

Machine Learning Project on Book Recommendation System

```
In [ ]: ## Importing all necessary libraries

import numpy as np
import pandas as pd

## for data visualisation

import matplotlib.pyplot as plt
import seaborn as sns

## for interactive plot
import ipywidgets
from ipywidgets import interact
from ipywidgets import interact_manual

## for Ignoring Warning ErrorMessage

from warnings import filterwarnings
filterwarnings('ignore')
```

Importing Dataset

```
In [ ]: df = pd.read_csv("Books.csv",error_bad_lines = False)
```

```
b'Skipping line 3350: expected 12 fields, saw 13\nSkipping line 4704: expected 12 fields, saw 13\nSkipping line 5879: expected 12 fields, saw 13\nSkipping line 8981: expected 12 fields, saw 13\n'
b'Skipping line 3350: expected 12 fields, saw 13\nSkipping line 4704: expected 12 fields, saw 13\nSkipping line 5879: expected 12 fields, saw 13\nSkipping line 8981: expected 12 fields, saw 13\n'
```

Dataset Description

- BookID: Unique identification number for each book
- title: Name under which book was published
- Author: Name of the author of the book
- average_rating: average rating of the book received in total.
- isbn : international standard book number
- isbn13: 13 digit isbn to identify the book
- language_book : primary language of the book
- num_pages : number of pages the book contains
- rating_counts: Total number of ratings of book received.
- text_reviews_count: total number of written reviews received.

- publication_date: date when the book was first published.
- publisher: name of publisher

In []:

df.head(5)

Out[]:

| | bookID | title | authors | average_rating | isbn | isbn13 | language_code | num_p |
|---|--------|----------------------------------------------------|-----------------------|----------------|------------|---------------|---------------|-------|
| 0 | 1 | Harry Potter and the Half-Blood Prince (Harry ...) | Rowling/Mary GrandPré | 4.57 | 0439785960 | 9780439785969 | eng | |
| 1 | 2 | Harry Potter and the Order of the Phoenix (Har... | Rowling/Mary GrandPré | 4.49 | 0439358078 | 9780439358071 | eng | |
| 2 | 4 | Harry Potter and the Chamber of Secrets (Harry... | J.K. Rowling | 4.42 | 0439554896 | 9780439554893 | eng | |
| 3 | 5 | Harry Potter and the Prisoner of Azkaban (Harr... | Rowling/Mary GrandPré | 4.56 | 043965548X | 9780439655484 | eng | |
| 4 | 8 | Harry Potter Boxed Set Books 1-5 (Harry Pott... | Rowling/Mary GrandPré | 4.78 | 0439682584 | 9780439682589 | eng | |

Out[]:

| | bookID | title | authors | average_rating | isbn | isbn13 | language_code | num_p |
|---|--------|----------------------------------------------------|-----------------------|----------------|------------|---------------|---------------|-------|
| 0 | 1 | Harry Potter and the Half-Blood Prince (Harry ...) | Rowling/Mary GrandPré | 4.57 | 0439785960 | 9780439785969 | eng | |

| bookID | title | authors | average_rating | isbn | isbn13 | language_code | num_pages |
|--------|-------------------------------------------------------------------------------------------------------------------|----------------------------|----------------|------------|---------------|---------------|-----------|
| 1 | 2 Harry Potter and the Order of the Phoenix (Harry Potter and the Phoenix (Harry Potter and the Phoenix (Harry... | J.K. Rowling/Mary GrandPré | 4.49 | 0439358078 | 9780439358071 | eng | 160 |
| 2 | 4 Chamber of Secrets (Harry Potter and the Chamber of Secrets (Harry... | J.K. Rowling | 4.42 | 0439554896 | 9780439554893 | eng | 160 |
| 3 | 5 Harry Potter and the Prisoner of Azkaban (Harry Potter and the Prisoner of Azkaban (Harry... | J.K. Rowling/Mary GrandPré | 4.56 | 043965548X | 9780439655484 | eng | 160 |
| 4 | 8 Harry Potter Boxed Set Books 1-5 (Harry Potter Boxed Set Books 1-5 (Harry... | J.K. Rowling/Mary GrandPré | 4.78 | 0439682584 | 9780439682589 | eng | 160 |



Checking Row & Column of Dataset

```
In [ ]: df.shape
```

```
Out[ ]: (11123, 12)
```

```
Out[ ]: (11123, 12)
```

Checking all columns of the dataset

```
In [ ]: df.columns
```

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

Removing Extra space from all columns name

```
In [ ]: df.columns = df.columns.str.strip()
```

Checking Datatypes of the dataset

```
In [ ]: df.dtypes
```

```
Out[ ]: 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

Out[ ]: 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
```

Checking Statistical Summary of all Numeric Columns

```
In [ ]: 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 |

| | bookID | average_rating | isbn13 | num_pages | ratings_count | text_reviews_count |
|------------|--------------|----------------|--------------|-------------|---------------|--------------------|
| 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 |

| | 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 |

Checking Statistical Summary of all Categorical Columns

In []: df.describe(include= 'object')

| | title | authors | isbn | language_code | publication_date | publisher |
|---------------|-----------|--------------|------------|---------------|------------------|-----------|
| count | 11123 | 11123 | 11123 | 11123 | 11123 | 11123 |
| unique | 10348 | 6639 | 11123 | 27 | 3679 | 2290 |
| top | The Iliad | Stephen King | 0439785960 | eng | 10/1/2005 | Vintage |
| freq | 9 | 40 | 1 | 8908 | 56 | 318 |

| | title | authors | isbn | language_code | publication_date | publisher |
|---------------|-----------|--------------|------------|---------------|------------------|-----------|
| count | 11123 | 11123 | 11123 | 11123 | 11123 | 11123 |
| unique | 10348 | 6639 | 11123 | 27 | 3679 | 2290 |
| top | The Iliad | Stephen King | 0439785960 | eng | 10/1/2005 | Vintage |
| freq | 9 | 40 | 1 | 8908 | 56 | 318 |

Checking Sum of all null value present in the dataset

In []: df.isnull().sum()

```
Out[ ]: bookID          0  
        title           0  
        authors         0  
        average_rating  0  
        isbn            0  
        isbn13          0  
        language_code   0  
        num_pages       0  
        ratings_count   0  
        text_reviews_count 0  
        publication_date 0  
        publisher        0  
        dtype: int64  
  
Out[ ]: bookID          0  
        title           0  
        authors         0  
        average_rating  0  
        isbn            0  
        isbn13          0  
        language_code   0  
        num_pages       0  
        ratings_count   0  
        text_reviews_count 0  
        publication_date 0  
        publisher        0  
        dtype: int64
```

Checking if any duplicate row present in dataset or not

```
In [ ]: df.duplicated().any()
```

```
Out[ ]: False  
Out[ ]: False
```

Checking Summary of dataset

```
In [ ]: 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
<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 all Columns present name dataset in dataset

```
In [ ]: df.columns
```

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

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

Checking all Unique rows present in isbn columns

```
In [ ]: df.isbn.nunique()
```

```
Out[ ]: 11123
```

```
Out[ ]: 11123
```

Checking all Unique rows present in isbn13 columns

```
In [ ]: df.isbn13.nunique()
```

```
Out[ ]: 11123
```

```
Out[ ]: 11123
```

```
In [ ]: df.columns
```

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

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

Droping unnecessary column present in dataset

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

Checking all Columns present name dataset in dataset

```
In [ ]: df.columns
```

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

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

Checking All rows present in publication_date

```
In [ ]: 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
Out[ ]: 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
```

Creating New year Columns

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

Checking top 2 rows from dataset

```
In [ ]: df.head(2)
```

| | title | authors | average_rating | language_code | num_pages | ratings_count | text_reviews_count |
|---|---------------------------------------------------|----------------------------|-----------------------|----------------------|------------------|----------------------|---------------------------|
| 0 | Harry Potter and the Half-Blood Prince (Harry ... | J.K. Rowling/Mary GrandPré | 4.57 | eng | 652 | 2095690 | 27591 |
| 1 | Harry Potter and the Order of the Phoenix (Har... | J.K. Rowling/Mary GrandPré | 4.49 | eng | 870 | 2153167 | 29221 |

| | title | authors | average_rating | language_code | num_pages | ratings_count | text_reviews_count |
|---|---------------------------------------------------|----------------------------|-----------------------|----------------------|------------------|----------------------|---------------------------|
| 0 | Harry Potter and the Half-Blood Prince (Harry ... | J.K. Rowling/Mary GrandPré | 4.57 | eng | 652 | 2095690 | 27591 |

| | title | authors | average_rating | language_code | num_pages | ratings_count | text_reviews_count |
|---|---------------------------------------------------|----------------------------|----------------|---------------|-----------|---------------|--------------------|
| 1 | Harry Potter and the Order of the Phoenix (Har... | J.K. Rowling/Mary GrandPré | 4.49 | eng | 870 | 2153167 | 29221 |

Checking dtype of all columns

In []:

```
df.dtypes
```

Out[]:

| | |
|--------------------|---------|
| 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 | object |
| dtype: object | |

| | |
|--------------------|---------|
| 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 | object |
| dtype: object | |

Changing Datatype of year Columns from object to integer.

In []:

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

Checking dtype of all columns

In []:

```
df.dtypes
```

Out[]:

| | |
|----------------|---------|
| 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
Out[ ]: 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

```

Checking all column name from dataset

```

In [ ]: df.columns
Out[ ]: Index(['title', 'authors', 'average_rating', 'language_code', 'num_pages',
               'ratings_count', 'text_reviews_count', 'publication_date', 'publisher',
               'year'],
              dtype='object')
Out[ ]: Index(['title', 'authors', 'average_rating', 'language_code', 'num_pages',
               'ratings_count', 'text_reviews_count', 'publication_date', 'publisher',
               'year'],
              dtype='object')

```

Checking minimum Year Present in a dataset.

```

In [ ]: df['year'].min()
Out[ ]: 1900
Out[ ]: 1900

```

Checking maximum Year Present in a dataset.

```

In [ ]: df['year'].max()
Out[ ]: 2020
Out[ ]: 2020

```

Checking all column name from dataset

```
In [ ]: df.columns
```

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

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

EDA (Exploratory Data Analysis)

Filter Year == 2022 and get required output based on input

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

| | title | authors | average_rating | language_code | publisher | num_pages |
|-------------|----------------------------|--------------|----------------|---------------|-----------|-----------|
| 9664 | A Quick Bite (Argeneau #1) | Lynsay Sands | 3.91 | eng | Avon | 360 |

| | title | authors | average_rating | language_code | publisher | num_pages |
|-------------|----------------------------|--------------|----------------|---------------|-----------|-----------|
| 9664 | A Quick Bite (Argeneau #1) | Lynsay Sands | 3.91 | eng | Avon | 360 |

Filter Year == 2022 and get required output based on input

```
In [ ]: df[df['year']==2018][['title', 'authors', 'average_rating', 'language_code', 'publisher', 'n
```

| | title | authors | average_rating | language_code | publisher | num_pages |
|--------------|--------------------------------------------------|---------------------------------------------------|----------------|---------------|----------------------------------|-----------|
| 3171 | Ariel: The Restored Edition | Sylvia Plath/Frieda Hughes | 4.27 | eng | Harper Perennial Modern Classics | 256 |
| 4080 | El Perfume: Historia De Un Asesino | Patrick Süskind | 4.02 | spa | Planeta Publishing | 312 |
| 4082 | The Perfume Factory | Alex Austin | 4.18 | eng | Kindle | 227 |
| 8068 | El diablo de la botella | Robert Louis Stevenson/Diana Castellanos/Eleon... | 3.74 | spa | Grupo Editorial Norma S.A. | 77 |
| 11085 | El alquimista: una fábula para seguir tus sueños | Paulo Coelho/Juan Godó Costa | 3.86 | eng | Rayo | 192 |

Out[]:

| | title | authors | average_rating | language_code | publisher | num_pages |
|-------|--------------------------------------------------|---------------------------------------------------|----------------|---------------|----------------------------------|-----------|
| 3171 | Ariel: The Restored Edition | Sylvia Plath/Frieda Hughes | 4.27 | eng | Harper Perennial Modern Classics | 256 |
| 4080 | El Perfume: Historia De Un Asesino | Patrick Süskind | 4.02 | spa | Planeta Publishing | 312 |
| 4082 | The Perfume Factory | Alex Austin | 4.18 | eng | Kindle | 227 |
| 8068 | El diablo de la botella | Robert Louis Stevenson/Diana Castellanos/Eleon... | 3.74 | spa | Grupo Editorial Norma S.A. | 77 |
| 11085 | El alquimista: una fábula para seguir tus sueños | Paulo Coelho/Juan Godó Costa | 3.86 | eng | Rayo | 192 |

Creating Groupby function based on year and title columns

In []:

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

Out[]:

```
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
```

Out[]:

```
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

```

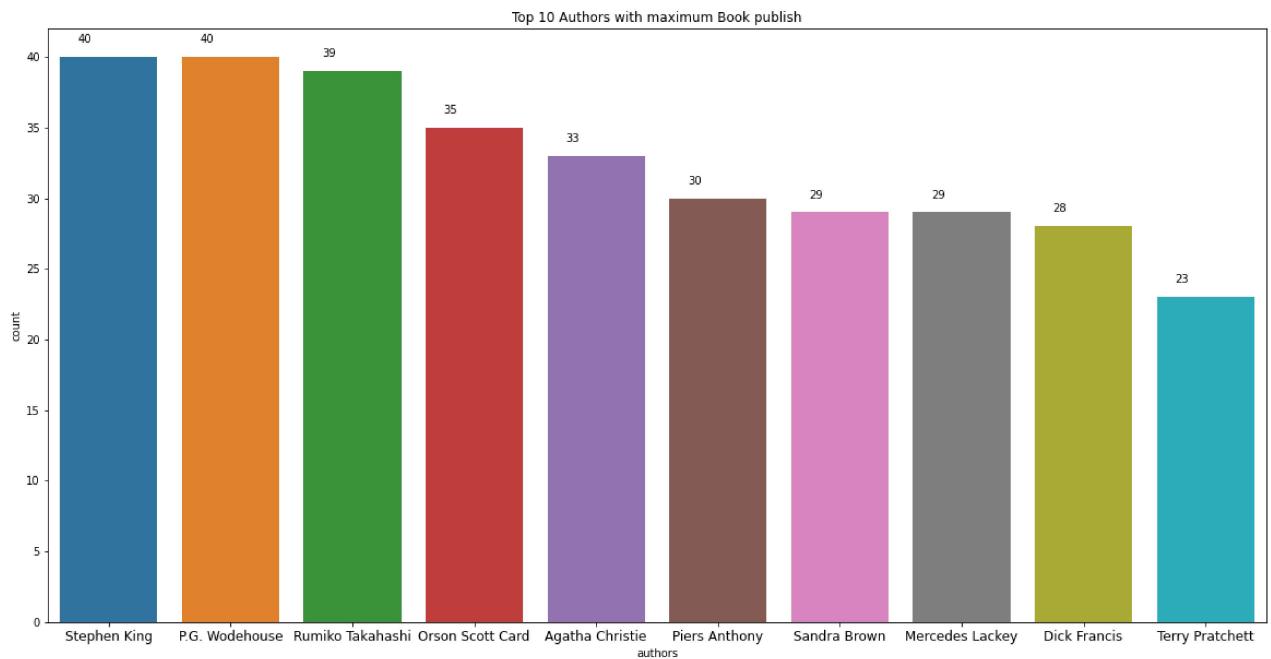
Plotting Countplot graph for "Top 10 Authors with maximum book publish"

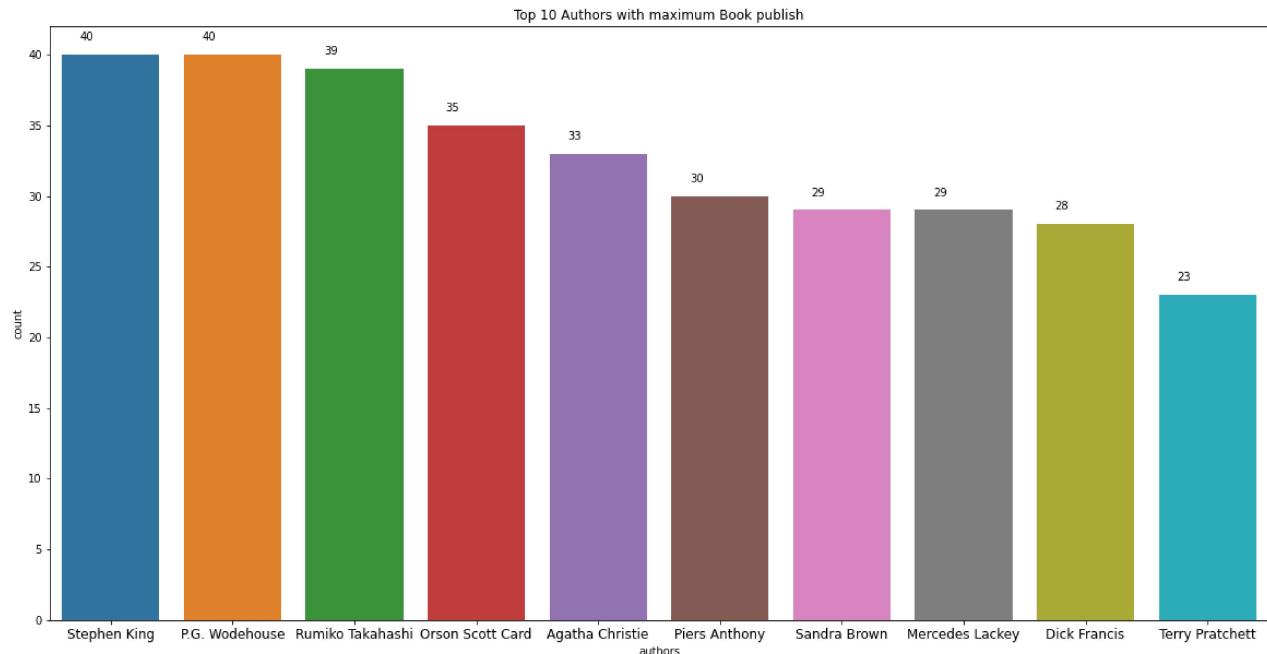
```

In [ ]:
plt.figure(figsize=(20,10))
ax = 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)

for p in ax.patches:
    ax.annotate(format(p.get_height()),(p.get_x()+0.15,p.get_height()+1))
plt.show()

```





Checking all column name from dataset

```
In [ ]: df.columns
```

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

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

Sort All Value Count of language_code

```
In [ ]: df.language_code.value_counts()
```

```
Out[ ]: 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
Out[ ]:
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

```

Creating Groupby Function base on Language_code Column and getting Required Output.

```
In [ ]: df.groupby(['language_code'])[['average_rating','ratings_count','text_reviews_count']]
```

```
Out[ ]:   average_rating  ratings_count  text_reviews_count
```

| 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 |

| language_code | average_rating | ratings_count | text_reviews_count |
|---------------|----------------|---------------|--------------------|
| 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 |
| por | 3.945000 | 165.100000 | 13.500000 |
| rus | 4.255000 | 4477.000000 | 98.500000 |
| spa | 3.929312 | 4636.114679 | 91.123853 |
| srp | 0.000000 | 0.000000 | 0.000000 |
| swe | 3.455000 | 2671.000000 | 157.000000 |
| tur | 4.420000 | 1000.000000 | 41.000000 |
| wel | 5.000000 | 1.000000 | 0.000000 |
| zho | 4.456429 | 20.428571 | 0.500000 |

Out[]:

| 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 |

| language_code | average_rating | ratings_count | text_reviews_count |
|---------------|----------------|---------------|--------------------|
| 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 |
| por | 3.945000 | 165.100000 | 13.500000 |
| rus | 4.255000 | 4477.000000 | 98.500000 |
| spa | 3.929312 | 4636.114679 | 91.123853 |
| srp | 0.000000 | 0.000000 | 0.000000 |
| swe | 3.455000 | 2671.000000 | 157.000000 |
| tur | 4.420000 | 1000.000000 | 41.000000 |
| wel | 5.000000 | 1.000000 | 0.000000 |
| zho | 4.456429 | 20.428571 | 0.500000 |

Checking Top 20 Value count of Title Column

```
In [ ]: book = df['title'].value_counts()[20:]
```

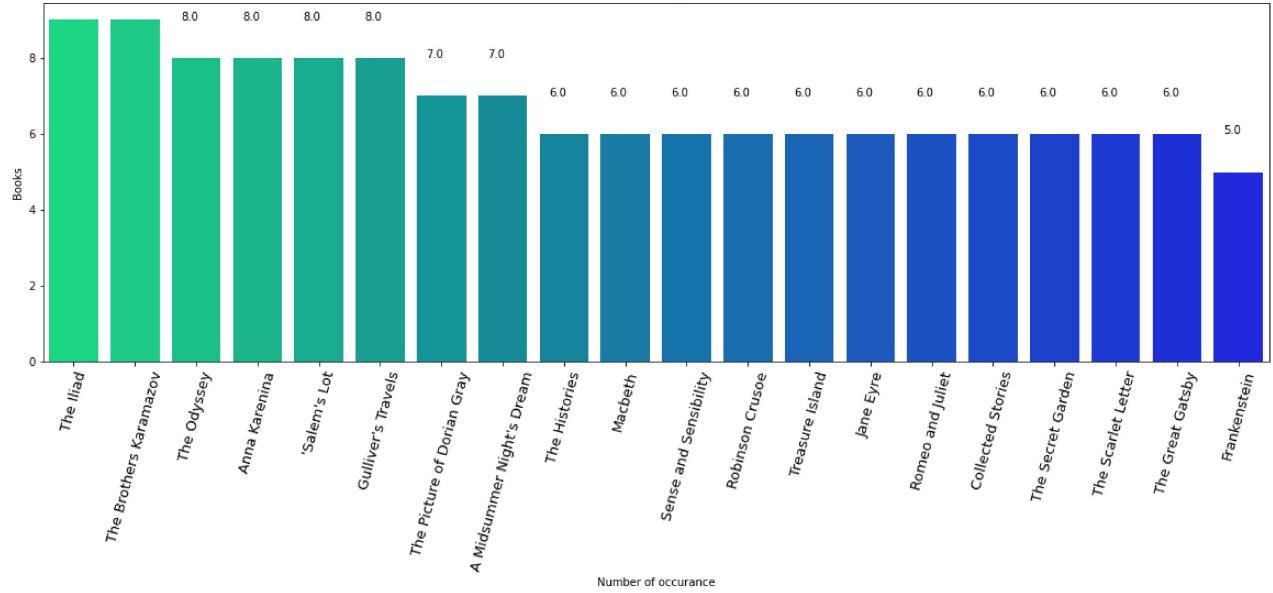
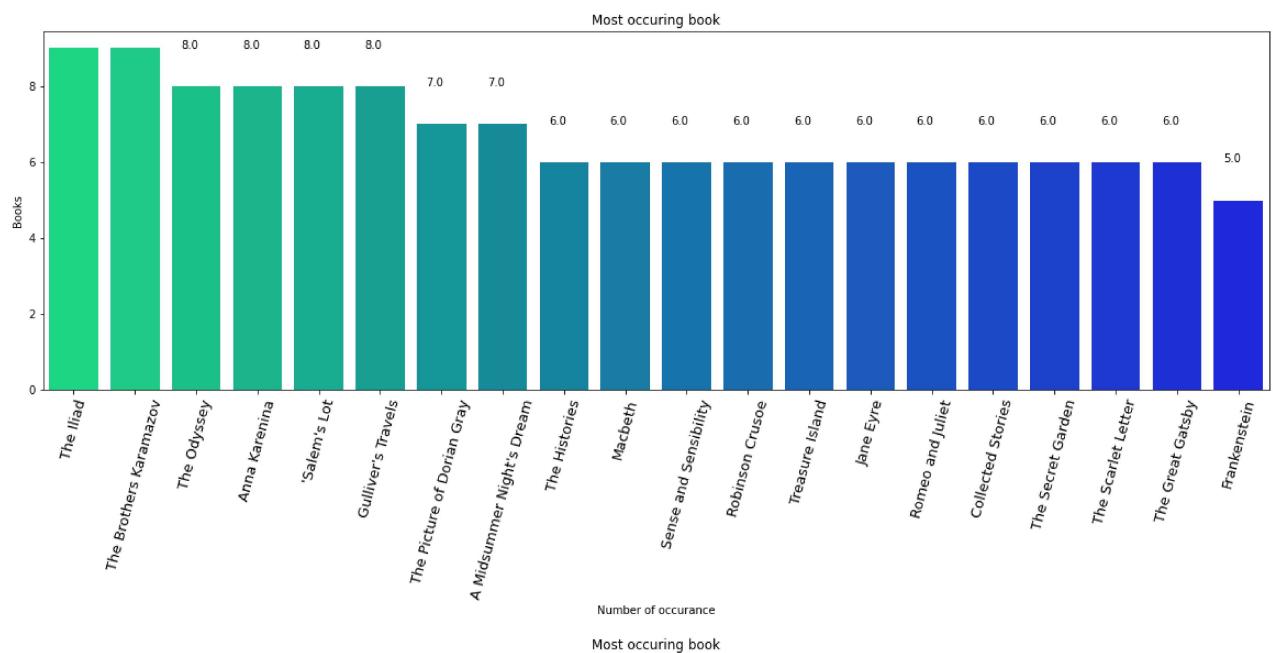
```
In [ ]: book
```

```
Out[ ]: The Iliad          9
        The Brothers Karamazov    9
        The Odyssey            8
        Anna Karenina          8
        'Salem's Lot           8
        Gulliver's Travels      8
        The Picture of Dorian Gray 7
        A Midsummer Night's Dream 7
        The Histories          6
        Macbeth                6
        Sense and Sensibility   6
        Robinson Crusoe         6
        Treasure Island          6
```

```
Jane Eyre          6
Romeo and Juliet  6
Collected Stories 6
The Secret Garden 6
The Scarlet Letter 6
The Great Gatsby   6
Frankenstein      5
Name: title, dtype: int64
Out[ ]: 
The Iliad          9
The Brothers Karamazov 9
The Odyssey        8
Anna Karenina      8
'Salem's Lot        8
Gulliver's Travels 8
The Picture of Dorian Gray 7
A Midsummer Night's Dream 7
The Histories       6
Macbeth            6
Sense and Sensibility 6
Robinson Crusoe    6
Treasure Island    6
Jane Eyre          6
Romeo and Juliet   6
Collected Stories  6
The Secret Garden  6
The Scarlet Letter 6
The Great Gatsby   6
Frankenstein      5
Name: title, dtype: int64
```

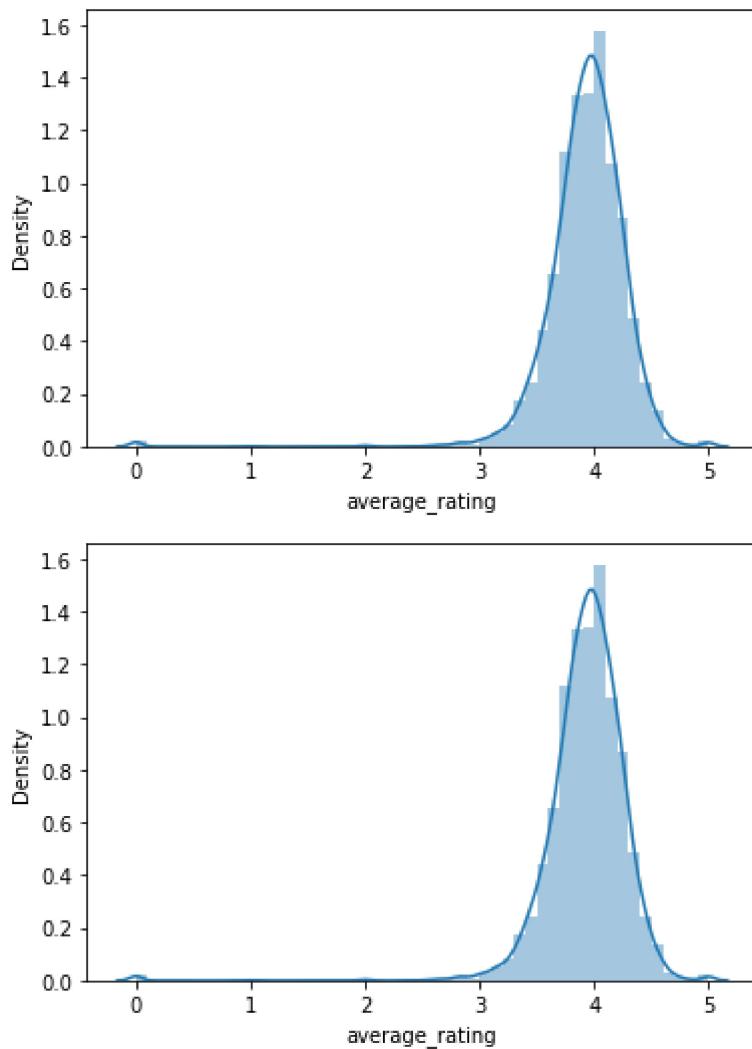
Plotting Barplot to find most occurring book in our data

```
In [ ]: plt.figure(figsize = (20, 6))
book = df['title'].value_counts()[:20]
ax = sns.barplot(x = book.index, y = book,
                  palette = 'winter_r')
plt.title("Most occurring book")
plt.xlabel("Number of occurrence")
plt.ylabel("Books")
plt.xticks(rotation = 75, fontsize=13)
for p in ax.patches:
    ax.annotate(format(p.get_height()), (p.get_x()+0.15, p.get_height()+1))
plt.show()
```



Ploting Distribution Graph on Average_rating.m

```
In [ ]: sns.distplot(df['average_rating'])
plt.show()
```



Sorting Dataset related with maximum average_rating Column

```
In [ ]: df[df.average_rating == df.average_rating.max()][['title','authors','average_rating','language_code','publishe
```

| | | title | authors | average_rating | language_code | publishe |
|------|--|---------------------------------------------------|--------------------------------------|----------------|---------------|-----------------------------|
| 624 | | Comoediae 1: Acharenses/Equites/Nubes/Vespae/P... | Aristophanes/F.W. Hall/W.M. Geldart | 5.0 | grc | Oxford University Press USA |
| 786 | | Willem de Kooning: Late Paintings | Julie Sylvester/David Sylvester | 5.0 | eng | Schirme Mose |
| 855 | | Literature Circle Guide: Bridge to Terabithia:... | Tara MacCarthy | 5.0 | eng | Teaching Resource |
| 1243 | | Middlesex Borough (Images of America: New Jersey) | Middlesex Borough Heritage Committee | 5.0 | eng | Arcadia Publishing |
| 4125 | | Zone of the Enders: The 2nd Runner Official St... | Tim Bogenn | 5.0 | eng | BradyGame |

| | | title | authors | average_rating | language_code | publishe |
|-------|---------------------------------------------------|-----------------------------------------------|---------|----------------|---------------|--------------------------------|
| 4788 | The Diamond Color Meditation: Color Pathway to... | John Diamond | | 5.0 | eng | Square One Publisher |
| 4933 | Bulgakov's the Master and Margarita: The Text ... | Elena N. Mahlow | | 5.0 | eng | Vantage Pres |
| 5023 | The Complete Theory Fun Factory: Music Theory ... | Ian Martin/Katie Elliott | | 5.0 | eng | Boosey & Hawkes Inc |
| 5474 | The Goon Show Volume 4: My Knees Have Fallen ... | NOT A BOOK | | 5.0 | eng | BBC Physica Audi |
| 5476 | The Goon Show Volume 11: He's Fallen in the W... | NOT A BOOK | | 5.0 | eng | BBC Physica Audi |
| 5647 | Winchester Shotguns | Dennis Adler/R.L. Wilson | | 5.0 | eng | Chartwel Book |
| 5648 | Colossians and Philemon: A Critical and Exeget... | R. McL. Wilson | | 5.0 | eng | T&T Clar Int' |
| 6184 | Taxation of Mineral Rents | Ross Garnaut | | 5.0 | eng | Oxford University Press USA |
| 6247 | The New Big Book of America | Todd Davis/Marc Frey | | 5.0 | eng | Courage Book |
| 6775 | Delwau Duon: Peintiadau Nicholas Evans = Symph... | Nicholas Evans/Rhonda Evans | | 5.0 | wel | Y Lolfa |
| 8544 | Fanning the Flame: Bible Cross and Mission | Chris Green/Chris Wright/Paul Douglas Gardner | | 5.0 | eng | Zondervan |
| 9282 | Oliver Wendell Holmes in Paris: Medicine Theo... | William C. Dowling | | 5.0 | eng | University Press o Nev England |
| 9324 | Tyrannosaurus Wrecks (Stanley #1) | Laura Driscoll/Alisa Klayman-Grodsky/Eric ... | | 5.0 | eng | Disney Pres |
| 9720 | The Irish Anatomist: A Study of Flann O'Brien | Keith Donohue | | 5.0 | eng | Academic Pres |
| 9847 | The American Campaign: U.S. Presidential Campa... | James E. Campbell | | 5.0 | eng | Texas A&M University Pres |
| 9893 | His Princess Devotional: A Royal Encounter Wit... | Sheri Rose Shepherd | | 5.0 | eng | Multnomal |
| 10262 | Bill Gates: Computer Legend (Famous Lives) | Sara Barton-Wood | | 5.0 | eng | Raintree |



Out[]:

| | title | authors | average_rating | language_code | publishe |
|--|-------|---------|----------------|---------------|----------|
|--|-------|---------|----------------|---------------|----------|

| | | title | authors | average_rating | language_code | publishe |
|------|--|---------------------------------------------------|-----------------------------------------------|----------------|---------------|--------------------------------|
| 624 | | Comoediae 1: Acharenses/Equites/Nubes/Vespae/P... | Aristophanes/F.W. Hall/W.M. Geldart | 5.0 | grc | Oxford University Press US/ |
| 786 | | Willem de Kooning: Late Paintings | Julie Sylvester/David Sylvester | 5.0 | eng | Schirme Mose |
| 855 | | Literature Circle Guide: Bridge to Terabithia:... | Tara MacCarthy | 5.0 | eng | Teaching Resource: |
| 1243 | | Middlesex Borough (Images of America: New Jersey) | Middlesex Borough Heritage Committee | 5.0 | eng | Arcadia Publishing |
| 4125 | | Zone of the Enders: The 2nd Runner Official St... | Tim Bogenn | 5.0 | eng | BradyGame: |
| 4788 | | The Diamond Color Meditation: Color Pathway to... | John Diamond | 5.0 | eng | Square One Publisher: |
| 4933 | | Bulgakov's the Master and Margarita: The Text ... | Elena N. Mahlow | 5.0 | eng | Vantage Pres: |
| 5023 | | The Complete Theory Fun Factory: Music Theory ... | Ian Martin/Katie Elliott | 5.0 | eng | Boosey & Hawkes Inc |
| 5474 | | The Goon Show Volume 4: My Knees Have Fallen ... | NOT A BOOK | 5.0 | eng | BBC Physical Audic |
| 5476 | | The Goon Show Volume 11: He's Fallen in the W... | NOT A BOOK | 5.0 | eng | BBC Physical Audic |
| 5647 | | Winchester Shotguns | Dennis Adler/R.L. Wilson | 5.0 | eng | Chartwell Book: |
| 5648 | | Colossians and Philemon: A Critical and Exeget... | R. McL. Wilson | 5.0 | eng | T&T Clark Int' |
| 6184 | | Taxation of Mineral Rents | Ross Garnaut | 5.0 | eng | Oxford University Press US/ |
| 6247 | | The New Big Book of America | Todd Davis/Marc Frey | 5.0 | eng | Courage Book: |
| 6775 | | Delwau Duon: Peintiadau Nicholas Evans = Symph... | Nicholas Evans/Rhonda Evans | 5.0 | wel | Y Lolfa: |
| 8544 | | Fanning the Flame: Bible Cross and Mission | Chris Green/Chris Wright/Paul Douglas Gardner | 5.0 | eng | Zondervan |
| 9282 | | Oliver Wendell Holmes in Paris: Medicine Theo... | William C. Dowling | 5.0 | eng | University Press o Nev England |

| | | title | authors | average_rating | language_code | publishe |
|-------|--|---------------------------------------------------|-----------------------------------------------|----------------|---------------|----------------------------|
| 9324 | | Tyrannosaurus Wrecks (Stanley #1) | Laura Driscoll/Alisa Klayman-Grodsky/Eric ... | 5.0 | eng | Disney Pres: |
| 9720 | | The Irish Anatomist: A Study of Flann O'Brien | Keith Donohue | 5.0 | eng | Academic Pres: |
| 9847 | | The American Campaign: U.S. Presidential Campa... | James E. Campbell | 5.0 | eng | Texas A&M University Pres: |
| 9893 | | His Princess Devotional: A Royal Encounter Wit... | Sheri Rose Shepherd | 5.0 | eng | Multnomah |
| 10262 | | Bill Gates: Computer Legend (Famous Lives) | Sara Barton-Wood | 5.0 | eng | Raintree |



Checking Top 20 Publisher in dataset

```
In [ ]: publisher = df['publisher'].value_counts()[:20]
publisher
```

```
Out[ ]: Vintage                318
Penguin Books            261
Penguin Classics          184
Mariner Books             150
Ballantine Books          144
Harper Perennial           112
HarperCollins              112
Pocket Books                111
Bantam                      110
VIZ Media LLC                 88
Berkley                      86
Dover Publications            85
Modern Library                  82
Del Rey                        80
Tor Books                         76
Grand Central Publishing        76
Oxford University Press USA      75
Oxford University Press          73
Scribner                      73
W. W. Norton Company            68
Name: publisher, dtype: int64
```

```
Out[ ]: Vintage                318
Penguin Books            261
Penguin Classics          184
Mariner Books             150
Ballantine Books          144
Harper Perennial           112
HarperCollins              112
Pocket Books                111
Bantam                      110
VIZ Media LLC                 88
```

| | |
|-----------------------------|----|
| Berkley | 86 |
| Dover Publications | 85 |
| Modern Library | 82 |
| Del Rey | 80 |
| Tor Books | 76 |
| Grand Central Publishing | 76 |
| Oxford University Press USA | 75 |
| Oxford University Press | 73 |
| Scribner | 73 |
| W. W. Norton Company | 68 |

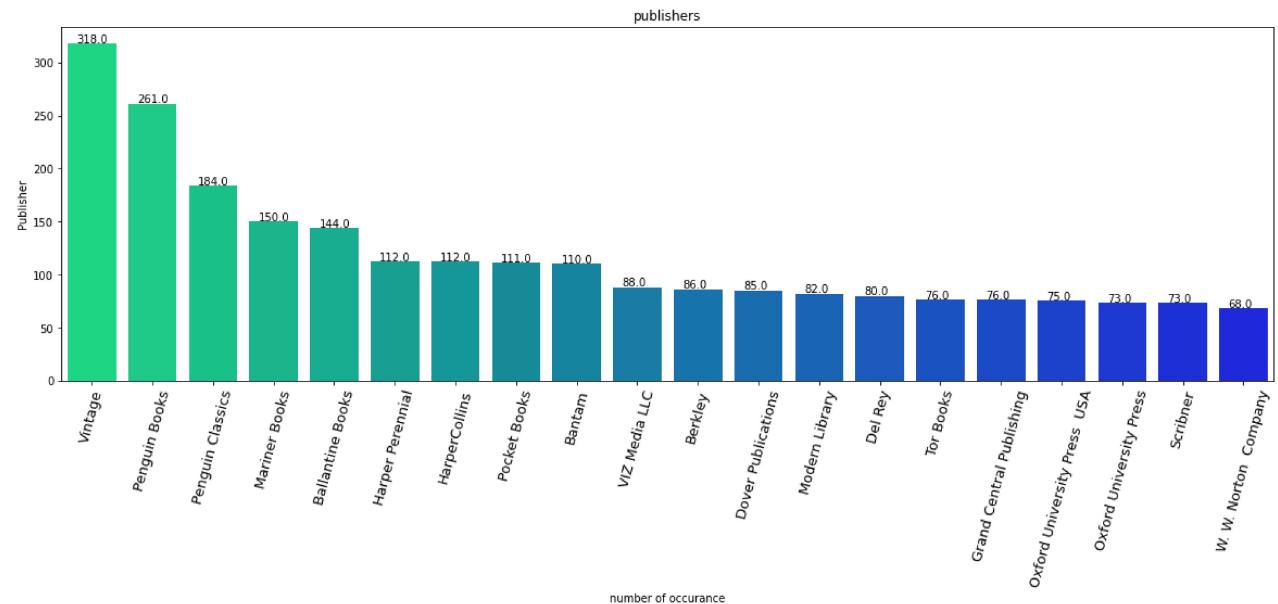
Name: publisher, dtype: int64

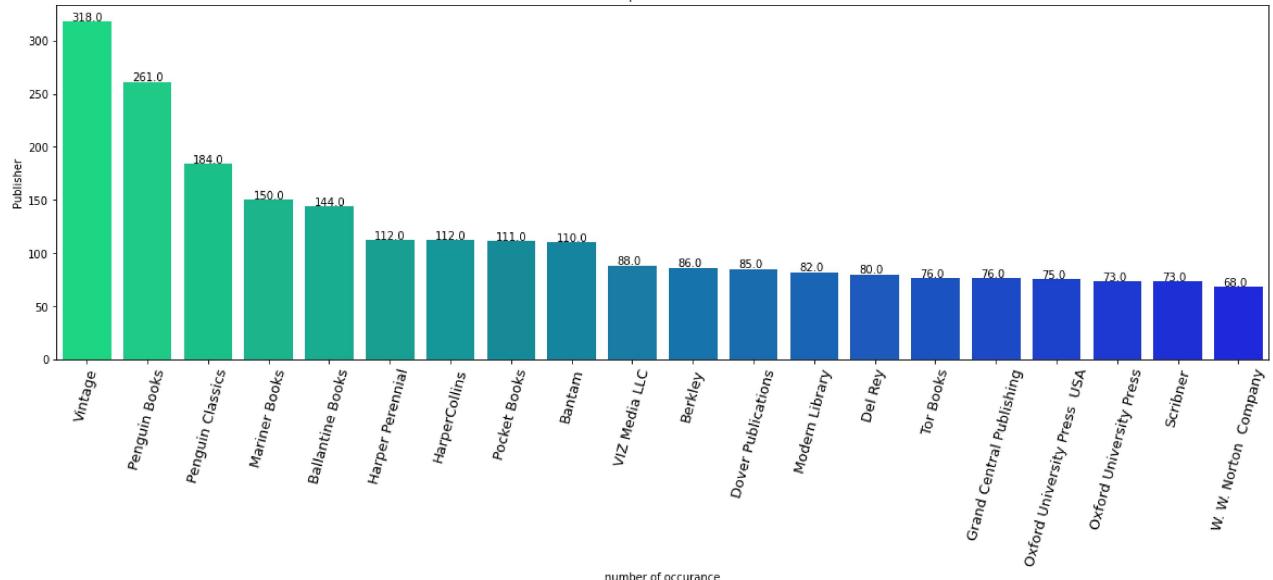
Plotting barplot for Top 20 Publisher

In []:

```
plt.figure(figsize=(20,6))
publisher = df['publisher'].value_counts()[:20]
ax = sns.barplot(x = publisher.index,y=publisher, palette='winter_r')
plt.title('publishers')
plt.xlabel('number of occurrence')
plt.ylabel('Publisher')
plt.xticks(rotation = 75, fontsize = 13)

for p in ax.patches:
    ax.annotate(format(p.get_height()),(p.get_x()+0.15,p.get_height()+1))
plt.show()
```





Now Book Recommendation System Implementation

- Recommendation System based on Publisher
- Recommendation system based on Authors
- Recommendation System based on Language

Recommendation System based on Publisher

```
In [ ]: df.publisher.value_counts()
```

```
Out[ ]:
Vintage                318
Penguin Books           261
Penguin Classics         184
Mariner Books            150
Ballantine Books          144
...
University of Calgary Press    1
Marlowe & Company          1
University Press of America    1
Abstract Studio             1
VeloPress                  1
Name: publisher, Length: 2290, dtype: int64
```

```
Out[ ]:
Vintage                318
Penguin Books           261
Penguin Classics         184
Mariner Books            150
Ballantine Books          144
...
University of Calgary Press    1
Marlowe & Company          1
University Press of America    1
Abstract Studio             1
VeloPress                  1
Name: publisher, Length: 2290, dtype: int64
```

```
In [ ]: df.columns
```

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

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

Defining Function For Recommendation System based on Publisher

```
In [ ]: def recomd_books_publishers(x):
    a = df[df['publisher'] == x][['title','average_rating','year']]
    a = a.sort_values(by = 'average_rating', ascending = False)
    return a.head(10)
```

```
In [ ]: recomd_books_publishers('Vintage')
```

```
Out[ ]:
```

| | | title | average_rating | year |
|--------------|---------------------------------------------------|-------|----------------|------|
| 7371 | Remembrance of Things Past: Volume II - The Gu... | | 4.53 | 1982 |
| 335 | The Power Broker: Robert Moses and the Fall of... | | 4.51 | 1975 |
| 10838 | The Civil War Vol. 1: Fort Sumter to Perryville | | 4.42 | 1986 |
| 1775 | The Son Avenger (The Master of Hestviken #4) | | 4.40 | 1995 |
| 1505 | A Fine Balance | | 4.36 | 2001 |
| 9626 | Nobody Knows My Name | | 4.35 | 1992 |
| 2267 | The Stories of Vladimir Nabokov | | 4.30 | 1996 |
| 3112 | All of Us: The Collected Poems | | 4.30 | 2000 |
| 8787 | Selected Stories | | 4.28 | 1995 |
| 4019 | Selected Stories | | 4.28 | 1997 |

```
Out[ ]:
```

| | | title | average_rating | year |
|--------------|---------------------------------------------------|-------|----------------|------|
| 7371 | Remembrance of Things Past: Volume II - The Gu... | | 4.53 | 1982 |
| 335 | The Power Broker: Robert Moses and the Fall of... | | 4.51 | 1975 |
| 10838 | The Civil War Vol. 1: Fort Sumter to Perryville | | 4.42 | 1986 |
| 1775 | The Son Avenger (The Master of Hestviken #4) | | 4.40 | 1995 |
| 1505 | A Fine Balance | | 4.36 | 2001 |
| 9626 | Nobody Knows My Name | | 4.35 | 1992 |
| 2267 | The Stories of Vladimir Nabokov | | 4.30 | 1996 |
| 3112 | All of Us: The Collected Poems | | 4.30 | 2000 |

| | | title | average_rating | year |
|------|--|------------------|----------------|------|
| 8787 | | Selected Stories | 4.28 | 1995 |
| 4019 | | Selected Stories | 4.28 | 1997 |

In []:

```
recomd_books_publishers('Penguin Books')
```

Out[]:

| | | title | average_rating | year |
|-------|--|-------------------------------------------------|----------------|------|
| 4244 | | The Complete Maus | 4.55 | 2003 |
| 5564 | | The Penguin Companion to European Literature | 4.50 | 1969 |
| 1381 | | Before The Mayflower A History of Black America | 4.44 | 1984 |
| 4602 | | Selected Non-Fictions | 4.43 | 2000 |
| 3011 | | The Read-Aloud Handbook | 4.41 | 2006 |
| 4551 | | Life With Jeeves (Jeeves #6 2 & 4) | 4.39 | 1983 |
| 1275 | | East of Eden | 4.37 | 2002 |
| 3304 | | Ludwig Wittgenstein: The Duty of Genius | 4.36 | 1991 |
| 4980 | | Life at Blandings | 4.35 | 1988 |
| 10867 | | The Portable Dorothy Parker | 4.34 | 2006 |

Out[]:

| | | title | average_rating | year |
|-------|--|-------------------------------------------------|----------------|------|
| 4244 | | The Complete Maus | 4.55 | 2003 |
| 5564 | | The Penguin Companion to European Literature | 4.50 | 1969 |
| 1381 | | Before The Mayflower A History of Black America | 4.44 | 1984 |
| 4602 | | Selected Non-Fictions | 4.43 | 2000 |
| 3011 | | The Read-Aloud Handbook | 4.41 | 2006 |
| 4551 | | Life With Jeeves (Jeeves #6 2 & 4) | 4.39 | 1983 |
| 1275 | | East of Eden | 4.37 | 2002 |
| 3304 | | Ludwig Wittgenstein: The Duty of Genius | 4.36 | 1991 |
| 4980 | | Life at Blandings | 4.35 | 1988 |
| 10867 | | The Portable Dorothy Parker | 4.34 | 2006 |

Creating Interactive Plotting Using ipywidgets

In []:

```
@interact
def recomd_books_publishers(publisher_name = list(df['publisher'].value_counts().index)
                           a = df[df['publisher'] == publisher_name][['title','average_rating']]
                           a = a.sort_values(by= 'average_rating',ascending = False)
                           return a.head(10)
```

```
In [ ]: df.columns
```

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

Recommending Books based on Authors

```
In [ ]: @interact
def recomd_books_authors(authors_name = list(df['authors'].value_counts().index)):
    a = df[df['authors'] == authors_name][['title','average_rating']]
    a = a.sort_values(by = 'average_rating', ascending = False)
    return a.head(10)
```

```
In [ ]: df.columns
```

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

Recommendation System based on Language

```
In [ ]: @interact
def recomd_books_lang (language = list(df['language_code'].value_counts().index)):
    a = df[df['language_code'] == language][['title','average_rating']]
    a = a.sort_values(by = 'average_rating', ascending = False)
    return a.head(15)
```

Data Processing

```
In [ ]: df.head()
```

```
Out[ ]:      title     authors  average_rating  language_code  num_pages  ratings_count  text_reviews_coun
```

| | title | authors | average_rating | language_code | num_pages | ratings_count | text_reviews_count |
|---|----------------------------------------------------|----------------------------|-----------------------|----------------------|------------------|----------------------|---------------------------|
| 0 | Harry Potter and the Half-Blood Prince (Harry ...) | J.K. Rowling/Mary GrandPré | 4.57 | eng | 652 | 2095690 | 27591 |
| 1 | Harry Potter and the Order of the Phoenix (Har... | J.K. Rowling/Mary GrandPré | 4.49 | eng | 870 | 2153167 | 29221 |
| 2 | Harry Potter and the Chamber of Secrets (Harry... | J.K. Rowling | 4.42 | eng | 352 | 6333 | 244 |
| 3 | Harry Potter and the Prisoner of Azkaban (Harr... | J.K. Rowling/Mary GrandPré | 4.56 | eng | 435 | 2339585 | 36321 |
| 4 | Harry Potter Boxed Set Books 1-5 (Harry Potte... | J.K. Rowling/Mary GrandPré | 4.78 | eng | 2690 | 41428 | 164 |

| ◀ | ▶ | | | | | | |
|----------------|----------------------------------------------------|----------------------------|-----------------------|----------------------|------------------|----------------------|---------------------------|
| Out[]: | | | | | | | |
| | title | authors | average_rating | language_code | num_pages | ratings_count | text_reviews_count |
| 0 | Harry Potter and the Half-Blood Prince (Harry ...) | J.K. Rowling/Mary GrandPré | 4.57 | eng | 652 | 2095690 | 27591 |
| 1 | Harry Potter and the Order of the Phoenix (Har... | J.K. Rowling/Mary GrandPré | 4.49 | eng | 870 | 2153167 | 29221 |

| | title | authors | average_rating | language_code | num_pages | ratings_count | text_reviews_coun |
|---|---------------------------------------------------|----------------------------|----------------|---------------|-----------|---------------|-------------------|
| 2 | Harry Potter and the Chamber of Secrets (Harry... | J.K. Rowling | 4.42 | eng | 352 | 6333 | 24 |
| 3 | Harry Potter and the Prisoner of Azkaban (Harr... | J.K. Rowling/Mary GrandPré | 4.56 | eng | 435 | 2339585 | 3632! |
| 4 | Harry Potter Boxed Set Books 1-5 (Harry Pott... | J.K. Rowling/Mary GrandPré | 4.78 | eng | 2690 | 41428 | 16 |

Creating Function for Converting number to object on Average_Rating Column.

```
In [ ]: 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)
```

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

```
Out[ ]: 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
Out[ ]: 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
```

In []:

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

Out[]:

| | 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 |

Out[]:

| | 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 |

In []:

```
df.columns
```

Out[]:

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

Out[]:

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

In []:

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

Out[]:

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

5 rows × 27 columns

Out[]:

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

5 rows × 27 columns

In []:

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

Out[]:

| | 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 | | | |
|--------------------------------------------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|-----|-----|-------|-------|-------|-----|-----|-----|-----|---|---|---|
| title | | | | | | | | | | | | | | | | | |
| 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 |
| 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 |
| 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 |

| | 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 |
|------------------------------------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|-----|-----|-----------|-----------|-----------|-----|-----|-----|-----|
| title | | | | | | | | | | | | | | |
| Harry Potter and the Prisoner of Azkaban (Harry Potter #3) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| Harry Potter Boxed Set Books 1-5 (Harry Potter #1-5) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| | | | | | | | | | | | | | | |

5 rows × 34 columns

| Out[]: | 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 |
|-------------------------------------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|-----|-----|-----------|-----------|-----------|-----|-----|-----|-----|
| title | | | | | | | | | | | | | | |
| Harry Potter and the Half-Blood Prince (Harry Potter #6) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| Harry Potter and the Order of the Phoenix (Harry Potter #5) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| Harry Potter and the Chamber of Secrets (Harry Potter #2) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| | | | | | | | | | | | | | | |

| | 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 |
|------------------------------------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|-----|-----|-----------|-----------|-----------|-----|-----|-----|-----|
| title | | | | | | | | | | | | | | |
| Harry Potter and the Prisoner of Azkaban (Harry Potter #3) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| Harry Potter Boxed Set Books 1-5 (Harry Potter #1-5) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| | | | | | | | | | | | | | | |

5 rows × 34 columns



```
In [ ]: from sklearn.preprocessing import MinMaxScaler
```

```
In [ ]: scaler = MinMaxScaler()
features_scaled = scaler.fit_transform(features)
```

```
In [ ]: features_scaled
```

```
Out[ ]: array([[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 9.1400000e-01, 4.55816060e-01],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 8.9800000e-01, 4.68317403e-01],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 8.8400000e-01, 1.37743803e-03],
...,
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 7.9200000e-01, 1.78351363e-04],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 7.4400000e-01, 1.67258779e-04],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 7.8200000e-01, 2.45776879e-05]])
```

```
Out[ ]: array([[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 9.1400000e-01, 4.55816060e-01],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 8.9800000e-01, 4.68317403e-01],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 8.8400000e-01, 1.37743803e-03],
...,
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 7.9200000e-01, 1.78351363e-04],
```

```
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 7.4400000e-01, 1.67258779e-04],
[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ...,
 0.0000000e+00, 7.8200000e-01, 2.45776879e-05]])
```

model Building

```
In [ ]: from sklearn import neighbors
```

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

```
In [ ]: df['title'].value_counts()
```

```
Out[ ]: The Iliad                                9
The Brothers Karamazov                         9
The Odyssey                                    8
Anna Karenina                                 8
'Salem's Lot                                  8
                                           ..
The Noonday Demon: An Atlas of Depression      1
The Noonday Demon: An Anatomy of Depression    1
My Secret: A PostSecret Book                  1
The Secret Lives of Men and Women: A PostSecret Book 1
Las aventuras de Tom Sawyer                   1
Name: title, Length: 10348, dtype: int64

Out[ ]: The Iliad                                9
The Brothers Karamazov                         9
The Odyssey                                    8
Anna Karenina                                 8
'Salem's Lot                                  8
                                           ..
The Noonday Demon: An Atlas of Depression      1
The Noonday Demon: An Anatomy of Depression    1
My Secret: A PostSecret Book                  1
The Secret Lives of Men and Women: A PostSecret Book 1
Las aventuras de Tom Sawyer                   1
Name: title, Length: 10348, dtype: int64
```

```
In [135...]: @interact
def BookRecomender(book_name = list(df['title'].value_counts().index)):
    book_list_name = []
    book_id = df[df['title'] == book_name]
    book_id = book_id.index[0]
    for newid in idlist[book_id]:
        book_list_name.append(df.iloc[newid].title)
    return book_list_name
```

Thank you

Mohammad Aamir Rangrez

In []: