Assignment 3

September 21, 2024

0.1 Assignment 3

TU Delft and WUR Q1 2024 Instructor: Theodoros Chatzivasileiadis **Instructor:** Hans Hoogenboom **TA:** Ka Yi Chua **Metropolitan Data 1**

This homework assignment document will guide you through five tasks in cleaning your data.

1. Reading and Summarizing the Data.
2. Subsetting the Data.
3. Manage Missing Data.
4. Shape the Data.
5. Saving the Results.

1 NB: From now on you should submit 1) your notebook with the answers, remember that comments are good practice, 2) a working Git page with your assignment

1.1 Exercise 1: Loading the data:

- Load the `goodreads.csv` file into Python
- Explore it by looking at first and last 5 rows
- Change the column names to `["rating", 'review_count', 'isbn', 'booktype','author_url', 'year', 'genre_urls', 'dir','rating_count', 'name']`

[38]:
```
import pandas as pd #import pandas library

goodreads_df=pd.read_csv("data/goodreads.csv") #load raw dataset
```

[39]:
```
goodreads_df.head() #inspect first 5 rows
```

[39]:
```
    4.40   136455  0439023483  good_reads:book  \
0  4.41  16648.0  0439358078  good_reads:book
1  3.56  85746.0  0316015849  good_reads:book
2  4.23  47906.0  0061120081  good_reads:book
3  4.23  34772.0  0679783261  good_reads:book
4  4.25  12363.0  0446675539  good_reads:book

  https://www.goodreads.com/author/show/153394.Suzanne_Collins    2008  \
```

```
0  https://www.goodreads.com/author/show/1077326…            2003.0
1  https://www.goodreads.com/author/show/941441.S…           2005.0
2  https://www.goodreads.com/author/show/1825.Har…           1960.0
3  https://www.goodreads.com/author/show/1265.Jan…           1813.0
4  https://www.goodreads.com/author/show/11081.Ma…           1936.0

  /genres/young-adult|/genres/science-
fiction|/genres/dystopia|/genres/fantasy|/genres/science-
fiction|/genres/romance|/genres/adventure|/genres/book-club|/genres/young-
adult|/genres/teen|/genres/apocalyptic|/genres/post-apocalyptic|/genres/action
\
0  /genres/fantasy|/genres/young-adult|/genres/fi…
1  /genres/young-adult|/genres/fantasy|/genres/ro…
2  /genres/classics|/genres/fiction|/genres/histo…
3  /genres/classics|/genres/fiction|/genres/roman…
4  /genres/classics|/genres/historical-fiction|/g…

                dir01/2767052-the-hunger-games.html    2958974  \
0  dir01/2.Harry_Potter_and_the_Order_of_the_Phoe…  1284478.0
1                          dir01/41865.Twilight.html  2579564.0
2              dir01/2657.To_Kill_a_Mockingbird.html  2078123.0
3                dir01/1885.Pride_and_Prejudice.html  1388992.0
4                dir01/18405.Gone_with_the_Wind.html   645470.0

             The Hunger Games (The Hunger Games, #1)
0  Harry Potter and the Order of the Phoenix (Har…
1                            Twilight (Twilight, #1)
2                              To Kill a Mockingbird
3                                Pride and Prejudice
4                                 Gone with the Wind
```

[40]:
```
goodreads_df.tail() #inspect last 5 rows
```

[40]:
```
      4.40  136455  0439023483  good_reads:book  \
5994  4.17  2226.0  0767913736  good_reads:book
5995  3.99   775.0  1416909427  good_reads:book
5996  3.78   540.0  1620612321  good_reads:book
5997  3.91   281.0         NaN  good_reads:book
5998  4.35    61.0  0786929081  good_reads:book

     https://www.goodreads.com/author/show/153394.Suzanne_Collins    2008  \
5994  https://www.goodreads.com/author/show/44565.Ca…            2005.0
5995  https://www.goodreads.com/author/show/151371.J…            2006.0
5996  https://www.goodreads.com/author/show/5761314…             2012.0
5997  https://www.goodreads.com/author/show/1201952…             2006.0
5998  https://www.goodreads.com/author/show/1023510…             2001.0
```

```
     /genres/young-adult|/genres/science-
fiction|/genres/dystopia|/genres/fantasy|/genres/science-
fiction|/genres/romance|/genres/adventure|/genres/book-club|/genres/young-
adult|/genres/teen|/genres/apocalyptic|/genres/post-apocalyptic|/genres/action
\
5994  /genres/history|/genres/non-fiction|/genres/bi…
5995  /genres/young-adult|/genres/realistic-fiction|…
5996  /genres/contemporary|/genres/romance|/genres/y…
5997  /genres/religion|/genres/islam|/genres/religio…
5998  /genres/fiction|/genres/fantasy|/genres/magic|…

                    dir01/2767052-the-hunger-games.html  2958974  \
5994                dir60/78508.The_River_of_Doubt.html  16618.0
5995                             dir60/259068.Shug.html   6179.0
5996                          dir60/13503247-flawed.html   2971.0
5997                               dir60/2750008.html   3083.0
5998  dir60/66677.Legacy_of_the_Drow_Collector_s_Edi…   3982.0

                The Hunger Games (The Hunger Games, #1)
5994                                  The River of Doubt
5995                                               Shug
5996                                             Flawed
5997               Ø£Ø³Ø¹Ø¯ Ø§Ù
Ø±Ø£Ø© Ù Ù  Ø§Ù Ø¹Ø§Ù Ù

5998  Legacy of the Drow Collector's Edition (Legacy…
```

[41]:
```
#load goodread_df with required column names (from lab-03-part-02)
goodreads_df=pd.read_csv("data/goodreads.csv", header=None, names=["rating", \
    'review_count', 'isbn', 'booktype','author_url', 'year', 'genre_urls',␣
 ↪'dir','rating_count', 'name'],)

goodreads_df.head() #display first 5 rows
```

[41]:
```
   rating  review_count        isbn         booktype  \
0    4.40      136455.0  0439023483  good_reads:book
1    4.41       16648.0  0439358078  good_reads:book
2    3.56       85746.0  0316015849  good_reads:book
3    4.23       47906.0  0061120081  good_reads:book
4    4.23       34772.0  0679783261  good_reads:book

                                          author_url    year  \
0  https://www.goodreads.com/author/show/153394.S…  2008.0
1  https://www.goodreads.com/author/show/1077326…  2003.0
2  https://www.goodreads.com/author/show/941441.S…  2005.0
3  https://www.goodreads.com/author/show/1825.Har…  1960.0
4  https://www.goodreads.com/author/show/1265.Jan…  1813.0
```

```
                                          genre_urls  \
0  /genres/young-adult|/genres/science-fiction|/g…
1  /genres/fantasy|/genres/young-adult|/genres/fi…
2  /genres/young-adult|/genres/fantasy|/genres/ro…
3  /genres/classics|/genres/fiction|/genres/histo…
4  /genres/classics|/genres/fiction|/genres/roman…

                                             dir  rating_count  \
0                dir01/2767052-the-hunger-games.html     2958974.0
1  dir01/2.Harry_Potter_and_the_Order_of_the_Phoe…     1284478.0
2                          dir01/41865.Twilight.html     2579564.0
3               dir01/2657.To_Kill_a_Mockingbird.html     2078123.0
4                 dir01/1885.Pride_and_Prejudice.html     1388992.0

                                          name
0           The Hunger Games (The Hunger Games, #1)
1  Harry Potter and the Order of the Phoenix (Har…
2                          Twilight (Twilight, #1)
3                            To Kill a Mockingbird
4                              Pride and Prejudice
```

1.2 Exercise 2: Subsetting the data

- Subset the data by creating new dataframe only with `["rating", 'isbn', 'author_url', 'year', 'genre_urls', 'name']`

[42]:
```
desired_columns = ["rating", 'isbn', 'author_url', 'year', 'genre_urls',␣
↪'name'] #put all desired subset columns in a list
subset_df = goodreads_df[desired_columns] #make new dataframe with these␣
↪desired columns

subset_df.head() #display first 5 rows of subset dataframe
```

[42]:
```
   rating        isbn                                          author_url  \
0    4.40  0439023483  https://www.goodreads.com/author/show/153394.S…
1    4.41  0439358078  https://www.goodreads.com/author/show/1077326…
2    3.56  0316015849  https://www.goodreads.com/author/show/941441.S…
3    4.23  0061120081  https://www.goodreads.com/author/show/1825.Har…
4    4.23  0679783261  https://www.goodreads.com/author/show/1265.Jan…

     year                                          genre_urls  \
0  2008.0  /genres/young-adult|/genres/science-fiction|/g…
1  2003.0  /genres/fantasy|/genres/young-adult|/genres/fi…
2  2005.0  /genres/young-adult|/genres/fantasy|/genres/ro…
3  1960.0  /genres/classics|/genres/fiction|/genres/histo…
4  1813.0  /genres/classics|/genres/fiction|/genres/roman…
```

```
                                                name
0            The Hunger Games (The Hunger Games, #1)
1  Harry Potter and the Order of the Phoenix (Har…
2                             Twilight (Twilight, #1)
3                               To Kill a Mockingbird
4                                 Pride and Prejudice
```

1.3 Exercise 3: Manage Missing Data

We've got a number of ways in general of dealing with missing data. These involve

1. Dropping off cases (or rows) in the data with any missing variables
2. Excluding variables in the data with any missing data
3. Selectively choosing indicators with only a limited amount of missing data
4. Replacing missing variables with averages, or other representative values
5. Creating a separate model to predict missing data

- Count the missing values in each column
- Manage the missing values (delete or replace values or leave them as they are) and briefly explain your choice for each column

[43]:
```
import numpy as np #numpy needed for count

def missing_value_count(dataframe): #define a function as want to view missing␣
 ↪values multiple times
    nan_counts={} #empty dictionary to store the counts of each columns
    for column in dataframe.columns: #for every column of subset dataframe
        nan_count=np.sum(dataframe[column].isnull())#from lab-03-part-02
        nan_counts[column] = nan_count #add missing count to dictionary
    return(nan_counts)

print(missing_value_count(subset_df)) #print NaN counts for current dataframe
```

```
{'rating': np.int64(2), 'isbn': np.int64(477), 'author_url': np.int64(2),
'year': np.int64(7), 'genre_urls': np.int64(62), 'name': np.int64(2)}
```

[44]:
```
#display  name missing rows
subset_df[subset_df.name.isnull()] #from lab-03-part-02
```

[44]:
```
      rating isbn author_url  year genre_urls name
3643     NaN  NaN        NaN   NaN        NaN  NaN
5282     NaN  NaN        NaN   NaN        NaN  NaN
```

[45]:
```
'''
Found that 2 rows have NaN values for all of them. Can delete these rows. These␣
 ↪rows show the entire NaN count for rating, author_url and name and deleting
should reduce their missing value count to 0.
```

```
'''
subset_df=subset_df.dropna(subset=["name"]) # dropna deletes an entire row␣
 ↪where missing value https://pandas.pydata.org/docs/reference/api/pandas.
 ↪DataFrame.dropna.html
print(missing_value_count(subset_df)) #should display 0's for rating,␣
 ↪author_url and name
```

```
{'rating': np.int64(0), 'isbn': np.int64(475), 'author_url': np.int64(0),
'year': np.int64(5), 'genre_urls': np.int64(60), 'name': np.int64(0)}
```

[46]:
```
subset_df[subset_df.year.isnull()] #look at the 5 rows where year is missing
```

[46]:

	rating	isbn	author_url	year	genre_urls	name
2442	4.23	NaN	https://www.goodreads.com/author/show/623606.A…	NaN	/genres/religion|/genres/islam|/genres/non-fic…	La Tahzan
2869	4.61	NaN	https://www.goodreads.com/author/show/8182217…	NaN	NaN	My Death Experiences - A Preacherâ s 18 Apoca…
5572	3.71	8423336603	https://www.goodreads.com/author/show/285658.E…	NaN	/genres/fiction	Ã rase una vez el amor pero tuve que matarlo. …
5658	4.32	NaN	https://www.goodreads.com/author/show/25307.Ro…	NaN	/genres/fantasy|/genres/fantasy|/genres/epic-f…	Assassin's Apprentice / Royal Assassin (Farsee…
5683	4.56	NaN	https://www.goodreads.com/author/show/3097905…	NaN	/genres/fantasy|/genres/young-adult|/genres/ro…	Tiger's Dream (The Tiger Saga, #5)

[47]:
```
'''
Since there is only 5 values missing for year we can find these ourselves.

(2442) La Tahzan - Aidh bin Abdullah al Qarni
Year: 2003 (https://www.amazon.co.uk/Dont-Sad-Aaidh-Abdullah-al-Qarni/dp/
 ↪9960850447) *cant find goodreads*

(2869) My Death Experiences - A Preacherâs 18 Apocalyptic Encounter with Death,␣
 ↪Heaven & Hell - Zion Odum
Year: 2014 (https://www.goodreads.com/book/show/
 ↪22031070-my-death-experiences---a-preacher-s-18-apocalyptic-encounter-with-death?
 ↪from_search=true&from_srp=true&qid=yuvTulgEpt&rank=1)
```

```
(5572) Ã rase una vez el amor pero tuve que matarlo. Musica de Sex Pistols y␣
 ↪Nirvana - Efraim_Medina_Reyes'
Year: 2002 (https://www.goodreads.com/book/show/890680.
 ↪_rase_una_vez_el_amor_pero_tuve_que_matarlo_M_sica_de_Sex_Pistols_y_Nirvana)

(5658) "Assassin's Apprentice / Royal Assassin (Farseer Trilogy, #1-2)"-␣
 ↪Robin_Hobb
Year: 2002 (https://www.goodreads.com/book/show/
 ↪5533041-assassin-s-apprentice-royal-assassin?
 ↪from_search=true&from_srp=true&qid=18tVU2qEF6&rank=1)

(5683) "Tiger's Dream (The Tiger Saga, #5)" - Colleen_Houck
Year: 2018 (https://www.goodreads.com/book/show/12474623-tiger-s-dream)
'''

subset_df.loc[2442, 'year'] = np.int64(2003) #manually entering year values to␣
 ↪dataframe, int64 to keep consistent format as other values in dataframe
subset_df.loc[2869, 'year'] = np.int64(2014)
subset_df.loc[5572, 'year'] = np.int64(2002)
subset_df.loc[5658, 'year'] = np.int64(2002)
subset_df.loc[5683, 'year'] = np.int64(2018)

print(missing_value_count(subset_df)) #should display year with 0 NaN
```

```
{'rating': np.int64(0), 'isbn': np.int64(475), 'author_url': np.int64(0),
'year': np.int64(0), 'genre_urls': np.int64(60), 'name': np.int64(0)}
```

[48]:
```
subset_df[subset_df.isbn.isnull()] #look at rows with isbn missing
```

[48]:

	rating	isbn	author_url	year	genre_urls
16	3.92	NaN	https://www.goodreads.com/author/show/498072.A…	2003.0	/genres/fiction|/genres/romance|/genres/fantas…
49	3.85	NaN	https://www.goodreads.com/author/show/5152.Vla…	1955.0	/genres/classics|/genres/fiction|/genres/liter…
85	4.16	NaN	https://www.goodreads.com/author/show/137902.R…	2007.0	/genres/young-adult|/genres/teen|/genres/young…
116	3.92	NaN	https://www.goodreads.com/author/show/957894.A…	1942.0	/genres/classics|/genres/fiction|/genres/philo…
156	4.03	NaN	https://www.goodreads.com/author/show/4785.Ale…	1843.0	
…	…	…	…	…	
5972	4.19	NaN	https://www.goodreads.com/author/show/4586597…	2011.0	
5976	4.23	NaN	https://www.goodreads.com/author/show/5160667…	2014.0	
5977	4.03	NaN	https://www.goodreads.com/author/show/5769580…	1987.0	
5991	4.20	NaN	https://www.goodreads.com/author/show/1112683._	2009.0	
5998	3.91	NaN	https://www.goodreads.com/author/show/1201952…	2006.0	

```
156   /genres/classics|/genres/fiction|/genres/histo…
…                                                   …
5972  /genres/romance|/genres/romance|/genres/contem…
5976  /genres/romance|/genres/science-fiction|/genre…
5977  /genres/fiction|/genres/novels|/genres/literat…
5991  /genres/novels|/genres/fiction|/genres/religio…
5998  /genres/religion|/genres/islam|/genres/religio…

                                        name
16                  The Time Traveler's Wife
49                                     Lolita
85     Vampire Academy (Vampire Academy, #1)
116                              The Stranger
156                     The Three Musketeers
…                                         …
5972     Perfection (Neighbor from Hell, #2)
5976                             Transcendence
5977                                ØSÙ Ø³Ù Ù
5991                        Ø£Ù Ù ØSØ  Ù Ø¯Ø³Ø±
5998  Ø£Ø³Ø¹Ø¯ ØSÙ
Ø±Ø£Ø© Ù Ù  ØSÙ Ø¹ØSÙ Ù

[475 rows x 6 columns]
```

[49]:
```
'''
475 isbn still missing, however will leave these as not necessary for our data␣
 ↪analysis and the books can be uniquely identified
through their author and name. Can just fill as "-". This could be solved␣
 ↪through webscraping and Beautiful Soup, but thats
a lot of work when the isbn is not necessary for our current data analysis.
'''

subset_df['isbn'].fillna('-', inplace=True) #fill all missing with '-' https://
 ↪pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html
```

```
/tmp/ipykernel_96834/3543925312.py:7: FutureWarning: A value is trying to be set
on a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.
```

```
  subset_df['isbn'].fillna('-', inplace=True)
#https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html
```

1.3.1 Machine learning, predicting genre based on name (extra)

The only column left with missing values is genre. In order to solve this issue we have decided to use machine learning with scikit learn to predict the multi label genres based on the name of the book. To achieve this we must convert the book name to sentence embeddings using the bert transformers model and then convert the multi label genre to 0's and 1's across all the possible unique genres. We can then train these bert embeddings to predict an array of 0's and 1's corresponding to an array of target genres.

[50]:
```
'''
Grab all the unique genres
'''

unique_genres=[]
unique_values = subset_df['genre_urls'].unique() #grab the unique values from␣
 ↪dataframe in case some books have exact repeat of combination of genres
for value in unique_values:
    try: #gained an error about "float" value but couldnt solve, not necessary␣
 ↪to when we just storing unique values therefore added 'try' to continue␣
 ↪through this error
        y=value.split("|") #split the genre string into a list where each item␣
 ↪is /genre/**genre**
        for x in y: #for every individual genre in this list
            if x not in unique_genres: #if we have not already added that genre␣
 ↪to the list
                unique_genres.append(x) #add that genre to the unique values.
    except:
        continue
print(unique_genres) #all the possible genres individually without repeats␣
 ↪(length 537)
```

```
['/genres/young-adult', '/genres/science-fiction', '/genres/dystopia',
'/genres/fantasy', '/genres/romance', '/genres/adventure', '/genres/book-club',
'/genres/teen', '/genres/apocalyptic', '/genres/post-apocalyptic',
'/genres/action', '/genres/fiction', '/genres/magic', '/genres/childrens',
'/genres/science-fiction-fantasy', '/genres/paranormal', '/genres/vampires',
'/genres/paranormal-romance', '/genres/supernatural', '/genres/urban-fantasy',
'/genres/classics', '/genres/historical-fiction', '/genres/academic',
'/genres/school', '/genres/literature', '/genres/read-for-school',
'/genres/novels', '/genres/high-school', '/genres/european-literature',
'/genres/british-literature', '/genres/classic-literature', '/genres/adult',
'/genres/historical-romance', '/genres/war', '/genres/military-history',
'/genres/civil-war', '/genres/christian', '/genres/religion', '/genres/picture-
books', '/genres/inspirational', '/genres/philosophy', '/genres/juvenile',
```

'/genres/short-stories', '/genres/politics', '/genres/humor', '/genres/comedy',
'/genres/funny', '/genres/cultural', '/genres/japan', '/genres/contemporary',
'/genres/adult-fiction', '/genres/mystery', '/genres/thriller',
'/genres/suspense', '/genres/mystery-thriller', '/genres/crime', '/genres/world-
war-ii', '/genres/holocaust', '/genres/history', '/genres/19th-century',
'/genres/plays', '/genres/drama', '/genres/france', '/genres/french-literature',
'/genres/time-travel', '/genres/womens-fiction', '/genres/chick-lit',
'/genres/epic-fantasy', '/genres/epic', '/genres/high-fantasy',
'/genres/horror', '/genres/gothic', '/genres/russia', '/genres/russian-
literature', '/genres/animals', '/genres/middle-grade', '/genres/chapter-books',
'/genres/canada', '/genres/coming-of-age', '/genres/spirituality',
'/genres/american', '/genres/fairy-tales', '/genres/southern',
'/genres/realistic-fiction', '/genres/magical-realism', '/genres/spanish-
literature', '/genres/latin-american', '/genres/kids', '/genres/literary-
fiction', '/genres/media-tie-in', '/genres/movies', '/genres/modern',
'/genres/love-story', '/genres/poetry', '/genres/food-and-drink',
'/genres/food', '/genres/angels', '/genres/shapeshifters', '/genres/werewolves',
'/genres/india', '/genres/mythology', '/genres/modern-classics',
'/genres/medieval', '/genres/space-opera', '/genres/speculative-fiction',
'/genres/new-york', '/genres/detective', '/genres/non-fiction',
'/genres/autobiography', '/genres/memoir', '/genres/ireland',
'/genres/biography', '/genres/biography-memoir', '/genres/irish-literature',
'/genres/psychology', '/genres/mental-health', '/genres/mental-illness',
'/genres/feminism', '/genres/africa', '/genres/african-american',
'/genres/dragons', '/genres/economics', '/genres/scotland', '/genres/greek-
mythology', '/genres/buddhism', '/genres/german-literature', '/genres/spain',
'/genres/young-adult-fantasy', '/genres/love', '/genres/womens',
'/genres/contemporary-romance', '/genres/glbt', '/genres/steampunk',
'/genres/travel', '/genres/science', '/genres/mathematics',
'/genres/environment', '/genres/nature', '/genres/czech-literature',
'/genres/self-help', '/genres/theatre', '/genres/western', '/genres/20th-
century', '/genres/writing', '/genres/books-about-books', '/genres/italy',
'/genres/italian-literature', '/genres/true-crime', '/genres/sequential-art',
'/genres/graphic-novels', '/genres/graphic-novels-comics', '/genres/comics',
'/genres/superheroes', '/genres/murder-mystery', '/genres/military',
'/genres/germany', '/genres/australia', '/genres/storytime', '/genres/china',
'/genres/asia', '/genres/holiday', '/genres/arthurian', '/genres/essays',
'/genres/islam', '/genres/prehistoric', '/genres/banned-books',
'/genres/witches', '/genres/dogs', '/genres/asian-literature',
'/genres/japanese-literature', '/genres/pirates', '/genres/christian-fiction',
'/genres/art', '/genres/erotica', '/genres/bdsm', '/genres/erotic-romance',
'/genres/christianity', '/genres/theology', '/genres/faith', '/genres/indian-
literature', '/genres/family', '/genres/journalism', '/genres/race',
'/genres/southern-gothic', '/genres/survival', '/genres/tragedy',
'/genres/ancient', '/genres/comic-book', '/genres/comic-strips',
'/genres/college', '/genres/17th-century', '/genres/demons', '/genres/18th-
century', '/genres/sociology', '/genres/abuse', '/genres/sweden',
'/genres/childrens-classics', '/genres/music', '/genres/reference',

'/genres/egypt', '/genres/space', '/genres/roman', '/genres/popular-science',
'/genres/physics', '/genres/new-adult', '/genres/african-literature',
'/genres/regency', '/genres/anthologies', '/genres/law', '/genres/taoism',
'/genres/eastern-philosophy', '/genres/legal-thriller', '/genres/cyberpunk',
'/genres/horses', '/genres/ghosts', '/genres/biology', '/genres/evolution',
'/genres/natural-history', '/genres/true-story', '/genres/european-history',
'/genres/french-revolution', '/genres/jewish', '/genres/road-trip',
'/genres/political-science', '/genres/dark', '/genres/low-fantasy',
'/genres/education', '/genres/christian-romance', '/genres/american-history',
'/genres/dark-fantasy', '/genres/aliens', '/genres/atheism', '/genres/english-
literature', '/genres/birds', '/genres/young-adult-romance', '/genres/young-
adult-contemporary', '/genres/robots', '/genres/fairies', '/genres/fae',
'/genres/swedish-literature', '/genres/christmas', '/genres/zombies',
'/genres/sports', '/genres/mountaineering', '/genres/outdoors',
'/genres/alternate-history', '/genres/presidents', '/genres/retellings',
'/genres/sports-and-games', '/genres/unicorns', '/genres/greece',
'/genres/african-american-literature', '/genres/personal-development',
'/genres/new-age', '/genres/health', '/genres/nutrition', '/genres/cooking',
'/genres/foodie', '/genres/spy-thriller', '/genres/espionage',
'/genres/folklore', '/genres/death', '/genres/noir', '/genres/judaism',
'/genres/queer', '/genres/historical-mystery', '/genres/gender', '/genres/lds',
'/genres/church', '/genres/business', '/genres/social-science',
'/genres/culture', '/genres/anthropology', '/genres/leadership',
'/genres/management', '/genres/productivity', '/genres/fairy-tale-retellings',
'/genres/astronomy', '/genres/art-history', '/genres/israel',
'/genres/marriage', '/genres/relationships', '/genres/hinduism',
'/genres/language', '/genres/communication', '/genres/young-adult-paranormal',
'/genres/nobel-prize', '/genres/scandinavian-literature', '/genres/denmark',
'/genres/danish', '/genres/lovecraftian', '/genres/cats', '/genres/animal-
fiction', '/genres/computer-science', '/genres/14th-century', '/genres/dc-
comics', '/genres/batman', '/genres/baseball', '/genres/wizards', '/genres/gay',
'/genres/romantic-suspense', '/genres/latin-american-literature',
'/genres/lesbian', '/genres/polish-literature', '/genres/iran', '/genres/weird-
fiction', '/genres/new-weird', '/genres/ukraine', '/genres/gardening',
'/genres/occult', '/genres/rabbits', '/genres/genetics', '/genres/portugal',
'/genres/portuguese-literature', '/genres/turkish-literature',
'/genres/collections', '/genres/wildlife', '/genres/world-history',
'/genres/medical', '/genres/kenya', '/genres/humanities', '/genres/government',
'/genres/serbian-literature', '/genres/manga', '/genres/comics-manga',
'/genres/social-movements', '/genres/social-justice', '/genres/social-issues',
'/genres/poverty', '/genres/15th-century', '/genres/sports-romance',
'/genres/american-civil-war', '/genres/wolves', '/genres/romantic',
'/genres/lds-non-fiction', '/genres/folk-tales', '/genres/fables',
'/genres/mormonism', '/genres/ecology', '/genres/sustainability',
'/genres/skepticism', '/genres/comix', '/genres/brazil', '/genres/medicine',
'/genres/turkish', '/genres/indonesian-literature', '/genres/time-travel-
romance', '/genres/hard-boiled', '/genres/anime', '/genres/shojo',
'/genres/young-adult-historical-fiction', '/genres/technology',

'/genres/poland', '/genres/theory', '/genres/regency-romance', '/genres/pulp',
'/genres/16th-century', '/genres/americana', '/genres/dutch-literature',
'/genres/criticism', '/genres/role-playing-games', '/genres/dungeons-and-
dragons', '/genres/forgotten-realms', '/genres/dragonlance', '/genres/utopia',
'/genres/teaching', '/genres/terrorism', '/genres/young-adult-science-fiction',
'/genres/gods', '/genres/dinosaurs', '/genres/pakistan', '/genres/finnish-
literature', '/genres/disability', '/genres/games', '/genres/heroic-fantasy',
'/genres/sword-and-sorcery', '/genres/parenting', '/genres/menage',
'/genres/metaphysics', '/genres/wilderness', '/genres/travelogue',
'/genres/bizarro-fiction', '/genres/christian-historical-fiction', '/genres/m-m-
romance', '/genres/social', '/genres/gender-studies', '/genres/romanian-
literature', '/genres/architecture', '/genres/food-writing', '/genres/canadian-
literature', '/genres/psychological-thriller', '/genres/lds-fiction',
'/genres/fitness', '/genres/counter-culture', '/genres/anarchism',
'/genres/satanism', '/genres/trans', '/genres/transgender', '/genres/diary',
'/genres/chinese-literature', '/genres/mysticism', '/genres/egyptian-
literature', '/genres/ghost-stories', '/genres/planetary-romance',
'/genres/star-wars', '/genres/poetry-plays', '/genres/international',
'/genres/engineering', '/genres/mermaids', '/genres/american-fiction',
'/genres/anthropomorphic', '/genres/lesbian-fiction', '/genres/pop-culture',
'/genres/female-authors', '/genres/western-romance', '/genres/cthulhu-mythos',
'/genres/shonen', '/genres/how-to', '/genres/belgium', '/genres/sci-fi-fantasy',
'/genres/school-stories', '/genres/boarding-school', '/genres/united-states',
'/genres/catholic', '/genres/fantasy-romance', '/genres/cozy-mystery',
'/genres/chess', '/genres/photography', '/genres/cycling', '/genres/m-f-m',
'/genres/musicians', '/genres/hard-science-fiction', '/genres/medieval-romance',
'/genres/cartoon', '/genres/beauty-and-the-beast', '/genres/god',
'/genres/sexuality', '/genres/church-history', '/genres/clean-romance',
'/genres/adoption', '/genres/maritime', '/genres/queer-lit', '/genres/hungarian-
literature', '/genres/hungary', '/genres/young-readers', '/genres/vegan',
'/genres/cities', '/genres/urban-planning', '/genres/geography',
'/genres/urbanism', '/genres/crafts', '/genres/crafty', '/genres/wicca',
'/genres/film', '/genres/guides', '/genres/womens-studies', '/genres/video-
games', '/genres/walking', '/genres/josei', '/genres/military-science-fiction',
'/genres/journal', '/genres/category-romance', '/genres/harlequin',
'/genres/rock-n-roll', '/genres/canon', '/genres/logic', '/genres/short-story-
collection', '/genres/gothic-horror', '/genres/dictionaries',
'/genres/prehistory', '/genres/illness', '/genres/slice-of-life',
'/genres/princesses', '/genres/drawing', '/genres/design', '/genres/cult-
classics', '/genres/us-presidents', '/genres/zen', '/genres/academia',
'/genres/romania', '/genres/neuroscience', '/genres/georgian-romance',
'/genres/belgian', '/genres/light-novel', '/genres/love-inspired',
'/genres/love-inspired-historical', '/genres/bulgarian-literature',
'/genres/bulgaria', '/genres/activism', '/genres/mine', '/genres/literary-
criticism', '/genres/christian-contemporary-fiction', '/genres/textbooks',
'/genres/marvel', '/genres/x-men', '/genres/surreal', '/genres/horse-racing',
'/genres/brain', '/genres/dying-earth', '/genres/post-colonial',
'/genres/classical-studies', '/genres/superman', '/genres/football',

```
'/genres/gay-romance', '/genres/gay-for-you', '/genres/bande-dessin%C3%A9e',
'/genres/esoterica', '/genres/favorites', '/genres/history-of-science',
'/genres/technical', '/genres/science-fiction-romance', '/genres/erotic-
historical-romance', '/genres/tasmania', '/genres/cookbooks',
'/genres/basketball', '/genres/culinary', '/genres/epic-poetry',
'/genres/cults', '/genres/society', '/genres/komik', '/genres/adolescence',
'/genres/holland', '/genres/gaming', '/genres/interracial-romance',
'/genres/soldiers', '/genres/swashbuckling', '/genres/food-history',
'/genres/fat', '/genres/fat-acceptance', '/genres/fat-studies',
'/genres/teachers', '/genres/near-future', '/genres/paganism', '/genres/gothic-
romance', '/genres/splatterpunk', '/genres/american-novels',
'/genres/scripture']
```

[51]:
```
'''
Make new training table of name and genre.
'''
filtered_df = subset_df[subset_df['name'].notna() & subset_df['genre_urls'].
 ↪notna()] #remove missing values as using this for training
new_df = filtered_df[['name', 'genre_urls']] #make fresh table separate to make␣
 ↪it clean for training
new_df.head() #display first 5 rows
```

[51]:

	name	genre_urls
0	The Hunger Games (The Hunger Games, #1)	/genres/young-adult|/genres/science-fiction|/g…
1	Harry Potter and the Order of the Phoenix (Har…	/genres/fantasy|/genres/young-adult|/genres/fi…
2	Twilight (Twilight, #1)	/genres/young-adult|/genres/fantasy|/genres/ro…
3	To Kill a Mockingbird	/genres/classics|/genres/fiction|/genres/histo…
4	Pride and Prejudice	/genres/classics|/genres/fiction|/genres/roman…

[52]:
```
'''
Make function to convert sentence to vector embedding using transformers bert␣
 ↪uncased.
'''

from transformers import BertTokenizer, BertModel
import torch

#source: https://www.geeksforgeeks.org/
 ↪how-to-generate-word-embedding-using-bert/
#source: previous work files
```

```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased') #load pretrained␣
 ↪tokenizer
model = BertModel.from_pretrained('bert-base-uncased') #load pretrained model␣
 ↪bert

def text_to_embedding(text): #function to make word embeddings
    inputs = tokenizer(text, return_tensors="pt", padding=True,␣
 ↪truncation=True, max_length=512) #tokenize the input text and convert to␣
 ↪tensors
    with torch.no_grad(): #no computing gradients
        outputs = model(**inputs) #get bert embeddings
    cls_embedding = outputs.last_hidden_state[:, 0, :]  # use the [CLS] token␣
 ↪representation as the embedding, represents entire sentence
    return cls_embedding
```

```
/mnt/nvme0n1p1/made/data1/env/lib/python3.10/site-
packages/transformers/tokenization_utils_base.py:1601: FutureWarning:
`clean_up_tokenization_spaces` was not set. It will be set to `True` by default.
This behavior will be depracted in transformers v4.45, and will be then set to
`False` by default. For more details check this issue:
https://github.com/huggingface/transformers/issues/31884
  warnings.warn(
```

[53]:
```
'''
Make and store vector embeddings and encoding of different genres for training.
'''

one_hot_encodings_genre=[] #define empty list to save binary encodings
embeddings=[] #define empty list to save each embeddings

for index, row in new_df.iterrows():
    genre_list=row['genre_urls'].split('|')  #split each item genres (multi␣
 ↪label) into a list
    one_hot = [1 if genre in genre_list else 0 for genre in unique_genres]␣
 ↪#from our unique genres list made previously for each index if the genre is␣
 ↪there add 1 else 0
    one_hot_encodings_genre.append(one_hot) #store this list containing 1's and␣
 ↪0's representing genre

    embedding = text_to_embedding(row['name']) #get embedding of name of book
    embeddings.append(embedding) #store embedding

#add new embeddings and encoding to dataframe
new_df['genre_one_hot'] = one_hot_encodings_genre
new_df['name_embeddings'] = embeddings
```

```
/tmp/ipykernel_96834/3515638456.py:17: SettingWithCopyWarning:
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_guide/indexing.html#returning-a-view-versus-a-copy
  new_df['genre_one_hot'] = one_hot_encodings_genre
/tmp/ipykernel_96834/3515638456.py:18: SettingWithCopyWarning:
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_guide/indexing.html#returning-a-view-versus-a-copy
  new_df['name_embeddings'] = embeddings
```

[54]:
```
new_df.head() #display dataframe to train
```

[54]:

	name	genre_urls	genre_one_hot	name_embeddings
0	The Hunger Games (The Hunger Games, #1)	/genres/young-adult|/genres/science-fiction|/g…	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, …	[[tensor(-0.2898), tensor(-0.6596), tensor(-0…
1	Harry Potter and the Order of the Phoenix (Har…	/genres/fantasy|/genres/young-adult|/genres/fi…	[1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, …	[[tensor(-0.6311), tensor(-0.4915), tensor(-0…
2	Twilight (Twilight, #1)	/genres/young-adult|/genres/fantasy|/genres/ro…	[1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, …	[[tensor(-0.4366), tensor(-0.5661), tensor(-0…
3	To Kill a Mockingbird	/genres/classics|/genres/fiction|/genres/histo…	[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, …	[[tensor(-0.1206), tensor(0.0739), tensor(-0.5…
4	Pride and Prejudice	/genres/classics|/genres/fiction|/genres/roman…	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, …	[[tensor(-0.3545), tensor(-0.0832), tensor(-1…

[55]:
```
'''
Preparing data for logistic regression scikit-learn training.
```

```
'''

bert_embeddings_tensor = torch.stack(new_df['name_embeddings'].to_list())  #␣
 ↪Convert list of tensors to a single tensor
one_hot_encodings_list = new_df['genre_one_hot'].to_list()  # This remains a␣
 ↪list of one-hot vectors

bert_embeddings_tensor = bert_embeddings_tensor.squeeze(1)  #(5938, 768),␣
 ↪removing extra dimension

bert_embeddings = bert_embeddings_tensor.numpy() #convert tensor to numpy for␣
 ↪scikit learn
one_hot_encodings = np.array(one_hot_encodings_list)  # (5938, 537)

print(one_hot_encodings.shape)
print(one_hot_encodings)

print(bert_embeddings.shape)
print(bert_embeddings)
```

```
(5938, 537)
[[1 1 1 … 0 0 0]
 [1 0 0 … 0 0 0]
 [1 0 0 … 0 0 0]
 …
 [1 0 0 … 0 0 0]
 [0 0 0 … 0 0 0]
 [0 0 0 … 0 0 0]]
(5938, 768)
[[-0.2897721  -0.6596172  -0.1429491  … -0.14120401  0.10775147
  -0.03917341]
 [-0.6310893  -0.49147648 -0.44651166 …  0.13486508  0.2583655
   0.5564113 ]
 [-0.4366131  -0.56608146 -0.22119586 … -0.18344577  0.29364842
   0.03336031]
 …
 [-0.32230896  0.1102947  -0.17088711 … -0.11993182  0.00252702
   0.21543483]
 [-0.2515789   0.25974703  0.07736662 … -0.1965177   0.28938103
   0.80860037]
 [-0.7774892  -1.118674   -0.41348854 …  0.28654087  0.13396198
  -0.09689084]]
```

[56]:
```
'''
Train model to map vector embeddings to array of 1's and 0's representing an␣
 ↪index of unique_genres.
'''
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.multioutput import MultiOutputClassifier

log_reg_clf = LogisticRegression(max_iter=1000)  # initilise logistic regression

multi_label_clf = MultiOutputClassifier(log_reg_clf) # get ready for multi␣
 ↪label classification as can be several genres

multi_label_clf.fit(bert_embeddings, one_hot_encodings) #train the model
```

[56]:
```
MultiOutputClassifier(estimator=LogisticRegression(max_iter=1000))
```

[57]:
```
subset_df[subset_df.genre_urls.isnull()] #look at rows with genre_url missing
```

[57]:

	rating	isbn	author_url
953	4.56	1477276068	https://www.goodreads.com/author/show/6621980…
1515	4.56	-	https://www.goodreads.com/author/show/394525.T…
1693	4.21	-	https://www.goodreads.com/author/show/3110785._
1752	3.85	140921818X	https://www.goodreads.com/author/show/7337562…
1942	4.66	0992382009	https://www.goodreads.com/author/show/7574275…
2034	4.92	147930414X	https://www.goodreads.com/author/show/6467808…
2067	4.16	0804844399	https://www.goodreads.com/author/show/6894841…
2145	5.00	1300589469	https://www.goodreads.com/author/show/6906561…
2170	4.65	-	https://www.goodreads.com/author/show/6565853…
2190	4.33	0987434853	https://www.goodreads.com/author/show/4443219…
2476	3.53	-	https://www.goodreads.com/author/show/5139623…
2734	4.08	0192833987	https://www.goodreads.com/author/show/128382.L…
2865	4.93	9789898541	https://www.goodreads.com/author/show/7458878…
2868	4.94	-	https://www.goodreads.com/author/show/8127793…
2869	4.61	-	https://www.goodreads.com/author/show/8182217…
2877	3.53	-	https://www.goodreads.com/author/show/8193887…
2903	5.00	0983002282	https://www.goodreads.com/author/show/6589034…
2909	5.00	0983002215	https://www.goodreads.com/author/show/6589034…
2920	4.93	1453634819	https://www.goodreads.com/author/show/4808225…
2928	4.73	0983002274	https://www.goodreads.com/author/show/6589034…
3225	4.00	-	https://www.goodreads.com/author/show/6520851…
3229	4.40	1625108737	https://www.goodreads.com/author/show/6978127…
3238	4.35	-	https://www.goodreads.com/author/show/7243654…
3258	4.63	-	https://www.goodreads.com/author/show/7058502…
3289	4.61	-	https://www.goodreads.com/author/show/7243654…
3688	4.53	-	https://www.goodreads.com/author/show/6440482…
3786	4.00	-	https://www.goodreads.com/author/show/7843984…
3803	3.98	1608622835	https://www.goodreads.com/author/show/4875841…
3868	4.43	1606191934	https://www.goodreads.com/author/show/2941486…
4383	4.29	-	https://www.goodreads.com/author/show/5077755…
4384	4.06	-	https://www.goodreads.com/author/show/5831594…

4387	4.36	1494884631	https://www.goodreads.com/author/show/7754245…
4389	4.29	-	https://www.goodreads.com/author/show/7801640…
4405	4.88	-	https://www.goodreads.com/author/show/8433703…
4410	4.37	-	https://www.goodreads.com/author/show/7380017…
4472	4.79	-	https://www.goodreads.com/author/show/6585024…
4473	5.00	-	https://www.goodreads.com/author/show/6896621…
4524	4.46	-	https://www.goodreads.com/author/show/4808225…
4763	4.56	-	https://www.goodreads.com/author/show/4087195…
5411	4.25	1450556566	https://www.goodreads.com/author/show/3335237…
5429	4.45	-	https://www.goodreads.com/author/show/7373870…
5433	4.60	-	https://www.goodreads.com/author/show/8187644…
5435	4.48	-	https://www.goodreads.com/author/show/5132169…
5443	4.15	0982816308	https://www.goodreads.com/author/show/4524500…
5458	4.25	1311059164	https://www.goodreads.com/author/show/8507331…
5479	4.66	-	https://www.goodreads.com/author/show/7456408…
5480	4.83	1495416739	https://www.goodreads.com/author/show/7399883…
5494	3.81	-	https://www.goodreads.com/author/show/5077755…
5498	4.80	1933455098	https://www.goodreads.com/author/show/5834131…
5500	4.71	-	https://www.goodreads.com/author/show/5895074…
5516	4.62	0974056049	https://www.goodreads.com/author/show/503388.B…
5532	4.86	1477504540	https://www.goodreads.com/author/show/5989528…
5560	4.00	-	https://www.goodreads.com/author/show/7518486._
5565	4.50	-	https://www.goodreads.com/author/show/8262567…
5584	4.75	1481959824	https://www.goodreads.com/author/show/5100743…
5618	4.44	1495907791	https://www.goodreads.com/author/show/7399883…
5692	5.00	-	https://www.goodreads.com/author/show/5989528…
5717	4.71	-	https://www.goodreads.com/author/show/5838022…
5729	4.83	-	https://www.goodreads.com/author/show/7058502…
5778	4.63	-	https://www.goodreads.com/author/show/4808225…

	year	genre_urls	name
953	2012.0	NaN	Crossing the Seas
1515	2013.0	NaN	Crashing Down to Earth
1693	2009.0	NaN	Ð¡Ð»Ñ Ð½Ñ Ðµ Ð½ÐµÐ´Ð¾Ñ ÐµÐ³Ð°ÐµÐ½Ð¾
1752	13.0	NaN	The Day Jesus Rode Into Croydon
1942	2014.0	NaN	Letters from your soul
2034	2012.0	NaN	Happy Halloween
2067	2012.0	NaN	A Capitalist in North Korea
2145	2012.0	NaN	A Book About Absolutely Nothing.
2170	2012.0	NaN	Curse of The Salute
2190	2013.0	NaN	Life Song
2476	2013.0	NaN	Miscellaneous Stuff & Stuff
2734	1869.0	NaN	War and Peace
2865	2012.0	NaN	LaÃ§os fortes e decisÃµes difÃceis
2868	2014.0	NaN	Nik Nassa & the Mark of Destiny
2869	2014.0	NaN	My Death Experiences - A Preacherâ s 18 Apoca…
2877	2014.0	NaN	Ø§Ù ØµÙ Øª Ø±Ù Ø

```
2903  2012.0        NaN              Obscured Darkness (Family Secrets #2)
2909  2011.0        NaN                                    Family Secrets
2920  2010.0        NaN                                The Years Distilled
2928  2012.0        NaN                                             Mortal
3225  2012.0        NaN  Take a Deep Breath - 21 Top Tips for Relaxed, …
3229  2013.0        NaN                             The Mystery of Revenge
3238  2013.0        NaN                           The Real Book of the Dead
3258  2013.0        NaN                         The Keeper (The Keeper, #3)
3289  2013.0        NaN                               Messages From Heaven
3688  2012.0        NaN                           The Oaks (Royal Oaks, #1)
3786  2014.0        NaN                                 The Eagle's Secret
3803  2011.0        NaN  Hillbilly Tales from the Smoky Mountains - And…
3868  2012.0        NaN  The Case of the Cosmological Killer (Displaced…
4383  2011.0        NaN                                    I am an Island
4384  2012.0        NaN                               Undying (Undying, #1)
4387  2014.0        NaN  It's Not What You Think (It's Not What You Thi…
4389  2013.0        NaN  We Got Zombies On The Lawn Again, Ma (Ax Hande…
4405  2014.0        NaN                                       My Two Cents
4410  2013.0        NaN                                  Hard Up, Ardon #1
4472  2013.0        NaN                         Tilt (Dreams of Chaos #2)
4473  2012.0        NaN                                Patience's Love
4524  2011.0        NaN                                             Fresh
4763  2010.0        NaN            In the Name of Revenge (Ivanovich, #1)
5411  2010.0        NaN                   A Terrace On The Tower Of Babel
5429  2013.0        NaN  It Comes Natural - Understanding Natural and I…
5433  2014.0        NaN                   New Zigon - The Founder's Curse
5435  2014.0        NaN                                Bound Anthology
5443  2010.0        NaN                Secrets of Jewish Wealth Revealed!
5458  2014.0        NaN                                          Liliana
5479  2013.0        NaN                                Churning Waters
5480  2014.0        NaN                             This Land of Streams
5494  2008.0        NaN             Huey Lambert's Walking Nuclear Circus
5498  2012.0        NaN                                 Syrian Folktales
5500  2011.0        NaN                              Fantacia (Voxian, #1)
5516  2006.0        NaN                   The Peace of the Spirit Within
5532  2013.0        NaN                            Call Of The Lost Ages
5560  2013.0        NaN                                  Ø§Ù
Ù Ø§Ù  Ù Ø¯Ù
Ø§Ø¡
5565  2014.0        NaN                          Five Years - The Meeting
5584  2013.0        NaN                                    Why Not-World
5618  2014.0        NaN                       The Afternoon When She Died
5692  2012.0        NaN  Abstraction In Theory - Laws Of Physical Trans…
5717  2012.0        NaN                                American Amaranth
5729  2014.0        NaN                       The Keeper (The Keeper, #5)
5778  2010.0        NaN                                       (Un) Spoken
```

[58]:
```
'''
Make name vector embeddings of all missing genre values, to then use to predict␣
 ↪the genre/s.
'''

missing_genre_name_list = subset_df[subset_df['genre_urls'].isnull()]['name'].
 ↪tolist() #get all missing genre_url names as a list
missing_genre_embeddings=[] #empty list to store name embeddings

for name in missing_genre_name_list:
    embedding=text_to_embedding(name) #make embedding of name
    missing_genre_embeddings.append(embedding) #add to store

# (same steps as when preparing embedding data for training)
missing_bert_embeddings_tensor = torch.stack(missing_genre_embeddings)  #␣
 ↪convert list of tensor embeddings to a single tensor
missing_bert_embeddings_tensor = missing_bert_embeddings_tensor.squeeze(1) ␣
 ↪#removing extra dimension
missing_bert_embeddings = missing_bert_embeddings_tensor.numpy() #convert␣
 ↪tensor to numpy for scikit learn
print(missing_bert_embeddings)
```

```
[[-0.31766725  0.13855387 -0.22493249 … -0.09315871  0.0146475
   0.05123395]
 [-0.667365    0.15916485 -0.19500211 … -0.4225353   0.20716761
  -0.22099356]
 [-0.39068276  0.4437609   0.12601301 … -0.09302449  0.54468113
   0.5817003 ]
 …
 [-0.4975458   0.2882989  -0.45175612 … -0.39606276  0.39692545
   0.11483536]
 [-0.5846143  -0.47402704 -0.1233344  … -0.07081988  0.3550252
   0.10488999]
 [-0.45655882  0.06432083 -0.46465328 … -0.26800245  0.10546472
   0.42697194]]
```

[59]:
```
'''
Predict the genres with trained model.
'''

predicted_genres=multi_label_clf.predict(missing_bert_embeddings) #using name␣
 ↪(embedding) make prediction with model for genres, outputs 1's and 0's
print(predicted_genres)
```

```
[[0 0 0 … 0 0 0]
 [1 0 0 … 0 0 0]
 [1 0 0 … 0 0 0]
```

```
…
 [0 0 0 … 0 0 0]
 [1 0 0 … 0 0 0]
 [0 0 0 … 0 0 0]]
```

[60]:
```
'''
Have got predictions in 1's and 0's for genre representing the index of a␣
 ↪specific genre from previously made "unique_genres".
Now must match these with the string so it is ready to insert into dataframe
'''
complete_predicted_genres=[] #store for strings of combined genres
for predicted_genre in predicted_genres: #for every encoding
    selected_genres = [genre for idx, genre in enumerate(unique_genres) if␣
 ↪predicted_genre[idx] == 1] #make list of the combination of genres
    combined_genre= '|'.join(selected_genres) #combine list in the correct␣
 ↪format using |
    complete_predicted_genres.append(combined_genre) #append to total list␣
 ↪representing each genres of missing values

print(complete_predicted_genres)
```

```
['/genres/classics|/genres/spirituality|/genres/travel', '/genres/young-
adult|/genres/classics|/genres/juvenile|/genres/poetry', '/genres/young-adult|/g
enres/fantasy|/genres/romance|/genres/fiction|/genres/russia|/genres/media-tie-
in|/genres/psychology|/genres/nobel-prize',
'/genres/fiction|/genres/classics|/genres/novels|/genres/philosophy',
'/genres/fiction|/genres/cultural|/genres/contemporary', '/genres/young-
adult|/genres/romance|/genres/fiction|/genres/contemporary',
'/genres/literature|/genres/humor|/genres/cultural|/genres/non-
fiction|/genres/autobiography|/genres/memoir|/genres/biography', '/genres/young-
adult|/genres/fiction|/genres/classics|/genres/academic|/genres/contemporary|/ge
nres/plays', '/genres/fiction|/genres/academic|/genres/school|/genres/novels',
'/genres/book-
club|/genres/fiction|/genres/classics|/genres/literature|/genres/novels', '/genr
es/fantasy|/genres/fiction|/genres/childrens|/genres/horror|/genres/poetry|/genr
es/writing|/genres/essays',
'/genres/fiction|/genres/classics|/genres/literature|/genres/european-
literature', '/genres/fiction|/genres/portuguese-literature', '/genres/fiction',
'/genres/fantasy|/genres/fiction|/genres/paranormal|/genres/religion|/genres/sho
rt-stories|/genres/spirituality|/genres/islam|/genres/ghosts',
'/genres/supernatural', '/genres/fantasy|/genres/romance|/genres/fiction|/genres
/paranormal|/genres/supernatural', '/genres/fiction', '/genres/literature',
'/genres/fantasy|/genres/romance|/genres/fiction', '/genres/book-club|/genres/fi
ction|/genres/inspirational|/genres/humor|/genres/funny|/genres/contemporary|/ge
nres/psychology|/genres/self-help|/genres/personal-development', '/genres/book-
club|/genres/literature|/genres/american|/genres/non-fiction',
'/genres/fantasy|/genres/fiction|/genres/horror', '/genres/young-
```

```
adult|/genres/fantasy|/genres/romance|/genres/fiction', '/genres/fiction', '/gen
res/fantasy|/genres/adventure|/genres/fiction|/genres/childrens|/genres/science-
fiction-fantasy|/genres/adult|/genres/juvenile|/genres/fairies|/genres/fae',
'/genres/fantasy|/genres/fiction|/genres/historical-fiction|/genres/novels',
'/genres/fiction|/genres/classics|/genres/literature|/genres/classic-
literature',
'/genres/fantasy|/genres/adventure|/genres/fiction|/genres/mystery',
'/genres/fiction|/genres/literature|/genres/novels|/genres/european-
literature|/genres/philosophy|/genres/comedy|/genres/non-fiction',
'/genres/young-
adult|/genres/fantasy|/genres/romance|/genres/fiction|/genres/paranormal-
romance|/genres/contemporary', '/genres/young-
adult|/genres/romance|/genres/fiction|/genres/contemporary', '/genres/young-
adult|/genres/fantasy|/genres/fiction|/genres/paranormal|/genres/supernatural',
'/genres/book-club|/genres/fiction|/genres/contemporary',
'/genres/fantasy|/genres/fiction',
'/genres/fantasy|/genres/romance|/genres/fiction|/genres/paranormal',
'/genres/fiction|/genres/historical-fiction', '/genres/fiction', '/genres/advent
ure|/genres/fiction|/genres/novels|/genres/cultural|/genres/thriller|/genres/sus
pense', '/genres/romance|/genres/fiction|/genres/classics|/genres/historical-
fiction|/genres/literature|/genres/novels|/genres/european-
literature|/genres/german-literature|/genres/italy',
'/genres/magic|/genres/adult|/genres/short-stories|/genres/non-
fiction|/genres/psychology|/genres/self-help', '/genres/fiction|/genres/magic',
'/genres/young-adult|/genres/fantasy|/genres/romance|/genres/fiction',
'/genres/childrens|/genres/classics|/genres/inspirational|/genres/humor',
'/genres/fiction|/genres/paranormal',
'/genres/romance|/genres/fiction|/genres/historical-
fiction|/genres/literature|/genres/novels|/genres/american', '/genres/book-
club|/genres/fiction|/genres/classics|/genres/novels',
'/genres/fantasy|/genres/fiction|/genres/humor', '/genres/fantasy|/genres/litera
ture|/genres/philosophy|/genres/cultural|/genres/mystery', '/genres/young-
adult|/genres/fantasy|/genres/romance|/genres/fiction', '/genres/book-
club|/genres/fiction|/genres/classics|/genres/historical-
fiction|/genres/literature|/genres/novels|/genres/religion',
'/genres/fantasy|/genres/adventure|/genres/fiction|/genres/historical-
fiction|/genres/literature', '/genres/fiction|/genres/historical-
fiction|/genres/academic|/genres/novels|/genres/modern', '/genres/book-
club|/genres/fiction|/genres/literature|/genres/novels|/genres/adult-
fiction|/genres/realistic-fiction', '/genres/realistic-fiction|/genres/non-
fiction', '/genres/fiction|/genres/adult|/genres/contemporary|/genres/womens-
fiction|/genres/chick-lit',
'/genres/classics|/genres/literature|/genres/european-
literature|/genres/philosophy|/genres/politics|/genres/history|/genres/non-
fiction|/genres/science|/genres/20th-century|/genres/theory',
'/genres/fantasy|/genres/book-club|/genres/drama', '/genres/young-
adult|/genres/fantasy|/genres/romance|/genres/fiction',
'/genres/fiction|/genres/classics|/genres/academic']
```

[61]:
```
'''
Replace null genre values with predicted and completed string format genres.
'''

null_indexes = subset_df[subset_df['genre_urls'].isnull()].index #grab all the␣
 ↪null indexes representing a value from complete_predicted_genres
subset_df.loc[null_indexes, 'genre_urls'] = complete_predicted_genres #add␣
 ↪these genres to dataframe
```

[62]:
```
missing_value_count(subset_df) #check missing counts should be 0 for everything
```

[62]:
```
{'rating': np.int64(0),
 'isbn': np.int64(0),
 'author_url': np.int64(0),
 'year': np.int64(0),
 'genre_urls': np.int64(0),
 'name': np.int64(0)}
```

[63]:
```
subset_df.head() #display dataframe, no missing values and clean
```

[63]:

	rating	isbn	author_url	\
0	4.40	0439023483	https://www.goodreads.com/author/show/153394.S…	
1	4.41	0439358078	https://www.goodreads.com/author/show/1077326…	
2	3.56	0316015849	https://www.goodreads.com/author/show/941441.S…	
3	4.23	0061120081	https://www.goodreads.com/author/show/1825.Har…	
4	4.23	0679783261	https://www.goodreads.com/author/show/1265.Jan…	

	year	genre_urls	\
0	2008.0	/genres/young-adult|/genres/science-fiction|/g…	
1	2003.0	/genres/fantasy|/genres/young-adult|/genres/fi…	
2	2005.0	/genres/young-adult|/genres/fantasy|/genres/ro…	
3	1960.0	/genres/classics|/genres/fiction|/genres/histo…	
4	1813.0	/genres/classics|/genres/fiction|/genres/roman…	

	name
0	The Hunger Games (The Hunger Games, #1)
1	Harry Potter and the Order of the Phoenix (Har…
2	Twilight (Twilight, #1)
3	To Kill a Mockingbird
4	Pride and Prejudice

1.4 Exercise 4: Shape the data

- Parse the `author_url` to create new column named `author`
- Sort the data by putting higher rates go first. If there are overlapping rates, try to put earlier years go first.
- **(Stretch Goal)** Examine how many books were published at each year and find lowest,

highest rate of each year.

[64]:
```
'''
Making new column "author" without the url.
'''

#from lab-03-part-02
def get_author(url):
    name = url.split('/')[-1].split('.')[1:][0]
    return name

subset_df['author'] = subset_df.author_url.map(get_author)
subset_df.head()
```

[64]:
```
   rating        isbn                                          author_url  \
0    4.40  0439023483  https://www.goodreads.com/author/show/153394.S…
1    4.41  0439358078  https://www.goodreads.com/author/show/1077326…
2    3.56  0316015849  https://www.goodreads.com/author/show/941441.S…
3    4.23  0061120081  https://www.goodreads.com/author/show/1825.Har…
4    4.23  0679783261  https://www.goodreads.com/author/show/1265.Jan…

     year                                          genre_urls  \
0  2008.0  /genres/young-adult|/genres/science-fiction|/g…
1  2003.0  /genres/fantasy|/genres/young-adult|/genres/fi…
2  2005.0  /genres/young-adult|/genres/fantasy|/genres/ro…
3  1960.0  /genres/classics|/genres/fiction|/genres/histo…
4  1813.0  /genres/classics|/genres/fiction|/genres/roman…

                                                 name           author
0           The Hunger Games (The Hunger Games, #1)  Suzanne_Collins
1  Harry Potter and the Order of the Phoenix (Har…      J_K_Rowling
2                          Twilight (Twilight, #1)  Stephenie_Meyer
3                            To Kill a Mockingbird       Harper_Lee
4                              Pride and Prejudice      Jane_Austen
```

[68]:
```
#rating descending, year ascending therefore False, True
sorted_df = subset_df.sort_values(by=['rating', 'year'], ascending=[False,␣
 ↪True])

sorted_df.head()
```

[68]:
```
      rating        isbn                                          author_url  \
2909     5.0  0983002215  https://www.goodreads.com/author/show/6589034…
2145     5.0  1300589469  https://www.goodreads.com/author/show/6906561…
2903     5.0  0983002282  https://www.goodreads.com/author/show/6589034…
4473     5.0           -  https://www.goodreads.com/author/show/6896621…
5692     5.0           -  https://www.goodreads.com/author/show/5989528…
```

```
        year                                          genre_urls  \
2909  2011.0                                    /genres/fiction
2145  2012.0  /genres/young-adult|/genres/fiction|/genres/cl…
2903  2012.0  /genres/fantasy|/genres/romance|/genres/fictio…
4473  2012.0         /genres/fiction|/genres/historical-fiction
5692  2012.0  /genres/classics|/genres/literature|/genres/eu…

                                                     name            author
2909                                      Family Secrets    Rebekah_McClew
2145                  A Book About Absolutely Nothing.         I_M_Nobody
2903              Obscured Darkness (Family Secrets #2)    Rebekah_McClew
4473                                   Patience's Love       Ronda_Paige
5692  Abstraction In Theory - Laws Of Physical Trans…  Subhajit_Ganguly
```

[71]:
```
#source: https://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.
 ↪core.groupby.DataFrameGroupBy.agg.html
#source ('size', 'max', 'min'): https://stackoverflow.com/questions/44952061/
 ↪getting-the-size-of-a-groupby-operation-in-pandas

#make table grouped by year of count of total rows for that year, the highest␣
 ↪rating and the lowest rating.
result = sorted_df.groupby('year').agg(count=('year', 'size'),\
                                       highest_rating=('rating', 'max'),␣
 ↪lowest_rating=('rating', 'min'))
print(result)
```

```
         count  highest_rating  lowest_rating
year
-1500.0      1            3.60           3.60
-800.0       2            4.01           3.68
-560.0       1            4.03           4.03
-512.0       1            3.92           3.92
-500.0       1            4.06           4.06
…          …              …              …
 2011.0    374            5.00           2.00
 2012.0    355            5.00           3.15
 2013.0    276            4.93           2.90
 2014.0     88            5.00           3.31
 2018.0      1            4.56           4.56

[294 rows x 3 columns]
```

1.5 Exercise 5: Saving the results

- Save the cleaned dataframe as 'hw-03-cleaned.csv' in data folder

[73]:
```
sorted_df.to_csv('hw-03-cleaned.csv', index=False) #no index as index values in␣
↪sorted table mean nothing
```

1.6 Exercise 6: Investigate the relationship between the number of reviews and the average rating for books in the dataset cleaned-goodreads.csv procided.

- Calculate the correlation coefficient. Give me a short definition of this coefficient
- Create a scatter plot showing the relationship between these two features.
- Based on the plot and the correlation, provide a brief interpretation of the relationship.

1.6.1 Python Tools: Use pandas and numpy for correlation, and matplotlib or seaborn for the scatter plot.

[74]:
```
cleaned_goodreads=pd.read_csv('data/cleaned-goodreads.csv') #open csv
cleaned_goodreads.head() #display csv
```

[74]:

	rating	review_count	isbn	booktype	\
0	4.40	136455	0439023483	good_reads:book	
1	4.41	16648	0439358078	good_reads:book	
2	3.56	85746	0316015849	good_reads:book	
3	4.23	47906	0061120081	good_reads:book	
4	4.23	34772	0679783261	good_reads:book	

	author_url	year	\
0	https://www.goodreads.com/author/show/153394.S…	2008	
1	https://www.goodreads.com/author/show/1077326…	2003	
2	https://www.goodreads.com/author/show/941441.S…	2005	
3	https://www.goodreads.com/author/show/1825.Har…	1960	
4	https://www.goodreads.com/author/show/1265.Jan…	1813	

	dir	rating_count	\
0	dir01/2767052-the-hunger-games.html	2958974	
1	dir01/2.Harry_Potter_and_the_Order_of_the_Phoe…	1284478	
2	dir01/41865.Twilight.html	2579564	
3	dir01/2657.To_Kill_a_Mockingbird.html	2078123	
4	dir01/1885.Pride_and_Prejudice.html	1388992	

	name	author	\
0	The Hunger Games (The Hunger Games, #1)	Suzanne_Collins	
1	Harry Potter and the Order of the Phoenix (Har…	J_K_Rowling	
2	Twilight (Twilight, #1)	Stephenie_Meyer	
3	To Kill a Mockingbird	Harper_Lee	
4	Pride and Prejudice	Jane_Austen	

	genres
0	young-adult|science-fiction|dystopia|fantasy|s…
1	fantasy|young-adult|fiction|fantasy|magic|chil…

```
2  young-adult|fantasy|romance|paranormal|vampire…
3  classics|fiction|historical-fiction|academic|s…
4  classics|fiction|romance|historical-fiction|li…
```

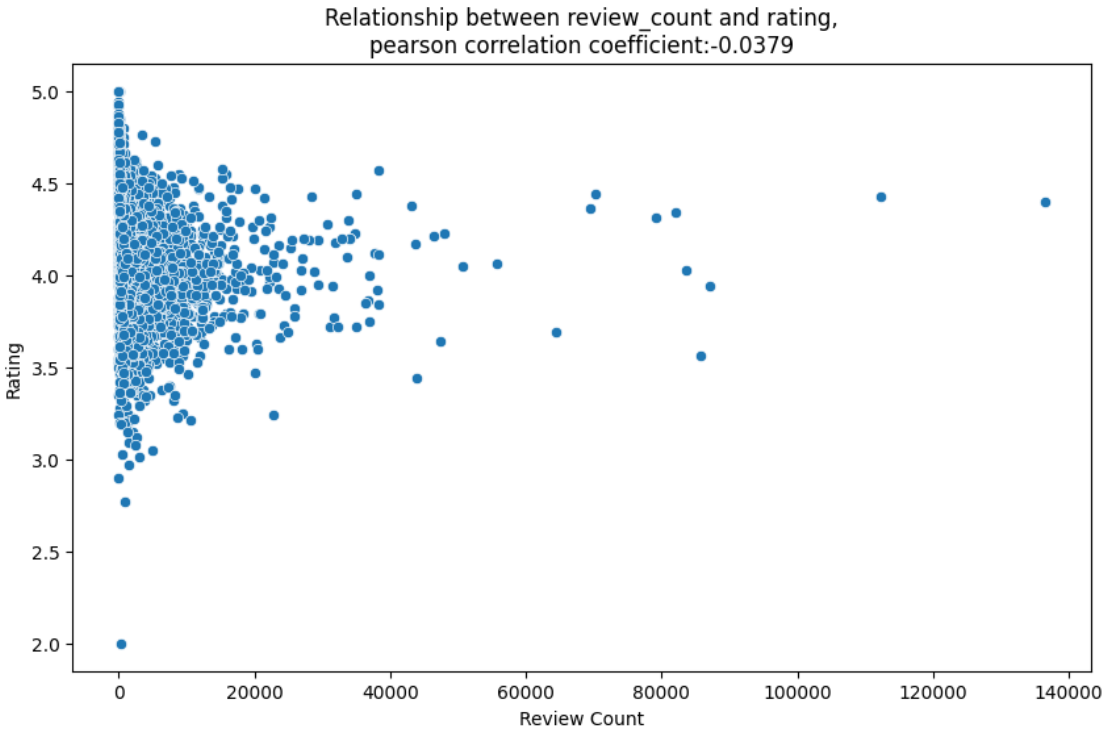
[77]:
```
'''
Make table of review_count and rating (average rating)
'''

rating_review_df=cleaned_goodreads[['rating', 'review_count']] #make small␣
 ↪table with desired variables
rating_review_df.head()
```

[77]:

	rating	review_count
0	4.40	136455
1	4.41	16648
2	3.56	85746
3	4.23	47906
4	4.23	34772

[91]:
```
import seaborn as sns
import matplotlib.pyplot as plt

#source: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html

correlation = rating_review_df.corr(method='pearson') #method is pearson␣
 ↪correlation coefficient
print(correlation)
correlation_value = correlation.loc['rating', 'review_count'] #since .corr␣
 ↪gives matrix, grab value comparing the two different
print(f"\nCorrelation coefficient between rating and review_count:␣
 ↪{correlation_value}") #print this value

plt.figure(figsize=(10, 6)) #prepare figure size
sns.scatterplot(x='review_count', y='rating', data=rating_review_df) #create␣
 ↪scatter plot with seaborn
plt.xlabel('Review Count') #label x axis
plt.ylabel('Rating') #label y axis (rating goes on y axis because we are␣
 ↪measuring how this changes with respect to review count)
plt.title(f'Relationship between review_count and rating,\npearson correlation␣
 ↪coefficient:{correlation_value:.4f}') #make title with correlation (to 4.s.f)
plt.show() #present graph
```

```
                rating  review_count
rating        1.000000     -0.037896
review_count -0.037896      1.000000

Correlation coefficient between rating and review_count: -0.037896372578744196
```



1.6.2 Correlation coefficient explanation

A correlation of -0.0379 shows a weak negative correlation between review count and rating. This means that as review count increases, rating decreases; however, since it is close to 0 this is not a strong relationship and there is almost no linear relationship.

1.7 Exercise 7: Calculate the following descriptive statistics for the numerical features (e.g., number of reviews, average rating, etc.):

- Mean
- Median
- Standard Deviation
- Range
- Create a histogram or box plot for at least one of the numerical features, highlighting any skewness or outliers.

1.7.1 Python Tools: Use pandas for data manipulation and matplotlib or seaborn for visualization.

[93]:
```
cleaned_goodreads.head()
```

[93]:

	rating	review_count	isbn	booktype	\
0	4.40	136455	0439023483	good_reads:book	
1	4.41	16648	0439358078	good_reads:book	

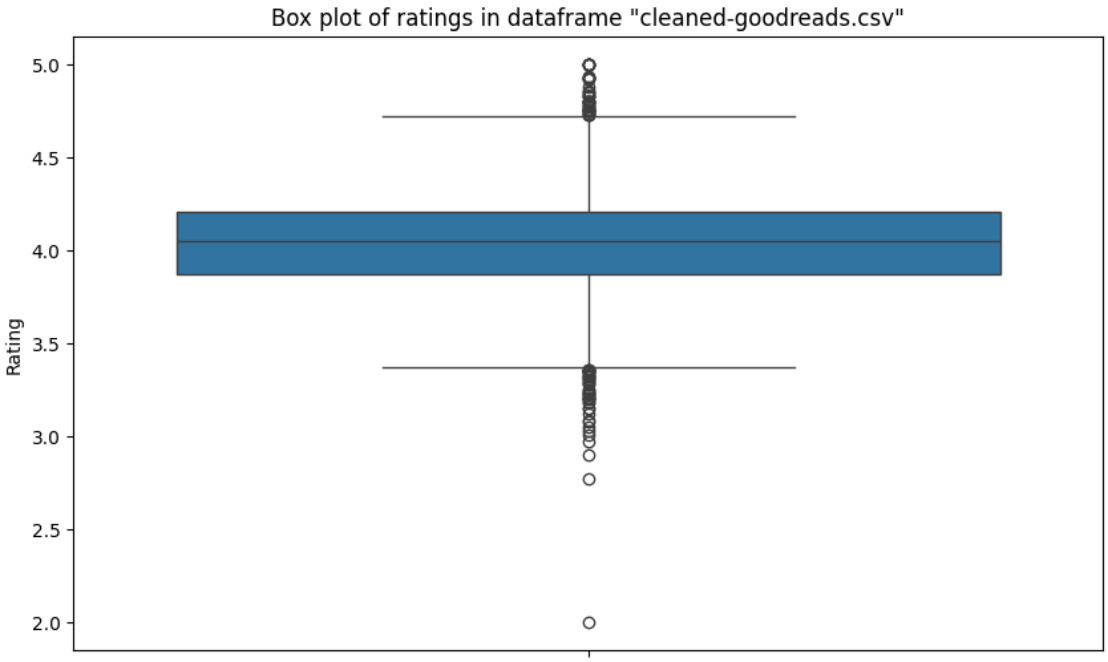
```
2    3.56         85746  0316015849  good_reads:book
3    4.23         47906  0061120081  good_reads:book
4    4.23         34772  0679783261  good_reads:book

                                          author_url  year  \
0  https://www.goodreads.com/author/show/153394.S…  2008
1  https://www.goodreads.com/author/show/1077326…  2003
2  https://www.goodreads.com/author/show/941441.S…  2005
3  https://www.goodreads.com/author/show/1825.Har…  1960
4  https://www.goodreads.com/author/show/1265.Jan…  1813

                                             dir  rating_count  \
0                dir01/2767052-the-hunger-games.html       2958974
1  dir01/2.Harry_Potter_and_the_Order_of_the_Phoe…       1284478
2                          dir01/41865.Twilight.html       2579564
3              dir01/2657.To_Kill_a_Mockingbird.html       2078123
4                dir01/1885.Pride_and_Prejudice.html       1388992

                                            name            author  \
0           The Hunger Games (The Hunger Games, #1)  Suzanne_Collins
1  Harry Potter and the Order of the Phoenix (Har…      J_K_Rowling
2                            Twilight (Twilight, #1)  Stephenie_Meyer
3                              To Kill a Mockingbird       Harper_Lee
4                                Pride and Prejudice      Jane_Austen

                                              genres
0  young-adult|science-fiction|dystopia|fantasy|s…
1  fantasy|young-adult|fiction|fantasy|magic|chil…
2  young-adult|fantasy|romance|paranormal|vampire…
3  classics|fiction|historical-fiction|academic|s…
4  classics|fiction|romance|historical-fiction|li…
```

[94]:
```python
def column_statistics(column_name, df): #make function so don't need to repeat the operations
    column_data = df[column_name] #grab data of that column
    mean_value = column_data.mean() #find mean
    median_value = column_data.median() #find median
    std_dev_value = column_data.std() #find standard deviation
    range_value = column_data.max() - column_data.min() #find range

    #print values (to 2 significant figures)
    print(f"Statistics for column: '{column_name}'")
    print(f"Mean: {mean_value:.2f}")
    print(f"Median: {median_value:.2f}")
    print(f"Standard Deviation: {std_dev_value:.2f}")
    print(f"Range: {range_value:.2f}\n")
```

```
column_statistics('rating', cleaned_goodreads)
column_statistics('review_count', cleaned_goodreads)
# column_statistics('year', cleaned_goodreads) year is not a numerical feature␣
 ↪even though it is a number
column_statistics('rating_count', cleaned_goodreads)
```

```
Statistics for column: 'rating'
Mean: 4.04
Median: 4.05
Standard Deviation: 0.26
Range: 3.00

Statistics for column: 'review_count'
Mean: 2374.33
Median: 936.00
Standard Deviation: 5493.09
Range: 136455.00

Statistics for column: 'rating_count'
Mean: 51183.90
Median: 18072.00
Standard Deviation: 137649.34
Range: 2958969.00
```

[96]:
```
#source: https://seaborn.pydata.org/generated/seaborn.boxplot.html

plt.figure(figsize=(10, 6)) #prepare figure
sns.boxplot(y='rating', data=cleaned_goodreads) #make boxplot (done␣
 ↪automatically)
plt.ylabel('Rating') #label rating yaxis
plt.title('Box plot of ratings in dataframe "cleaned-goodreads.csv"') #title␣
 ↪boxplot
plt.show()
```


Box plot of ratings in dataframe "cleaned-goodreads.csv"
Rating
5.0
4.5
4.0
3.5
3.0
2.5
2.0