In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
books=pd.read_csv('Books_data.csv', sep=";",error_bad_lines =False,encoding='latin-1')
```

```
C:\Users\91766\AppData\Local\Temp\ipykernel_10436\502342762.py:1: FutureWarning: The
error_bad_lines argument has been deprecated and will be removed in a future version.
Use on_bad_lines in the future.


  books=pd.read_csv('Books_data.csv', sep=";",error_bad_lines =False,encoding='latin-
1')
b'Skipping line 6452: expected 8 fields, saw 9\nSkipping line 43667: expected 8 field
s, saw 10\nSkipping line 51751: expected 8 fields, saw 9\n'
b'Skipping line 92038: expected 8 fields, saw 9\nSkipping line 104319: expected 8 fie
lds, saw 9\nSkipping line 121768: expected 8 fields, saw 9\n'
b'Skipping line 144058: expected 8 fields, saw 9\nSkipping line 150789: expected 8 fi
elds, saw 9\nSkipping line 157128: expected 8 fields, saw 9\nSkipping line 180189: ex
pected 8 fields, saw 9\nSkipping line 185738: expected 8 fields, saw 9\n'
b'Skipping line 209388: expected 8 fields, saw 9\nSkipping line 220626: expected 8 fi
elds, saw 9\nSkipping line 227933: expected 8 fields, saw 11\nSkipping line 228957: e
xpected 8 fields, saw 10\nSkipping line 245933: expected 8 fields, saw 9\nSkipping li
ne 251296: expected 8 fields, saw 9\nSkipping line 259941: expected 8 fields, saw 9\n
Skipping line 261529: expected 8 fields, saw 9\n'
C:\Users\91766\AppData\Local\Temp\ipykernel_10436\502342762.py:1: DtypeWarning: Colum
ns (3) have mixed types. Specify dtype option on import or set low_memory=False.
  books=pd.read_csv('Books_data.csv', sep=";",error_bad_lines =False,encoding='latin-
1')
```

In [3]:
```
books.columns
```

Out[3]:
```
Index(['ISBN', 'Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher',
       'Image-URL-S', 'Image-URL-M', 'Image-URL-L'],
      dtype='object')
```

In [4]:
```
books = books[['ISBN','Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher']
```

In [5]:
```
books.rename(columns={'Book-Title':'title', 'Book-Author':'author', 'Year-Of-Publicati
books.head(1)
```

Out[5]:

	ISBN	title	author	year	publisher
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press

In [6]:
```
users= pd.read_csv('Book_user.csv',sep =";",error_bad_lines = False,encoding='latin-1'
users.columns
```

```
C:\Users\91766\AppData\Local\Temp\ipykernel_10436\2620640758.py:1: FutureWarning: The
error_bad_lines argument has been deprecated and will be removed in a future version.
Use on_bad_lines in the future.


  users= pd.read_csv('Book_user.csv',sep =";",error_bad_lines = False,encoding='latin
-1')
```

Out[6]:
```
Index(['User-ID', 'Location', 'Age'], dtype='object')
```

In [7]:
```
users.rename(columns={'User-ID':'user_id', 'Location':'location', 'Age':'age'},inplace
users.head(2)
```

Out[7]:

	user_id	location	age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0

In [8]:
```
ratings = pd.read_csv('Book_rating.csv',sep =";",error_bad_lines = False,encoding='lat
ratings.columns
```

```
C:\Users\91766\AppData\Local\Temp\ipykernel_10436\404777886.py:1: FutureWarning: The
error_bad_lines argument has been deprecated and will be removed in a future version.
Use on_bad_lines in the future.


  ratings = pd.read_csv('Book_rating.csv',sep =";",error_bad_lines = False,encoding
='latin-1')
```

Out[8]:
```
Index(['User-ID', 'ISBN', 'Book-Rating'], dtype='object')
```

In [9]:
```
ratings.rename(columns= {'User-ID':'user_id', 'Book-Rating':'rating'},inplace=True)
ratings.head(3)
```

Out[9]:

	user_id	ISBN	rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0

In [10]:
```
books.shape
```

Out[10]:
```
(271360, 5)
```

In [11]:
```
ratings.shape
```

Out[11]:
```
(1149780, 3)
```

In [12]:
```
users.shape
```

Out[12]:
```
(278858, 3)
```

In [13]:
```
y = ratings['user_id'].value_counts()>200
```

In [14]:
```
x =y[y].index
x
```

Out[14]:
```
Int64Index([ 11676, 198711, 153662,  98391,  35859, 212898, 278418,  76352,
            110973, 235105,
            ...
            260183,  73681,  44296, 155916,   9856, 274808,  28634,  59727,
            268622, 188951],
           dtype='int64', length=899)
```

In [15]:
```
ratings = ratings[ratings['user_id'].isin(x)]
ratings.shape
```

Out[15]:
```
(526356, 3)
```

In [16]:
```
ratings.head()
```

Out[16]:

	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 [17]:
```
ratings_with_books = ratings.merge(books, on ='ISBN')
ratings_with_books
```

Out[17]:

	user_id	ISBN	rating	title	author	year	publisher
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc
1	3363	002542730X	0	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc
2	11676	002542730X	6	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc
3	12538	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc
4	13552	002542730X	0	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc
...	...	...	...	...	...	...	...
487666	275970	1892145022	0	Here Is New York	E. B. White	1999	Little Bookroom
487667	275970	1931868123	0	There's a Porcupine in My Outhouse: Misadventu...	Mike Tougias	2002	Capital Books (VA)
487668	275970	3411086211	10	Die Biene.	Sybil GrÃ? Â¤fin SchÃ? Â¶nfeldt	1993	Bibliographisches Institut, Mannheim
487669	275970	3829021860	0	The Penis Book	Joseph Cohen	1999	Konemann
487670	275970	4770019572	0	Musashi	Eiji Yoshikawa	1995	Kodansha International (JPN)

487671 rows × 7 columns

In [18]:
```
number_rating = ratings_with_books.groupby('title')['rating'].count().reset_index()
```

In [19]:
```
number_rating.rename(columns = {'rating':'number of ratings'},inplace = True)
```

In [20]:
```
final_rating= ratings_with_books.merge(number_rating, on ='title')
final_rating.columns
```

Out[20]:
```
Index(['user_id', 'ISBN', 'rating', 'title', 'author', 'year', 'publisher',
       'number of ratings'],
      dtype='object')
```

In [21]:
```
final_rating
```

Out[21]:

	user_id	ISBN	rating	title	author	year	publisher	number of ratings
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	82
1	3363	002542730X	0	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	82
2	11676	002542730X	6	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	82
3	12538	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	82
4	13552	002542730X	0	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	82
...	...	...	...	...	...	...	...	...
487666	275970	1892145022	0	Here Is New York	E. B. White	1999	Little Bookroom	1
487667	275970	1931868123	0	There's a Porcupine in My Outhouse: Misadventu...	Mike Tougias	2002	Capital Books (VA)	1
487668	275970	3411086211	10	Die Biene.	Sybil GrÃ?Â¤fin SchÃ?Â¶nfeldt	1993	Bibliographisches Institut, Mannheim	1
487669	275970	3829021860	0	The Penis Book	Joseph Cohen	1999	Konemann	1
487670	275970	4770019572	0	Musashi	Eiji Yoshikawa	1995	Kodansha International (JPN)	1

487671 rows × 8 columns

In [22]:
```
final_rating=final_rating[final_rating['number of ratings']>=50]
```

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

Out[23]: (61853, 8)

In [24]:
```
final_rating.drop_duplicates(['user_id','title'],inplace = True)
```

```
C:\Users\91766\AppData\Local\Temp\ipykernel_10436\1092272760.py:1: SettingWithCopyWar
ning:
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/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  final_rating.drop_duplicates(['user_id','title'],inplace = True)
```

In [25]:
```
final_rating.shape
```

Out[25]: (59850, 8)

In [26]:
```
book_pivot = final_rating.pivot_table(columns ='user_id',index ='title',values='rating
```

In [27]:
```
book_pivot
```

Out[27]:

user_id	254	2276	2766	2977	3363	3757	4017	4385	6242	6251	...	274004	274061
title													
1984	9.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
1st to Die: A Novel	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
2nd Chance	NaN	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
4 Blondes	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	...	NaN	NaN
84 Charing Cross Road	NaN	NaN	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	...	NaN	NaN
You Belong To Me	NaN	NaN	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	...	NaN	NaN
Zoya	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
\O\" Is for Outlaw"	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

742 rows × 888 columns

In [28]:
```
book_pivot.fillna(0,inplace= True)
```

In [29]:
```
book_pivot
```

Out[29]:

user_id	254	2276	2766	2977	3363	3757	4017	4385	6242	6251	...	274004	274061
title													
1984	9.0	0.0	0.0	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	...	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	...	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	...	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	...	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	...	0.0	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	...	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	...	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	...	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	...	0.0	0.0

742 rows × 888 columns

In [30]:
```
from scipy.sparse import csr_matrix
book_sparse = csr_matrix(book_pivot)
```

In [31]:
```
type(book_sparse)
```

Out[31]:
```
scipy.sparse._csr.csr_matrix
```

In [32]:
```
from sklearn.neighbors import NearestNeighbors
model = NearestNeighbors(algorithm = 'brute')
```

In [33]:
```
model.fit(book_sparse)
```

Out[33]:
```
NearestNeighbors(algorithm='brute')
```

In [34]:
```
distances , suggestions = model.kneighbors(book_pivot.iloc[54, :].values.reshape(1,-1)
```

In [35]:
```
distances
```

Out[35]:
```
array([[ 0.        , 42.34383072, 43.48562981, 43.50861984, 43.56604182,
        43.89760814]])
```

In [36]:
```
suggestions
```

Out[36]:
```
array([[ 54, 184, 291, 440, 393, 372]], dtype=int64)
```

In [37]:
```
for i in range (len(suggestions)):
    print(book_pivot.index[suggestions[i]])
```

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

In [38]:
```
np.where(book_pivot.index=='Animal Farm')[0][0]
```

Out[38]:
```
54
```

In [41]:
```
def recommend_book(book_name):
    book_id = np.where(book_pivot.index==book_name)[0][0]
    distances , suggestions = model.kneighbors(book_pivot.iloc[54, :].values.reshape(1
    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]])
```

recommend_book('Animal Farm')

In [ ]:
```

```