Book Recommendation System- Milestone Report

Problem Statement:

Today, the e-commerce industry is growing at an unprecedented rate. Thousands of items are being added to the e-catalog every month and the contents available for users to explore are overwhelming, hence there is a need to shift and prioritize user preferences by analyzing relevant information on items and users. Most of the big e-commerce companies started its online marketplace by selling books but later diversified to sell electronics, apparels, food, toys and jewelry. Success of online book sales has made great impact to the e-commerce industry and new machine learning techniques are being introduced to drive the sales up north.

One of the techniques which can make meaningful impact to online book sales is the Recommendation system. A Recommendation system helps users to discover items that they may like. The system has become fundamental applications in e- commerce websites by providing suggestions that effectively snip large information so that users are directed toward those items that best meet their needs and preferences. The main goal is to build the system which can make a best suggestion of books.

Datasets:

Book-Crossings dataset which has been publicly shared from University of Freiburg related website was utilized for the project. Book-Crossings is a book ratings dataset compiled by Cai-Nicolas Ziegler based on data from bookcrossing.com. It contains 1.1 million ratings of 270,000 books by 90,000 users. The ratings are on a scale from 1 to 10, and implicit ratings are also included. The dataset consists of three tables namely BX-Users, BX-Books and BX-Book-ratings which effectively divides users, books and ratings data in separate tables.

The BX_Books table has information of book title along with author information for each ISBN. Additionally, it contains book image URL information.

The BX-Users table contains demographic information of book users.

The BX-Book-Ratings table consists of user rating information along with User ID and ISBN to identify each book and its respective user.

Data Cleaning:

Books data:

1) Upon reviewing the data type for each column in the books table, it was found that the column ‘Year_of_Publication’ had incorrectly captured publisher information for couple of records having ISBN# 078946697X, 0789466953 and 2070426769. The records for the following ISBN#s were updated by reviewing the data.
2) As the data was compiled in the year 2004, any information of books having ‘Year_of_Publication’ beyond 2004 was replaced by the mean value of publication year.

Additionally, Year_of_Publication' having values '0' and NAN were also replaced by the mean value.

Users Data:

1) The location column having concatenated information of State, City and Country were split in to three separate columns so that demographic information of the user was better utilized.
2) The User table didn't include Age information for all the Users. Additionally, the Age column had some values which were above 100 years and below 5 years. In order to capture the right tech-savvy population, Users having age less than 5 years were set to 5 years and age above 80 years were set to 80 years. Missing values were replaced by the mean age of the population.

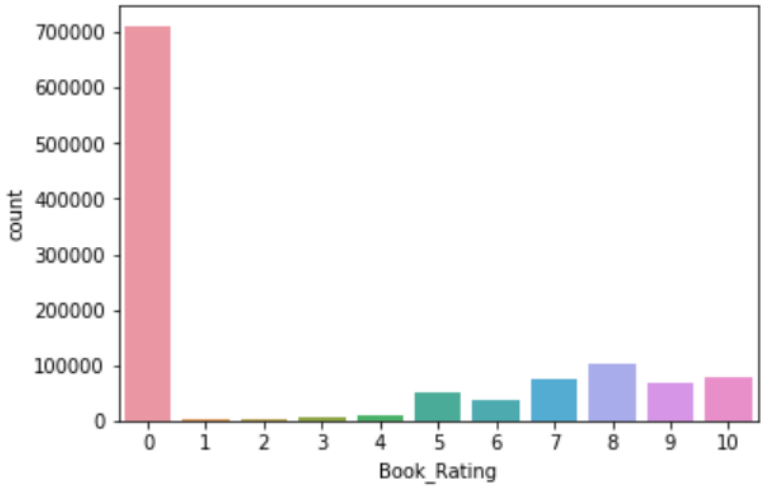
Book ratings data:

1) Ratings table had ratings for books which were not part of books table. A new ratings DataFrame was created to capture only those books and users which are part of other tables.

Data Visualization:

In order to capture more insights from the data, a count plot was plotted to capture how the ratings are distributed across all the books.

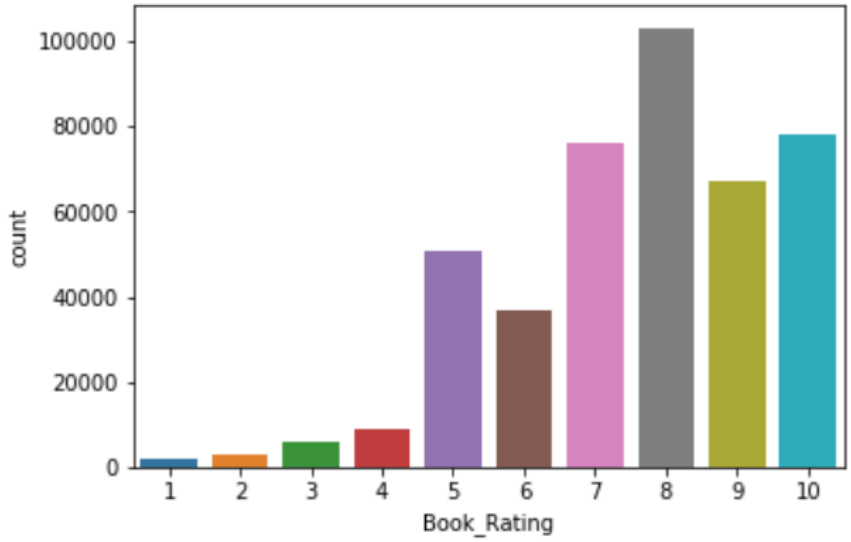
```
#Firstly, we will analyze how the ratings are distributed across all the books.
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(new_ratings.Book_Rating)
plt.show()
```



From the countplot, it was clear that the implicit and explicit have unequal representation in the ratings table with books having around 700k implicit ratings recorded out of 1.14M ratings.

Since the focus was on the explicit ratings to model the recommendation system, count plot was plotted to check how it is distributed across.

In [62]:
```
sns.countplot(explicit_ratings.Book_Rating)
plt.show()
explicit_ratings.shape
```



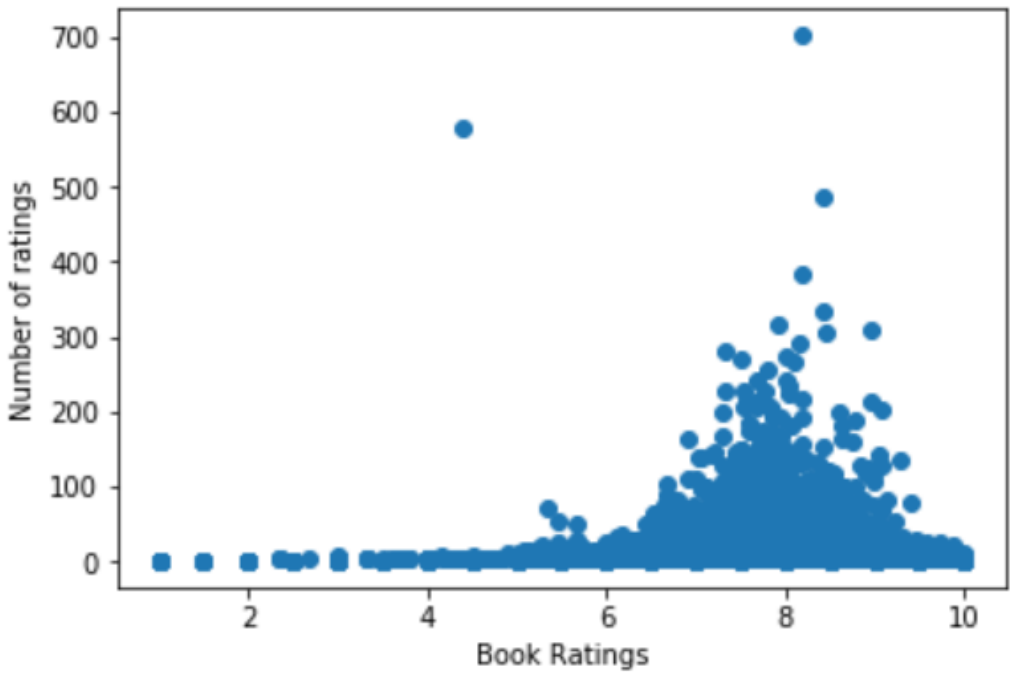
Out[62]: (430913, 3)

From the plot, it was clear that majority of the books had favorable rating which is greater 7.

Correlation:

The number of ratings and average ratings for each book was analyzed to check how the ratings correlated.

A scatter plot was plotted to check the distribution of data points.




```
Pearsons correlation: 0.022
```

From the plot, it was clear that there was no correlation between the variables. The Pearsons correlation co-efficient backed the assumption by calculating a low co-efficient value of 0.022 for the two variables.