BOOK RECOMMENDATION SYSTEM

ROBOTICS AND AUTOMATION

Submitted by

VINAY MALKAR – 21BEC1430

SUBHAYU KABIRAJ - 21BEC1757

SHAILENDRA KUMAR – 21BEC1486

To

Dr. PRITAM BHATTACHARJEE SIR

ELECTRONICS AND COMMUNICATION ENGINEERING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRONICS ENGINEERING

Introduction :

In today's fast-paced world, where information is abundant and easily accessible, the task of finding the right book to read can sometimes feel overwhelming. Whether you're a casual reader looking for a new novel to dive into or a student searching for academic resources, the sheer volume of available books can make the selection process daunting. This is where the importance of book recommendation systems comes into play. By leveraging the power of machine learning and data analytics, these systems aim to provide personalized recommendations tailored to individual preferences and interests, thereby helping users discover new books that align with their tastes.

Research Background :

The concept of recommendation systems isn't new and has been extensively studied and applied in various domains, including e-commerce, music streaming, and movie recommendations. However, with the ever-expanding digital library of books and the rise of online platforms for book browsing and purchasing, there's a growing need for effective book recommendation systems. Traditional approaches to book recommendations often relied on manual curation, user reviews, or bestseller lists. While

these methods have their merits, they may not always capture the diverse and evolving preferences of readers accurately.

In recent years, advancements in machine learning algorithms, coupled with the availability of large-scale book datasets, have paved the way for more sophisticated recommendation systems. These systems can analyze vast amounts of data, including book metadata, user ratings, reading history, and even textual content, to generate personalized recommendations. By understanding the underlying patterns and preferences of readers, these systems can suggest books that are not only relevant but also likely to be enjoyed by the user.

Motivation :

The motivation behind developing a book recommendation system stems from the desire to enhance the book browsing and discovery experience for readers. With the proliferation of digital platforms and online bookstores, readers are often inundated with choices, making it challenging to navigate through the vast sea of available titles. A well-designed recommendation system can alleviate this burden by presenting users with tailored suggestions based on their unique reading habits, genre preferences, and past interactions.

Moreover, from a business perspective, book recommendation systems hold immense potential for online retailers and publishers. By facilitating book discovery and increasing user engagement, these systems can drive sales and customer satisfaction. Additionally, by leveraging data analytics, retailers can gain valuable insights into consumer behavior and market trends, enabling them to optimize their inventory, marketing strategies, and promotional campaigns.

Problem Statement :

Despite the advancements in recommendation algorithms and data analytics techniques, building an effective book recommendation system presents several challenges and complexities. One of the primary challenges is the inherent subjectivity and diversity of reading preferences among users. Unlike movies or music, which may have more standardized genres or categories, books span a wide range of topics, styles, and formats, making it harder to capture individual tastes accurately.

Another challenge lies in the availability and quality of data. While large-scale book datasets are increasingly becoming accessible, they may suffer from issues such as incomplete metadata, biased ratings, or sparse user interactions. Moreover,

the cold-start problem, where new books or users lack sufficient data for accurate recommendations, poses a significant hurdle for recommendation systems.

Furthermore, the scalability and computational complexity of recommendation algorithms can be limiting factors, especially for real-time or large-scale applications. As the volume of available books and users continues to grow, the need for scalable and efficient recommendation models becomes more pronounced.

Addressing these challenges requires a multidisciplinary approach that combines expertise in machine learning, data engineering, and domain knowledge of the publishing industry. By developing innovative algorithms, leveraging rich data sources, and incorporating user feedback mechanisms, it's possible to create robust and adaptive book recommendation systems that enhance the reading experience for users while driving business value for stakeholders.

Methodology

1. Data Understanding and Preprocessing

The project begins with a comprehensive exploration of the dataset, requiring the importation of essential libraries and the examination of the dataset titled books.csv. This dataset provides extensive information about various books, encompassing attributes such as the unique identification number for each book (bookID), the title under which the book was published, the authors' names, the average rating received by the book, the International Standard Book Number (ISBN) and its 13-digit counterpart (ISBN13) used for book identification, the primary language in which the book is written, the number of pages contained within the book, the total count of ratings received by the book, the number of written reviews it has garnered, the publication date marking its initial release, and finally, the name of the publisher responsible for bringing the book to print.

2. Exploratory Data Analysis (EDA)

Following the initial data understanding phase, the next step involves delving into Exploratory Data Analysis (EDA). This crucial phase entails a deeper examination of the dataset to uncover patterns, trends, and potential insights that may inform subsequent stages of the project. Through visualizations, statistical summaries, and other analytical techniques, EDA facilitates a nuanced understanding of the dataset's characteristics, distributions, and relationships between variables. Moreover, EDA serves as a vital precursor to feature engineering and model development, guiding the selection of relevant features and informing the construction of predictive models.

3. Feature Engineering

Feature engineering constitutes a pivotal aspect of the methodology, focusing on the creation and selection of informative features that enhance the predictive power of the recommendation system. This stage involves transforming raw data into meaningful representations that capture essential aspects of books and user interactions. Feature engineering techniques may encompass text processing for extracting key information from book titles, author names, and textual content,

as well as numerical transformations to standardize or scale numerical attributes such as ratings, page counts, and review counts. Additionally, feature engineering may involve the creation of novel features derived from domain knowledge or external data sources, enriching the dataset with supplementary information relevant to book recommendations.

4. Model Selection and Evaluation

With the curated dataset and engineered features in hand, the subsequent phase revolves around model selection and evaluation. This stage entails the exploration of various recommendation algorithms, ranging from traditional collaborative filtering methods to state-of-the-art deep learning architectures. The selection of an appropriate model depends on factors such as the nature of the data, the computational resources available, and the desired balance between accuracy and scalability. Once candidate models are identified, rigorous evaluation procedures are employed to assess their performance in terms of recommendation quality, computational efficiency, and robustness to diverse user preferences and scenarios. Evaluation metrics such as precision, recall, and mean average precision are commonly employed to quantify the effectiveness

of recommendation models across different evaluation settings and datasets.

5. Model Deployment and Integration

Following the selection and evaluation of the recommendation model, the final phase involves model deployment and integration into real-world applications or platforms. This step necessitates the seamless integration of the recommendation system with existing infrastructure, ensuring compatibility with web interfaces, mobile applications, or other user-facing interfaces. Additionally, considerations such as scalability, latency, and user privacy must be addressed to deliver a responsive and user-friendly experience. Furthermore, ongoing monitoring and maintenance are essential to ensure the continued performance and relevance of the recommendation system in response to evolving user behavior and preferences.

6. Continuous Improvement and Optimization

The deployment of the recommendation system marks the beginning of an iterative process of continuous improvement and optimization. Through user feedback mechanisms, A/B testing,

and monitoring of key performance indicators, insights are gleaned to refine the recommendation algorithms, enhance user experience, and drive business outcomes. This iterative approach enables the recommendation system to adapt to changing user preferences, incorporate new books or features, and mitigate potential biases or shortcomings in the underlying data or models. Moreover, ongoing optimization efforts may involve leveraging advanced techniques such as reinforcement learning or multi-armed bandit algorithms to dynamically adjust recommendations in real-time based on user interactions and feedback.

7. Ethical Considerations and Bias Mitigation

In the development and deployment of book recommendation systems, it is imperative to address ethical considerations and mitigate potential biases that may arise. Recommendations have the power to influence users' choices and perceptions, underscoring the importance of fairness, transparency, and accountability in recommendation algorithms. Measures such as diversity-aware recommendation strategies, fairness-aware evaluation metrics, and bias detection mechanisms are essential for mitigating biases related to factors such as gender, race, or

cultural background. Moreover, ongoing efforts to promote diversity, equity, and inclusion within the recommendation ecosystem are crucial for fostering a more inclusive and representative reading experience for users from diverse backgrounds and perspect

8. k-Nearest Neighbors (k-NN) :

algorithm for book recommendations. k-NN is a simple yet effective method based on similarity measurement. It works by identifying books similar to those a user has interacted with and recommending them based on their proximity in feature space

To implement k-NN, we start by defining a distance metric like Euclidean distance or cosine similarity. Then, for a given user or book, the algorithm finds the k nearest neighbors from the dataset based on similarity. These neighbors are used to generate recommendations, often by aggregating their preferences or ratings.

Training the k-NN model involves storing feature vectors of all books and precomputing distances or similarities for efficient retrieval during recommendation. Evaluation focuses on accuracy, coverage, and diversity, typically through cross-validation.

In practice, k-NN complements complex recommendation algorithms, offering transparency and interpretability. By leveraging similarities between books and users, it provides personalized recommendations aligned with individual preferences, enriching the book discovery experience. Integrating k-NN expands the recommendation system's capabilities, offering users diverse and relevant suggestions tailored to their tastes.

The project starts with importing the necessary libraries and reading the dataset (`books.csv`). The dataset contains information about books, including attributes like:

1. **bookID:** Unique identification number fro each book
2. **title:** Name under which book was published
3. **authors:** Name of the Authors of the book
4. **average_rating:**Avarage rating of the book recevied in total.
5. **isbn:** International standarded book number
6. **isbn13:**13 digitisbn to identify the book
7. **language_code:** Primary Language of the book
8. **num_pages**: Number of pages the book containes
9. **ratings_count:** Total Number of ratings the book recevied.

10. **text_reviews_count:** Total number of written reviews recevied.
11. **publication_date:** Date when the book was first published
12. **publisher:** Name of the Pulishers

Now we performed data preprocessing steps:

- Checking for null values.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11123 entries, 0 to 11122
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   bookID              11123 non-null  int64  
 1   title               11123 non-null  object 
 2   authors             11123 non-null  object 
 3   average_rating      11123 non-null  float64
 4   isbn                11123 non-null  object 
 5   isbn13              11123 non-null  int64  
 6   language_code       11123 non-null  object 
 7     num_pages         11123 non-null  int64  
 8   ratings_count       11123 non-null  int64  
 9   text_reviews_count  11123 non-null  int64  
 10  publication_date    11123 non-null  object 
 11  publisher           11123 non-null  object 
dtypes: float64(1), int64(5), object(6)
memory usage: 1.0+ MB
```

- Checking the data types and basic statistics.

```
df.dtypes
```

```
bookID                int64
title                object
authors              object
average_rating      float64
isbn                 object
isbn13                int64
language_code        object
num_pages             int64
ratings_count         int64
text_reviews_count    int64
publication_date     object
publisher            object
dtype: object
```

- Getting descriptive statistics for the numerical columns of the dataset (to understand the data better).

```
df.describe()
```

	bookID	average_rating	isbn13	num_pages	ratings_count	text_reviews_count
count	11123.000000	11123.000000	1.112300e+04	11123.000000	1.112300e+04	11123.000000
mean	21310.856963	3.934075	9.759880e+12	336.405556	1.794285e+04	542.048099
std	13094.727252	0.350485	4.429758e+11	241.152626	1.124992e+05	2576.619589
min	1.000000	0.000000	8.987060e+09	0.000000	0.000000e+00	0.000000
25%	10277.500000	3.770000	9.780345e+12	192.000000	1.040000e+02	9.000000
50%	20287.000000	3.960000	9.780582e+12	299.000000	7.450000e+02	47.000000
75%	32104.500000	4.140000	9.780872e+12	416.000000	5.000500e+03	238.000000
max	45641.000000	5.000000	9.790008e+12	6576.000000	4.597666e+06	94265.000000

Then we moved to feature extraction where we: extracted important features, reduced its size, and created some new features from the existring ones:

- First we re-visualized the features of the dataset and then checked the number of unique entries in: ‘isbn’and ‘isbn13’.

```
df.columns
```

```
Index(['bookID', 'title', 'authors', 'average_rating', 'isbn', 'isbn13',
       'language_code', 'num_pages', 'ratings_count', 'text_reviews_count',
       'publication_date', 'publisher'],
      dtype='object')
```

```
df.isbn.nunique()
```

```
11123
```

```
df.isbn13.nunique()
```

```
11123
```

- Dropedthose columns (`bookID`, `isbn`, `isbn13`) that are not necessary for model building in this case.

```
df.drop(['bookID', 'isbn', 'isbn13'], axis = 1, inplace = True)
```

```
df.columns
```

```
Index(['title', 'authors', 'average_rating', 'language_code', 'num_pages',
       'ratings_count', 'text_reviews_count', 'publication_date', 'publisher'],
      dtype='object')
```

- Then we visualized and extracted the publication year from the `publication_date` column.

```
df.publication_date
```

```
0         9/16/2006
1          9/1/2004
2         11/1/2003
3          5/1/2004
4         9/13/2004
            ...
11118    12/21/2004
11119     12/1/1988
11120      8/1/1993
11121     2/27/2007
11122     5/28/2006
Name: publication_date, Length: 11123, dtype: object
```

```
df['year'] = df['publication_date'].str.split('/')
df['year'] = df['year'].apply(lambda x: x[2])
```

```
df.head(2)
```

	title	authors	average_rating	language_code	num_pages	ratings_count	text_reviews_count	publication_date	publisher	year
0	Harry Potter and the Half-Blood Prince (Harry ...	J.K. Rowling/Mary GrandPré	4.57	eng	652	2095690	27591	9/16/2006	Scholastic Inc.	2006
1	Harry Potter and the Order of the Phoenix (Har...	J.K. Rowling/Mary GrandPré	4.49	eng	870	2153167	29221	9/1/2004	Scholastic Inc.	2004

- Finally we convert the ‘year’ into ‘int’ and analyse the max and min values of the same.

```
df['year'] = df['year'].astype('int')
```

```
df.dtypes
```

```
title                 object
authors               object
average_rating       float64
language_code         object
num_pages              int64
ratings_count          int64
text_reviews_count     int64
publication_date      object
publisher             object
year                   int32
dtype: object
```

```
df.columns
```

```
Index(['title', 'authors', 'average_rating', 'language_code', 'num_pages',
       'ratings_count', 'text_reviews_count', 'publication_date', 'publisher',
       'year'],
      dtype='object')
```

```
df['year'].min()
```

```
1900
```

```
df['year'].max()
```

```
2020
```

2. Exploratory Data Analysis (EDA)

EDA is performed to gain insights into the dataset and understand the relationships between different attributes. Key EDA steps which we performed include:

- Analyzing the distribution of books across publication years.

```
df[df['year'] == 2020][['title', 'authors','average_rating','language_code','publisher' ]]
```

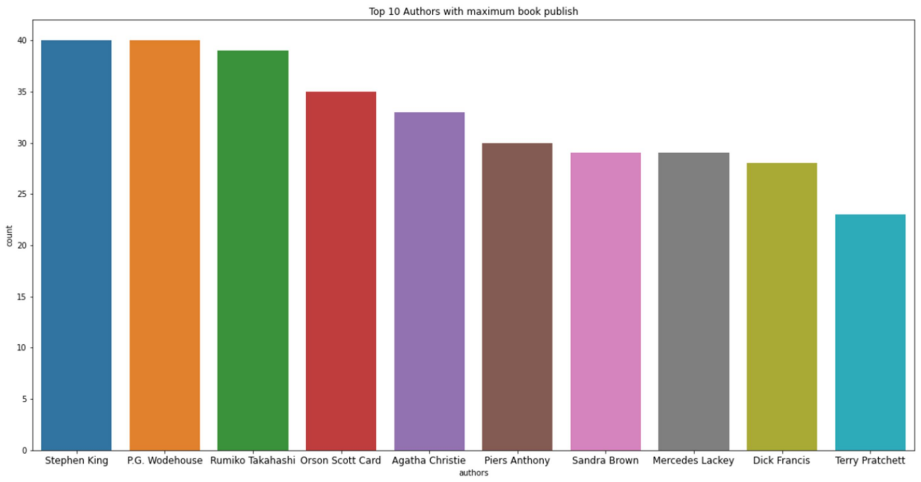
	title	authors	average_rating	language_code	publisher
9664	A Quick Bite (Argeneau #1)	Lynsay Sands	3.91	eng	Avon

```
df.groupby(['year'])['title'].agg('count').sort_values(ascending = False).head(20)
```

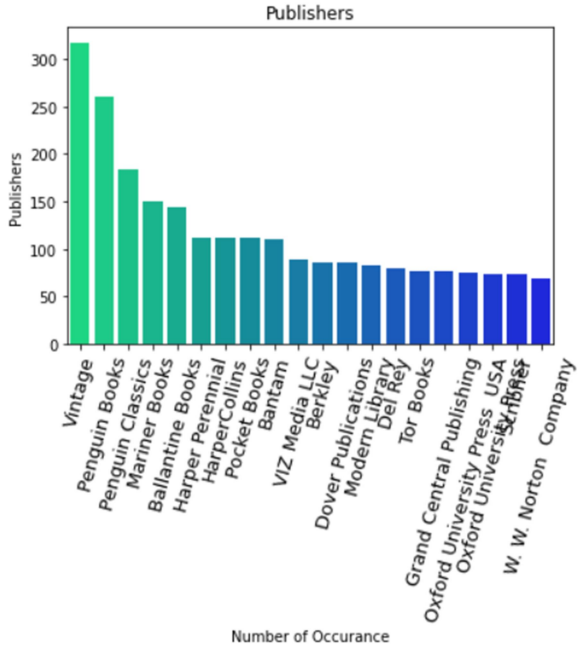
```
year
2006    1700
2005    1260
2004    1069
2003     931
2002     798
2001     656
2000     534
2007     518
1999     450
1998     396
1997     290
1996     250
1995     249
1994     220
1992     183
1993     165
1991     151
1989     118
1990     117
1987      88
Name: title, dtype: int64
```

- Identifying top authors and publishers based on the number of books published.

```
plt.figure(figsize = (20, 10))
sns.countplot(x = 'authors', data = df,
              order = df['authors'].value_counts().iloc[:10].index)
plt.title("Top 10 Authors with maximum book publish")
plt.xticks(fontsize = 12)
plt.show()
```

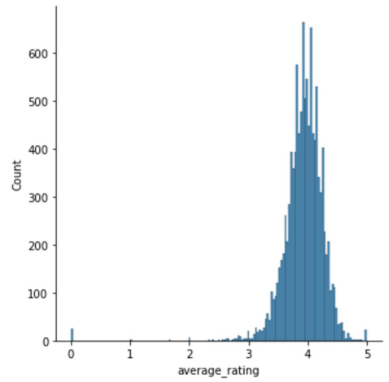



```
publisher = df['publisher'].value_counts()[:20]
sns.barplot(x = publisher.index, y = publisher, palette = 'winter_r')
plt.title("Publishers")
plt.xlabel("Number of Occurance")
plt.ylabel("Publishers")
plt.xticks(rotation = 75, fontsize = 13)
plt.show()
```



- Visualizing the distribution of book ratings and identifying books with the highest ratings.

```
sns.displot(df['average_rating'])
plt.show()
```

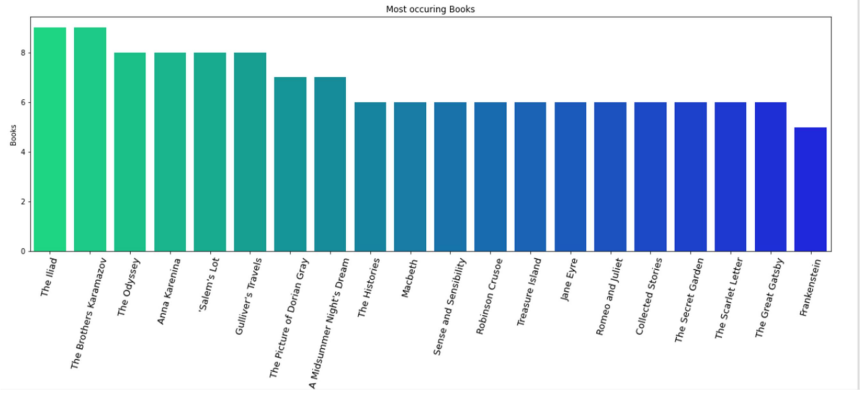



```
df[df.average_rating == df.average_rating.max()][['title','authors','language_code','publisher']]
```

	title	authors	language_code	publisher
624	Comoediae 1: Acharenses/Equites/Nubes/Vespae/P...	Aristophanes/F.W. Hall/W.M. Geldart	grc	Oxford University Press USA
786	Willem de Kooning: Late Paintings	Julie Sylvester/David Sylvester	eng	Schirmer Mosel
855	Literature Circle Guide: Bridge to Terabithia:...	Tara MacCarthy	eng	Teaching Resources
1243	Middlesex Borough (Images of America: New Jersey)	Middlesex Borough Heritage Committee	eng	Arcadia Publishing
4125	Zone of the Enders: The 2nd Runner Official St...	Tim Bogenn	eng	BradyGames
4788	The Diamond Color Meditation: Color Pathway to...	John Diamond	eng	Square One Publishers
4933	Bulgakov's the Master and Margarita: The Text ...	Elena N. Mahlow	eng	Vantage Press
5023	The Complete Theory Fun Factory: Music Theory ...	Ian Martin/Katie Elliott	eng	Boosey & Hawkes Inc
5474	The Goon Show Volume 4: My Knees Have Fallen ...	NOT A BOOK	eng	BBC Physical Audio
5476	The Goon Show Volume 11: He's Fallen in the W...	NOT A BOOK	eng	BBC Physical Audio
5647	Winchester Shotguns	Dennis Adler/R.L. Wilson	eng	Chartwell Books
5648	Colossians and Philemon: A Critical and Exeget...	R. McL. Wilson	eng	T&T Clark Int'l
6184	Taxation of Mineral Rents	Ross Garnaut	eng	Oxford University Press USA
6247	The New Big Book of America	Todd Davis/Marc Frey	eng	Courage Books
6775	Delwau Duon: Peintiadau Nicholas Evans = Symph...	Nicholas Evans/Rhonda Evans	wel	Y Lolfa
8544	Fanning the Flame: Bible Cross and Mission	Chris Green/Chris Wright/Paul Douglas Gardner	eng	Zondervan
9282	Oliver Wendell Holmes in Paris: Medicine Theo...	William C. Dowling	eng	University Press of New England
9324	Tyrannosaurus Wrecks (Stanley #1)	Laura Driscoll/Alisa Klayman-Grodsky/Eric ...	eng	Disney Press
9720	The Irish Anatomist: A Study of Flann O'Brien	Keith Donohue	eng	Academica Press
9847	The American Campaign: U.S. Presidential Campa...	James E. Campbell	eng	Texas A&M University Press
9893	His Princess Devotional: A Royal Encounter Wit...	Sheri Rose Shepherd	eng	Multnomah
10262	Bill Gates: Computer Legend (Famous Lives)	Sara Barton-Wood	eng	Raintree

- Exploring the most popular books based on occurrence in the dataset.

```
plt.figure(figsize = (20, 6))
book = df['title'].value_counts()[:20]
sns.barplot(x = book.index, y = book,
            palette = 'winter_r')
plt.title("Most occuring Books")
plt.xlabel("Number of Occurance")
plt.ylabel("Books")
plt.xticks(rotation = 75, fontsize = 13)
plt.show()
```



- Analyzing the distribution of books across different languages.

```
df.language_code.value_counts()
```

```
eng      8908
en-US    1408
spa       218
en-GB     214
fre       144
ger        99
jpn        46
mul        19
zho        14
grc        11
por        10
en-CA       7
ita         5
enm         3
lat         3
swe         2
rus         2
srp         1
nl          1
msa         1
glg         1
wel         1
ara         1
nor         1
tur         1
gla         1
ale         1
Name: language_code, dtype: int64
```

```
df.groupby(['language_code'])[['average_rating',
                               'ratings_count',
                               'text_reviews_count']].agg('mean').style.background_gradient(cmap = 'Wistia')
```

language_code	average_rating	ratings_count	text_reviews_count
ale	4.360000	102.000000	16.000000
ara	3.550000	122.000000	12.000000
en-CA	4.025714	4086.714286	324.428571
en-GB	3.923411	2463.691589	104.060748
en-US	3.914659	3773.906960	160.357244
eng	3.934062	21570.272564	645.156601
enm	3.873333	3233.666667	84.000000
fre	3.971528	3277.319444	64.513889
ger	3.950101	234.727273	8.232323
gla	4.470000	11.000000	0.000000
glg	3.360000	36.000000	2.000000
grc	3.707273	52.454545	2.454545
ita	4.078000	3234.400000	55.800000
jpn	4.268696	68.304348	3.152174
lat	4.353333	114.666667	12.333333
msa	4.110000	28.000000	6.000000
mul	4.126316	386.631579	19.263158
nl	4.180000	67.000000	9.000000
nor	3.600000	86.000000	8.000000

3. Further Processing

Now to build the model some further processing is needed. So we begin by:

- Creating a categorical feature ‘rating_obj’ based on ‘average_rating’.

```
def num_to_obj(x):
    if x >0 and x <=1:
        return "between 0 and 1"
    if x > 1 and x <= 2:
        return "between 1 and 2"
    if x > 2 and x <=3:
        return "between 2 and 3"
    if x >3 and x<=4:
        return "between 3 and 4"
    if x >4 and x<=5:
        return "between 4 and 5"
df['rating_obj'] = df['average_rating'].apply(num_to_obj)
```

```
df['rating_obj'].value_counts()
```

```
between 3 and 4    6285
between 4 and 5    4735
between 2 and 3      69
between 1 and 2       7
between 0 and 1       2
Name: rating_obj, dtype: int64
```

- Converting the categorical variables ‘rating_obj’ and ‘language_code’ into dummy variables.

```
rating_df = pd.get_dummies(df['rating_obj'])
rating_df.head()
```

	between 0 and 1	between 1 and 2	between 2 and 3	between 3 and 4	between 4 and 5
0	0	0	0	0	1
1	0	0	0	0	1
2	0	0	0	0	1
3	0	0	0	0	1
4	0	0	0	0	1

```
df.columns
```

```
Index(['title', 'authors', 'average_rating', 'language_code', 'num_pages',
       'ratings_count', 'text_reviews_count', 'publication_date', 'publisher',
       'year', 'rating_obj'],
      dtype='object')
```

```
language_df = pd.get_dummies(df['language_code'])
language_df.head()
```

	ale	ara	en-CA	en-GB	en-US	eng	enm	fre	ger	gla	...	nl	nor	por	rus	spa	srp	swe	tur	wel	zho
0	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 27 columns

- Creating a feature matrix (`features`) by concatenating the encoded features and numeric attributes (`average_rating`, `ratings_count`) with the book titles as the index.

```
features = pd.concat([rating_df,language_df, df['average_rating'],
                      df['ratings_count'], df['title']], axis = 1)
features.set_index('title', inplace= True)
features.head()
```

title	between 0 and 1	between 1 and 2	between 2 and 3	between 3 and 4	between 4 and 5	ale	ara	en-CA	en-GB	en-US	...	por	rus	spa	srp	swe	tur	wel	zho	average_rating	ratings_count
Harry Potter and the Half-Blood Prince (Harry Potter #6)	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	4.57	2095690
Harry Potter and the Order of the Phoenix (Harry Potter #5)	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	4.49	2153167
Harry Potter and the Chamber of Secrets (Harry Potter #2)	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	4.42	6333
Harry Potter and the Prisoner of Azkaban (Harry	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	4.56	2339585

- Scaling the feature matrix using `MinMaxScaler` for normalization.

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler()
features_scaled = scaler.fit_transform(features)
```

```
features_scaled
```

```
array([[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
        0.00000000e+00, 9.14000000e-01, 4.55816060e-01],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
        0.00000000e+00, 8.98000000e-01, 4.68317403e-01],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
        0.00000000e+00, 8.84000000e-01, 1.37743803e-03],
       ...,
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
        0.00000000e+00, 7.92000000e-01, 1.78351363e-04],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
        0.00000000e+00, 7.44000000e-01, 1.67258779e-04],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,
        0.00000000e+00, 7.82000000e-01, 2.45776879e-05]])
```

4. Model Building

The model for book recommendations is built using the `K-NearestNeighbors` algorithm from `sklearn.neighbors`. Key model building steps include:

- Initializing the k-NN model with the parameters as (`n_neighbors`, `algorithm`, `metric`).

```
from sklearn import neighbors
```

```
model = neighbors.NearestNeighbors(n_neighbors=5, algorithm = 'ball_tree',
                                   metric = 'euclidean')
model.fit(features_scaled)
dist, idlist = model.kneighbors(features_scaled)
```

- Fitting the model on the scaled feature matrix (`features_scaled`).
- Using the fitted model to find similar books for a given input book using the `BookRecommender` function.

```
[66] @interact
     def BookRecomender(book_name = list(df['title'].value_counts().index)):
         book_list_name = []
         book_id = df[df['title'] == book_name].index
         book_id = book_id[0]
         for newid in idlist[book_id]:
             book_list_name.append(df.loc[newid].title)
         return book_list_name
```

book_name: The Iliad

```
['The Iliad',
 'The Call of the Wild',
 "She's Come Undone",
 'The Fountainhead',
 'Beloved']
```

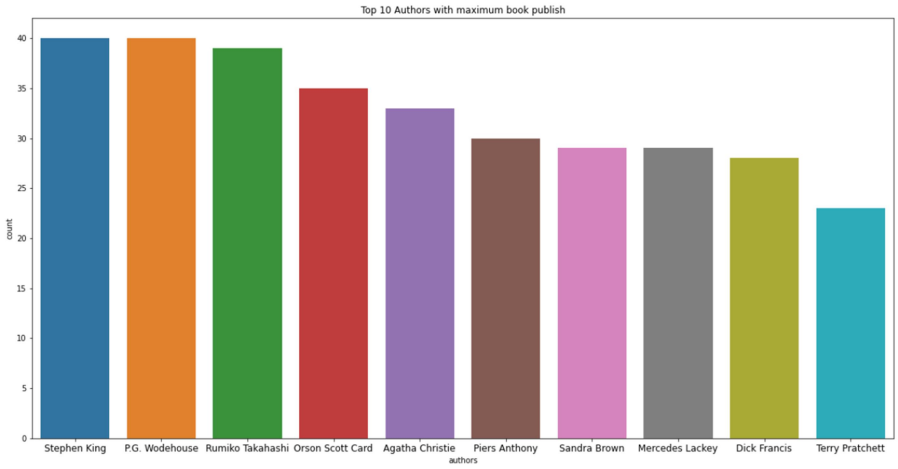
Results and Discussion

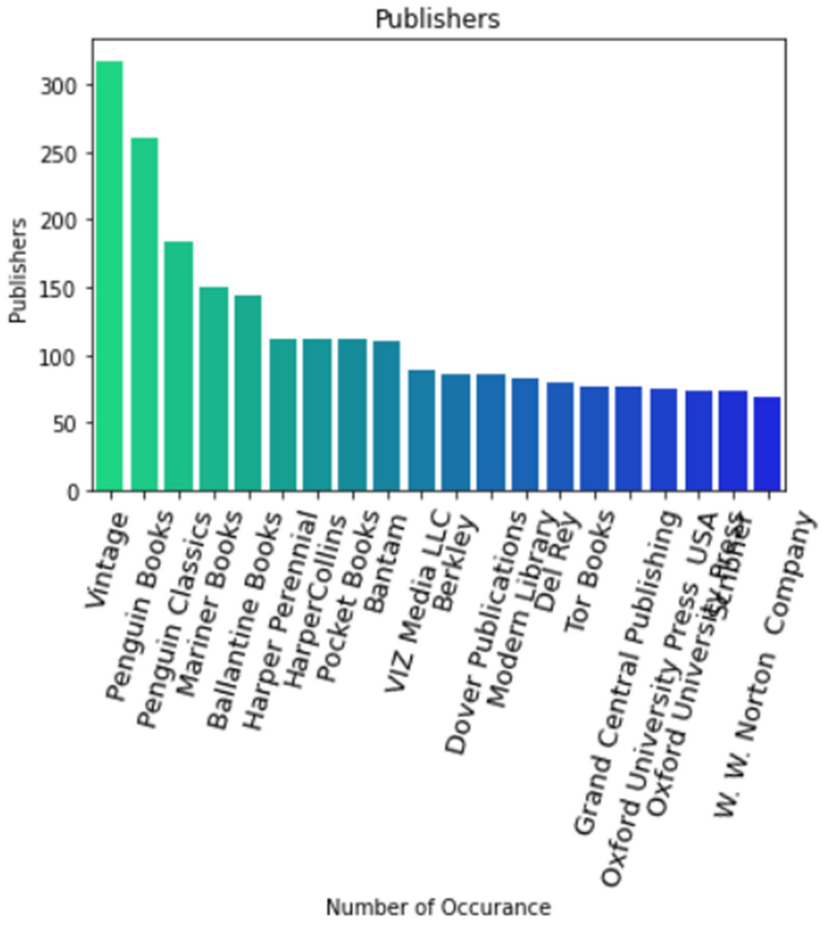
1. Exploratory Data Analysis Findings

- Publication Trends: The dataset shows a distribution of books across different publication years, with a significant number of books published around recent years.

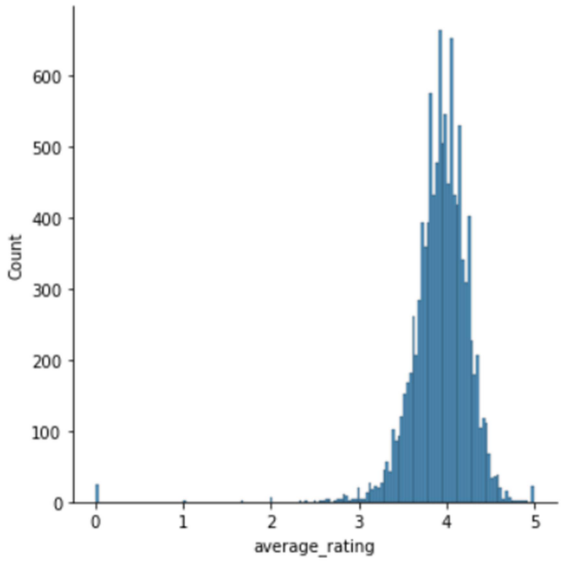
```
year
2006    1700
2005    1260
2004    1069
2003     931
2002     798
2001     656
2000     534
2007     518
1999     450
1998     396
1997     290
1996     250
1995     249
1994     220
1992     183
1993     165
1991     151
1989     118
1990     117
1987      88
Name: title, dtype: int64
```

- Author and Publisher Insights: Several authors and publishers stand out based on the number of books they've produced. Here are the top 10 Authors and top 20 Publishers based on the data.


Top 10 Authors with maximum book publish
count
authors
Stephen King
P.G. Wodehouse
Rumiko Takahashi
Orson Scott Card
Agatha Christie
Piers Anthony
Sandra Brown
Mercedes Lackey
Dick Francis
Terry Pratchett


Publishers
Publishers
Number of Occurance
Vintage
Penguin Books
Penguin Classics
Mariner Books
Ballantine Books
Harper Perennial
HarperCollins
Pocket Books
Bantam
VIZ Media LLC
Berkley
Dover Publications
Modern Library
Del Rey
Tor Books
Grand Central Publishing
Oxford University Press USA
W. W. Norton Company

- Book Ratings: The distribution of book ratings reveals that most books have ratings above 3, with a few outliers having exceptionally high ratings.

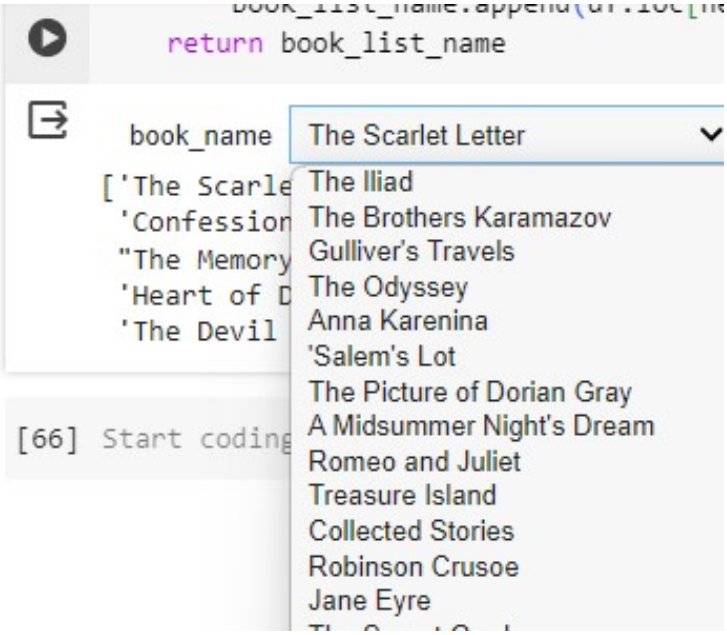


- Language Distribution: The dataset contains most books in English, followed by other languages like Spanish and French. It also has separate entries for US and UK English.

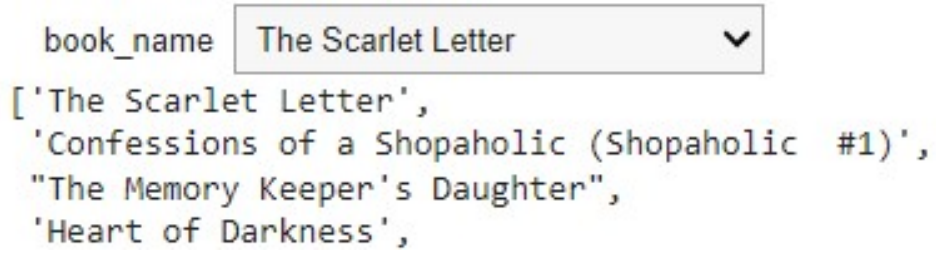
```
eng      8908
en-US    1408
spa       218
en-GB     214
fre       144
ger        99
jpn        46
mul        19
zho        14
grc        11
por        10
en-CA       7
ita         5
enm         3
lat         3
swe         2
rus         2
srp         1
nl          1
msa         1
glg         1
wel         1
ara         1
nor         1
tur         1
gla         1
ale         1
Name: language_code, dtype: int64
```

2. Book Recommendation System

- Recommendation Functionality: The interactive recommendation system allows users to input a book title which he/she has read and get a list of similar books based on the model's predictions.



- Model Performance: The k-Nearest Neighbors model performs well in finding similar books based on a combination of features like ratings, language, and more.



3. Limitations and Future Work

- Data Coverage: The dataset primarily focuses on certain languages and genres, limiting the diversity of recommendations.
- Model Scalability: As the dataset grows, the scalability of the model and recommendation system should be considered.
- Personalization: Enhancing the recommendation system to incorporate user preferences and behaviors for a longer period of time for more personalized recommendations.