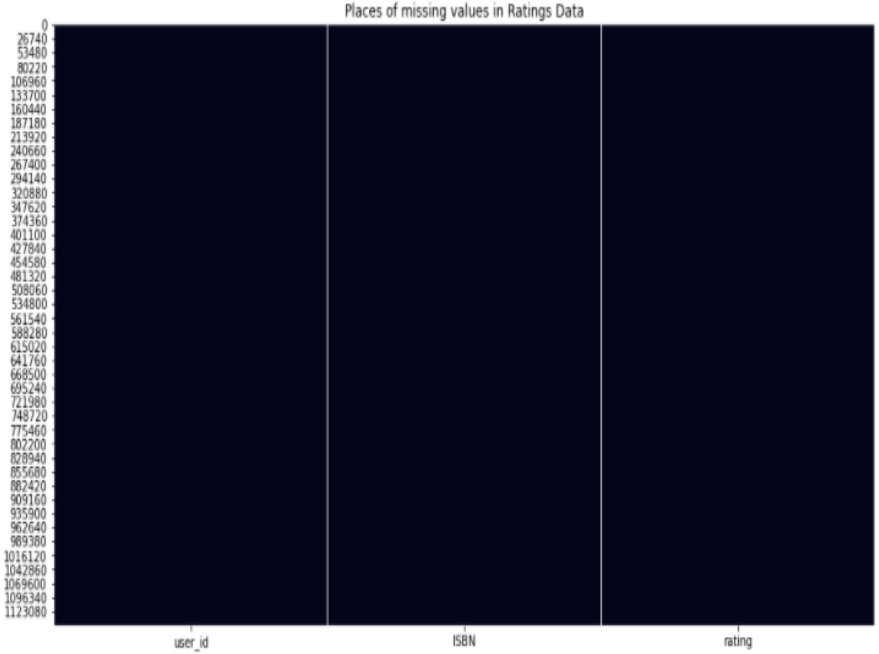
- User-ID
- ISBN
- Book-Rating
- Shape of Dataset - (1149780, 3)

Where are the missing values?

The book dataset ‘year’ column have **4690** missing values.

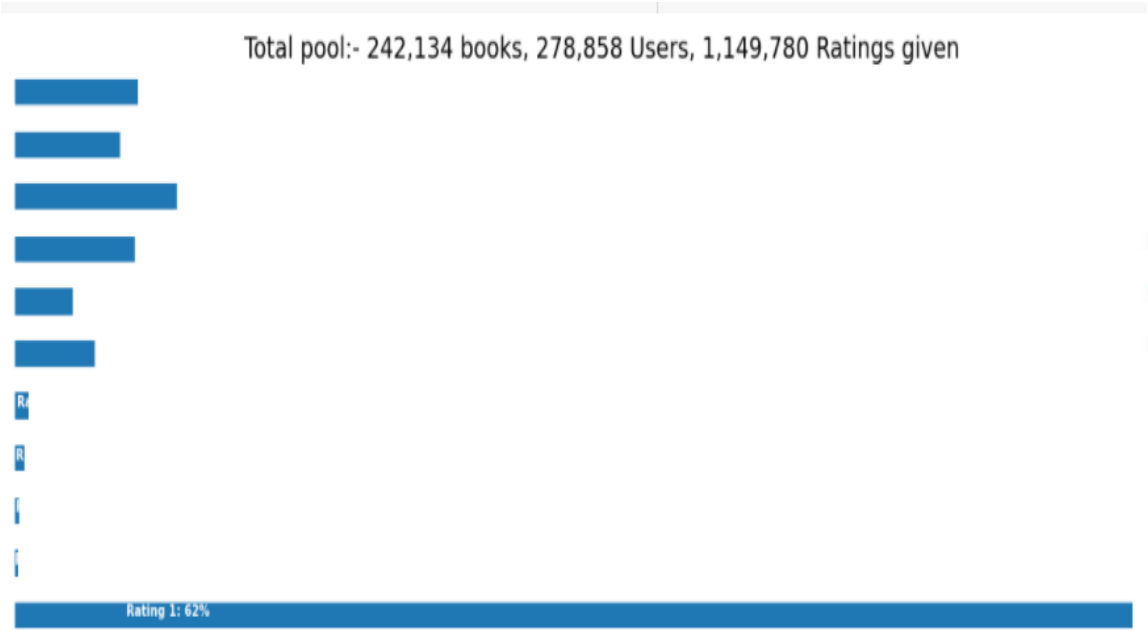
Contd..





The users dataset ‘Age’ column have missing values.
The ratings dataset don’t have missing values.

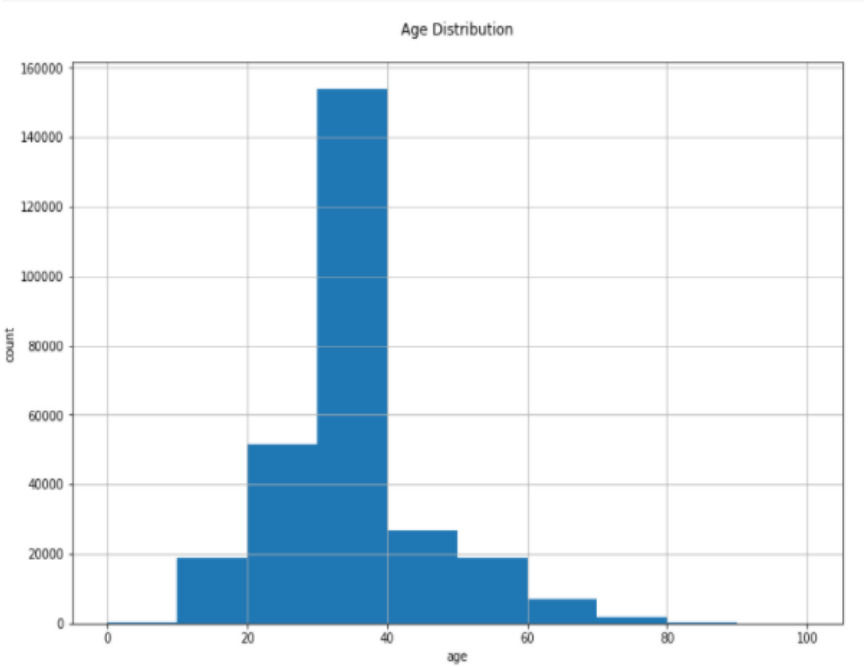
Visualizing the data

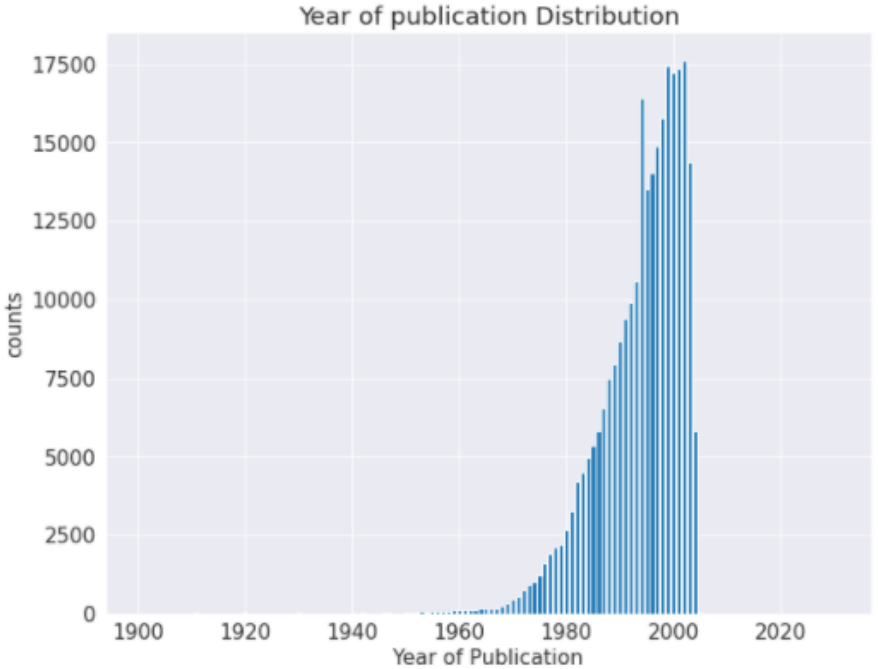


Here we can see that we have data

- 1)Number of books - 242134
- 2)Number of users -278858
- 3)Total Number of Ratings-1149780

Distribution of ‘age’ and ‘year of publication’ variable

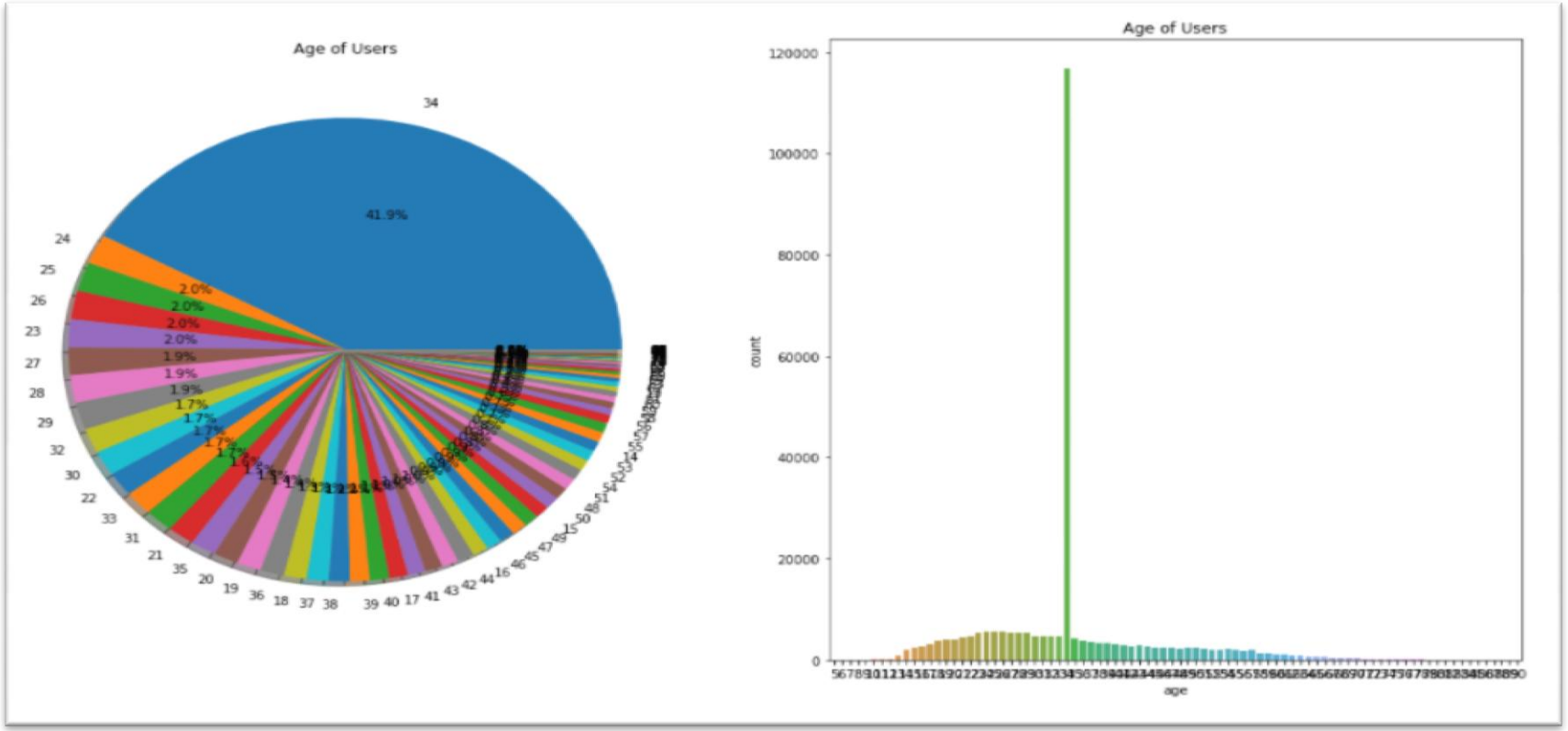




Most active readers lie in age group 20-40.

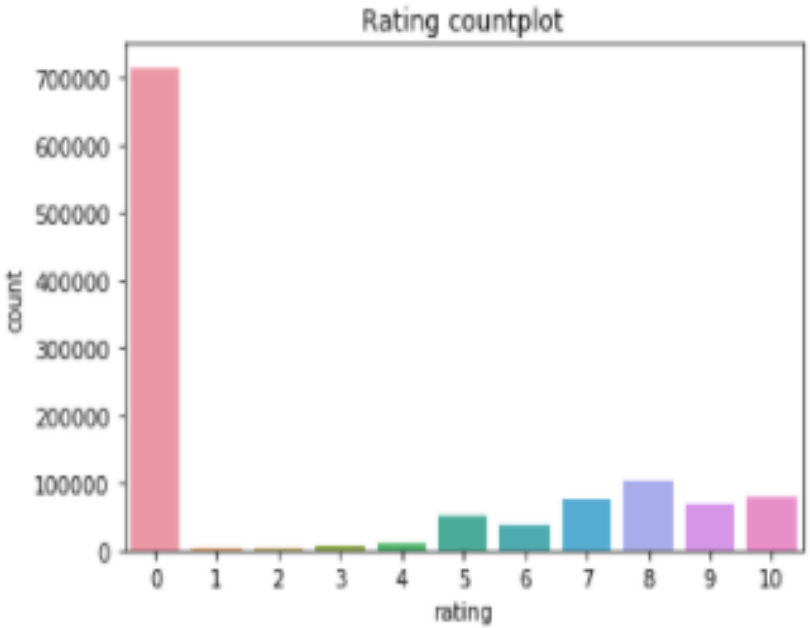
Publication years are somewhat between **1950 - 2005** here.

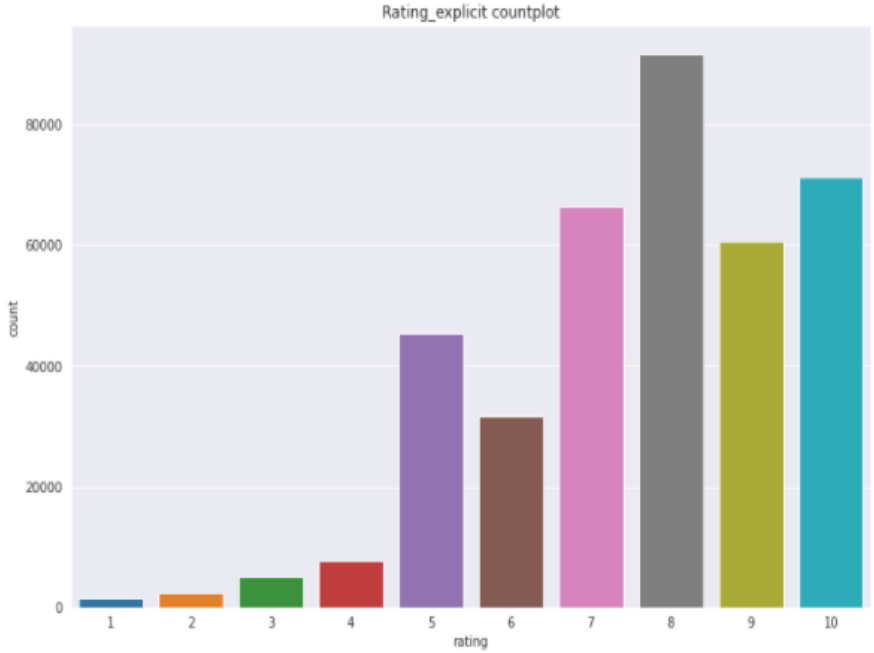
Contd..



We observed that 41.9% of age 34 group read more books compared to other age groups. Also the users with the age 60 and above do not read more books.

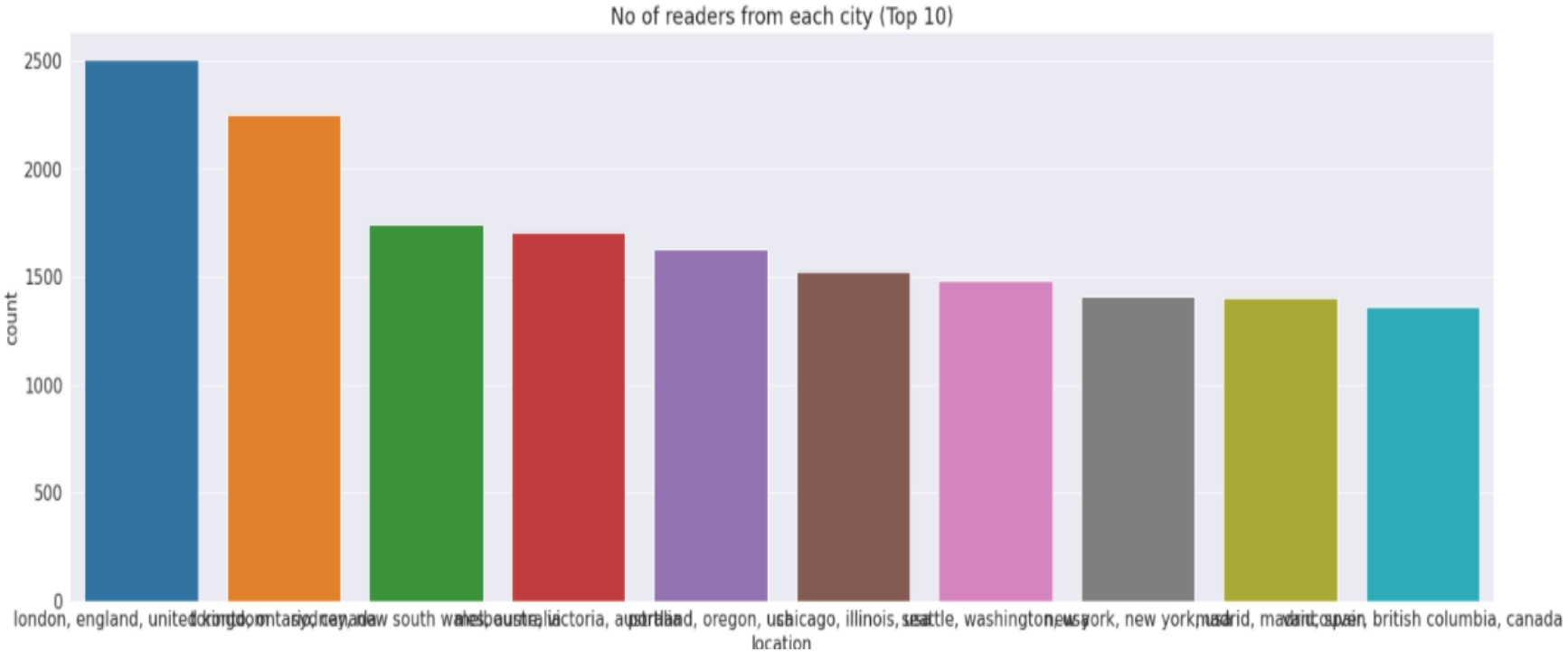
Analysis of Rating feature





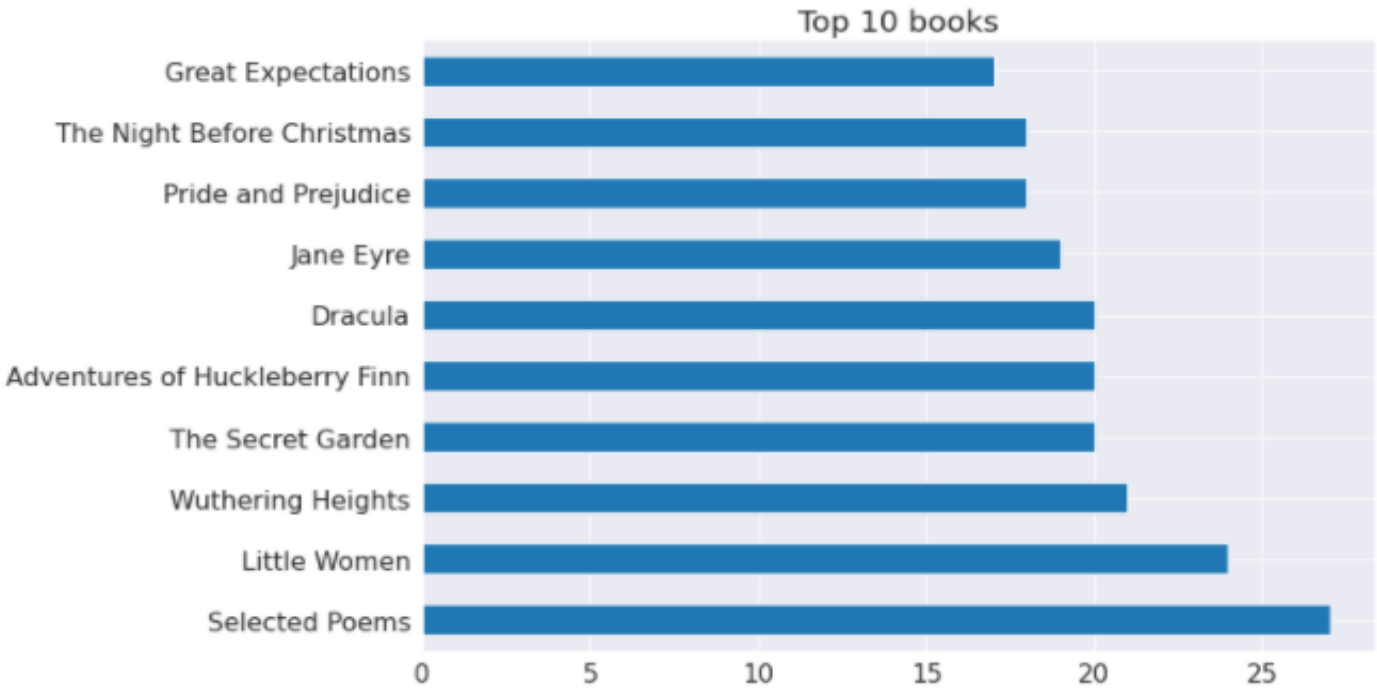
- First plot shows users have rated 0 the most, which can mean they haven't rated books at all.
- On second plot we have separated the explicit ratings represented by 1–10.

Exploring Locations of top Users:



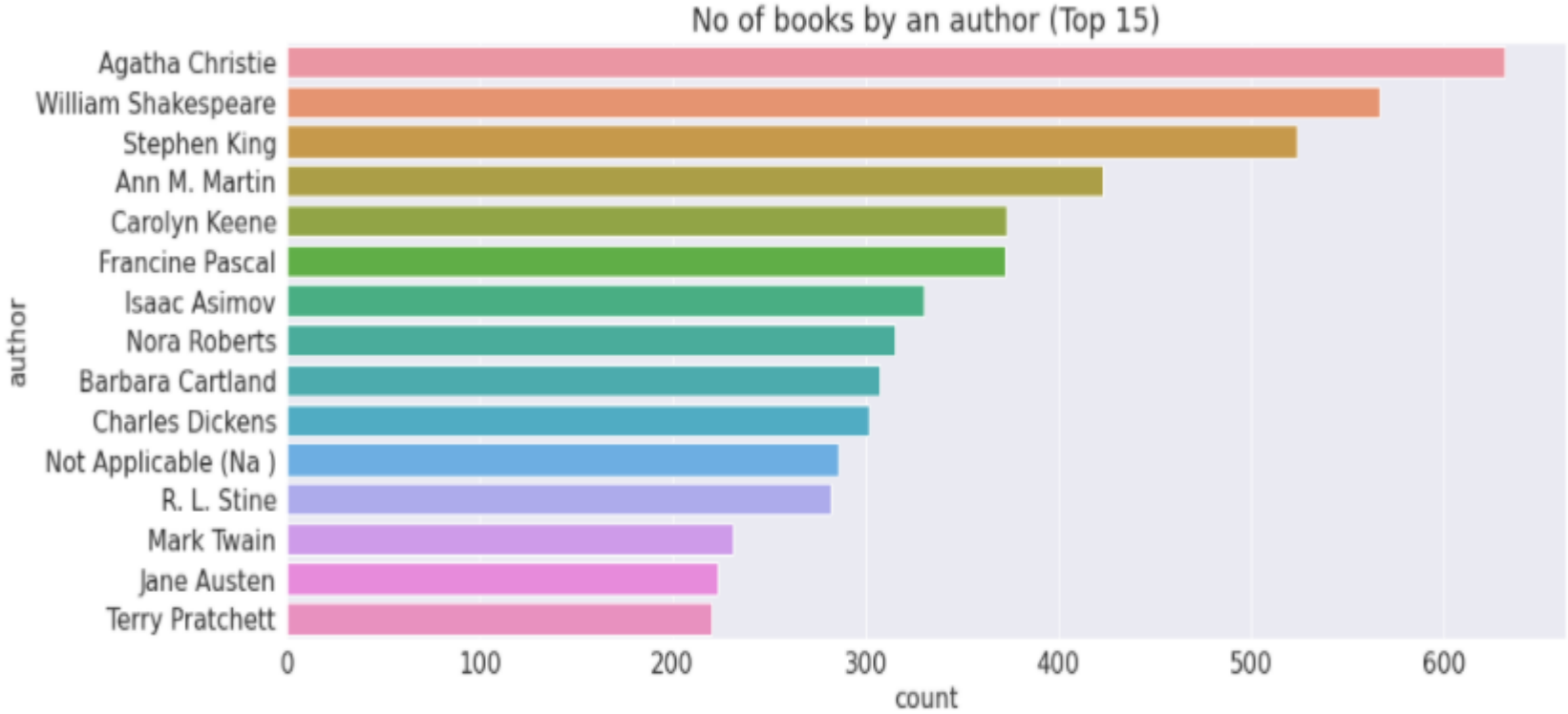
Here we can observe that user with locations London, england, united kingdom, Toronto, ontario, canda are high in numbers.

Exploring Top 10 Books:



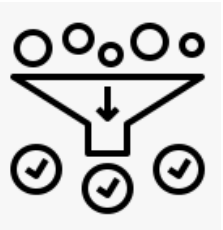
Bar plot shows us popular books.From above plot we can see that selected poems is highest in number.

Exploring Top Authors:



Agatha Christie wrote highest number of books in our given dataset.

Data Preprocessing:



- First, we've renamed the columns of each file. Because the name of the column contains space, and uppercase letters so we will correct as to make it easy to use.

Book Dataset:

- In the **books** dataset, we have some extra columns which are not required for our task like 'Image-URL-S', 'Image-URL-M' and 'Image-URL-L' so we exclude that column.
- In 'year' column it looks like 'publisher' names '**DK Publishing Inc**' and '**Gallimard**' have been incorrectly loaded as year in dataset due to some errors in csv file. Also some of the entries are strings and same years have been entered as numbers in some places.

	ISBN	title	author	year	publisher
209538	078946697X	DK Readers: Creating the X-Men, How It All Began (Level 4: Proficient Readers)\";Michael Teitelbaum"	2000	DK Publishing Inc	http://images.amazon.com/images/P/078946697X.01.THUMBZZZ.jpg
220731	2070426769	Peuple du ciel, suivi de 'Les Bergers\";Jean-Marie Gustave Le ClÃ?Â©zio"	2003	Gallimard	http://images.amazon.com/images/P/2070426769.01.THUMBZZZ.jpg
221678	0789466953	DK Readers: Creating the X-Men, How Comic Books Come to Life (Level 4: Proficient Readers)\";James Buckley"	2000	DK Publishing Inc	http://images.amazon.com/images/P/0789466953.01.THUMBZZZ.jpg

Contd..

Users dataset:

- In users data ‘age’ column has some invalid entries like nan, 0 and very high values like 100 and above. In our view values below **5** and above **90** do not make much sense for our book rating case...hence replacing these by NaNs.
- Then we replaced NaNs with mean of age.

Ratings Dataset:

- We‘ve only taken the ISBNs that also belongs to the main **books** set(as unique_rating).
- Also we’ve taken the rating from users which exist in **users** dataset, unless new users or book are added to users dataset.
- We have separated the explicit ratings represented by 1–10 and implicit ratings represented by 0. Because users who have rated 0 the most, which can mean they haven't rated books at all.

ratings dataset	unique_ratings dataset	explicit_ratings dataset
11,49,780 rows	10,31,135 rows	3,83,841 rows

Contd..

- For **Popularity based recommendation system** we only consider **ISBNs** that were explicitely rated than we **merge** ratings_explicit dataset with books dataset on **ISBNs features.** After that we grouped the data based on ‘title’ and aggregate based on ‘rating’.
- For **Collaborative Filtering using kNN(k-Nearest Neighbors)** to ensure statistical significance, users who has given less than 200 ratings, and books with less than 100 ratings are excluded.
- The dataset ‘**ratings'** and ‘**books'** have common column 'ISBN‘, so we create new data-frame by merging the two data-frames on ISBN (as **book_with_rating**) than we group by book titles and create a new column for total rating count and store in new data-frame **rating_count_df** also done the renaming on rating column as total_ratings.

Contd..

- Now we will combine the books_with_rating with the rating_count_df data, this gives us exactly what we need to find out which books are popular and filter out lesser-known books.
- Then we've considered only books having minimum total 50 ratings(Threshold) by creating new data-frame as **rating_popular_book** after that we've merged the **'users**' dataframe with **'rating_popular_book**' dataframe and stored them into '**combined'** data-frame.
- Also we've dropped the duplicate values from '**combined'** data-frame.

	user_id	ISBN	rating	title	author	year	publisher	total_ratings	location	age
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Tales for Our Life and Times	James Finn Garner	1994	John Wiley & Sons Inc	82	gilbert, arizona, usa	48
1	3363	002542730X	0	Politically Correct Bedtime Stories: Modern Tales for Our Life and Times	James Finn Garner	1994	John Wiley & Sons Inc	82	knoxville, tennessee, usa	29
2	11676	002542730X	6	Politically Correct Bedtime Stories: Modern Tales for Our Life and Times	James Finn Garner	1994	John Wiley & Sons Inc	82	n/a, n/a, n/a	34
3	12538	002542730X	10	Politically Correct Bedtime Stories: Modern Tales for Our Life and Times	James Finn Garner	1994	John Wiley & Sons Inc	82	byron, minnesota, usa	18
4	13552	002542730X	0	Politically Correct Bedtime Stories: Modern Tales for Our Life and Times	James Finn Garner	1994	John Wiley & Sons Inc	82	cordova, tennessee, usa	32

Contd..

- For **SVD(Singular Value Decomposition) Based recommendation System** we've used **Surprise** (it is a Python sci-kit for building and analyzing recommender systems that deal with explicit rating data.)
- The name **Surprise** (roughly) stands for Simple Python Recommendation System Engine.
- From surprise we imported '**Reader'** to set the limit of rating (in our case 1 to 10) & '**Dataset'** to load **ratings_explicit** data.

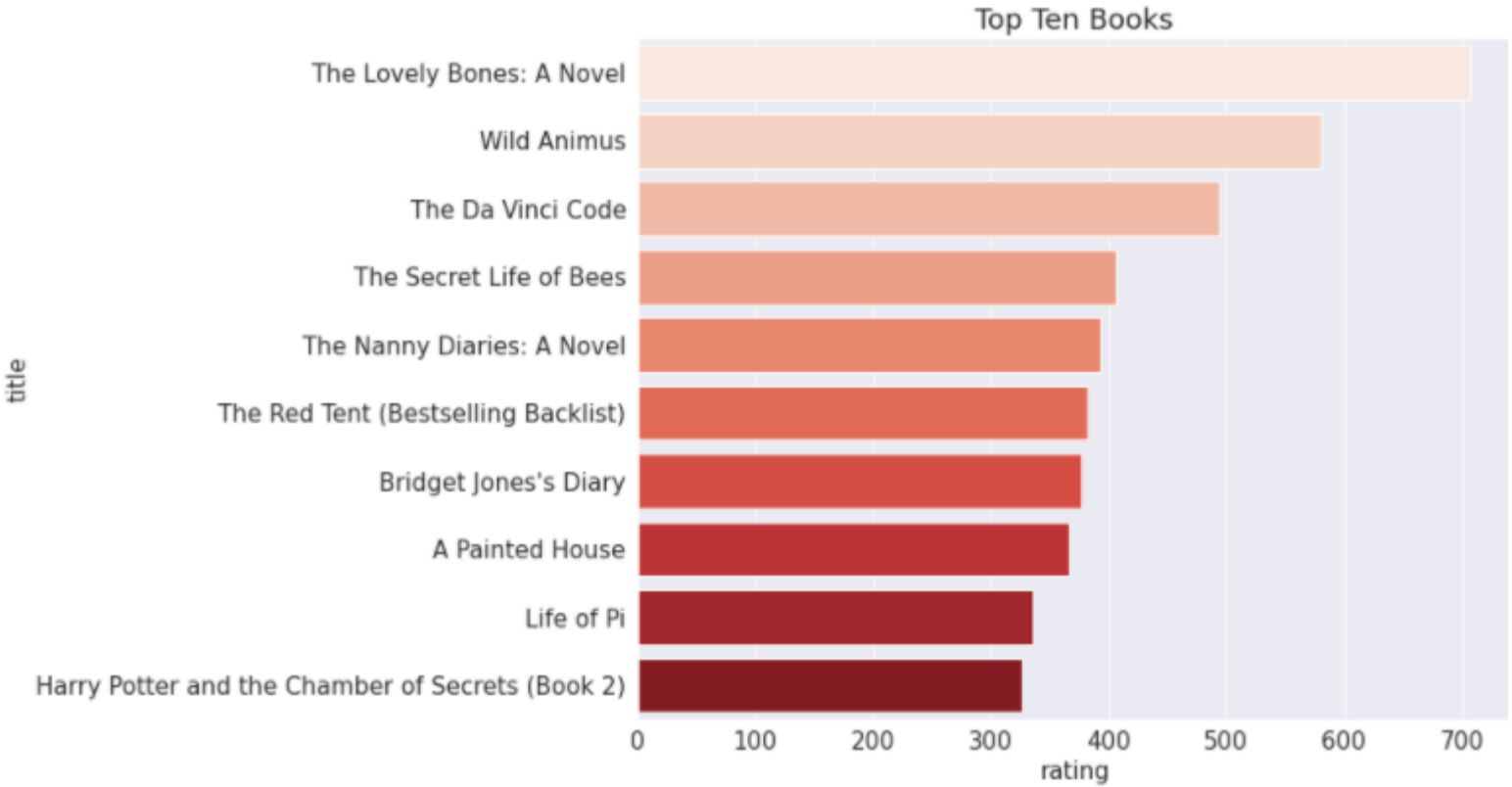
Model building, Predictions:

Here we've build 3 models for Book Recommendation :

1. **Popularity Based Recommendation(Base Model):** These model check about the product or movie which are in trend or are most popular among the users and directly recommend those.
2. **Collaborative Filtering Using KNN (k-Nearest Neighbors):** kNN is a machine learning algorithm to find clusters of similar users based on common book ratings, and make predictions using the average rating of top-k nearest neighbors.
3. **SVD(Singular Value Decomposition) Based recommendation System:** The **SVD(Singular Value Decomposition)** is used as a collaborative filtering technique. It uses a matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.

Predictions

1. Popularity Based Recommendation(Base Model)



2. Collaborative Filtering Using KNN (k-Nearest Neighbors)

- **Best parameters:**
  {algorithm='brute', leaf_size=30, metric='cosine', metric_params=None, n_jobs=None, n_neighbors=5, p=2, radius=1.0}

Recommendation for random book:

```
Recommendations for Cat's Cradle:

1: Slaughterhouse Five or the Children's Crusade: A Duty Dance With Death, with distance of 0.9103445085990503:
2: Catch 22, with distance of 0.9135549316310942:
3: Cryptonomicon, with distance of 0.9187612511299216:
4: Stargirl, with distance of 0.9218718063164879:
5: The Two Towers (The Lord of the Rings, Part 2), with distance of 0.9286713029531773:
```

Contd..

Recommendation for known book from dataset:

```
Recommendations for 1984:

1: Animal Farm, with distance of 0.8441751059913439:
2: Brave New World, with distance of 0.8630224130863633:
3: The Vampire Lestat (Vampire Chronicles, Book II), with distance of 0.9098446731344783:
4: Slaughterhouse Five or the Children's Crusade: A Duty Dance With Death, with distance of 0.9102283484822355:
5: American Psycho (Vintage Contemporaries), with distance of 0.9126168848307201:
```

3. SVD(Singular Value Decomposition) Based recommendation System:

Performed Cross Validation on **SVD** by providing parameters as:

(model, data, measures=['RMSE'], cv=5, verbose=True)

```
Evaluating RMSE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    1.6360  1.6411  1.6271  1.6421  1.6378  1.6368  0.0053
Fit time          21.28   21.52   21.45   21.40   21.64   21.46   0.12
Test time         1.03    0.73    0.78    0.80    0.74    0.82    0.11
CPU times: user 1min 59s, sys: 725 ms, total: 2min
Wall time: 2min
{'fit_time': (21.282174825668335,
  21.521875381469727,
  21.45281481742859,
  21.399600744247437,
  21.64139199256897),
 'test_rmse': array([1.63600669, 1.64105883, 1.62710693, 1.64208561, 1.63784546]),
 'test_time': (1.0336084365844727,
  0.7289626598358154,
  0.7829821109771729,
  0.8042428493499756,
  0.7400891780853271)}
```

Contd..

Recommendation by giving user-id as input:

On picking a random user_id = 116866 our model recommend this books

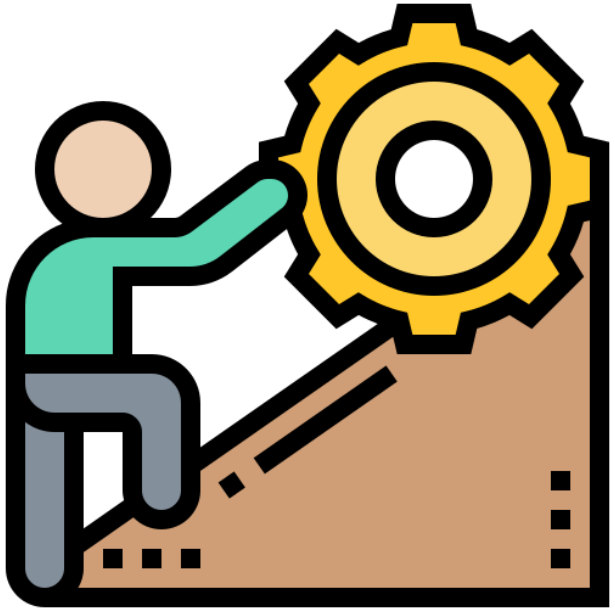
```
October Light: 7.815437160287302
Lucy: The Beginnings of Humankind: 7.815437160287302
An Introduction to Stochastic Modeling: 7.815437160287302
The Biosphere.: 7.815437160287302
Algebra and Trigonometry, Unit Circle (6th Edition): 7.815437160287302
River Why: 7.815437160287302
Excel: 7.815437160287302
The Progress of Love (King Penguin): 7.557823608083711
Metamagical Themas: Questing for the Essence of Mind and Pattern: 7.423558267082783
```

Evaluation of models:

- **kNN (k-Nearest Neighbor)** model gives the cosine distance value near to 1 for all book that are recommended.
- In **SVD** on an average the Root Mean Square Error(RMSE) for our test data set prediction is near about **1.64** which is pretty good.

Challenges:

- Understanding the metric for evaluation was a challenge as well.
- Decision making on missing value imputations quite challenging.
- Handling of sparsity was a major challenge.



Conclusion:

- Recommendation system is unturned to exist in the e-commerce businesses with the help of collaborative or content-based filtering to predict different items and yes, users are most satisfied with the products recommended to them.
- While performing Exploratory Data Analysis we observed that almost **42%** of readers with **age-34** read more books compared to other age group of readers.
- Books with publication years are somewhat between **1950 - 2005**.
- Also the readers mostly give 8 ratings(on scale 1-10) to books followed by 10 and 7.
- There are more readers from locations London, england, united kingdom, toronto, ontario, canada compare to other locations.
- KNN model gives good recommendation for books.
- The best collaborative book recommender model is **SVD(Singular value decompostion)** with best accuracy on test data which give stronger recommendations.
- We can deploy this model.

Contributors Role

Saurabh Yadav:

- Performed Data preprocessing and **EDA** (Exploratory Data Analysis)
- Build **Popularity Based Recommendation** and **SVD(Singular Value Decomposition) Based recommendation System** model.

Shubham Deshmukh:

- Performed Data preprocessing and **EDA** (Exploratory Data Analysis)
- Build **Collaborative Filtering Using KNN (k-Nearest Neighbors)** model.

Thank You Q &A