Title:Book Recommandation

Problem Statement:

With the rapid growth of digital content and online book platforms, users often face the challenge of choosing which books to read next. The sheer volume of available books makes it difficult to discover titles that match individual preferences. To address this, a recommendation system is needed that can suggest books to users based on their reading behavior and preferences.This project aims to build a Book Recommendation System using collaborative filtering techniques, leveraging datasets of books, user ratings, and user demographics. The goal is to provide personalized book recommendations by analyzing patterns in user ratings, identifying similar users or books, and suggesting titles that users are likely to enjoy.

Data Description

Column Name Description
ISBN --Unique identifier for each book
Book-Title-- Title of the book Book-Author --Author(s) of the book Year-Of-Publication-- Year the book was published Publishe--Publisher of the book Image-URL-S, Image-URL-M, Image-URL-L -- URLs of small, medium, and large images of the book cover

In [1]:
```
# Import essential libraries for data manipulation and analysis
import numpy as np
import pandas as pd
```

In [2]:
```
# Load the Books dataset into a DataFrame
books = pd.read_csv("Books.csv")
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_2764\2715094300.py:2: DtypeWarnin
g: Columns (3) have mixed types. Specify dtype option on import or set low_memo
ry=False.
  books = pd.read_csv("Books.csv")
```

In [3]:
```
# Preview the first few rows of the dataset
books.head()
```

Out[3]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amaz images/P/019515
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amaz images/P/000200
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amaz images/P/006097
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amaz images/P/037415
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amaz images/P/039304

In [4]:
```
# Display the column names of the dataset
books.columns
```

Out[4]:
```
Index(['ISBN', 'Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publishe
r',
       'Image-URL-S', 'Image-URL-M', 'Image-URL-L'],
      dtype='object')
```

In [5]:
```
# Select only relevant columns from the Books dataset
books = books[['ISBN', 'Book-Title', 'Book-Author', 'Year-Of-Publication', 'Pu
```

In [6]:
```
# Preview the first few rows of the dataset
books.head()
```

Out[6]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company

In [7]:
```
# Display the column names of the dataset
books.rename(columns={'Book-Title':'title','Book-Author':'author','Year-Of-Pub
```

In [8]:
```
# Preview the first few rows of the dataset
books.head()
```

Out[8]:

	ISBN	title	author	year	publish
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company

In [9]:
```
# Load the Users dataset into a DataFrame
users= pd.read_csv("Users.csv")
```

In [10]:
```
# Preview the first few rows of the dataset
users.head()
```

Out[10]:

	User-ID	Location	Age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0
2	3	moscow, yukon territory, russia	NaN
3	4	porto, v.n.gaia, portugal	17.0
4	5	farnborough, hants, united kingdom	NaN

In [11]:
```
# Display the column names of the dataset
users.rename(columns={'User-ID':'user_id','Location':'location','Age':'age'},i
```

In [12]:
```
# Preview the first few rows of the dataset
users.head()
```

Out[12]:

	user_id	location	age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0
2	3	moscow, yukon territory, russia	NaN
3	4	porto, v.n.gaia, portugal	17.0
4	5	farnborough, hants, united kingdom	NaN

In [13]:
```
ratings = pd.read_csv('Ratings.csv')
```

In [14]:
```
# Preview the first few rows of the dataset
ratings.head()
```

Out[14]:

	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

In [15]:
```
# Display the column names of the dataset
ratings.rename(columns={'User-ID':'user_id','Book-Rating':'rating'},inplace=Tr
```

In [16]:
```
# Preview the first few rows of the dataset
ratings.head()
```

Out[16]:

	user_id	ISBN	rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

In [17]:
```
books.shape
```

Out[17]:
```
(271360, 5)
```

In [18]:
```
users.shape
```

Out[18]:
```
(278858, 3)
```

In [19]:
```
ratings.shape
```

Out[19]:
```
(1149780, 3)
```

In [20]:
```
ratings['user_id'].value_counts()
```

Out[20]:
```
user_id
11676     13602
198711     7550
153662     6109
98391      5891
35859      5850
          ...
116180        1
116166        1
116154        1
116137        1
276723        1
Name: count, Length: 105283, dtype: int64
```

In [21]:
```
ratings['user_id'].value_counts().shape
```

Out[21]:
```
(105283,)
```

In [22]:
```
x=ratings['user_id'].value_counts() > 200
```

In [23]:
```
x[x].shape
```

Out[23]:
```
(899,)
```

In [24]:
```
y=x[x].index
```

In [25]:
```
y
```

Out[25]:
```
Index([ 11676, 198711, 153662,  98391,  35859, 212898, 278418,  76352, 110973,
       235105,
       ...
       260183,  73681,  44296, 155916,   9856, 274808,  28634,  59727, 268622,
       188951],
      dtype='int64', name='user_id', length=899)
```

In [26]:
```
ratings=ratings[ratings['user_id'].isin(y)]
```

In [27]:
```
ratings.shape
```

Out[27]:
```
(526356, 3)
```

In [28]:
```
# Preview the first few rows of the dataset
ratings.head()
```

Out[28]:

	user_id	ISBN	rating
1456	277427	002542730X	10
1457	277427	0026217457	0
1458	277427	003008685X	8
1459	277427	0030615321	0
1460	277427	0060002050	0

In [29]:
```
ratings_with_books = ratings.merge(books, on ='ISBN')
```

In [30]:
```
ratings_with_books
```

Out[30]:

	user_id	ISBN	rating	title	author	year	p
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	Joh & S
1	277427	0026217457	0	Vegetarian Times Complete Cookbook	Lucy Moll	1995	Joh &am
2	277427	003008685X	8	Pioneers	James Fenimore Cooper	1974	Th L
3	277427	0030615321	0	Ask for May, Settle for June (A Doonesbury book)	G. B. Trudeau	1982	Her &a
4	277427	0060002050	0	On a Wicked Dawn (Cynster Novels)	Stephanie Laurens	2002	Avor
...	...	...	...	...	...	...	
487666	275970	1931868123	0	There's a Porcupine in My Outhouse: Misadventu...	Mike Tougias	2002	Capita
487667	275970	3411086211	10	Die Biene.	Sybil GrÃ?Â¤fin SchÃ?Â¶nfeldt	1993	Bibliograp I Ma
487668	275970	3829021860	0	The Penis Book	Joseph Cohen	1999	Kon
487669	275970	4770019572	0	Musashi	Eiji Yoshikawa	1995	Ko Interr
487670	275970	9626340762	8	Northanger Abbey (Classic Literature with Clas...	Jane Austen	1996	Audiobo

487671 rows × 7 columns

In [31]:
```
number_rating=ratings_with_books.groupby('title')['rating'].count().reset_inde
```

In [32]:
```
# Display the column names of the dataset
number_rating.rename(columns={'rating':'number of ratings'},inplace=True)
```

In [33]:
```
number_rating
```

Out[33]:

	title	number of ratings
0	A Light in the Storm: The Civil War Diary of ...	2
1	Always Have Popsicles	1
2	Apple Magic (The Collector's series)	1
3	Beyond IBM: Leadership Marketing and Finance ...	1
4	Clifford Visita El Hospital (Clifford El Gran...	1
...	...	...
160264	Ã?Â?ber die Pflicht zum Ungehorsam gegen den S...	3
160265	Ã?Â?lpiraten.	1
160266	Ã?Â?rger mit Produkt X. Roman.	1
160267	Ã?Â?stlich der Berge.	1
160268	Ã?Â?thique en toc	1

160269 rows × 2 columns

In [34]:
```
final_rating = ratings_with_books.merge(number_rating,on='title')
```

In [35]:
```
final_rating
```

Out[35]:

	user_id	ISBN	rating	title	author	year	p
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	Joh & S
1	277427	0026217457	0	Vegetarian Times Complete Cookbook	Lucy Moll	1995	Joh &am
2	277427	003008685X	8	Pioneers	James Fenimore Cooper	1974	Th L
3	277427	0030615321	0	Ask for May, Settle for June (A Doonesbury book)	G. B. Trudeau	1982	Her &a
4	277427	0060002050	0	On a Wicked Dawn (Cynster Novels)	Stephanie Laurens	2002	Avor
...	...	...	...	...	...	...	
487666	275970	1931868123	0	There's a Porcupine in My Outhouse: Misadventu...	Mike Tougias	2002	Capita
487667	275970	3411086211	10	Die Biene.	Sybil GrÃ?Â¤fin SchÃ?Â¶nfeldt	1993	Bibliograp I Ma
487668	275970	3829021860	0	The Penis Book	Joseph Cohen	1999	Kon
487669	275970	4770019572	0	Musashi	Eiji Yoshikawa	1995	Ko Interr
487670	275970	9626340762	8	Northanger Abbey (Classic Literature with Clas...	Jane Austen	1996	Audiobo

487671 rows × 8 columns

In [36]:
```
final_rating.shape
```

Out[36]: (487671, 8)

In [37]:
```
final_rating=final_rating[final_rating['number of ratings'] >=50]
```

In [38]:
```
final_rating.shape
```

Out[38]: (61853, 8)

In [39]:
```
final_rating.drop_duplicates(['user_id','title'],inplace=True)
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_2764\1573401051.py:1: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy
  final_rating.drop_duplicates(['user_id','title'],inplace=True)
```

In [40]:
```
final_rating.shape
```

Out[40]: (59850, 8)

In [41]:
```
# Display the column names of the dataset
book_pivot = final_rating.pivot_table(columns='user_id',index ='title',values=
```

In [42]:
```
book_pivot
```

Out[42]:

user_id	254	2276	2766	2977	3363	3757	4017	4385	6242	6251	...
title											
1984	9.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
1st to Die: A Novel	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
2nd Chance	NaN	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
4 Blondes	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	...
84 Charing Cross Road	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
...	...	...	...	...	...	...	...	...	...	...	...
Year of Wonders	NaN	NaN	NaN	7.0	NaN	NaN	NaN	NaN	7.0	NaN	...
You Belong To Me	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
Zen and the Art of Motorcycle Maintenance: An Inquiry into Values	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN	0.0	...
Zoya	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
\O\" Is for Outlaw"	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...

742 rows × 888 columns

In [43]:
```
book_pivot.shape
```

Out[43]: (742, 888)

In [44]:
```
book_pivot.fillna(0,inplace =True)
```

In [45]:
```
book_pivot
```

Out[45]:

user_id	254	2276	2766	2977	3363	3757	4017	4385	6242	6251	...
title											
1984	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1st to Die: A Novel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
2nd Chance	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
4 Blondes	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
84 Charing Cross Road	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
Year of Wonders	0.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	7.0	0.0	...
You Belong To Me	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
Zen and the Art of Motorcycle Maintenance: An Inquiry into Values	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
Zoya	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
\O\" Is for Outlaw"	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

742 rows × 888 columns

In [46]:
```
from scipy.sparse import csr_matrix
book_sparse=csr_matrix(book_pivot)
```

In [47]:
```
type(book_sparse)
```

Out[47]:
```
scipy.sparse._csr.csr_matrix
```

In [48]:
```
from sklearn.neighbors import NearestNeighbors
model=NearestNeighbors(algorithm='brute')
```

In [49]:
```
model.fit(book_sparse)
```

Out[49]:
```
NearestNeighbors
NearestNeighbors(algorithm='brute')
```

In [50]:
```
distances, suggestions=model.kneighbors(book_pivot.iloc[237, :].values.reshape
```

In [51]:
```
suggestions
```

Out[51]:
```
array([[237, 238, 240, 241, 184, 536]], dtype=int64)
```

In [52]:
```
for i in range(len(suggestions)):
    print(book_pivot.index[suggestions[i]])
```

```
Index(['Harry Potter and the Chamber of Secrets (Book 2)',
       'Harry Potter and the Goblet of Fire (Book 4)',
       'Harry Potter and the Prisoner of Azkaban (Book 3)',
       'Harry Potter and the Sorcerer's Stone (Book 1)', 'Exclusive',
       'The Cradle Will Fall'],
      dtype='object', name='title')
```

In [53]:
```
book_pivot.index[237]
```

Out[53]:
```
'Harry Potter and the Chamber of Secrets (Book 2)'
```

In [54]:
```
np.where(book_pivot.index =='Animal Farm')[0][0]
```

Out[54]:
```
54
```

In [55]:
```
def recommend_book(book_name):
    book_id=np.where(book_pivot.index ==book_name)[0][0]
    distances, suggestions=model.kneighbors(book_pivot.iloc[book_id, :].values

    for i in range(len(suggestions)):
        if i ==0:
            print("The suggestion for", book_name,"are :")
        if not i:
            print(book_pivot.index[suggestions[i]])
```

In [56]:
```
recommend_book('Animal Farm')
```

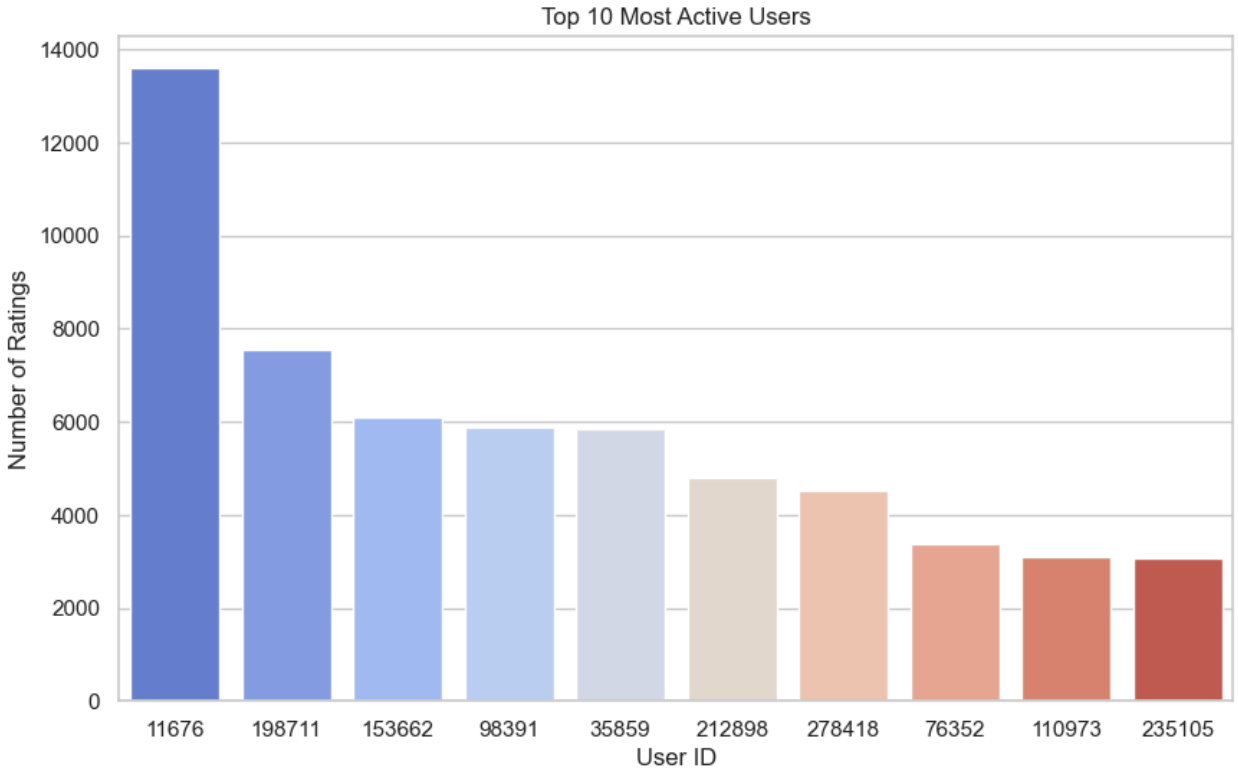
```
The suggestion for Animal Farm are :
Index(['Animal Farm', 'Exclusive', 'Jacob Have I Loved', 'Second Nature',
       'Pleading Guilty', 'No Safe Place'],
      dtype='object', name='title')
```

In [60]:
```
# Import libraries for data visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Set Seaborn style for better visuals
sns.set(style='whitegrid')
```

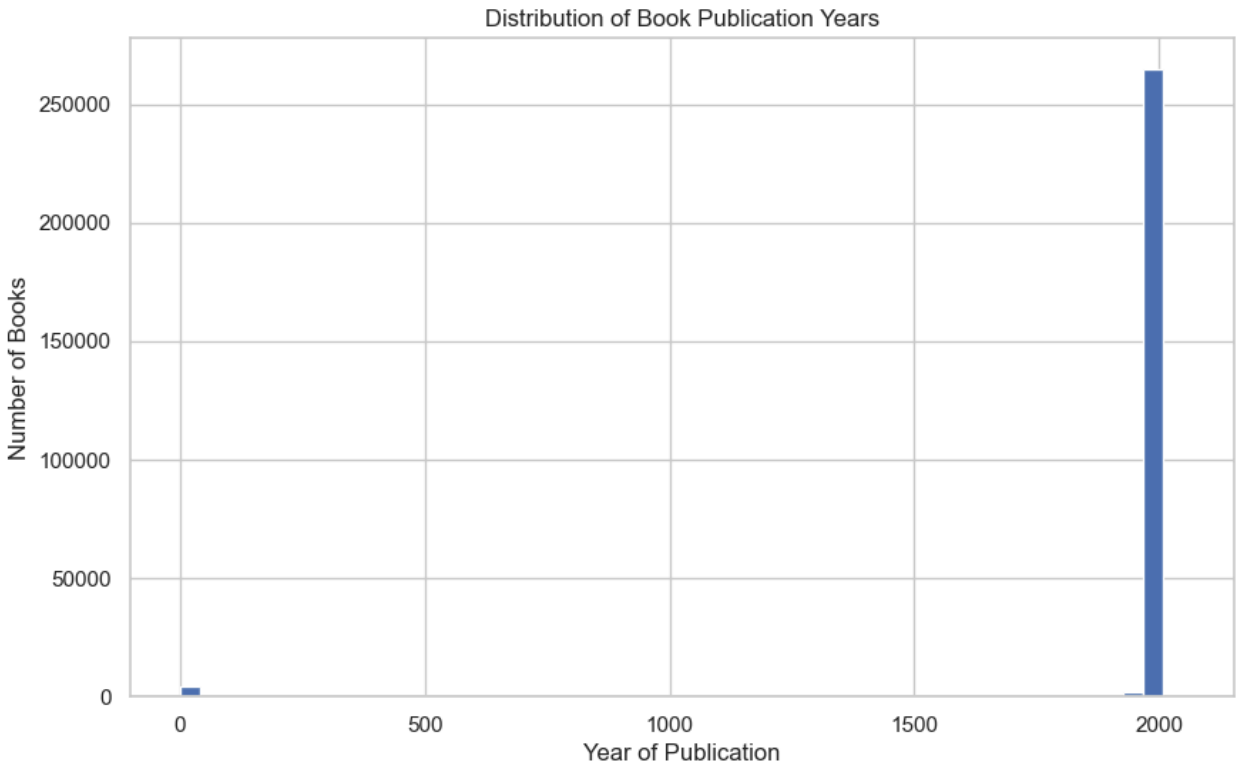
In [61]:
```
# Visualize top users by number of ratings
most_active_users = ratings['user_id'].value_counts().head(10)

plt.figure(figsize=(10,6))
sns.barplot(x=most_active_users.index.astype(str), y=most_active_users.values,
plt.title('Top 10 Most Active Users')
```

```
plt.xlabel('User ID')
plt.ylabel('Number of Ratings')
plt.show()
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_2764\2131958494.py:5: FutureWarnin
g:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same e
ffect.

  sns.barplot(x=most_active_users.index.astype(str), y=most_active_users.value
s, palette='coolwarm')
```



In [72]:
```
# Visualize distribution of publication years
plt.figure(figsize=(10,6))
books['year'] = pd.to_numeric(books['year'], errors='coerce')
books['year'].dropna().astype(int).hist(bins=50)
plt.title('Distribution of Book Publication Years')
plt.xlabel('Year of Publication')
plt.ylabel('Number of Books')
plt.show()
```


Distribution of Book Publication Years
Number of Books
250000
200000
150000
100000
50000
0
0
500
1000
1500
2000
Year of Publication

In [ ]: