In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
df_ratings = pd.read_csv('BX-Book-Ratings.csv', encoding = 'latin-1')
df_books = pd.read_csv('BX-Books.csv', encoding = 'latin1')
df_user = pd.read_csv('BX-Users.csv' , encoding = 'latin-1')
```

```
C:\Users\Netrika\anaconda.3\lib\site-packages\IPython\core\interactiveshell.py:3444: Dtype
Warning: Columns (3) have mixed types.Specify dtype option on import or set low_memory=Fal
se.
  exec(code_obj, self.user_global_ns, self.user_ns)
C:\Users\Netrika\anaconda.3\lib\site-packages\IPython\core\interactiveshell.py:3444: Dtype
Warning: Columns (0) have mixed types.Specify dtype option on import or set low_memory=Fal
se.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

In [3]:

```
df_books
```

Out[3]:

	isbn	book_title	book_author	year_of_publication	publisher
0	195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	2005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	60973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company
...	...	...	...	...	...
271374	440400988	There's a Bat in Bunk Five	Paula Danziger	1988	Random House Childrens Pub (Mm)
271375	525447644	From One to One Hundred	Teri Sloat	1991	Dutton Books
271376	006008667X	Lily Dale : The True Story of the Town that Ta...	Christine Wicker	2004	HarperSanFrancisco
271377	192126040	Republic (World's Classics)	Plato	1996	Oxford University Press
271378	767409752	A Guided Tour of Rene Descartes' Meditations o...	Christopher Biffle	2000	McGraw-Hill Humanities/Social Sciences/Languages

271379 rows × 5 columns

In [4]:

```
df_ratings
```

Out[4]:

	user_id	isbn	rating
0	276725	034545104X	0
1	276726	155061224	5
2	276727	446520802	0
3	276729	052165615X	3
4	[illegible]	521795028	6

Loading [MathJax]/extensions/Safe.js

	user_id	isbn	rating
...	...	...	...
1048570	250764	451410777	0
1048571	250764	452264464	8
1048572	250764	048623715X	0
1048573	250764	486256588	0
1048574	250764	515069434	0

1048575 rows × 3 columns

In [5]:

```
df_user
```

Out[5]:

	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
...	...	...	...
278854	278854	portland, oregon, usa	NaN
278855	278855	tacoma, washington, united kingdom	50.0
278856	278856	brampton, ontario, canada	NaN
278857	278857	knoxville, tennessee, usa	NaN
278858	278858	dublin, n/a, ireland	NaN

278859 rows × 3 columns

In [6]:

```
df_recommend = pd.read_csv ('Recommend.csv', encoding= 'latin-1',names = ['id','num','rati
```

In [7]:

```
df_recommend
```

Out[7]:

	id	num	rating	no
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596
...	...	...	...	...
99995	880	476	3	880175444
99996	716	204	5	879795543
99997	276	1090	1	874795795
			2	882399156

	id	num	rating	no
99999	12	203	3	879959583

100000 rows × 4 columns

In [8]:

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.neighbors import NearestNeighbors
from sklearn.feature_extraction import DictVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [9]:

```
print('books shape',df_books.shape)
print('ratings',df_ratings.shape)
print('users',df_user.shape)
```

```
books shape (271379, 5)
ratings (1048575, 3)
users (278859, 3)
```

In [10]:

```
df_books.columns
```

Out[10]:

```
Index(['isbn', 'book_title', 'book_author', 'year_of_publication',
       'publisher'],
      dtype='object')
```

In [11]:

```
df_books.head()
```

Out[11]:

	isbn	book_title	book_author	year_of_publication	publisher
0	195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	2005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	60973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company

In [12]:

```
missing_values_percentage=(df_books.isnull().sum()/df_books.shape[0])*100
missing_values_percentage
```

Out[12]:

```
isbn                   0.000000
book_title             0.000000
book_author            0.000368
year_of_publication    0.000000
publisher              0.000737
dtype: float64
```

In [13]:

```
df_books.loc[df_books['book_author'].isnull(),:]
```

Out[13]:

	isbn	book_title	book_author	year_of_publication	publisher

	isbn	book_title	book_author	year_of_publication	publisher
187700	9627982032	The Credit Suisse Guide to Managing Your Perso...	NaN	1995	Edinburgh Financial Publishing

In [14]:

```
df_books.loc[df_books['publisher'].isnull(),:]
```

Out[14]:

	isbn	book_title	book_author	year_of_publication	publisher
128896	193169656X	Tyrant Moon	Elaine Corvidae	2002	NaN
129043	1931696993	Finders Keepers	Linnea Sinclair	2001	NaN

In [15]:

```
df_books[(df_books.publisher>'2000')|(df_books.publisher=='Christine Wicker')]
```

Out[15]:

	isbn	book_title	book_author	year_of_publication	publisher
0	195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	2005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	60973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company
...	...	...	...	...	...
271374	440400988	There's a Bat in Bunk Five	Paula Danziger	1988	Random House Childrens Pub (Mm)
271375	525447644	From One to One Hundred	Teri Sloat	1991	Dutton Books
271376	006008667X	Lily Dale : The True Story of the Town that Ta...	Christine Wicker	2004	HarperSanFrancisco
271377	192126040	Republic (World's Classics)	Plato	1996	Oxford University Press
271378	767409752	A Guided Tour of Rene Descartes' Meditations o...	Christopher Biffle	2000	McGraw-Hill Humanities/Social Sciences/Languages

271204 rows × 5 columns

In [16]:

```
names=['Mark P. O. Morford','Richard Bruce Wright']
df_books[df_books.book_author.isin(names)]
```

Out[16]:

	isbn	book_title	book_author	year_of_publication	publisher
0	195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	2005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
69229	771597185	The teacher's daughter	Richard Bruce Wright	1982	Macmillan of Canada
95235	801319536	Classical Mythology	Mark P. O. Morford	1998	John Wiley &amp; Sons

In [17]:

```
df_books.loc[df_books['book_author'].isnull(),:]
```

Out[17]:

	isbn	book_title	book_author	year_of_publication	publisher
187700	9627982032	The Credit Suisse Guide to Managing Your Perso...	NaN	1995	Edinburgh Financial Publishing

In [18]:

```
df_books.loc[df_books['publisher'].isnull(),:]
```

Out[18]:

	isbn	book_title	book_author	year_of_publication	publisher
128896	193169656X	Tyrant Moon	Elaine Corvidae	2002	NaN
129043	1931696993	Finders Keepers	Linnea Sinclair	2001	NaN

In [19]:

```
df_books['publisher']=df_books['publisher'].replace(np.nan,'other')
df_books.loc[df_books['publisher'].isnull(),:]
```

Out[19]:

isbn	book_title	book_author	year_of_publication	publisher

In [20]:

```
df_books['book_author']=df_books['book_author'].replace(np.nan,'other')
df_books.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271379 entries, 0 to 271378
Data columns (total 5 columns):
 #   Column               Non-Null Count   Dtype 
---  ------               --------------   ----- 
 0   isbn                 271379 non-null  object
 1   book_title           271379 non-null  object
 2   book_author          271379 non-null  object
 3   year_of_publication  271379 non-null  object
 4   publisher            271379 non-null  object
dtypes: object(5)
memory usage: 10.4+ MB
```

In [21]:

```
df_books['year_of_publication'].unique()
```

Out[21]:

```
array(['2002', '2001', '1991', '1999', '2000', '1993', '1996', '1988',
       '2004', '1998', '1994', '2003', '1997', '1983', '1979', '1995',
       '1982', '1985', '1992', '1986', '1978', '1980', '1952', '1987',
       '1990', '1981', '1989', '1984', '0', '1968', '1961', '1958',
       '1974', '1976', '1971', '1977', '1975', '1965', '1941', '1970',
       '1962', '1973', '1972', '1960', '1966', '1920', '1956', '1959',
       '1953', '1951', '1942', '1963', '1964', '1969', '1954', '1950',
       '1967', '2005', '1957', '1940', '1937', 'John Peterman', '1955',
       '1946', '1936', '1930', '2011', '1925', '1948', '1943', '1947',
       '1945', '1923', '2020', '1939', '1926', '1938', '2030',
       '\\"Freedom Song\\""', '1911', '1904', '1949', 'Frank Muir',
       '1932', '1928', '1929', '1927', '1931', '1914', '2050', '1934',
       '1910', 'ROBERT A. WILSON', '1933', '1902', 'Karen T. Whittenburg',
       '1924', '1921', '1900', '2038', '2026', 'George H. Scherr', '1944',
       '1917', '1901', 'Salvador de Madariaga', '2010',
       'K.C. Constantine', 'Stan Berenstain', '1908', '1906', '1935',
       '1806', 'Francine Pascal', '2021', 'Luella Hill', '2012', '2006',
       'John Alderson Foote', 'DK Publishing Inc', 'Jules Janin',
       'Gallimard', '1909', '2008', '1378', ' &amp', ' Learning"', '1919',
       '1922', '1897', 'Isadora Duncan', '2024', 'Beatrix Potter', '1376',
       '2037', 'Bart Rulon', 'Alan Rich', 2000, 1982, 1983, 1989, 1993,
       1991, 1990, 1998, 1994, 1995, 1986, 1987, 1974, 1984, 0, 1977,
       1996, 1997, 1980, 1988, 2002, 2001, 1981, 1999, 1992, 2003, 2004,
       1972, 1976, 1985, 1978, 1979, 1970, 1962, 1975, 1901, 1973, 1955,
       1971, 1964, 1963, 1958, 1968, 1969, 1966, 1946, 1943, 1967, 1949,
       1965, 1961, 1960, 1930, 1951, 1957, 1959, 1952, 1953, 1956, 1950,
       1954, 1920, 2005, 1940, 1929], dtype=object)
```

In [22]:

```
books=df_books.copy()
books['year_of_publication']=pd.to_datetime(books['year_of_publication'],errors='ignore')
books.loc[books['year_of_publication']=='John Peterman',:]
```

Out[22]:

	isbn	book_title	book_author	year_of_publication	publisher
6450	735201994	Peterman Rides Again: Adventures Continue with...	the Catalog Business"	John Peterman	2000

In [23]:

```
books.loc[books.isbn=='735201994','year_of_publication']=2000
books.loc[books.isbn=='735201994','book_author']='John Peterman'
books.loc[books.isbn=='735201994','publisher']='the Catalog Business'
```

In [24]:

```
books.loc[books['year_of_publication']=='\\"Freedom Song\\""',:]
```

Out[24]:

	isbn	book_title	book_author	year_of_publication	publisher
43665	330482750	Three Novels: \A Strange and Sublime Address\"	\"Afternoon Raag\"	\"Freedom Song\""	Amit Chaudhuri

In [25]:

```
books.loc[books.isbn=='330482750','year_of_publication']=2000
books.loc[books.isbn=='330482750','book_author']='Amit Chaudhuri'
books.loc[books.isbn=='330482750','publisher']='other'
```

In [26]:

```
books.loc[books['year_of_publication']=='Frank Muir',:]
```

Out[26]:

	isbn	book_title	book_author	year_of_publication	publisher
51749	413326608	Upon my word!: More stories from \My word!\" a...	Tony Shryane"	Frank Muir	1974

In [27]:

```
books.loc[books.isbn=='413326608','year_of_publication']=1974
books.loc[books.isbn=='413326608','book_author']='Tony Shryane'
books.loc[books.isbn=='413326608','publisher']='Frank Muir'
```

In [28]:

```
books.loc[books['year_of_publication']=='ROBERT A. WILSON',:]
```

Out[28]:

	isbn	book_title	book_author	year_of_publication	publisher
92036	440500702	Schrodinger's Cat Trilogy : \The Universe Next...	\"The Homing Pigeons\""	ROBERT A. WILSON	1988

In [29]:

```
books.loc[books.isbn=='440500702','year_of_publication']=1988
books.loc[books.isbn=='413326608','book_author']='ROBERT A. WILSON'
books.loc[books.isbn=='413326608','publisher']='The Homing Pigeons'
```

In [30]:

```
books.loc[books['year_of_publication']=='Karen T. Whittenburg',:]
```

Out[30]:

	isbn	book_title	book_author	year_of_publication	publisher
104317	373166982	Please Say \I Do\" (Three Weddings &amp	A Hurricane) (Harlequin American Romance, No ...	Karen T. Whittenburg	1997

In [31]:

```
books.loc[books.isbn=='373166982','year_of_publication']=1997
books.loc[books.isbn=='373166982','book_author']='Karen T. Whittenburg'
books.loc[books.isbn=='373166982','publisher']='Harlequin'
books.loc[books.isbn=='373166982','book_title']='Please Say "I Do" (Three Weddings & A Hur
```

In [32]:

```
books.loc[books['year_of_publication']=='George H. Scherr',:]
```

Out[32]:

	isbn	book_title	book_author	year_of_publication	publisher
121766	894805959	The Best of the Journal of Irreproducible Resu...	Unfounded Findings\""	George H. Scherr	1989

In [33]:

```
books.loc[books.isbn=='894805959','year_of_publication']=1989
books.loc[books.isbn=='894805959','book_author']='George H. Scherr'
books.loc[books.isbn=='894805959','publisher']='Unfounded Findings'
```

In [34]:

```
books.loc[books['year_of_publication']=='Salvador de Madariaga',:]
```

Out[34]:

	isbn	book_title	book_author	year_of_publication	publisher
144056	8423920143	GuÃa del lector del \Quijote\": Ensayo psicol...	14 : Ensayo)"	Salvador de Madariaga	1976

In [35]:

```
books.loc[books.isbn=='8423920143','year_of_publication']=1976
books.loc[books.isbn=='8423920143','book_author']='Salvador de Madariaga'
books.loc[books.isbn=='8423920143','publisher']='Ensayo'
```

In [36]:

```
books.loc[books['year_of_publication']=='K.C. Constantine',:]
```

Out[36]:

	isbn	book_title	book_author	year_of_publication	publisher
150787	034050823X	The Double Detective: \The Blank Page\" &amp	\"A Fix Like This\" (Crime Club)"	K.C. Constantine	1990

In [37]:

```
books.loc[books.isbn=='034050823X','year_of_publication']=1990
books.loc[books.isbn=='034050823X','book_author']='K.C. Constantine'
books.loc[books.isbn=='034050823X','publisher']='other'
```

In [38]:

```
books.loc[books['year_of_publication']=='Stan Berenstain',:]
```

Out[38]:

	isbn	book_title	book_author	year_of_publication	publisher
157126	039482492X	C is for Clown: A Circus of \C\" Words, (Brigh...	early books for beginning beginners)"	Stan Berenstain	1972

In [39]:

```
books.loc[books.isbn=='039482492X','year_of_publication']=1972
books.loc[books.isbn=='039482492X','book_author']='Stan Berenstain'
ks.isbn=='039482492X','publisher']='other'
```

Loading [MathJax]/extensions/Safe.js

In [40]:

```
books.loc[books['year_of_publication']=='Francine Pascal',:]
```

Out[40]:

	isbn	book_title	book_author	year_of_publication	publisher
180187	553570722	\R\" for Revenge: Written by Kate William	Created by Francine Pascal (Sweet Valley High...	Francine Pascal	1997

In [41]:

```
books.loc[books.isbn=='553570722','year_of_publication']=1997
books.loc[books.isbn=='553570722','book_author']='Francine Pascal'
books.loc[books.isbn=='553570722','publisher']='other'
```

In [42]:

```
books.loc[books['year_of_publication']=='Luella Hill',:]
```

Out[42]:

	isbn	book_title	book_author	year_of_publication	publisher
185736	096401811X	Solid as a rock \I\" stand: Inspirational poet...	short stories"	Luella Hill	1998

In [43]:

```
books.loc[books.isbn=='096401811X','year_of_publication']=1998
books.loc[books.isbn=='096401811X','book_author']='     Luella Hill'
books.loc[books.isbn=='096401811X','publisher']='other'
```

In [44]:

```
books.loc[books['year_of_publication']=='John Alderson Foote',:]
```

Out[44]:

	isbn	book_title	book_author	year_of_publication	publisher
209386	085409878X	\Pie-powder\"	being dust from the law courts,"	John Alderson Foote	1973

In [45]:

```
books.loc[books.isbn=='085409878X','year_of_publication']=1973
books.loc[books.isbn=='085409878X','book_author']='John Alderson Foote'
books.loc[books.isbn=='085409878X','publisher']='other'
```

In [46]:

```
books.loc[books['year_of_publication']=='DK Publishing Inc',:]
```

Out[46]:

	isbn	book_title	book_author	year_of_publication	publisher
209550	078946697X	DK Readers: Creating the X-Men, How It All Beg...	2000	DK Publishing Inc	http://images.amazon.com/images/P/078946697X.0...
221691	789466953	DK Readers: Creating the X-Men, How Comic Book...	2000	DK Publishing Inc	http://images.amazon.com/images/P/0789466953.0...

In [47]:

```
books.loc[books.isbn=='078946697X','year_of_publication']=2000
books.loc[books.isbn=='078946697X','book_author']='James Buckley'
```

Loading [MathJax]/extensions/Safe.js

```
books.loc[books.isbn=='078946697X','publisher']='DK Publishing Inc'
books.loc[books.isbn=='078946697X','book_title']='DK Readers: Creating the X-Men, How Comi
```

In [48]:

```
books.loc[books.isbn=='789466953','year_of_publication']=2000
books.loc[books.isbn=='789466953','book_author']='Michael Teitelbaum'
books.loc[books.isbn=='789466953','publisher']='DK Publishing Inc'
books.loc[books.isbn=='789466953','book_title']='DK Readers: Creating the X-Men, How It Al
```

In [49]:

```
books.loc[books['year_of_publication']=='Jules Janin',:]
```

Out[49]:

	isbn	book_title	book_author	year_of_publication	publisher
220624	1874100055	\The Dead Donkey\" &amp	\"The Guillotined Woman\""	Jules Janin	0

In [50]:

```
books.loc[books.isbn=='1874100055','year_of_publication']=2000
books.loc[books.isbn=='1874100055','book_author']='Jules Janin'
books.loc[books.isbn=='1874100055','publisher']='other'
books.loc[books.isbn=='1874100055','book_title']='The Guillotined Woman'
```

In [51]:

```
books.loc[books['year_of_publication']=='Gallimard',:]
```

Out[51]:

	isbn	book_title	book_author	year_of_publication	publis
220744	2070426769	Peuple du ciel, suivi de 'Les Bergers\";Jean-M...	2003	Gallimard	http://images.amazon.com/images/P/2070426769

In [52]:

```
books.loc[books.isbn=='2070426769','year_of_publication']=2003
books.loc[books.isbn=='2070426769','book_author']='Gallimard'
books.loc[books.isbn=='2070426769','publisher']='other'
books.loc[books.isbn=='2070426769','book_title']='other'
```

In [53]:

```
books.loc[books['year_of_publication']==' &amp',:]
```

Out[53]:

	isbn	book_title	book_author	year_of_publication	publisher
227931	6479839	The LaVyrle Spencer Collection: Three Classic...	\"Forsaking All Others\"	&amp	\"A Promise to Cherish\")"

In [54]:

```
books.loc[books.isbn=='6479839','year_of_publication']=2000
books.loc[books.isbn=='6479839','book_author']='Other'
books.loc[books.isbn=='6479839','publisher']='other'
```

In [55]:

```
books.loc[books['year_of_publication']==' Learning"',:]
```

Out[55]:

	isbn	book_title	book_author	year_of_publication	publisher
228955	807735132	\The Having of Wonderful Ideas\" &amp	Other Essays on Teaching &amp	Learning"	Eleanor Duckworth

In [56]:

Loading [MathJax]/extensions/Safe.js

```
ks.isbn=='807735132','year_of_publication']=2000
```

```
books.loc[books.isbn=='807735132','book_author']='Eleanor Duckworth'
books.loc[books.isbn=='807735132','publisher']='other'
```

In [57]:

```
books.loc[books['year_of_publication']=='Isadora Duncan',:]
```

Out[57]:

	isbn	book_title	book_author	year_of_publication	publisher
245931	394720784	\Your Isadora\": The love story of Isadora Dun...	Gordon Craig"	Isadora Duncan	1976

In [58]:

```
books.loc[books.isbn=='394720784','year_of_publication']=1976
books.loc[books.isbn=='394720784','book_author']='Isadora Duncan'
books.loc[books.isbn=='394720784','publisher']='Gordon Craig'
```

In [59]:

```
books.loc[books['year_of_publication']=='Beatrix Potter',:]
```

Out[59]:

	isbn	book_title	book_author	year_of_publication	publisher
251294	723245827	\The World of Peter Rabbit &amp	Friends\" Complete Story Collection"	Beatrix Potter	1999

In [60]:

```
books.loc[books.isbn=='723245827','year_of_publication']=1999
books.loc[books.isbn=='723245827','book_author']='Beatrix Potter'
books.loc[books.isbn=='723245827','publisher']='other'
```

In [61]:

```
books.loc[books['year_of_publication']=='Bart Rulon',:]
```

Out[61]:

	isbn	book_title	book_author	year_of_publication	publisher
259939	1581801653	Artist\s Photo Reference: Water &amp	Skies (Artists Photo Reference)"	Bart Rulon	2002

In [62]:

```
books.loc[books.isbn=='1581801653','year_of_publication']=2002
books.loc[books.isbn=='1581801653','book_author']='Bart Rulon'
books.loc[books.isbn=='1581801653','publisher']='other'
```

In [63]:

```
books.loc[books['year_of_publication']=='Alan Rich',:]
```

Out[63]:

	isbn	book_title	book_author	year_of_publication	publisher
261527	006263545X	Ludwig van Beethoven: Play by Play/Symphony No...	The \"Egmont\" Overture"	Alan Rich	1995

In [64]:

```
books.loc[books.isbn=='006263545X','year_of_publication']=1995
books.loc[books.isbn=='006263545X','book_author']='Alan Rich'
books.loc[books.isbn=='006263545X','publisher']='other'
```

In [65]:

```
books['year_of_publication'].unique()
```

Out[65]:

```
array(['2002', '2001', '1991', '1999', '2000', '1993', '1996', '1988',
       '2004', '1998', '1994', '2003', '1997', '1983', '1979', '1995',
       '1982', '1985', '1992', '1986', '1978', '1980', '1952', '1987',
```

Loading [MathJax]/extensions/Safe.js

```
       '1990', '1981', '1989', '1984', '0', '1968', '1961', '1958',
       '1974', '1976', '1971', '1977', '1975', '1965', '1941', '1970',
       '1962', '1973', '1972', '1960', '1966', '1920', '1956', '1959',
       '1953', '1951', '1942', '1963', '1964', '1969', '1954', '1950',
       '1967', '2005', '1957', '1940', '1937', 2000, '1955', '1946',
       '1936', '1930', '2011', '1925', '1948', '1943', '1947', '1945',
       '1923', '2020', '1939', '1926', '1938', '2030', '1911', '1904',
       '1949', 1974, '1932', '1928', '1929', '1927', '1931', '1914',
       '2050', '1934', '1910', 1988, '1933', '1902', 1997, '1924', '1921',
       '1900', '2038', '2026', 1989, '1944', '1917', '1901', 1976, '2010',
       1990, 1972, '1908', '1906', '1935', '1806', '2021', 1998, '2012',
       '2006', 1973, 2003, '1909', '2008', '1378', '1919', '1922', '1897',
       '2024', 1999, '1376', '2037', 2002, 1995, 1982, 1983, 1993, 1991,
       1994, 1986, 1987, 1984, 0, 1977, 1996, 1980, 2001, 1981, 1992,
       2004, 1985, 1978, 1979, 1970, 1962, 1975, 1901, 1955, 1971, 1964,
       1963, 1958, 1968, 1969, 1966, 1946, 1943, 1967, 1949, 1965, 1961,
       1960, 1930, 1951, 1957, 1959, 1952, 1953, 1956, 1950, 1954, 1920,
       2005, 1940, 1929], dtype=object)
```

In [66]:

```
books.loc[books['year_of_publication']==0,:]
```

Out[66]:

	isbn	book_title	book_author	year_of_publication	publisher
262165	091680013X	Masterpieces of Erotic Photography	David Bailey	0	Harbor House Publishers
262239	190261500X	Ultimate Decorating Book	Judy Spours	0	Kilns House Books
262306	297826980	Get a Grip On Philosophy	Neil Turnbull	0	Trafalgar Square
262390	965766950	How to Do Almost Anything the Learn Guide	Jason Roberts	0	Villard Books
262433	1582430012	A Drink Called Paradise	Terese Svoboda	0	Counterpoint Press
...	...	...	...	...	...
270813	014029953X	Foe (Essential.penguin S.)	J.M. Coetzee	0	Penguin Books Ltd
270932	340571187	Postmens House	Maggie Hemingway	0	Trafalgar Square
271113	8427201079	El Misterio De Sittaford	Agatha Christie	0	Editorial Molino
271201	887781721	Tom Penny	Tony German	0	P. Martin Associates
271215	3150013763	Der Hofmeister	Jakob Lenz	0	Philipp Reclam, Jun Verlag GmbH

151 rows × 5 columns

In [67]:

```
books['year_of_publication']=pd.to_numeric(books['year_of_publication'],errors='coerce')
print(sorted(list(books['year_of_publication'].unique())))
```

```
[0, 1376, 1378, 1806, 1897, 1900, 1901, 1902, 1904, 1906, 1908, 1909, 1910, 1911, 1914, 19
17, 1919, 1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 19
33, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 19
48, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 19
63, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 19
78, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 19
93, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2008, 20
10, 2011, 2012, 2020, 2021, 2024, 2026, 2030, 2037, 2038, 2050]
```

In [68]:

```
#Converting values above 2021 year and equal to 0
books.loc[(books['year_of_publication']>2021)|(books['year_of_publication']==0),'year_of_p
```

Loading [MathJax]/extensions/Safe.js

In [69]:

```
#Replacing Nans with Median values of year of publications
books['year_of_publication'].fillna(round(books['year_of_publication'].median()),inplace=T
```

In [70]:

```
books.isnull().sum()
```

Out[70]:

```
isbn                   0
book_title             0
book_author            0
year_of_publication    0
publisher              0
dtype: int64
```

In [71]:

```
books.loc[books['year_of_publication']==1376,:]
```

Out[71]:

	isbn	book_title	book_author	year_of_publication	publisher
253767	964442011X	Tasht-i khun	IsmaÂ°il Fasih	1376.0	Nashr-i Alburz

In [72]:

```
books.loc[books.isbn=='964442011X','year_of_publication']=2000
```

In [73]:

```
books.loc[books['year_of_publication']==1378,:]
```

Out[73]:

	isbn	book_title	book_author	year_of_publication	publisher
227544	9643112136	Dalan-i bihisht (Dastan-i Irani)	Nazi Safavi	1378.0	Intisharat-i Quqnus

In [74]:

```
books.loc[books.isbn=='9643112136','year_of_publication']=2000
df_ratings
```

Out[74]:

	user_id	isbn	rating
0	276725	034545104X	0
1	276726	155061224	5
2	276727	446520802	0
3	276729	052165615X	3
4	276729	521795028	6
...	...	...	...
1048570	250764	451410777	0
1048571	250764	452264464	8
1048572	250764	048623715X	0
1048573	250764	486256588	0
1048574	250764	515069434	0

1048575 rows × 3 columns

In [75]:

```
#Ratings dataset should have books only which exist in our books dataset
ratings=df_ratings[df_ratings.isin(books.isbn)]
ratings
```

Out[75]:

	user_id	isbn	rating
0	276725	034545104X	0
1	276726	155061224	5
2	276727	446520802	0
3	276729	052165615X	3
4	276729	521795028	6
...	...	...	...
1048570	250764	451410777	0
1048571	250764	452264464	8
1048572	250764	048623715X	0
1048573	250764	486256588	0
1048574	250764	515069434	0

941148 rows × 3 columns

In [76]:

```
df_ratings.shape,ratings.shape
#It can be seen that many rows having book ISBN not part of books dataset got dropped off
```

Out[76]:

```
((1048575, 3), (941148, 3))
```

In [77]:

```
df_user
```

Out[77]:

	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
...	...	...	...
278854	278854	portland, oregon, usa	NaN
278855	278855	tacoma, washington, united kingdom	50.0
278856	278856	brampton, ontario, canada	NaN
278857	278857	knoxville, tennessee, usa	NaN
278858	278858	dublin, n/a, ireland	NaN

278859 rows × 3 columns

In [78]:

```
users=df_user.copy()
```

In [79]:

```
users
```

Out[79]:

	user_id	Location	Age

Loading [MathJax]/extensions/Safe.js

	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
...	...	...	...
278854	278854	portland, oregon, usa	NaN
278855	278855	tacoma, washington, united kingdom	50.0
278856	278856	brampton, ontario, canada	NaN
278857	278857	knoxville, tennessee, usa	NaN
278858	278858	dublin, n/a, ireland	NaN

278859 rows × 3 columns

In [80]:

```
def missing_values(df):
    mis_val=df.isnull().sum()
    miss_val_percent=round(df.isnull().mean()*100,2)
    mz_table=pd.concat([mis_val,miss_val_percent],axis=1)
    mz_table=mz_table.rename(columns={df.index.name:'col_name',0:'Missing values',1:'% of
    mz_table['Data_type']=df.dtypes
    mz_table=mz_table.sort_values('% of total values',ascending=False)
    return mz_table.reset_index()
```

In [81]:

```
missing_values(users)
```

Out[81]:

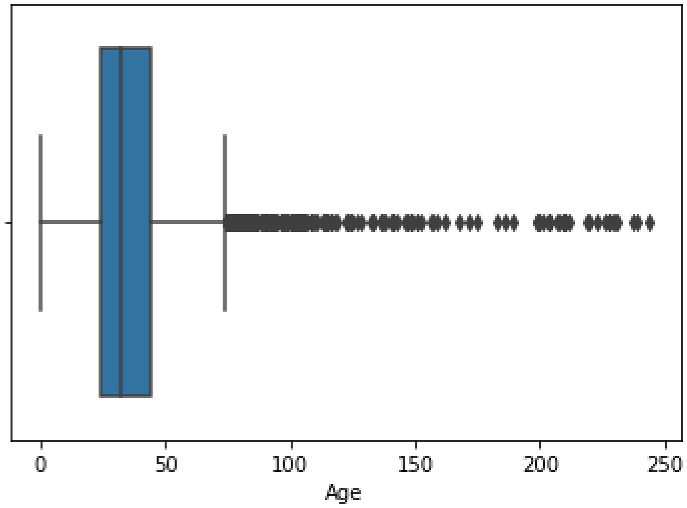
	index	Missing values	% of total values	Data_type
0	Age	110763	39.72	float64
1	user_id	0	0.00	object
2	Location	1	0.00	object

In [82]:

```
sns.boxplot(x='Age',data=users)
```

Out[82]:

```
<AxesSubplot:xlabel='Age'>
```



In [83]:

```
print(sorted(list(users.Age.unique())))
```

```
[nan, 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.
0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.
0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.
0, 46.0, 47.0, 48.0, 49.0, 50.0, 51.0, 52.0, 53.0, 54.0, 55.0, 56.0, 57.0, 58.0, 59.0, 60.
0, 61.0, 62.0, 63.0, 64.0, 65.0, 66.0, 67.0, 68.0, 69.0, 70.0, 71.0, 72.0, 73.0, 74.0, 75.
0, 76.0, 77.0, 78.0, 79.0, 80.0, 81.0, 82.0, 83.0, 84.0, 85.0, 86.0, 87.0, 88.0, 89.0, 90.
0, 91.0, 92.0, 93.0, 94.0, 95.0, 96.0, 97.0, 98.0, 99.0, 100.0, 101.0, 102.0, 103.0, 104.
0, 105.0, 106.0, 107.0, 108.0, 109.0, 110.0, 111.0, 113.0, 114.0, 115.0, 116.0, 118.0, 11
9.0, 123.0, 124.0, 127.0, 128.0, 132.0, 133.0, 136.0, 137.0, 138.0, 140.0, 141.0, 143.0, 1
46.0, 147.0, 148.0, 151.0, 152.0, 156.0, 157.0, 159.0, 162.0, 168.0, 172.0, 175.0, 183.0,
186.0, 189.0, 199.0, 200.0, 201.0, 204.0, 207.0, 208.0, 209.0, 210.0, 212.0, 219.0, 220.0,
223.0, 226.0, 228.0, 229.0, 230.0, 231.0, 237.0, 239.0, 244.0]
```

In [84]:

```
#We have outliers in data
```

In [85]:

```
users.Location
```

Out[85]:

```
0                             nyc, new york, usa
1                      stockton, california, usa
2                moscow, yukon territory, russia
3                      porto, v.n.gaia, portugal
4             farnborough, hants, united kingdom
                               ...
278854                     portland, oregon, usa
278855    tacoma, washington, united kingdom
278856                 brampton, ontario, canada
278857                 knoxville, tennessee, usa
278858                      dublin, n/a, ireland
Name: Location, Length: 278859, dtype: object
```

In [86]:

```
for i in users:
    users['country']=users.Location.str.extract(r'\,+\s?(\w*\s?\w*)\"*$')
```

In [87]:

```
users.drop('Location',axis=1,inplace=True)
```

In [88]:

```
users['country']=users['country'].astype('str')
```

In [89]:

```
country.unique())
```

Loading [MathJax]/extensions/Safe.js

```
a=set(a)
a=list(a)
a=[x for x in a if x is not None]
a.sort()
print(a)
```

```
['', '01776', '02458', '19104', '23232', '30064', '85021', '87510', '_ brasil', 'a', 'aa
a', 'adsgfdr', 'afghanistan', 'ahrensburg', 'alachua', 'albania', 'alderney', 'algeria',
'algérie', 'allen', 'ama lurra', 'america', 'american samoa', 'andorra', 'angola', 'antarc
tica', 'aotearoa', 'argentina', 'armenia', 'aroostook', 'aruba', 'asturies', 'auckland',
'austbritania', 'australia', 'australii', 'austria', 'autralia', 'az', 'azerbaijan', 'bade
mn würtemberg', 'bahamas', 'bahrain', 'baltimore', 'bangladesh', 'barbados', 'basilicata',
'basque country', 'bavaria', 'baxter', 'bbbzzzzz', 'belarus', 'belgi', 'belgique', 'belgiu
m', 'belize', 'bell', 'benin', 'berguedà', 'bermuda', 'bernalillo', 'bfe', 'bhutan', 'bi
h', 'bkk', 'bladen', 'bolivia', 'bosnia', 'botswana', 'brasil', 'brazil', 'brunei', 'brune
i darussalam', 'bucks', 'bulgaria', 'buncombe', 'burkina faso', 'burkinafasu', 'burlingto
n', 'burma', 'butler', 'c', 'cadd0', 'calabria', 'california', 'cambodia', 'camden', 'came
roon', 'can', 'canada', 'canada eh', 'cananda', 'canda', 'cape may', 'cape town', 'cape ve
rde', 'caribbean sea', 'carter', 'cass', 'catalonia', 'catalunya', 'catalunya spain', 'cay
man islands', 'chad', 'channel islands', 'cherokee', 'chian', 'chile', 'china', 'china öð¹
ú', 'chinaöð¹ú', 'clackamas', 'clinton', 'cn', 'cnina', 'collin', 'colombia', 'comoros',
'congo', 'cook', 'copenhagen', 'costa rica', 'country', 'courtenay', 'croatia', 'csa', 'cu
ba', 'cyprus', 'czech republic', 'côte d', 'dauphin', 'davidson', 'de', 'dekalb', 'denmar
k', 'der', 'deutsches reich', 'deutschland', 'dfg', 'distrito federal', 'diy', 'djibouti',
'dominica', 'dominican republic', 'doodedoo', 'dublin', 'east africa', 'ecuador', 'effingh
am', 'egypt', 'el salvador', 'england', 'england uk', 'english', 'equatorial geuinea', 'er
itrea', 'españa', 'essex', 'estonia', 'ethiopia', 'europa', 'europe', 'euskadi', 'euskal h
erria', 'evil empire', 'fairyland', 'faroe islands', 'ferrara', 'fifi', 'fiji', 'finland',
'fontana', 'fort bend', 'fortaleza', 'framingham', 'france', 'fred', 'frome', 'fulton', 'g
abon', 'galiza', 'galiza neghra', 'gb', 'geermany', 'georgia', 'germany', 'germay', 'ghan
a', 'goteborg', 'granville', 'greece', 'grenada', 'guam', 'guatemala', 'guernsey', 'guine
a', 'guyana', 'haiti', 'hamilton', 'hampden', 'harvatija', 'heaven', 'hennipen', 'henry',
'hernando', 'hidalgo', 'hillsborough', 'holland', 'holy see', 'honduras', 'hong kong', 'ho
ngkong', 'hornyonic', 'houston', 'hungary', 'iceland', 'id', 'il canada', 'in', 'india',
'indiai', 'indonesia', 'input error', 'iran', 'iraq', 'ireland', 'israel', 'isreal', 'ital
ia', 'italien', 'italy', 'itlay', 'jackson', 'jamaica', 'japan', 'jersey', 'johnson', 'jor
dan', 'k1c7b1', 'kazakhstan', 'kenya', 'kern', 'kiribati', 'korea', 'kosovo', 'ksa', 'kuwa
it', 'kyrgyzstan', 'kz', 'l', 'la argentina', 'la belgique', 'la france', 'la suisse', 'la
svizzera', 'lake', 'lane', 'laos', 'latvia', 'lazio', 'le canada', 'le madagascar', 'leban
on', 'lecce', 'lee', 'lesotho', 'liaoning', 'liberia', 'libya', 'lithuania', 'liushi', 'li
vingston', 'lj', 'lkjlj', 'lleida', 'lombardia', 'london', 'lornsenland', 'luxembourg', 'm
acau', 'macedonia', 'madagascar', 'madrid', 'maine', 'malawi', 'malaysia', 'malaysian', 'm
aldives', 'mali', 'malta', 'maracopa', 'maricopa', 'maroc', 'marrion', 'marshall islands',
'mauritania', 'mauritius', 'mexico', 'micronesia', 'milano', 'minnehaha', 'missouri', 'mol
dova', 'monaco', 'mongolia', 'monroe', 'monterrey', 'mordor', 'morgan', 'morocco', 'mozamb
ique', 'moçambique', 'my', 'myanmar', 'mérida', 'méxico', 'namibia', 'nan', 'nederlands',
'nepal', 'netherlands', 'netherlands antilles', 'neverland', 'new london', 'new zealand',
'newzealand', 'nh', 'nicaragua', 'niger', 'nigeria', 'nl', 'no more', 'none', 'north carol
ina', 'north korea', 'northern ireland', 'norway', 'nowhere', 'nrw', 'nyc', 'nyhamnsläge',
'nz', 'oakland', 'ohio', 'oman', 'onondaga nation', 'orange co', 'orense', 'osceola', 'our
anos', 'pa', 'pakistan', 'palau', 'palestine', 'palm beach', 'panama', 'paraguay', 'pasc
o', 'pender', 'peru', 'perãº', 'petrolwar nation', 'phila', 'philippine', 'philippines',
'philippinies', 'phillipines', 'phils', 'phippines', 'pinallas', 'pippo', 'pistlavakia',
'poland', 'polk', 'polska', 'portugal', 'pr', 'prc', 'prince william', 'pueblo', 'puerto r
ico', 'qatar', 'queensland', 'queenspark', 'quit', 'rapides', 'rice', 'richmond country',
'rmi', 'roma', 'romania', 'rosario', 'rosello', 'rsa', 'russia', 'russian federation', 'ru
therford', 'rwanda', 'saint loius', 'saint lucia', 'saint luica', 'samoa', 'san bernardin
o', 'san franicsco', 'san marino', 'san mateo', 'santa barbara', 'sardegna', 'sardinia',
'saudi arabia', 'saudia arabia', 'scotland', 'senegal', 'serbia', 'shelby', 'sicilia', 'si
erra leone', 'singapore', 'slo', 'slovak republik', 'slovakia', 'slovenia', 'smith', 'sola
no', 'solomon islands', 'somalia', 'somewherein space', 'south africa', 'south korea', 'sp
ace', 'spain', 'spotsylvania', 'sri lanka', 'srilanka', 'strongbadia', 'sudan', 'suisse',
'suriname', 'swaziland', 'swazilandia', 'sweden', 'switzerland', 'syria', 'taiwan', 'tajik
istan', 'tanzania', 'tdzimi', 'texas', 'thailand', 'thailoand', 'the', 'the gambia', 'the
```

Loading [MathJax]/extensions/Safe.js

```
netherlands', 'the philippines', 'thing', 'tobago', 'togo', 'tonga', 'toscana', 'travellin
g', 'trinidad', 'tunisia', 'turkei', 'turkey', 'turkmenistan', 'tyrtyrt', 'türkiye', 'u',
'u k', 'ua', 'uae', 'uganda', 'uk', 'ukrain', 'ukraine', 'unit', 'unite states', 'united k
indgdom', 'united kindgonm', 'united kingdom', 'united sates', 'united staes', 'united sta
te', 'united statea', 'united stated', 'united states', 'universe', 'unknown', 'urugua',
'uruguay', 'us', 'usa', 'ussurfing', 'ust', 'uusa', 'uyo', 'uzbekistan', 'valtesse', 'van
wert', 'vanuatu', 'vatican city', 'venezuela', 'ventura', 'ventura county', 'vicenza', 'vi
ctoria', 'vietnam', 'vorce', 'wales', 'washtenaw', 'west yorkshire', 'whatcom', 'windham',
'wonderful usa', 'wood', 'worcester', 'x', 'xinyu', 'xxxxxx', 'yakima', 'yanhill', 'yeme
n', 'ysa', 'yugoslavia', 'yunling', 'z', 'zambia', 'zhengjiang', 'zimbabwe', 'öð¹ú']
```

In [90]:

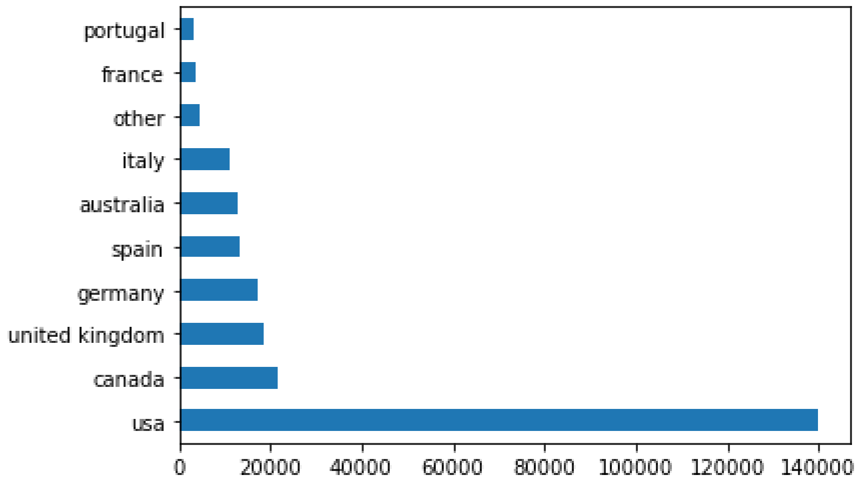
```
users['country'].replace(['','01776','02458','19104','23232','30064','85021','87510','alac
                                        ['other','usa','usa','usa','usa','usa','usa','usa','usa','usa',
```

In [91]:

```
users['country'].value_counts().sort_values(ascending=False)[:10].plot(kind='barh')
```

Out[91]:

```
<AxesSubplot:>
```



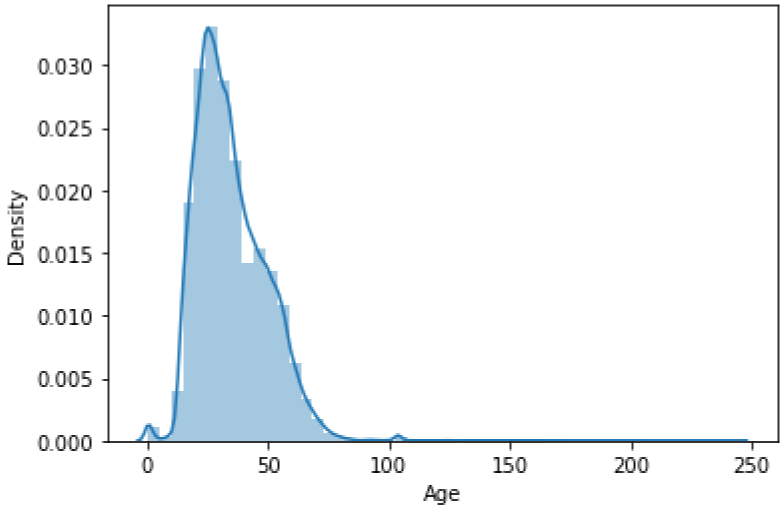
In [92]:

```
sns.distplot(users['Age'])
```

```
C:\Users\Netrika\anaconda.3\lib\site-packages\seaborn\distributions.py:2619: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Please ada
pt your code to use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[92]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```



Loading [MathJax]/extensions/Safe.js

```
ers.Age>100)|(users.Age<5),'Age']=np.nan
```

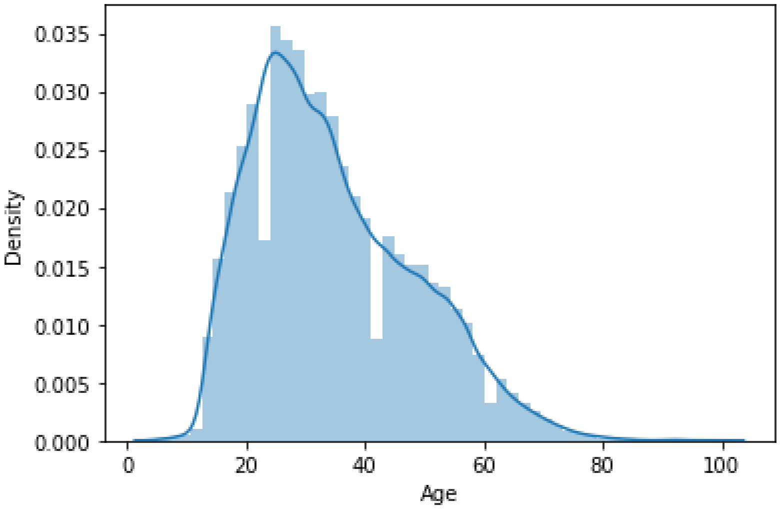
In [94]:

```
sns.distplot(users['Age'])
```

```
C:\Users\Netrika\anaconda.3\lib\site-packages\seaborn\distributions.py:2619: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Please ada
pt your code to use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[94]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```



In [95]:

```
users['Age']=users['Age'].fillna(users.groupby('country')['Age'].transform('median'))
```

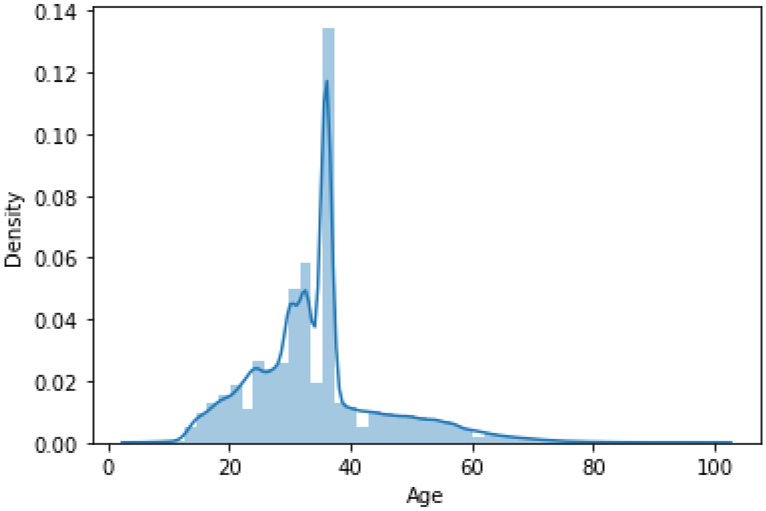
In [96]:

```
sns.distplot(users['Age'])
```

```
C:\Users\Netrika\anaconda.3\lib\site-packages\seaborn\distributions.py:2619: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Please ada
pt your code to use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[96]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```



In [97]:

```
users.isnull().sum()
```

Out[97]:

```
user_id       0
Age         276
```

Loading [MathJax]/extensions/Safe.js

```
country      0
dtype: int64
```

In [98]:

```
users['Age']=users['Age'].fillna(users['Age'].mean())
users.isnull().sum()
```

Out[98]:

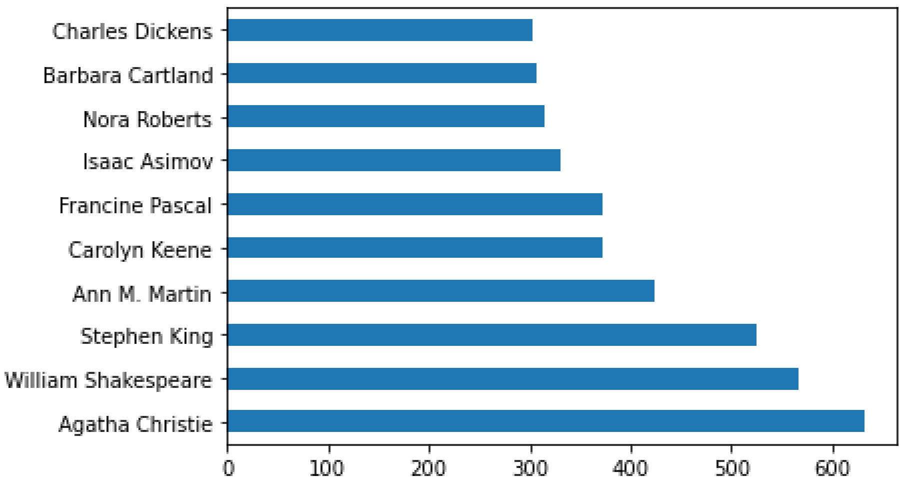
```
user_id    0
Age        0
country    0
dtype: int64
```

In [99]:

```
books.book_author.value_counts().sort_values(ascending=False)[:10].plot(kind='barh')
```

Out[99]:

```
<AxesSubplot:>
```



In [100…

```
books.columns
```

Out[100…

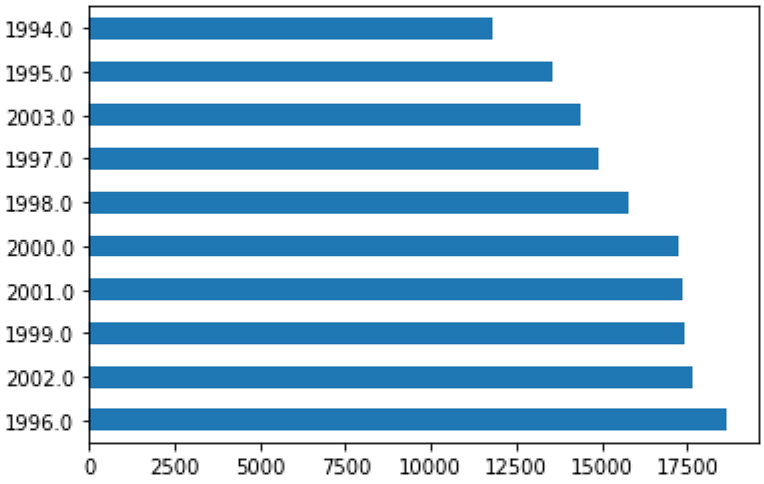
```
Index(['isbn', 'book_title', 'book_author', 'year_of_publication',
       'publisher'],
      dtype='object')
```

In [101…

```
books.year_of_publication.value_counts().sort_values(ascending=False).head(10).plot(kind='
```

Out[101…

```
<AxesSubplot:>
```

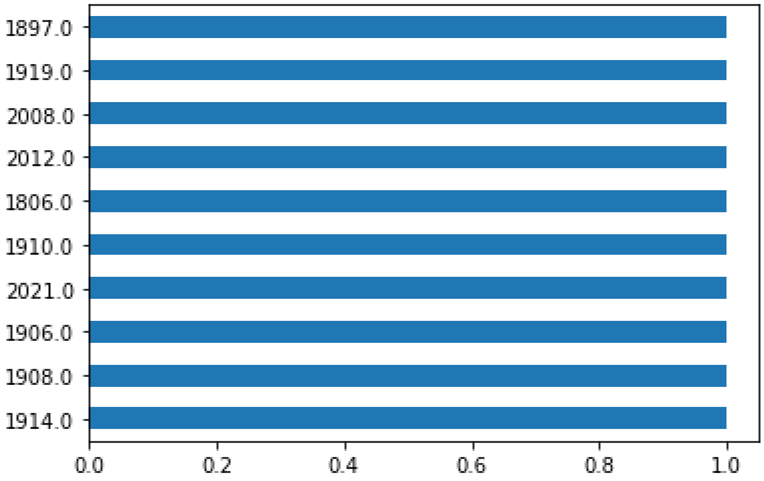


In [102…

```
books.year_of_publication.value_counts().sort_values(ascending=False).tail(10).plot(kind='
```

Out[102…

```
<AxesSubplot:>
```

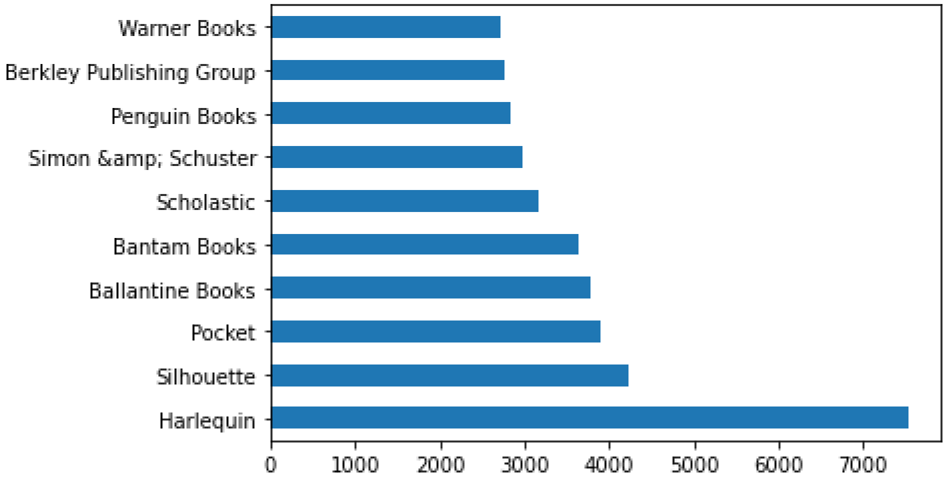


In [103…

```
books.publisher.value_counts().sort_values(ascending=False)[:10].plot(kind='barh')
```

Out[103…

```
<AxesSubplot:>
```



In [104…

```
ratings
```

Out[104…

	user_id	isbn	rating
0	276725	034545104X	0
1	276726	155061224	5
2	276727	446520802	0
3	276729	052165615X	3
4	276729	521795028	6
...	...	...	...
1048570	250764	451410777	0
1048571	250764	452264464	8
1048572	250764	048623715X	0
1048573	250764	486256588	0
1048574	250764	515069434	0

941148 rows × 3 columns

Loading [MathJax]/extensions/Safe.js

```
atings[ratings['user_id'].isin(users['user_id'])]
```

```
ratings_new
```

Out[105…

	user_id	isbn	rating
9561	2	195153448	0
9563	8	2005018	5
9564	8	60973129	0
9565	8	374157065	0
9566	8	393045218	0
...	...	...	...
1048570	250764	451410777	0
1048571	250764	452264464	8
1048572	250764	048623715X	0
1048573	250764	486256588	0
1048574	250764	515069434	0

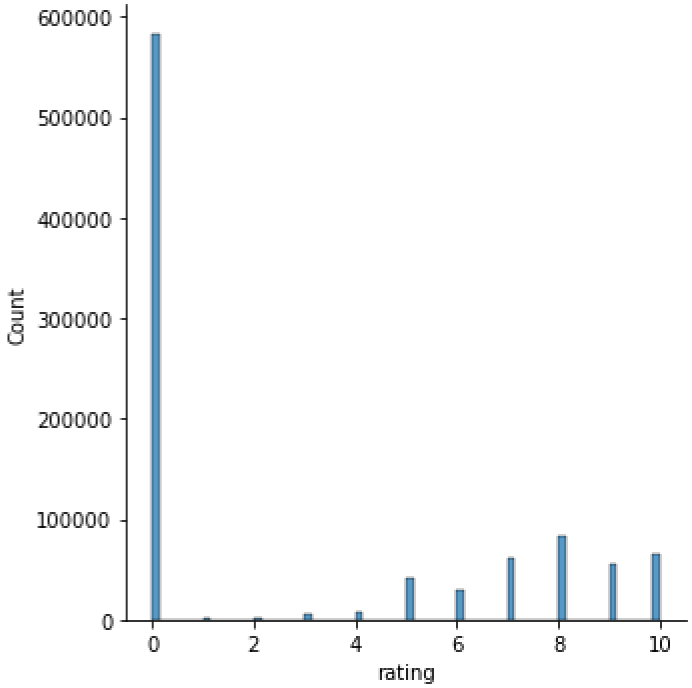
932713 rows × 3 columns

In [106…

```
sns.displot(ratings_new['rating'])
```

Out[106…

```
<seaborn.axisgrid.FacetGrid at 0x1bacd4edeb0>
```



In [107…

```
ratings_explicit=ratings_new[ratings_new['rating']!=0]
ratings_implicit=ratings_new[ratings_new['rating']==0]
```

In [108…

```
ratings_explicit.shape
```

Out[108…

```
(349765, 3)
```

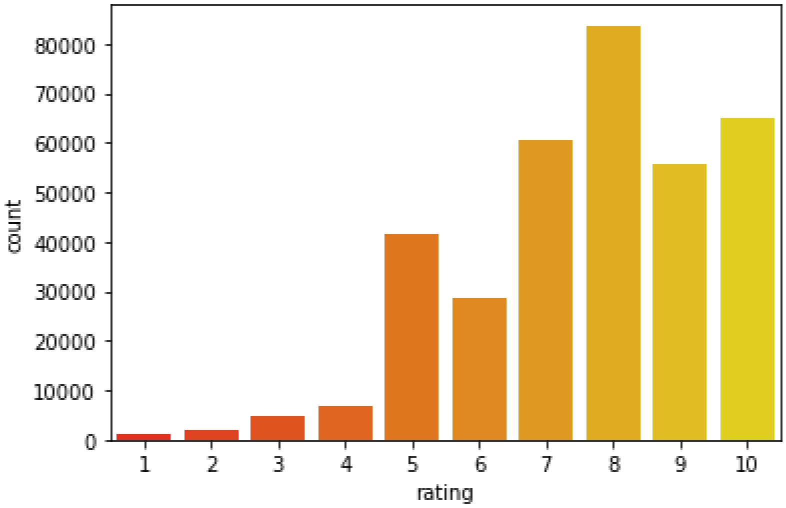
In [109…

```
ratings_implicit.shape
```

Out[109…
```
(582948, 3)
```

In [110…
```python
sns.countplot(data=ratings_explicit,x='rating',palette='autumn')
```

Out[110…
```
<AxesSubplot:xlabel='rating', ylabel='count'>
```



In [111…
```python
rating_count=pd.DataFrame(ratings_explicit.groupby('isbn')['rating'].count())
rating_count.sort_values(ascending=False,by='rating').head()
```

Out[111…

	rating
isbn	
316666343	632
971880107	524
385504209	450
312195516	351
60928336	292

In [112…
```python
books.loc[books.isbn=='316666343',:]
```

Out[112…

	isbn	book_title	book_author	year_of_publication	publisher
408	316666343	The Lovely Bones: A Novel	Alice Sebold	2002.0	Little, Brown

In [113…
```python
most_rated_books=pd.DataFrame(['316666343','971880107','385504209','312195516','60928336'
                             columns=['isbn'])
most_rated_books_summary=pd.merge(most_rated_books,books,on='isbn')
most_rated_books_summary
#top 5 rated books
```

Out[113…

	isbn	book_title	book_author	year_of_publication	publisher
0	316666343	The Lovely Bones: A Novel	Alice Sebold	2002.0	Little, Brown
1	971880107	Wild Animus	Rich Shapero	2004.0	Too Far
2	385504209	The Da Vinci Code	Dan Brown	2003.0	Doubleday
3	312195516	The Red Tent (Bestselling Backlist)	Anita Diamant	1998.0	Picador USA

	isbn	book_title	book_author	year_of_publication	publisher
4	60928336	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997.0	Perennial

In [114…

```
ratings_explicit['Ave_rating']=ratings_explicit.groupby('isbn')['rating'].transform('mean'
ratings_explicit['Total_users_rated']=ratings_explicit.groupby('isbn')['rating'].transform
```

```
C:\Users\Netrika\AppData\Local\Temp/ipykernel_16324/1386431096.py:1: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  ratings_explicit['Ave_rating']=ratings_explicit.groupby('isbn')['rating'].transform('mea
n')
C:\Users\Netrika\AppData\Local\Temp/ipykernel_16324/1386431096.py:2: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  ratings_explicit['Total_users_rated']=ratings_explicit.groupby('isbn')['rating'].transfo
rm('count')
```

In [115…

```
Final_dataset=users.copy()
Final_dataset=pd.merge(Final_dataset,ratings_explicit,on='user_id')
Final_dataset=pd.merge(Final_dataset,books,on='isbn')
Final_dataset.head()
```

Out[115…

	user_id	Age	country	isbn	rating	Ave_rating	Total_users_rated	book_title	book_author	year_of_publica
0	8	33.0	canada	2005018	5	7.875	8	Clara Callan	Richard Bruce Wright	20
1	11676	28.0	nan	2005018	8	7.875	8	Clara Callan	Richard Bruce Wright	20
2	67544	30.0	canada	2005018	8	7.875	8	Clara Callan	Richard Bruce Wright	20
3	116866	32.0	other	2005018	9	7.875	8	Clara Callan	Richard Bruce Wright	20
4	123629	33.0	canada	2005018	9	7.875	8	Clara Callan	Richard Bruce Wright	20

In [116…

```
missing_values(Final_dataset)
```

Out[116…

	index	Missing values	% of total values	Data_type
0	user_id	0	0.0	object
1	Age	0	0.0	float64
2	country	0	0.0	object
3	isbn	0	0.0	object
4	rating	0	0.0	int64
5	Ave_rating	0	0.0	float64
	ated	0	0.0	int64

	index	Missing values	% of total values	Data_type
7	book_title	0	0.0	object
8	book_author	0	0.0	object
9	year_of_publication	0	0.0	float64
10	publisher	0	0.0	object

In [117…

```
Final_dataset.shape
```

Out[117…

```
(349765, 11)
```

In [121…

```
bookName=input('Enter a book name: ')
Number=int(input('Enter number of books to recommend: '))
```

```
Enter a book name: Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
Enter number of books to recommend: 5
```

In [122…

```
def popularity_based(df,n):
    if n>=1 and n<=len(df):
        data=pd.DataFrame(df.groupby('isbn')['rating'].count()).sort_values('rating',ascer
        result=pd.merge(data,books,on='isbn')
        return result
    return 'Invalid number of books entered'
```

In [123…

```
print('top', Number, 'popular books are:')
popularity_based(Final_dataset,5)
```

```
top 5 popular books are:
```

Out[123…

	isbn	rating	book_title	book_author	year_of_publication	publisher
0	316666343	632	The Lovely Bones: A Novel	Alice Sebold	2002.0	Little, Brown
1	971880107	524	Wild Animus	Rich Shapero	2004.0	Too Far
2	385504209	450	The Da Vinci Code	Dan Brown	2003.0	Doubleday
3	312195516	351	The Red Tent (Bestselling Backlist)	Anita Diamant	1998.0	Picador USA
4	60928336	292	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997.0	Perennial

In [124…

```
def place_based(df,place):
    place=place.lower()
    if place in list(df['country'].unique()):
        return df.loc[df['country']==place]
    else:
        return 'Invalid Entry'
```

In [125…

```
place=input('Enter the name of place: ')
data=place_based(Final_dataset,place)
if isinstance(data,pd.DataFrame):
    data=popularity_based(data,Number)
data
```

Loading [MathJax]/extensions/Safe.js

```
of place: USA
```

Out[125…

	isbn	rating	book_title	book_author	year_of_publication	publisher
0	316666343	479	The Lovely Bones: A Novel	Alice Sebold	2002.0	Little, Brown
1	385504209	366	The Da Vinci Code	Dan Brown	2003.0	Doubleday
2	312195516	278	The Red Tent (Bestselling Backlist)	Anita Diamant	1998.0	Picador USA
3	971880107	271	Wild Animus	Rich Shapero	2004.0	Too Far
4	142001740	247	The Secret Life of Bees	Sue Monk Kidd	2003.0	Penguin Books

In [126…

```
def printbook(df,n):
    z=df['book_title'].unique()
    for x in range(len(z)):
        print(z[x])
        if x>=n-1:
            break
```

In [127…

```
def get_books(dataframe,name,n):
    print('\nbooks by same Author:\n')
    au=dataframe['book_author'].unique()
    data=Final_dataset[Final_dataset['book_title']!=name]
    if au[0] in list(data['book_author'].unique()):
        k2=data[data['book_author']==au[0]]
    k2=k2.sort_values(by=['rating'])
    printbook(k2,n)
    print('\n\nbooks by same publisher\:n')
    au=dataframe['publisher'].unique()
    if au[0] in list(data['publisher'].unique()):
        k2=pd.DataFrame(data[data['publisher']==au[0]])
    k2=k2.sort_values(by='rating')
    printbook(k2,n)
```

In [128…

```
if bookName in list(Final_dataset['book_title'].unique()):
    d=Final_dataset[Final_dataset['book_title']==bookName]
    get_books(d,bookName,Number)
else:
    print('Invalid book Name!!')
```

```
books by same Author:

Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Order of the Phoenix (Book 5)
Harry Potter y el cÃ¡liz de fuego
Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Prisoner of Azkaban (Book 3)


books by same publisher\:n
The Seeing Stone
The Slightly True Story of Cedar B. Hartley: Who Planned to Live an Unusual Life
Harry Potter and the Chamber of Secrets (Harry Potter)
Book! Book! Book!
The Mouse and His Child
```

In [129…

```
data=pd.DataFrame(Final_dataset.groupby('isbn')['rating'].count()).sort_values('rating',as
data=pd.merge(data,books,on='isbn')
data
```

Loading [MathJax]/extensions/Safe.js

sbn	rating	book_title	book_author	year_of_publication	publisher

	isbn	rating	book_title	book_author	year_of_publication	publisher
0	316666343	632	The Lovely Bones: A Novel	Alice Sebold	2002.0	Little, Brown
1	971880107	524	Wild Animus	Rich Shapero	2004.0	Too Far
2	385504209	450	The Da Vinci Code	Dan Brown	2003.0	Doubleday
3	312195516	351	The Red Tent (Bestselling Backlist)	Anita Diamant	1998.0	Picador USA
4	60928336	292	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997.0	Perennial
...	...	...	...	...	...	...
141535	373291663	1	Virtuous Cyprian (Harlequin Historical Series,...	Nicola Cornick	2001.0	Harlequin
141536	373291655	1	Sea Sprite (Sirens Of The Sea) (Harlequin Hist...	Ruth Langan	2001.0	Harlequin
141537	373291620	1	Taming The Duke (Harlequin Historical Series, ...	Jackie Manning	2001.0	Harlequin
141538	373291582	1	Night Hawk'S Bride (Tyler) (Harlequin Historic...	Jillian Hart	2001.0	Harlequin
141539	B000234N3A	1	Fraud	David Rakoff	2001.0	Doubleday

141540 rows × 6 columns

In [130…

```
years=set()
indices=[]
for ind, row in data.iterrows():
    if row['year_of_publication'] in years:
        indices.append(ind)
    else:
        years.add(row['year_of_publication'])
data=data.drop(indices)
data=data.sort_values('year_of_publication')
pd.set_option("display.max_rows",None, "display.max_columns", None)
data
```

Out[130…

	isbn	rating	book_title	book_author	year_of_publication	publisher
82262	781228956	1	Complete Works 10 Volumes [2,6,7,8,9] (Notable...	Benjamin Franklin	1806.0	Reprint Services Corp
113516	1551103982	1	The Cycling Adventures of Coconut Head: A Nort...	Ted Schredd	1900.0	Graphic Arts Center Pub Co
46560	671825356	1	W D HSE PLANTS	Jd Hersey	1901.0	Simon &amp; Schuster
26088	373226888	2	Tommy's Mom	Linda O. Johnston	1902.0	Harlequin
105939	038528120X	1	CATCH 22	JOSEPH HELLER	1904.0	Delta
119028	404089119	1	Charlotte Bronte and Her Sisters	Clement K. Shorter	1906.0	Ams Pr
23334	911662251	3	Kybalion: A Study of the Hermetic Philosophy o...	Three Initiates	1908.0	Yoga Publication Society
74953	684719274	1	Ethan Frome	Edith Wharton	1910.0	Scribner Book Company

Loading [MathJax]/extensions/Safe.js

	isbn	rating	book_title	book_author	year_of_publication	publisher
59134	877850836	1	The Spiritual Life/The Word of God	Emanuel Swedenborg	1911.0	Steiner Books
4332	684717603	10	The GREAT GATSBY (Scribner Classic)	F. Scott Fritzgerald	1920.0	Scribner Paper Fiction
18741	440772095	3	RAMONA THE PEST (Ramona Quimby (Paperback))	BEVERLY CLEARY	1923.0	Yearling
126343	486600815	1	Principle of Relativity (Dover Books on Physics)	Albert Einstein	1924.0	Dover Publications
42652	23381205	2	Great Gatsby, The	F. Scott Fitzgerald	1925.0	Prentice Hall
26007	684718081	2	The Sun Also Rises	Ernest Hemingway	1926.0	Scribner Book Company
8569	448089033	6	The Secret of the Old Mill (The Hardy Boys Mys...	Franklin W. Dixon	1927.0	Grosset &amp; Dunlap
55077	60830484	1	Point Counter Point	Aldous Huxley	1928.0	Harpercollins
5120	684717972	9	Farewell to Arms	Ernest Hemingway	1929.0	Charles Scribner's Sons
1804	448095017	20	The Secret of the Old Clock (Nancy Drew Myster...	Carolyn Keene	1930.0	Grosset &amp; Dunlap
100934	074752694X	1	Bloomsbury Good Reading Guide	Kenneth McLeish	1931.0	Trafalgar Square Publishing
14320	448095076	4	The Clue in the Diary (Nancy Drew Mystery Stor...	Carolyn Keene	1932.0	Imagination Studio
79536	804617279	1	Legends of Angria,	Charlotte Bronte	1933.0	Gateway/Associated Faculty Pr
66142	9997507002	1	A House Divided	Pearl S. Buck	1935.0	Harpercollins
14542	068483068X	4	Gone With the Wind	Margaret Mitchell	1936.0	Scribner
17157	802812201	3	Descent into Hell	Charles Williams	1937.0	Wm. B. Eerdmans Publishing Company
7955	395071224	6	The Hobbit	J. R. R. Tolkien	1938.0	Houghton Mifflin Company
66141	999750805X	1	The Happy Highwayman	Leslie Charteris	1939.0	Doubleday
26505	771595921	2	Garbage Delight	De Lee	1940.0	Nelson Canada
43745	486206173	2	Rubinstein's Chess Masterpieces	Akiba Rubinstein	1941.0	Dover Publications
9160	60901012	6	Brave New World &amp; Brave New World Revisited	Aldous Huxley	1942.0	Perennial
34238	448095203	2	The Clue in the Jewel Box (Nancy Drew Mystery ...	Carolyn Keene	1943.0	Grosset &amp; Dunlap
117576	394906012	1	The Black Stallion	WALTER FARLEY	1944.0	Random House Books for Young Readers
10722	60263954	5	Stuart Little 60th Anniversary Edition	E. B. White	1945.0	HarperCollins
7115	448060043	7	Alice in Wonderland and Through the Looking Gl...	Charles Lutwidge Dodgson	1946.0	Grosset &amp; Dunlap

	isbn	rating	book_title	book_author	year_of_publication	publisher
14315	448095246	4	The Clue in the Old Album (Nancy Drew Mystery ...	Carolyn Keene	1947.0	Grosset &amp; Dunlap
21568	717802418	3	Manifesto of the Communist Party	Karl Marx	1948.0	International Publishers
17563	448095262	3	Clue of the Leaning Chimney (Nancy Drew (Hardc...	Carolyn Keene	1949.0	Grosset &amp; Dunlap
8945	553213601	6	Cyrano De Bergerac (Bantam Classics)	EDMOND ROSTAND	1950.0	Bantam
4537	081120068X	10	Siddhartha	Hermann Hesse	1951.0	New Directions Publishing Corporation
4106	60263857	11	Charlotte's Web	E. B. White	1952.0	Harpercollins Juvenile Books
1109	64400018	27	Little House in the Big Woods	Laura Ingalls Wilder	1953.0	HarperTrophy
1808	395083567	20	My Antonia	Willa Sibert Cather	1954.0	Houghton Mifflin
4727	156838389	10	Sophocles, The Oedipus Cycle: Oedipus Rex, Oed...	Dudley Fitts	1955.0	Harvest Books
3339	872860175	13	Howl and Other Poems (Pocket Poets)	Allen Ginsberg	1956.0	City Lights Publishers
1166	039480001X	26	The Cat in the Hat	Dr. Seuss	1957.0	Random House Books for Young Readers
3879	394800028	11	The Cat in the Hat Comes Back (I Can Read It A...	Dr. Seuss	1958.0	Random House Books for Young Readers
116	399501487	101	Lord of the Flies	William Gerald Golding	1959.0	Perigee Trade
1838	394800168	20	Green Eggs and Ham (I Can Read It All by Mysel...	Dr. Seuss	1960.0	Random House Books for Young Readers
3456	394800206	12	Go, Dog, Go (I Can Read It All by Myself Begin...	Philip D. Eastman	1961.0	Random House Children's Books
7006	374386137	7	A Wrinkle in Time	Madeleine L'Engle	1962.0	Farrar Straus Giroux
2751	039480029X	15	Hop on Pop (I Can Read It All by Myself Beginn...	Dr. Seuss	1963.0	Random House Books for Young Readers
1947	60256656	19	The Giving Tree	Shel Silverstein	1964.0	HarperCollins Publishers
769	553210092	35	The Scarlet Letter	NATHANIEL HAWTHORNE	1965.0	Bantam
2526	055314300X	16	Bourne Identity	Robert Ludlum	1966.0	Bantam Doubleday Dell
5986	140430229	8	Gulliver's Travels (Penguin Classics)	Jonathan Swift	1967.0	Penguin Books
484	156528207	47	The Little Prince	Antoine de Saint-ExupÃ©ry	1968.0	Harcourt
3977	440435749	11	The High King (Chronicles of Prydain (Paperback))	LLOYD ALEXANDER	1969.0	Yearling

Loading [MathJax]/extensions/Safe.js

	isbn	rating	book_title	book_author	year_of_publication	publisher
1121	20442203	27	Lion, the Witch and the Wardrobe	C.S. Lewis	1970.0	MacMillan Publishing Company.
719	440439884	36	Island of the Blue Dolphins	SCOTT O'DELL	1971.0	Random House Children's Books
1454	64400581	23	Julie of the Wolves (Julie of the Wolves)	Jean Craighead George	1972.0	HarperTrophy
1819	394404289	20	The Prophet	Kahlil Gibran	1973.0	Alfred A. Knopf
314	64400557	59	Charlotte's Web (Trophy Newbery)	E. B. White	1974.0	HarperTrophy
1583	038000321X	22	Alive : The Story of the Andes Survivors (Avon...	Piers Paul Read	1975.0	Avon
158	380002930	85	Watership Down	Richard Adams	1976.0	Avon
2552	380014300	15	Nine Princes In Amber	Roger Zelazny	1977.0	Avon
275	380018179	64	The Thorn Birds	Colleen McCullough	1978.0	Avon
1985	440901588	19	A Swiftly Tilting Planet	Madeleine L'Engle	1979.0	Laure Leaf
482	671250671	47	ROAD LESS TRAVELED : A New Psychology of Love,...	M. Scott Peck	1980.0	Touchstone
526	553208845	45	Siddhartha	Hermann Hesse	1981.0	Bantam Classics
168	553272535	84	Night	Elie Wiesel	1982.0	Bantam Books
132	140067477	94	The Tao of Pooh	Benjamin Hoff	1983.0	Penguin Books
190	553250426	81	The Clan of the Cave Bear (Earth's Children (P...	Jean M. Auel	1984.0	Bantam Books
377	671617028	54	The Color Purple	Alice Walker	1985.0	Pocket Books
48	345339681	146	The Hobbit : The Enchanting Prelude to The Lor...	J.R.R. TOLKIEN	1986.0	Del Rey
45	345342968	148	Fahrenheit 451	RAY BRADBURY	1987.0	Del Rey
24	446310786	193	To Kill a Mockingbird	Harper Lee	1988.0	Little Brown &amp; Company
98	60915544	110	The Bean Trees	Barbara Kingsolver	1989.0	Perennial
36	345361792	167	A Prayer for Owen Meany	John Irving	1990.0	Ballantine Books
15	316769487	220	The Catcher in the Rye	J.D. Salinger	1991.0	Little, Brown
21	044021145X	197	The Firm	John Grisham	1992.0	Bantam Dell Publishing Group
17	345337662	207	Interview with the Vampire	Anne Rice	1993.0	Ballantine Books
31	804106304	176	The Joy Luck Club	Amy Tan	1994.0	Prentice Hall (K-12)
12	067976402X	231	Snow Falling on Cedars	David Guterson	1995.0	Vintage Books USA
26	671003755	184	She's Come Undone (Oprah's Book Club (Paperback))	Wally Lamb	1996.0	Washington Square Press
4	60928336	292	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997.0	Perennial

	isbn	rating	book_title	book_author	year_of_publication	publisher
3	312195516	351	The Red Tent (Bestselling Backlist)	Anita Diamant	1998.0	Picador USA
5	059035342X	283	Harry Potter and the Sorcerer's Stone (Harry P...	J. K. Rowling	1999.0	Arthur A. Levine Books
16	375727345	211	House of Sand and Fog	Andre Dubus III	2000.0	Vintage Books
8	044023722X	258	A Painted House	John Grisham	2001.0	Dell Publishing Company
0	316666343	632	The Lovely Bones: A Novel	Alice Sebold	2002.0	Little, Brown
2	385504209	450	The Da Vinci Code	Dan Brown	2003.0	Doubleday
1	971880107	524	Wild Animus	Rich Shapero	2004.0	Too Far
3216	1881273156	13	The Five Love Languages: Five Love Languages	Gary Chapman	2005.0	Moody Publishers
87461	1903436346	1	The Winter's Tale - Arden Shakespeare : Third ...	John Pitcher	2006.0	Arden
71482	743457226	1	Deep Waters	Jayne Ann Krentz	2010.0	Pocket
118370	394172116	1	Monkey (An Evergreen Book, E-112)	Cheng-En Wu	2011.0	Grove Press
100474	068107468X	1	Edgar Allen Poe Collected Poems	Edgar Allan Poe	2020.0	Bausch &amp; Lombard
35484	590085417	2	Heidi	Johanna Spyri	2021.0	Scholastic

In [131…

```
df=pd.DataFrame(Final_dataset['book_title'].value_counts())
df['Total-Ratings']=df['book_title']
df['book_title']=df.index
df.reset_index(level=0,inplace=True)
df=df.drop('index',axis=1)
df=Final_dataset.merge(df,left_on='book_title',right_on='book_title',how='left')
df=df.drop(['year_of_publication','publisher','Age','country'],axis=1)
```

In [132…

```
popularity_threshold=50
popular_book=df[df['Total-Ratings']>=popularity_threshold]
popular_book=popular_book.reset_index(drop=True)
```

In [133…

```
testdf = pd.DataFrame()
testdf['isbn'] = popular_book['isbn']
testdf['rating'] = popular_book['rating']
testdf['user_id'] = popular_book['user_id']
testdf = testdf[['user_id','rating']].groupby(testdf['isbn'])
```

In [134…

```
listOfDictonaries=[]
indexMap = {}
reverseIndexMap = {}
ptr=0

for groupKey in testdf.groups.keys():
    tempDict={}
    groupDF = testdf.get_group(groupKey)
    for i in range(0,len(groupDF)):
        tempDict[groupDF.iloc[i,0]] = groupDF.iloc[i,1]
                          ptr]=groupKey
```

Loading [MathJax]/extensions/Safe.js

```
    reverseIndexMap[groupKey] = ptr
    ptr=ptr+1
    listOfDictonaries.append(tempDict)

dictVectorizer = DictVectorizer(sparse=True)
vector = dictVectorizer.fit_transform(listOfDictonaries)
pairwiseSimilarity = cosine_similarity(vector)
```

In [135…

```
def printBookDetails(bookID):
    print(Final_dataset[Final_dataset['isbn']==bookID]['book_title'].values[0])
    """
    print("Title:", dataset1[dataset1['ISBN']==bookID]['Book-Title'].values[0])
    print("Author:",dataset1[dataset['ISBN']==bookID]['Book-Author'].values[0])
    #print("Printing Book-ID:",bookID)
    print("\n")
    """

def getTopRecommandations(bookID):
    collaborative = []
    row = reverseIndexMap[bookID]
    print("Input Book:")
    printBookDetails(bookID)

    print("\nRECOMMENDATIONS:\n")

    mn = 0
    similar = []
    for i in np.argsort(pairwiseSimilarity[row])[:-2][::-1]:
          if Final_dataset[Final_dataset['isbn']==indexMap[i]]['book_title'].values[0] not
                if mn>=Number:
                      break
                mn+=1
                similar.append(Final_dataset[Final_dataset['isbn']==indexMap[i]]['book_tit
                printBookDetails(indexMap[i])
                collaborative.append(Final_dataset[Final_dataset['isbn']==indexMap[i]]['bo
    return collaborative
```

In [136…

```
k = list(Final_dataset['book_title'])
m = list(Final_dataset['isbn'])

collaborative = getTopRecommandations(m[k.index(bookName)])
```

```
Input Book:
Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))

RECOMMENDATIONS:

Harry Potter and the Prisoner of Azkaban (Book 3)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Order of the Phoenix (Book 5)
Harry Potter and the Chamber of Secrets (Book 2)
Fried Green Tomatoes at the Whistle Stop Cafe
```

In [137…

```
popularity_threshold = 80
popular_book = df[df['Total-Ratings'] >= popularity_threshold]
popular_book = popular_book.reset_index(drop = True)
popular_book.shape
```

Out[137…

```
(36913, 8)
```

In [138…
```
tf = TfidfVectorizer(ngram_range=(1, 2), min_df = 1, stop_words='english')
tfidf_matrix = tf.fit_transform(popular_book['book_title'])
tfidf_matrix.shape
```

Out[138…
```
(36913, 986)
```

In [139…
```
normalized_df = tfidf_matrix.astype(np.float32)
cosine_similarities = cosine_similarity(normalized_df, normalized_df)
cosine_similarities.shape
```

Out[139…
```
(36913, 36913)
```

In [140…
```
print("Recommended Books:\n")
isbn = books.loc[books['book_title'] == bookName].reset_index(drop = True).iloc[0]['isbn']
content = []

idx = popular_book.index[popular_book['isbn'] == isbn].tolist()[0]
similar_indices = cosine_similarities[idx].argsort()[::-1]
similar_items = []
for i in similar_indices:
    if popular_book['book_title'][i] != bookName and popular_book['book_title'][i] not in
        similar_items.append(popular_book['book_title'][i])
        content.append(popular_book['book_title'][i])

for book in similar_items:
    print(book)
```

```
Recommended Books:

Harry Potter and the Sorcerer's Stone (Book 1)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Prisoner of Azkaban (Book 3)
Harry Potter and the Order of the Phoenix (Book 5)
```

In [143…
```
df.to_excel('my assessment.xlsx' )
```

In [ ]:

Loading [MathJax]/extensions/Safe.js