Capstone Project
Book Recommendation System

Sandesh Gaikwad

Content

- Problem Statement
- Exploratory Data Analysis
- Feature Engineering
- Modeling and comparing results
- Conclusions

Problem Statement

Problem :The main objective is to create a recommendation system to recommend relevant books to users based on popularity and user interests.

During the last few decades, with the rise of Youtube, Amazon, Netflix, and many other such web services, recommender systems have become much more important in our lives in terms of providing highly personalized and relevant content

Data Summary

- The dataset is comprised of three Tables:- User, Books, Ratings
- User :- Shape of Dataset - (278858, 3)
  - User-ID (unique for each user)
  - Location (contains city, state and country separated by commas)
  - Age
- Books_dataset :- Shape of Dataset - (278858, 3)
  - ISBN (unique for each book)
  - Book-Title
  - Book-Author
  - Year-Of-Publication
  - Publisher
  - Image-URL-S
  - Image-URL-M
  - Image-URL-L

Data Summary

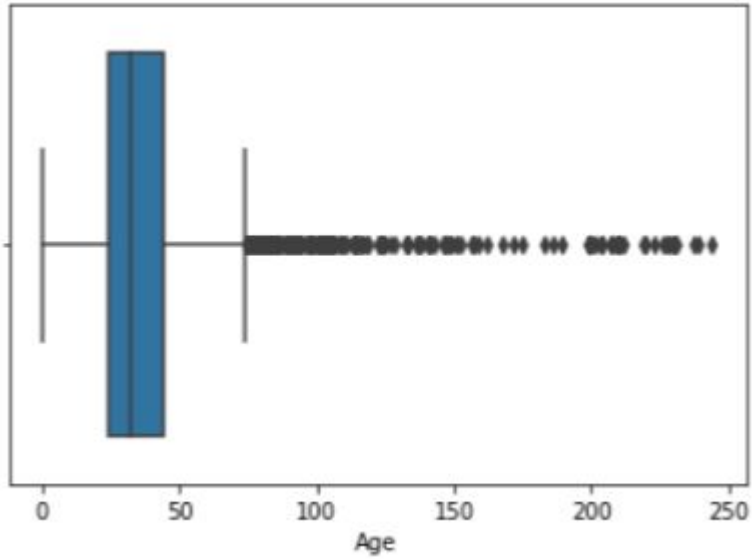
- Ratings_dataset- Shape of Dataset - (1149780, 3)
  - User-ID
  - ISBN
  - Book-Rating

Data cleaning

- We don’t need publisher , image url so we drop all the column related to that
- There are some errors with the values in column we will use google search to find and fix the few missing rows
- Age column has 40% of null values but we don’t need age column so we don’t need to impute values for age

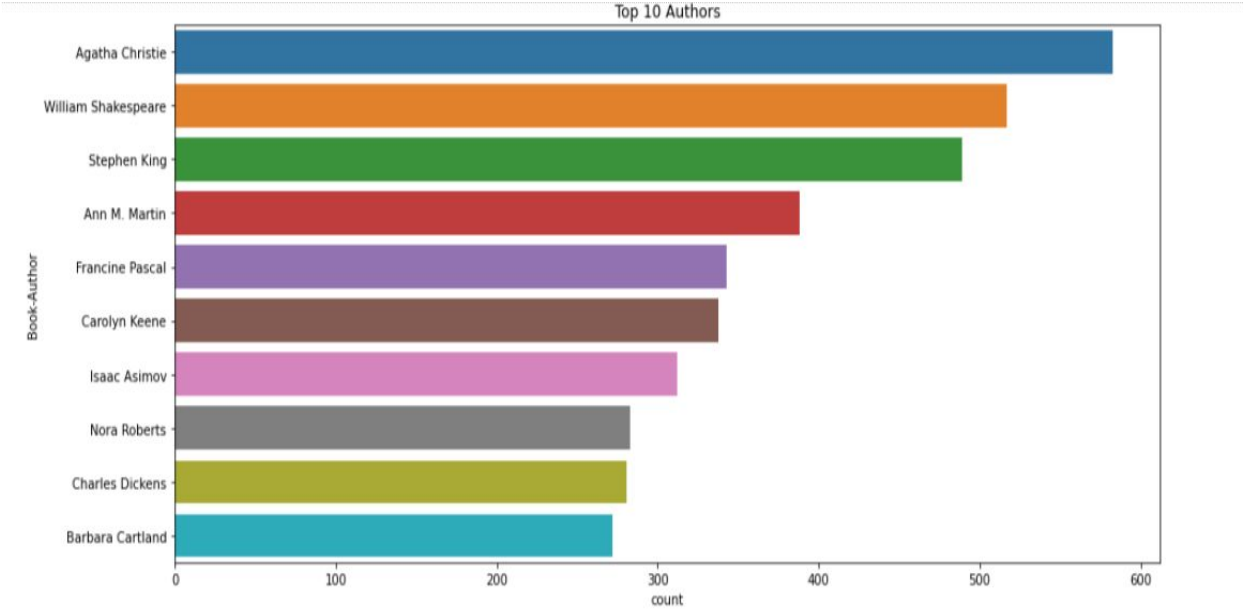
Age

- Age has positive Skewness (right tail) so we can use median to fill Nan value
- Majority of the readers were of the age bracket 20-40
- People above age of 100 doesn’t make sense we will replace these values with max value of 80



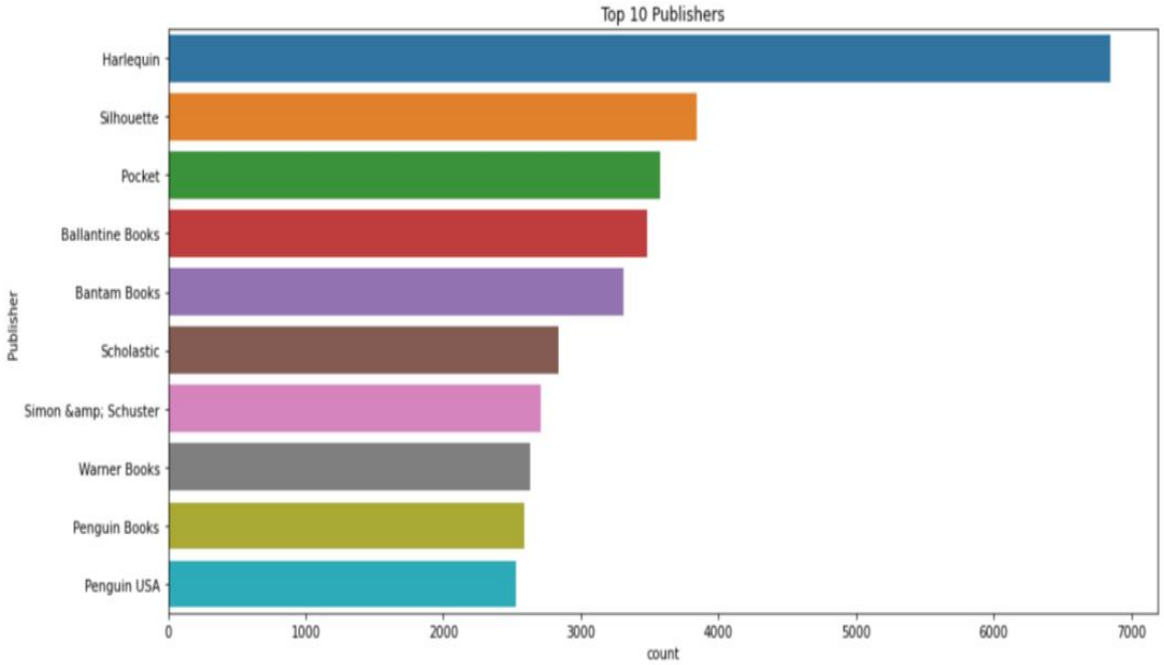
Top Authors

- Agatha Christie, William shakespeare, Stephen king Ann M. Martine are top authors



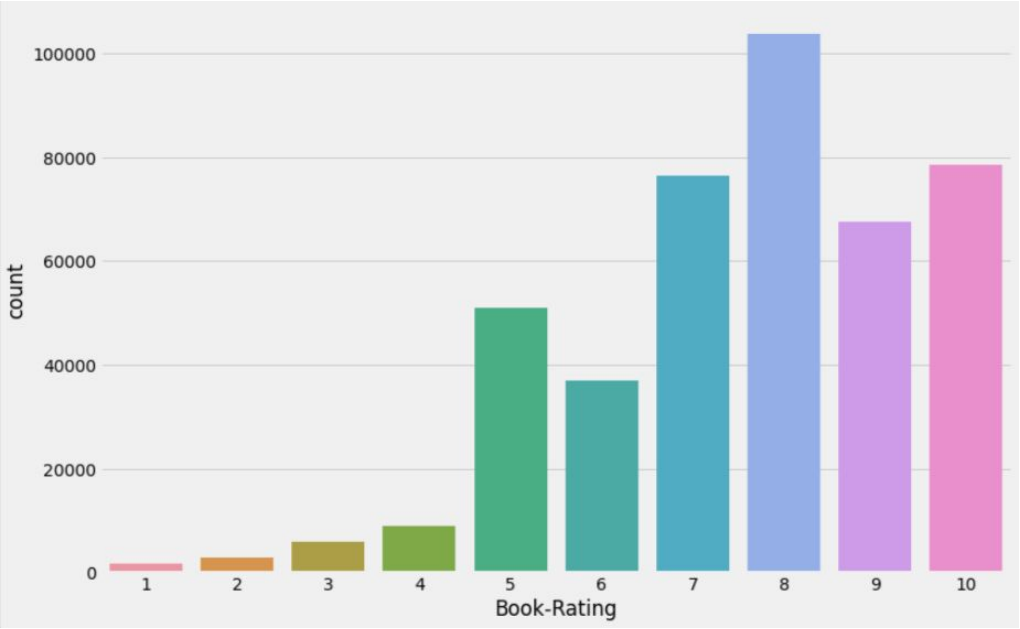
Top Publishers

- Harlequin , silhouette, pocket, ballantine books, Scholastic are top publishers



Ratings

- Most user have given 5 plus ratings
- 8 is the highest rated value



Feature Engineering

- We will create new weighted rating for the books

- Wr =(vR + mC) / (v + m)
  - v is the number of votes for books
  - m is the minimum votes required to be listed in the chart
  - R is the average rating of the book
  - C is the mean vote across the whole report

Popularity Based Recommendations

- We will only consider top 5 % books. Using the weighted rating we recommend top rated books to all the new users.

Book name	Total ratings	Avg rating	Weighted ratings
To Kill a Mockingbird	214	8.944	8.353
The Da Vinci Code	487	8.435	8.230
The Secret Life of Bees	307	8.452	8.170
The Lovely Bones: A Novel	707	8.185	8.187
The Red Tent (Bestselling Backlist)	383	8.183	8.032

Collaborative Filtering

- **Memory based** -
  - This approach uses the memory of previous users interactions to compute users similarities based on items they've interacted (user-based approach) or compute items similarities based on the users that have interacted with them (item-based approach).

- **Model Based** -
  - In this approach, models are developed using different machine learning algorithms to recommend items to users.
  - There are many model-based CF algorithms, like Neural Networks, Bayesian Networks, Clustering Techniques, and Latent Factor Models such as Singular Value Decomposition (SVD) and Probabilistic Latent Semantic Analysis.

Matrix Factorization

- Latent factor models compress the user-item matrix into a low-dimensional representation in terms of latent factors.
- One advantage of using this approach is that instead of having a high-dimensional matrix containing an abundant number of missing values we will be dealing with a much smaller matrix in lower-dimensional space.
- An important decision is choosing the number of factors to factor the user-item matrix. The higher the number of factors, the more precise is the factorization in the original matrix reconstructions.
- If the model is allowed to memorize too much details of the original matrix, it may not generalize well for data it was not trained on. Reducing the number of factors increases the model generalization.

Evaluation Method

- Take each user each item the user has interacted in test set
- Sample 100 other items the user has never interacted.
- Ask the recommender model to produce a ranked list of recommended items, from a set composed of one interacted item and the 100 non-interacted items
- Compute the Top-N accuracy metrics for this user and interacted item from the recommendations ranked list
- Aggregate the global Top-N accuracy metrics

Evaluation results Conclusion

- Hits in top 5 - 26 %
- Hits in top 10 - 36 %

- The results are pretty decent as we are using only svd based recommender

Challenges and Scope

- High volume of data
- We need to build more complex model for recommender system to improve the accuracy
- The model should account age , location and we can ask the user for interest in genre to further narrow down the results