MACHINE LEARNING

LAB ASSIGNMENT 4

TOPIC CHOSEN: Book Recommendation System

DATASET LINK: https://www.kaggle.com/code/hetulmehta/book-recommendation-system

DATASET NAME:

Books.csv,Ratings.csv, Users.csv

Dataset details:

The Book-Crossing dataset comprises 3 files.

- Users
  Contains the users. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL-values.
- Books
  Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (ImageURL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon web site.
- Ratings
  Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

Link:
https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

Importing Libraries:

```
import numpy as np
import pandas as pd
import seaborn as sns
```

```
import matplotlib.pyplot as plt
pd.set_option('mode.chained_assignment', None)
pd.set_option('display.max_colwidth',None)
```

Uploading Dataset:

```
users = pd.read_csv('Users.csv',low_memory=False)
ratings = pd.read_csv('Ratings.csv',low_memory=False)
books = pd.read_csv('Books.csv',low_memory=False)
```

Reading Dataset:

```
print(books.shape)
books.columns=['ISBN','Title','Author','Year_Of_Publication','Publisher','Image_URL_S','Image_URL_M','Image_URL_L']
books.drop(['Image_URL_S','Image_URL_L'],axis=1,inplace=True)
books.head()
```

```
(221735, 8)
```

	ISBN	Title	Author	Year_Of_Publication	Publisher	Imag
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.01.MZZZ
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.01.MZZZ
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.01.MZZZ
3	0374157065	Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.01.MZZZ

Check for Null Values

```
L=((books.isnull().sum()).sort_values()).to_dict()
for i in L:
    print(i,"--->",L[i])
```

```
ISBN ---> 0
Title ---> 0
Year_Of_Publication ---> 0
Author ---> 1
Image_URL_M ---> 1
Publisher ---> 2
```

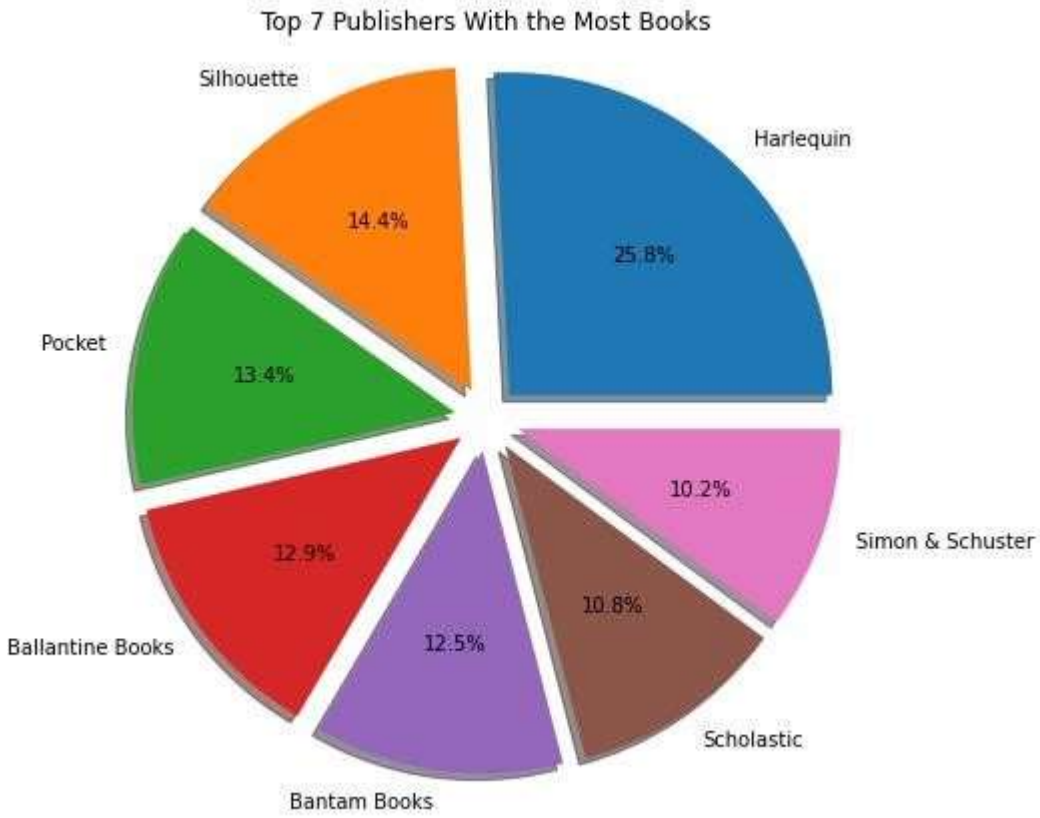
Details of Books:

```
books.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 221735 entries, 0 to 221734
Data columns (total 6 columns):
 #   Column               Non-Null Count   Dtype 
---  ------               --------------   ----- 
 0   ISBN                 221735 non-null  object
 1   Title                221735 non-null  object
 2   Author               221734 non-null  object
 3   Year_Of_Publication  221735 non-null  object
 4   Publisher            221733 non-null  object
 5   Image_URL_M          221734 non-null  object
dtypes: object(6)
memory usage: 10.2+ MB
```

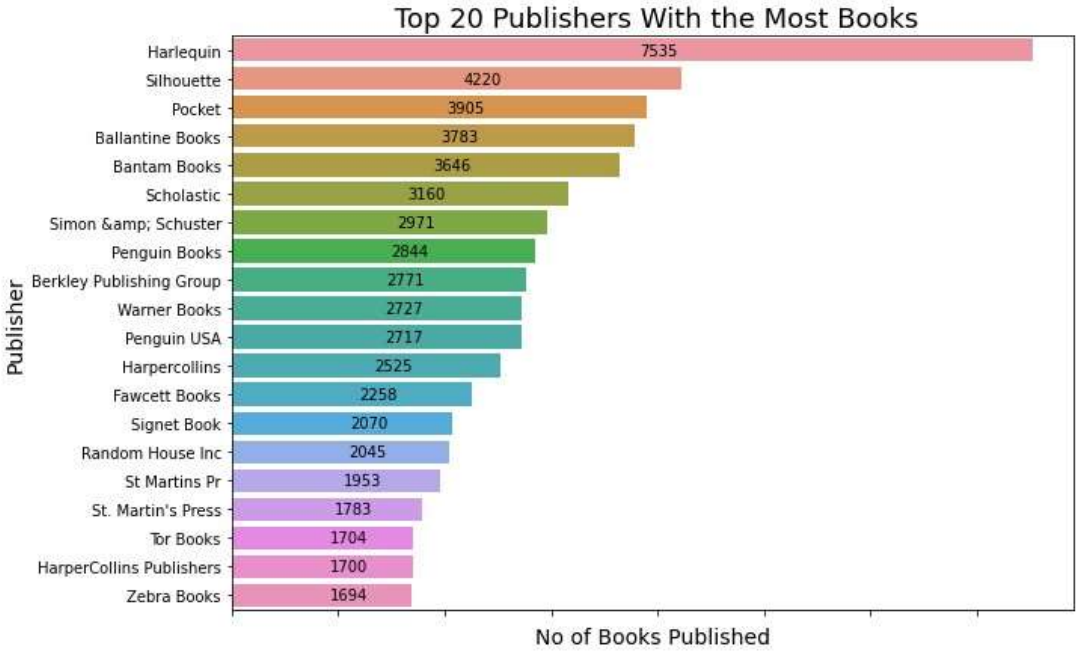
Pie chart for Top 7 publishers:

```
my_dict=(books['Publisher'].value_counts()).to_dict()
count= pd.DataFrame(list(my_dict.items()),columns = ['c','count'])
a = count.sort_values(by=['count'], ascending = False)
a.head(7)
labels = 'Harlequin','Silhouette','Pocket','Ballantine Books','Bantam Books','
Scholastic','Simon & Schuster'
sizes = [count['count'].iloc[0],count['count'].iloc[1],count['count'].iloc[2],
count['count'].iloc[3],count['count'].iloc[4],
         count['count'].iloc[5],count['count'].iloc[6]]
explode = (0.1, 0.1, 0.1, 0.1,0.1, 0.1,0.1 )
fig1 , ax1 = plt.subplots(figsize=(7,7))
ax1.pie(sizes,
        explode = explode,
        labels = labels,
        autopct = '%1.1f%%',
        shadow = True,
        startangle = 0)
plt.title("Top 7 Publishers With the Most Books")
ax1.axis ('equal')
plt.show()
```



Bar chart for Top 20 publishers:

```
b = count.sort_values(by=['count'], ascending = False)
b = b.head(20)
x =['Harlequin','Silhouette','Pocket','Ballantine Books','Bantam Books','Scholastic','Simon &amp;amp; Schuster']
y = [7537,4220,3905,3783,3646,3160,2971]
fig=plt.figure(figsize=(10,7))
ax = sns.barplot(x = 'count',y = 'c' , data = b)
ax.set_xticklabels(ax.get_xticklabels(), rotation=90,horizontalalignment='center')
plt.xlabel("No of Books Published", size=14)
plt.ylabel("Publisher", size=14)
plt.title(" Top 20 Publishers With the Most Books", size=18)
for p in ax.patches:
    ax.annotate("%.0f" % p.get_width(), xy=(p.get_width()/2, p.get_y()+p.get_height()/2),
                xytext=(5, 0), textcoords='offset points', ha="left", va="center")
plt.show()
```



Histogram of the year of publication:

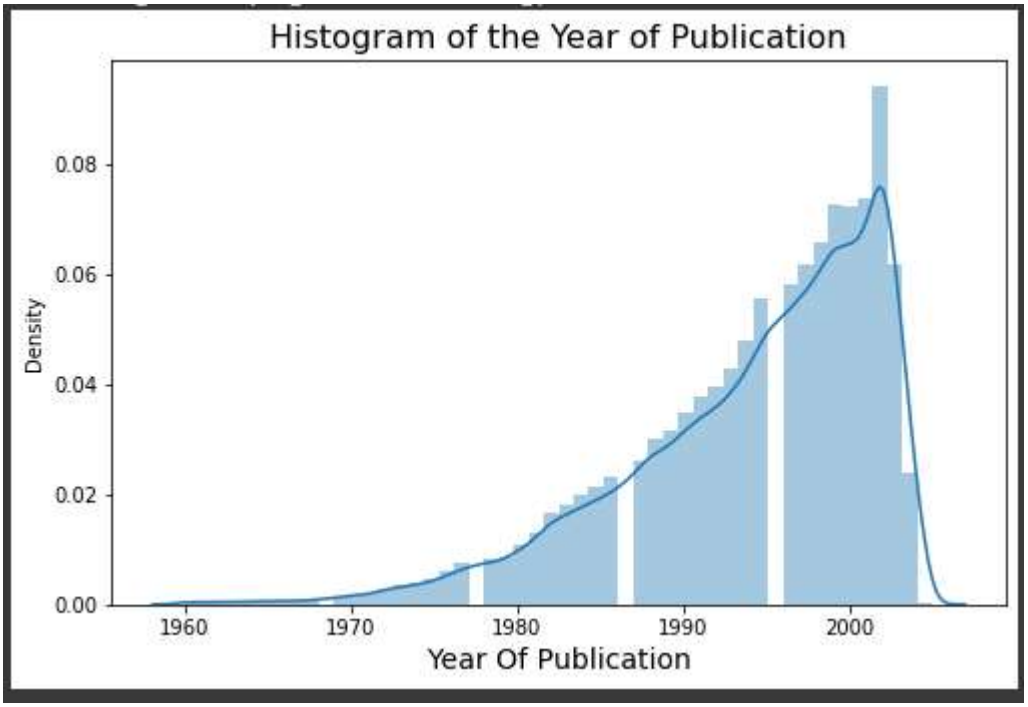
```
np.set_printoptions(threshold=np.inf)
books['Year_Of_Publication'].unique()
```

```
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', '1955', '1946', '1936',
       '1930', '2011', '1925', '1948', '1943', '1947', '1945', '1923',
       '2020', '1939', '1926', '1938', '2030', '1911', '1904', '1949',
       '1932', '1928', '1929', '1927', '1931', '1914', '2050', '1934',
       '1910', '1933', '1902', '1924', '1921', '1900', '2038', '2026',
       '1944', '1917', '1901', '2010', '1908', '1906', '1935', '1806',
       '2021', '2012', '2006', 'DK Publishing Inc', 'Gallimard'],
      dtype=object)
```

```
index=books.loc[books['Year_Of_Publication']=='DK Publishing Inc'].index
books.drop(index,inplace=True)
index=books.loc[books['Year_Of_Publication']=='Gallimard'].index
books.drop(index,inplace=True)
books['Year_Of_Publication'].replace({'0':books['Year_Of_Publication'].value_c
ounts().idxmax()},inplace=True)
books['Year_Of_Publication'] = books['Year_Of_Publication'].astype(str).astype
(int)
books['Year_Of_Publication'].unique()
```

```
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, 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, 1955, 1946, 1936, 1930, 2011, 1925,
       1948, 1943, 1947, 1945, 1923, 2020, 1939, 1926, 1938, 2030, 1911,
       1904, 1949, 1932, 1928, 1929, 1927, 1931, 1914, 2050, 1934, 1910,
       1933, 1902, 1924, 1921, 1900, 2038, 2026, 1944, 1917, 1901, 2010,
       1908, 1906, 1935, 1806, 2021, 2012, 2006])
```

```
fig=plt.figure(figsize=(8,5))
y1 = books[books['Year_Of_Publication'] >= 1960]
y1 = y1[y1['Year_Of_Publication'] <= 2005]
sns.distplot(y1['Year_Of_Publication'])
plt.xlabel('Year Of Publication',size=14)
plt.title('Histogram of the Year of Publication',size=16)
plt.show()
```



User Dataset:

Analyse users dataset:

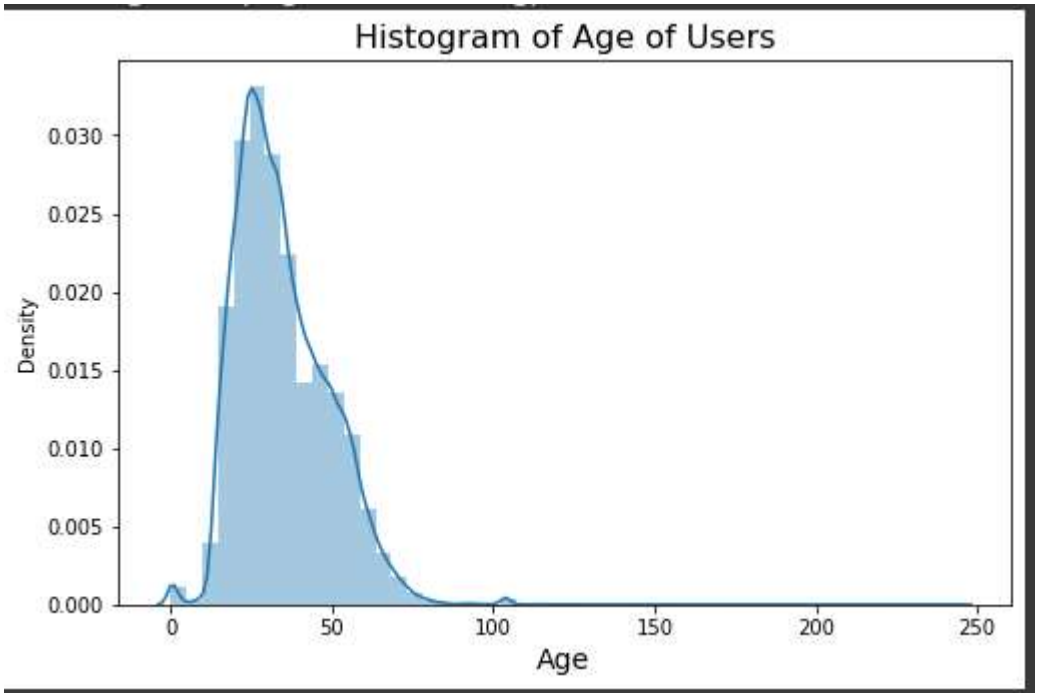
```
print(users.shape)
users.columns=['UserID','Location','Age']
users.head()
```

```
(278858, 3)
```

	UserID	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

Histogram of age of users:

```
fig=plt.figure(figsize=(8,5))
sns.distplot(users['Age'])
plt.xlabel('Age',size=14)
plt.title('Histogram of Age of Users',size=16)
plt.show()
```



User Location:

```
users['Location']
```

```
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
                         ...
278853                portland, oregon, usa
278854    tacoma, washington, united kingdom
278855             brampton, ontario, canada
278856             knoxville, tennessee, usa
278857                  dublin, n/a, ireland
Name: Location, Length: 278858, dtype: object
```

Pie chart for user id with highest rating:

```
users[['city','state','country','nan','nan','nan','nan','nan','nan']]=users['Location'].apply(lambda x: pd.Series(str(x).split(",")))
users.drop(['Location','nan'],axis=1,inplace=True)
users
```

	UserID	Age	city	state	country
0	1	NaN	nyc	new york	usa
1	2	18.0	stockton	california	usa
2	3	NaN	moscow	yukon territory	russia
3	4	17.0	porto	v.n.gaia	portugal
4	5	NaN	farnborough	hants	united kingdom
...	...	...	...	...	...
278853	278854	NaN	portland	oregon	usa
278854	278855	50.0	tacoma	washington	united kingdom
278855	278856	NaN	brampton	ontario	canada
278856	278857	NaN	knoxville	tennessee	usa
278857	278858	NaN	dublin	n/a	ireland

278858 rows × 5 columns

```
print(ratings.shape)
ratings.columns=['UserID','ISBN','Rating']
ratings.head()
```

```
(1149780, 3)
```

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

```
filter1 = ratings[ratings["UserID"].isin(users["UserID"])]
df_ratings=filter1[filter1["ISBN"].isin(books["ISBN"])]
df=pd.merge(users,df_ratings,on='UserID')
df
```

	UserID	Age	city	state	country	ISBN	Rating
0	2	18.0	stockton	california	usa	0195153448	0
1	8	NaN	timmins	ontario	canada	0002005018	5
2	8	NaN	timmins	ontario	canada	0060973129	0
3	8	NaN	timmins	ontario	canada	0374157065	0
4	8	NaN	timmins	ontario	canada	0393045218	0
...	...	...	...	...	...	...	...
975951	278854	NaN	portland	oregon	usa	0425163393	7
975952	278854	NaN	portland	oregon	usa	0515087122	0
975953	278854	NaN	portland	oregon	usa	0553275739	6
975954	278854	NaN	portland	oregon	usa	0553578596	0
975955	278854	NaN	portland	oregon	usa	0553579606	8

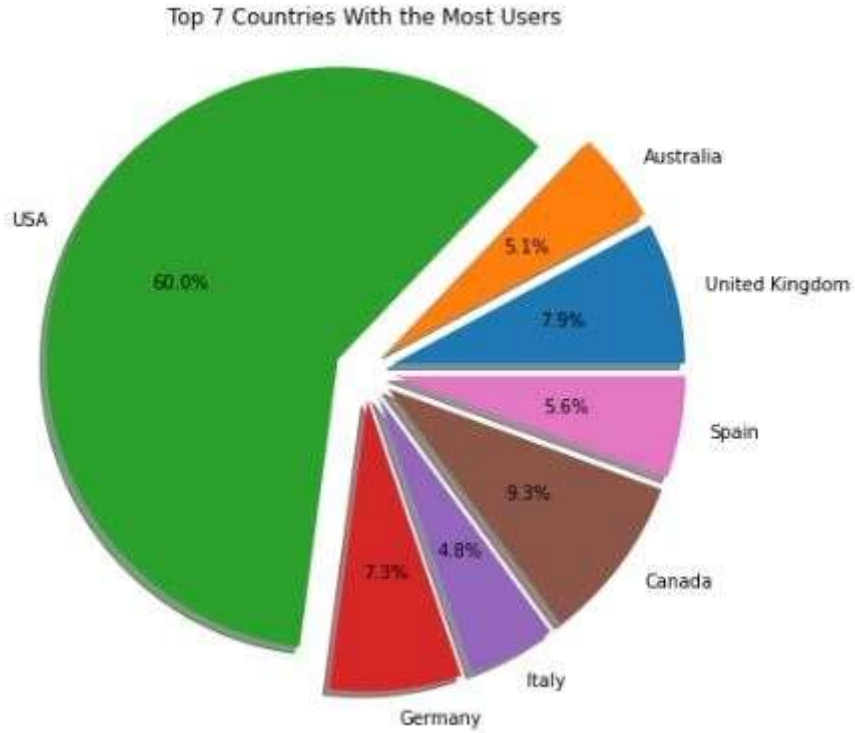
975956 rows × 7 columns

```
my_dict=(users['country'].value_counts()).to_dict()
count= pd.DataFrame(list(my_dict.items()),columns = ['c','count'])
a = count.sort_values(by=['count'], ascending = False)
a.head(7)
labels = 'United Kingdom','Australia','USA','Germany','Italy','Canada','Spain'
sizes = [count['count'].iloc[2],count['count'].iloc[5],count['count'].iloc[0],coun
t['count'].iloc[3],count['count'].iloc[6],
         count['count'].iloc[1],count['count'].iloc[4]]
explode = (0.1, 0.1, 0.1, 0.1,0.1, 0.1,0.1 )

fig1 , ax1 = plt.subplots(figsize=(7,7))

ax1.pie(sizes,
        explode = explode,
        labels = labels,
        autopct = '%1.1f%%',
        shadow = True,
        startangle = 0)
plt.title("Top 7 Countries With the Most Users")
ax1.axis ('equal')

plt.show()
```


Top 7 Countries With the Most Users
USA
60.0%
Australia
5.1%
United Kingdom
7.9%
Spain
5.6%
Canada
9.3%
Italy
4.8%
Germany
7.3%