Beyond The Bestseller List

Finding Your Next Book Using Data Science

Irene Buck

Computer Science Major

CS 332, Section 400

buckir@oregonstate.edu

Table Of Contents

Introduction 3

Data Gathering 5

Direct Download..... 5

Using an API 7

Data Cleaning..... 10

Goodreads Top 100 dataset.....10

New York Times Nonfiction Bestsellers dataset.....17

Goodreads User Export dataset 22

Unsupervised Learning Using KMeans Clustering.....26

Supervised Learning Using Decision Trees.....33

Decision Tree Visual.....39

Conclusion.....41

References.....43

Introduction

Meet Tara: a stay-at-home mom. She left a busy and challenging career, surrounded by educated coworkers and working with all walks of life. Soon after starting her homemaker life, she found herself restless, lonely, and eager for mental stimulation. When spending time with friends, she found she had less to talk about beyond her children. Meet George: a retired 75-year-old widower living on his own. He has a limited budget for fun and an even more limited social circle. He enjoys his daily walks to stave off the inevitable health decline from aging but worries about his memory declining. Meet Sam: a recent transplant from across the country. They graduated from college 20 years ago and feel eager to learn something new. They have a limited budget and limited time.

What do all these people have in common? They could benefit from regular reading. According to Dr. Paul Wright, M.D. at the Neuroscience Institute, just 20 minutes of reading every day is an intellectual workout that reduces stress, improves sleep quality, boosts memory and cognition function, builds resilience, develops empathy, and even lowers blood pressure. Book clubs also offer valuable social connections and a sense of community. Most importantly, reading a great book is fun!

Finding interesting, well-written books can be a challenge due to the sheer number of books available. So how do you find a great book? This data science project aims to address just that by developing an analytical framework to identify compelling books based on specific criteria and personal preferences. The project focuses on the author as the primary user, leveraging their preference for nonfiction books that have less than 1000 pages and haven't been read previously. Three datasets will be used to find compelling books: the New York Times Nonfiction Bestseller’s List, the Top 100 books of each year rated by Goodreads readers, and the author’s personal Goodreads "Books Read" export.

The goal of this analysis is to answer specific questions: Which books will appear in both the Top 100 Goodreads and the NY Times datasets but not in the Goodreads – User Export dataset, and will distinct patterns emerge? Follow along as we answer these questions and find some excellent books to read!

Data Gathering

Direct Data Download

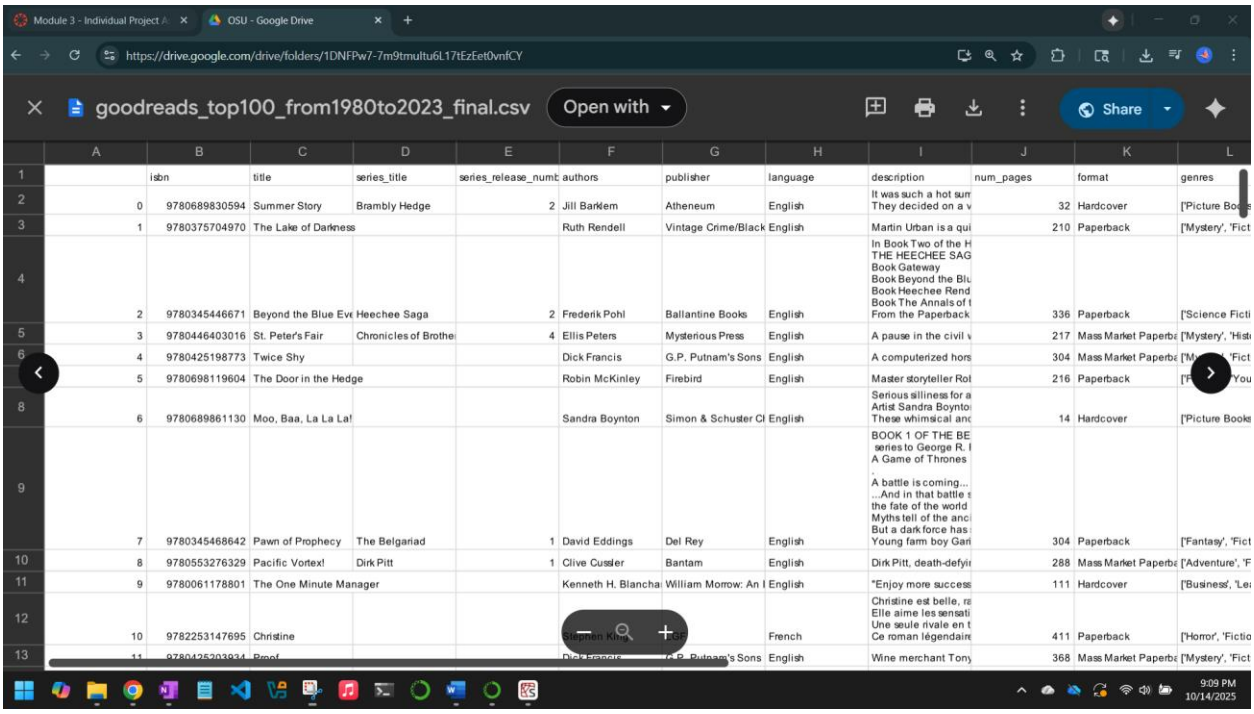
Dataset 1 - Top 100 books from Goodreads, 1980-2023

goodreads

Link to Goodreads Top 100 1980 - 2023 dataset

This data was sourced from kaggle.com and includes the top 100 books each year based on Goodreads reviews and number of votes from 1980 through 2023. This dataset will be sorted down to Nonfiction books and include 9 necessary columns: Title, Authors, Description, Number of Pages, Genres, Publication Date, Rating Score, Number of Ratings, and Number of Reviews.

The data is labelled, and has both quantitative (Number of pages, Rating Score, etc.) and qualitative (Title, Authors, etc.) data. The label is the book title.

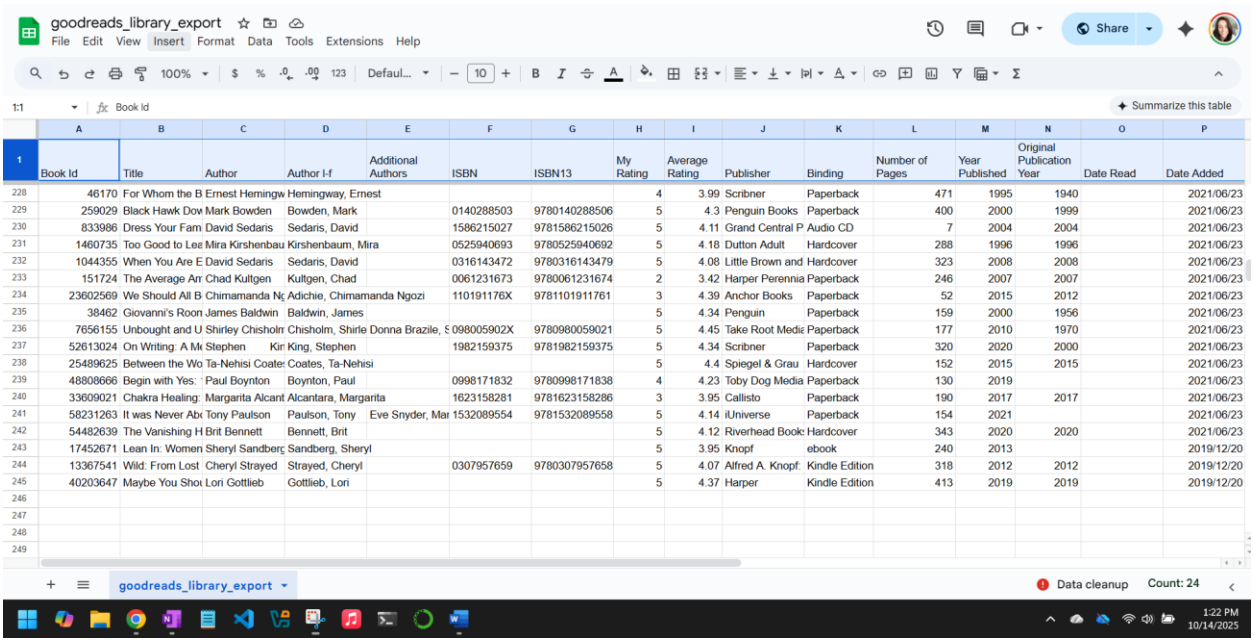


Dataset 2 – Goodreads personal data list of books read

[Link to my Goodreads - Books Read data]

The Goodreads account was accessed from a non-mobile computer to export the book library. The "My Books" tab in the top menu bar was selected, the "Import and export" link under the "Tools" link on the left sidebar was clicked, and the "Export Library" button was clicked. After a brief wait, a link with the date and time of the export appeared, which was used to download the resulting CSV file.

The data includes the title, author, ISBN, the user's rating, the Goodreads average rating, number of pages, year published, and more. The label is the book title. There are over 240 rows and 24 columns of data listing 239 books read.



Book Id	Title	Author	Author l-f	Additional Authors	ISBN	ISBN13	My Rating	Average Rating	Publisher	Binding	Number of Pages	Year Published	Original Publication Year	Date Read	Date Added
46170	For Whom the B	Ernest Hemingw	Hemingway, Ernest				4	3.99	Scribner	Paperback	471	1995	1940		2021/06/23
259029	Black Hawk Dow	Mark Bowden	Bowden, Mark		0140288503	9780140288506	5	4.3	Penguin Books	Paperback	400	2000	1999		2021/06/23
833986	Dress Your Fam	David Sedaris	Sedaris, David		1586215027	9781586215026	5	4.11	Grand Central P	Audio CD	7	2004	2004		2021/06/23
1460735	Too Good to Lea	Mira Kirshenbau	Kirshenbaum, Mira		0525940693	9780525940692	5	4.18	Dutton Adult	Hardcover	288	1996	1996		2021/06/23
1044355	When You Are E	David Sedaris	Sedaris, David		0316143472	9780316143479	5	4.08	Little Brown and	Hardcover	323	2008	2008		2021/06/23
151724	The Average Am	Chad Kultgen	Kultgen, Chad		0061231673	9780061231674	2	3.42	Harper Perennia	Paperback	246	2007	2007		2021/06/23
23602569	We Should All B	Chimamanda Ng	Adichie, Chimamanda Ngozi		110191176X	9781101911761	3	4.39	Anchor Books	Paperback	52	2015	2012		2021/06/23
38462	Giovanni's Roon	James Baldwin	Baldwin, James				5	4.34	Penguin	Paperback	159	2000	1956		2021/06/23
7656155	Unbought and U	Shirley Chisholm	Chisholm, Shirle	Donna Brazile, S	098005902X	9780980059021	5	4.45	Take Root Media	Paperback	177	2010	1970		2021/06/23
52613024	On Writing: A Me	Stephen Kin	King, Stephen		1982159375	9781982159375	5	4.34	Scribner	Paperback	320	2020	2000		2021/06/23
25489625	Between the Wo	Ta-Nehisi Coate	Coates, Ta-Nehisi				5	4.4	Spiegel & Grau	Hardcover	152	2015	2015		2021/06/23
48808666	Begin with Yes:	Paul Boynton	Boynton, Paul		0998171832	9780998171838	4	4.23	Toby Dog Media	Paperback	130	2019			2021/06/23
33609021	Chakra Healing:	Margarita Alcant	Alcantara, Margarita		1623158281	9781623158286	3	3.95	Callisto	Paperback	190	2017	2017		2021/06/23
58231263	It was Never Ab	Tony Paulson	Paulson, Tony	Eve Snyder, Mar	1532089554	9781532089558	5	4.14	iUniverse	Paperback	154	2021			2021/06/23
54482639	The Vanishing H	Brit Bennett	Bennett, Brit				5	4.12	Riverhead Book	Hardcover	343	2020	2020		2021/06/23
17452671	Lean In: Women	Sheryl Sandberg	Sandberg, Sheryl				5	3.95	Knopf	ebook	240	2013			2019/12/20
13367541	Wild: From Lost	Cheryl Strayed	Strayed, Cheryl		0307957659	9780307957658	5	4.07	Alfred A. Knopf	Kindle Edition	318	2012	2012		2019/12/20
40203647	Maybe You Shou	Lori Gottlieb	Gottlieb, Lori				5	4.37	Harper	Kindle Edition	413	2019	2019		2019/12/20

Data Gathering Using an API

Dataset 3 – New York Times Nonfiction Bestsellers 2015 - 2019

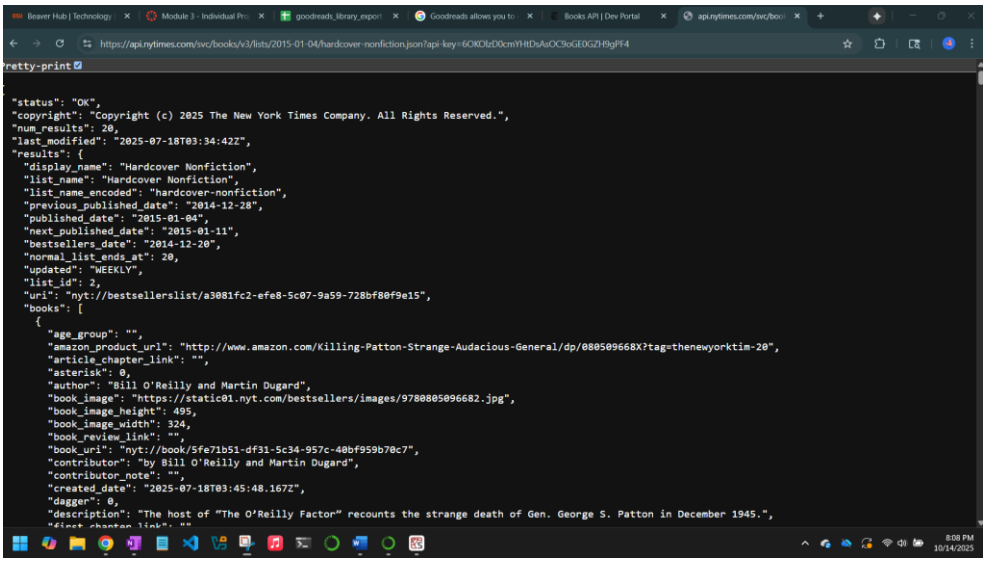
Link to New York Times Hardcover Nonfiction Bestsellers

This data is a list of all Hardcover Nonfiction books on the New York Times Best Sellers lists gathered using the Books API in the NYTimes Developer Network at https://developer.nytimes.com. The Get Started page walks through setting up an account, signing in, registering an app in My Apps, and getting an API key for the app.

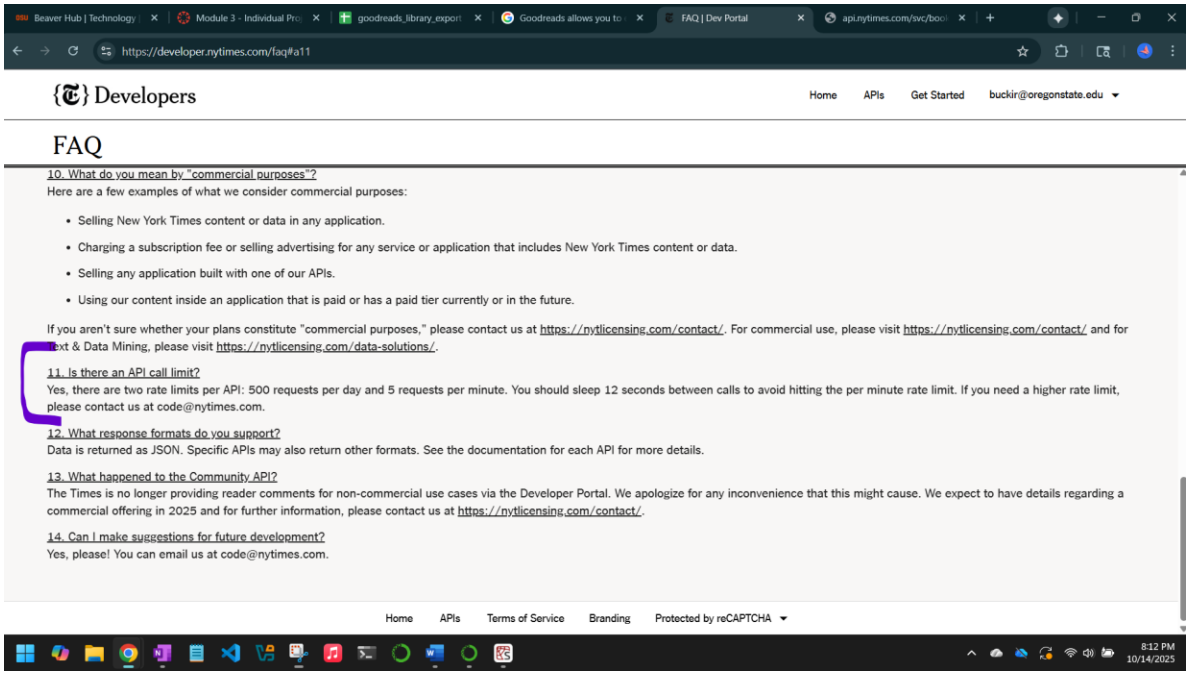
The Books API overview within the NYTimes Developer website shared example calls to the API like

https://api.nytimes.com/svc/books/v3/lists/2025-05-04/hardcover-fiction.json?api-key=yourkey

and schemas to understand how the JSON response is formatted. It also described that these bestseller Hardcover Nonfiction lists are pulled on Sundays every week and published the following Wednesday (10 days later). The date range is January 1, 2015, through December 31, 2019. Below is the response from the API call when dropped in the web address using the first date of 1/4/2015 and Hardcover Nonfiction. It displays how the JSON is formatted:

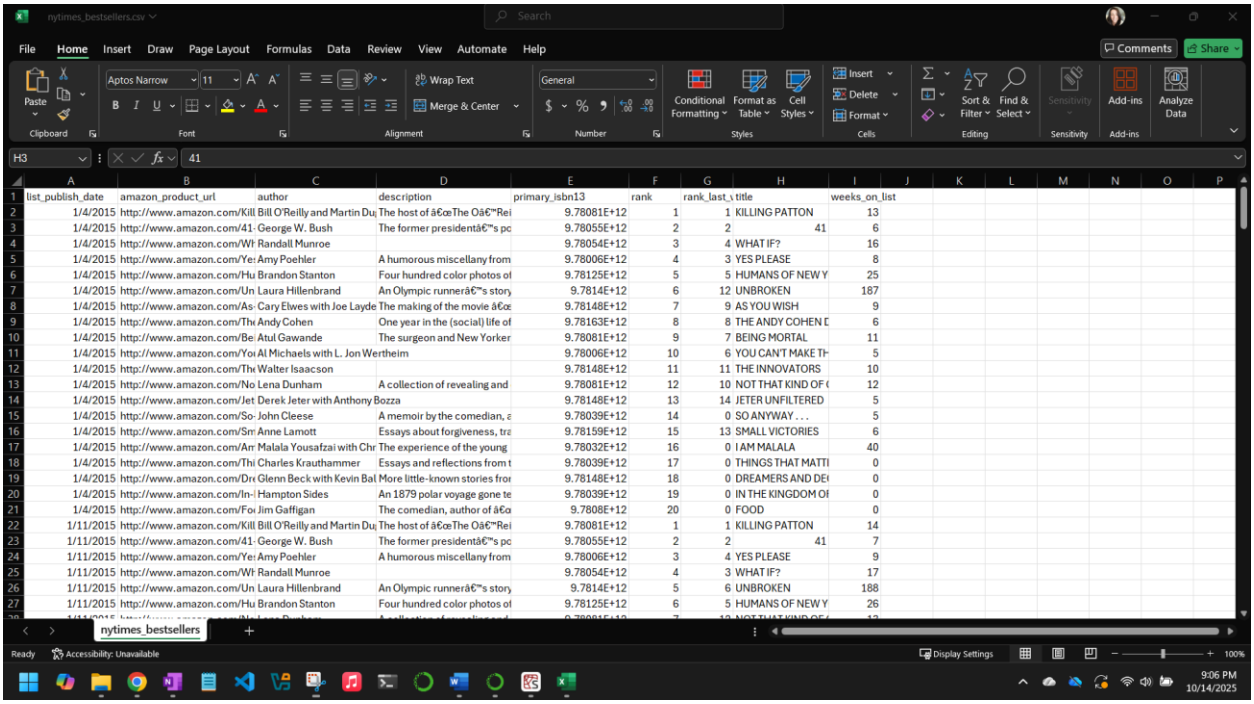


Their bestsellers history was removed recently, so separate API calls for each week's bestsellers list were made. In total, 52 API calls were made for all Hardcover Nonfiction bestsellers lists in each of the 5 years, or 260 calls total. The FAQs for the NYTimes API show there is five calls per minute and 500 calls per day limit. A 12 second sleep was implemented between calls to not hit the limit.



Using Python, functions were created to format the date for the API call and to pull the book data from the JSON response and then populate a blank csv file with the columns named in the JSON response. Next, a while loop was implemented using hardcover nonfiction bestseller lists published between 1/4/2015(the first Sunday in 2015) and 12/31/2019 as the qualifying condition. In the loop, API calls were made to https://api.nytimes.com/svc/books/v3/lists/{list_date}/{list_name}.json?api-key=6OKOlzD0cmYHtDsAsOC9oGE0GZH9gPF4, the JSON response was parsed for the elements wanted, the elements were printed to a .csv file, and the 12 seconds sleep was implemented before looping again with a new date and API call.

Because of the time limitation of 5 calls per minute, it took 52 minutes to run the program and collect the data. The created csv file includes 4460 rows and 9 columns of data.

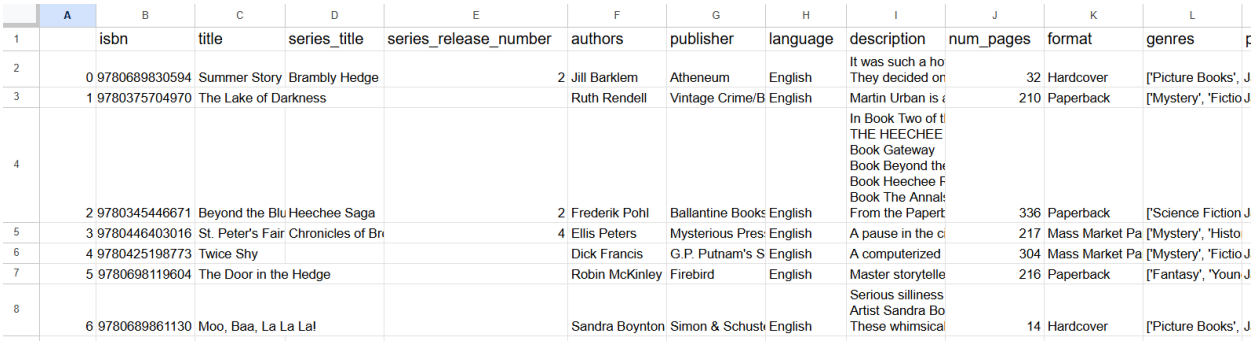


list_publish_date	amazon_product_url	author	description	primary_isbn13	rank	rank_last_	title	weeks_on_list
1/4/2015	http://www.amazon.com/Kill	Bill O'Reilly and Martin Du	The host of â€œThe Oâ€™Rei	9.78081E+12	1	1	KILLING PATTON	13
1/4/2015	http://www.amazon.com/41	George W. Bush	The former presidentâ€™s pc	9.78055E+12	2	2	41	6
1/4/2015	http://www.amazon.com/Wh	Randall Munroe		9.78054E+12	3	4	WHAT IF?	16
1/4/2015	http://www.amazon.com/Ye	Amy Poehler	A humorous miscellany from	9.78006E+12	4	3	YES PLEASE	8
1/4/2015	http://www.amazon.com/Hu	Brandon Stanton	Four hundred color photos of	9.78125E+12	5	5	HUMANS OF NEW Y	25
1/4/2015	http://www.amazon.com/Un	Laura Hillenbrand	An Olympic runnerâ€™s story	9.7814E+12	6	12	UNBROKEN	187
1/4/2015	http://www.amazon.com/As	Cary Elwes with Joe Layde	The making of the movie â€œ	9.78148E+12	7	9	AS YOU WISH	9
1/4/2015	http://www.amazon.com/Th	Andy Cohen	One year in the (social) life of	9.78163E+12	8	8	THE ANDY COHEN D	6
1/4/2015	http://www.amazon.com/Be	Atul Gawande	The surgeon and New Yorker	9.78081E+12	9	7	BEING MORTAL	11
1/4/2015	http://www.amazon.com/Yo	Al Michaels with L. Jon Wertheim		9.78006E+12	10	6	YOU CAN'T MAKE TH	5
1/4/2015	http://www.amazon.com/Th	Walter Isaacson		9.78148E+12	11	11	THE INNOVATORS	10
1/4/2015	http://www.amazon.com/No	Lena Dunham	A collection of revealing and	9.78081E+12	12	10	NOT THAT KIND OF	12
1/4/2015	http://www.amazon.com/Je	Derek Jeter with Anthony Bozza		9.78148E+12	13	14	JETER UNFILTERED	5
1/4/2015	http://www.amazon.com/So	John Cleese	A memoir by the comedian, a	9.78039E+12	14	0	SO ANYWAY . . .	5
1/4/2015	http://www.amazon.com/Sm	Anne Lamott	Essays about forgiveness, tra	9.78159E+12	15	13	SMALL VICTORIES	6
1/4/2015	http://www.amazon.com/Ar	Malala Yousafzai with Chr	The experience of the young	9.78032E+12	16	0	I AM MALALA	40
1/4/2015	http://www.amazon.com/Thi	Charles Krauthammer	Essays and reflections from t	9.78039E+12	17	0	THINGS THAT MATTI	0
1/4/2015	http://www.amazon.com/Dr	Glenn Beck with Kevin Bal	More little-known stories fro	9.78148E+12	18	0	DREAMERS AND DE	0
1/4/2015	http://www.amazon.com/In-	Hampton Sides	An 1879 polar voyage gone te	9.78039E+12	19	0	IN THE KINGDOM OF	0
1/4/2015	http://www.amazon.com/Fo	Jim Gaffigan	The comedian, author of â€œ	9.7808E+12	20	0	FOOD	0
1/11/2015	http://www.amazon.com/Kill	Bill O'Reilly and Martin Du	The host of â€œThe Oâ€™Rei	9.78081E+12	1	1	KILLING PATTON	14
1/11/2015	http://www.amazon.com/41	George W. Bush	The former presidentâ€™s pc	9.78055E+12	2	2	41	7
1/11/2015	http://www.amazon.com/Ye	Amy Poehler	A humorous miscellany from	9.78006E+12	3	4	YES PLEASE	9
1/11/2015	http://www.amazon.com/Wh	Randall Munroe		9.78054E+12	4	3	WHAT IF?	17
1/11/2015	http://www.amazon.com/Un	Laura Hillenbrand	An Olympic runnerâ€™s story	9.7814E+12	5	6	UNBROKEN	188
1/11/2015	http://www.amazon.com/Hu	Brandon Stanton	Four hundred color photos of	9.78125E+12	6	5	HUMANS OF NEW Y	26

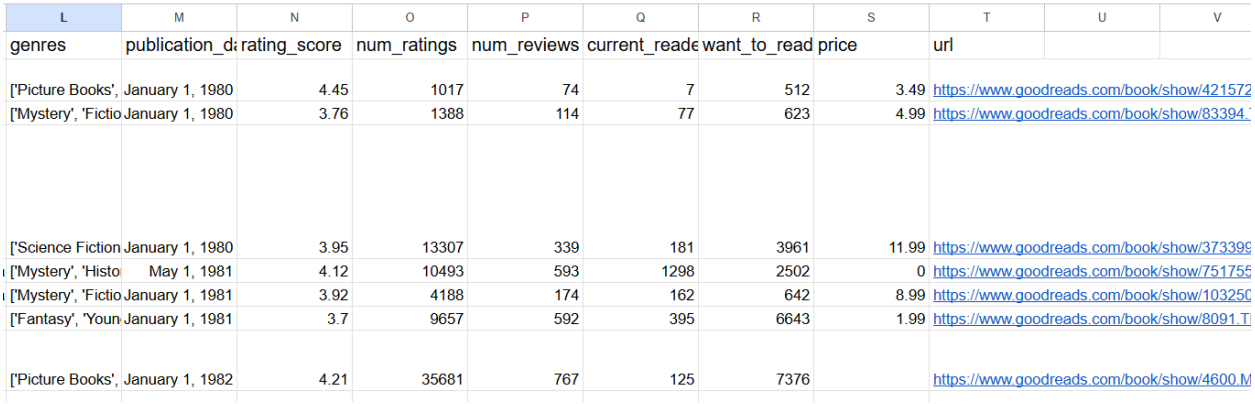
Data Cleaning

Goodreads Top 100 Ranked Books Dataset Before Cleaning

Two images are used to show the first eight lines and all variables in the “Before” dataset. Genres is at the end of the first image and the beginning of the second.



	A	B	C	D	E	F	G	H	I	J	K	L
1		isbn	title	series_title	series_release_number	authors	publisher	language	description	num_pages	format	genres
2	0	9780689830594	Summer Story	Brambly Hedge	2	Jill Barklem	Atheneum	English	It was such a ho They decided on	32	Hardcover	['Picture Books', J
3	1	9780375704970	The Lake of Darkness			Ruth Rendell	Vintage Crime/B	English	Martin Urban is a	210	Paperback	['Mystery', 'Fictio J
4	2	9780345446671	Beyond the Blu	Heechee Saga	2	Frederik Pohl	Ballantine Books	English	In Book Two of t THE HEECHEE Book Gateway Book Beyond the Book Heechee F Book The Annals From the Papert	336	Paperback	['Science Fiction J
5	3	9780446403016	St. Peter's Fair	Chronicles of Br	4	Ellis Peters	Mysterious Pres	English	A pause in the c	217	Mass Market Pa	['Mystery', 'Histo
6	4	9780425198773	Twice Shy			Dick Francis	G.P. Putnam's S	English	A computerized	304	Mass Market Pa	['Mystery', 'Fictio J
7	5	9780698119604	The Door in the Hedge			Robin McKinley	Firebird	English	Master storytelle	216	Paperback	['Fantasy', 'Youn J
8	6	9780689861130	Moo, Baa, La La La!			Sandra Boynton	Simon & Schust	English	Serious silliness Artist Sandra Bo These whimsical	14	Hardcover	['Picture Books', J



	L	M	N	O	P	Q	R	S	T	U	V
	genres	publication_d	rating_score	num_ratings	num_reviews	current_reade	want_to_read	price	url		
	['Picture Books',	January 1, 1980	4.45	1017	74	7	512	3.49	https://www.goodreads.com/book/show/421572		
	['Mystery', 'Fictio	January 1, 1980	3.76	1388	114	77	623	4.99	https://www.goodreads.com/book/show/83394.		
	['Science Fiction	January 1, 1980	3.95	13307	339	181	3961	11.99	https://www.goodreads.com/book/show/373399		
	['Mystery', 'Histo	May 1, 1981	4.12	10493	593	1298	2502	0	https://www.goodreads.com/book/show/751755		
	['Mystery', 'Fictio	January 1, 1981	3.92	4188	174	162	642	8.99	https://www.goodreads.com/book/show/10325(		
	['Fantasy', 'Youn	January 1, 1981	3.7	9657	592	395	6643	1.99	https://www.goodreads.com/book/show/8091.T		
	['Picture Books',	January 1, 1982	4.21	35681	767	125	7376		https://www.goodreads.com/book/show/4600.M		

Variable and Exploratory Data Analysis

Before cleaning, the Goodreads Top 100 dataset contained 20 variables as seen in the two images above. They include an unnamed index column, isbn, title, series_title, series_release_number, authors, publisher, language, description, num_pages, format, genres, publication_date, rating_score, num_ratings, num_reviews, current_readers, want_to_read, price, and url.

There are several columns/variables that do not assist in the project's goals. These unhelpful columns include the int64 datatype variable unnamed that is a duplicate of the index for the book/instance; the object datatype variables series_title, series_release_number, publisher, language, format, and url as nonfiction titles are not written in series and the other four are not pertinent information; and the float64 datatype variables current_readers, want_to_read, and price as these do not influence the results. The isbn variable is an object datatype. ISBN stands for International Standard Book Number and is a 10 or 13-digit number that identifies published books after 1970. Once assigned to a book, an ISBN can never be reused. When books have a new edition, ISBN is different. This column is also not needed as title will identify each book. After removing these eleven variables, nine remain.

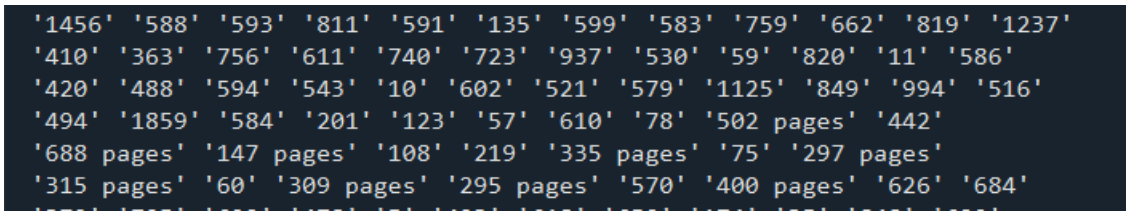
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4400 entries, 0 to 4399
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   title             4400 non-null   object
 1   authors           4400 non-null   object
 2   description       4398 non-null   object
 3   num_pages         4379 non-null   object
 4   genres            4400 non-null   object
 5   publication_date  4399 non-null   object
 6   rating_score      4400 non-null   float64
 7   num_ratings       4400 non-null   float64
 8   num_reviews       4400 non-null   float64
dtypes: float64(3), object(6)
memory usage: 309.5+ KB
None
```

The title and authors variables are object datatypes. The object datatype is the correct choice as text strings are the values in these columns. There are 4400 values in the 4400 rows in these two columns showing no missing values; this is correct as each book should have a title and an author(s) value. Duplicate removal is the only step needed to clean these columns.

The description variable has an object datatype. The object datatype is the correct choice as text strings should be the only value in this column and outliers as a result are not possible. There are 4398 values in the 4400 rows in each column showing two missing values. No cleaning is necessary in this column; it will be used to review which books may pique

interest once selected. If a description is missing, there is no accurate way to replace it without manually addressing each with a web search. It will be helpful to have it if available.

The num_pages variable has an object datatype. This should be changed to an integer, as the value should be a page count between 1 and 1000.



When visually spot checking the unique values in my console using the .unique() method, some values show a number followed by " pages". There are 4315 of 4400 possible values; the blanks and zeroes will be changed to a mean average for the column value. The " pages" suffix will be removed and then the datatype will be changed to integer. Outliers for page count (1000 pages or longer) will be dropped.

The genres column is an object datatype and contains a list of genres associated with a book; this is the correct datatype. There are 4400 values in the 4400 rows in each column showing no missing values. After sorting through the list in this column, if "Nonfiction" is not listed the book will be deleted. Because this is a list of strings, outliers are not a concern.

The publication_date is an object datatype in a string format, like November 1, 2025. There are 4399 values in the 4400 rows in each column showing only 1 missing value. The vector missing a value will be dropped. Next, the portion of the date string before the year will be removed and conversion of the remaining 4-digit string to an integer datatype will happen. The month and day of publishing holds no significance. Lastly, books published before 2000 will be dropped.

publication_date	
January 1, 1986	60
January 1, 1984	58
January 1, 1983	57
January 1, 1990	56
January 1, 1982	53
	..
December 7, 2023	1
June 20, 2023	1
July 20, 2023	1
June 9, 2023	1
May 30, 2023	1

The rating_score is a float64 datatype; this is correct as the value can fall anywhere between 0.0 and 5.0. There are 4400 values in the 4400 rows showing no missing values. Using the .describe() method, scores range from 2.97 to 4.81. There are no outliers. Books below 4.0 are dropped.

	rating_score	num_ratings	num_reviews
count	4400.00000	4.400000e+03	4400.000000
mean	4.02337	1.476433e+05	8887.657045
std	0.24228	4.015620e+05	18632.823791
min	2.97000	4.000000e+00	0.000000
25%	3.87000	1.372150e+04	743.000000
50%	4.03000	4.415300e+04	2545.500000
75%	4.18000	1.217560e+05	8383.500000
max	4.81000	9.911798e+06	259109.000000

The num_ratings score is a float64 datatype. When a reader marks a book as read in the account, they can select to rank a book on a scale of 1 to 5 starts or not rank it. This variable should be corrected to an integer as you cannot partially review a book; either it is rated or not, and each star ranking counts as one rating. There are 4400 values in the 4400 rows showing no missing values. Outliers would not be a consideration as a wildly popular book could have many more reviews than the next closest value and it would be accurate.

The num_reviews score is a float64. When a reader marks a book as read and rates it, they have the option of writing a review. Writing a review typically happens when someone has a strong opinion of a book. This should be corrected to an integer datatype as the reader either

wrote a review or did not; the value should be a whole number. There are 4400 values in the 4400 rows showing no missing values.

Cleaning the Goodreads Top 100 Dataset

Using the drop_duplicates() method, multiple instances of the same book are removed looking only at title and authors columns; it would be possible for the same book to make it to the Top 100 in multiple years with different review counts. The updated data frame now has 4,335 books, a loss of 155 duplicate occurrences. Using the keep='last' parameter to have an accurate num_ratings and num_reviews count; these counts would likely increase when a book is on the list in a future year.

Next, missing values are addressed. In the publication_date column, the book missing a publication date was dropped. To address the num_pages missing values, the " pages" suffix was removed from some values had and then all num_pages values were changed to numbers. Then, the mean average of the column replaced any missing values with the mean average rounded down to the nearest whole number.

For incorrect values and values in the wrong format, the num_pages datatype was changed from the newly changed float64 to integer, the num_ratings and num_reviews changed from float64 to integers datatypes, the formatting for publication_date changed to only have a 4-digit year, and then looped in reverse to remove values outside of desired ranges (allowing page counts between 1 and 1000, publication years after 2000, and ratings_score over 4.0) or did not include Nonfiction in the genres list.

This takes the count of books down to 110. The index was reset and the updated dataframe was written into a csv file.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110 entries, 0 to 109
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   title             110 non-null    object
 1   authors           110 non-null    object
 2   description       110 non-null    object
 3   num_pages         110 non-null    int64
 4   genres            110 non-null    object
 5   publication_date  110 non-null    int64
 6   rating_score      110 non-null    float64
 7   num_ratings       110 non-null    int64
 8   num_reviews       110 non-null    int64
dtypes: float64(1), int64(4), object(4)
memory usage: 7.9+ KB
None
```

To create a normalized dataset from the cleaned data, scikit-learn's MinMaxScaler was used and the fit_transform() method applied to the publication_date, num_pages, num_ratings, and num_reviews variables. The before and after outputs are listed below in order:

```
     publication_date  num_pages  num_ratings  num_reviews
0                2001        751       359949         7166
1                2001        216        42378         1267
2                2001        277        27693         1511
3                2003        187        74931         3144
4                2004        358         9507         1176
..                ...        ...          ...          ...
105              2021         29        31241         2999
106              2021        304        39037         4021
107              2022        320       881457        94122
108              2023        329        75313         7984
109              2023        284        28248         4007

[110 rows x 4 columns]
     publication_date  num_pages  num_ratings  num_reviews
0            0.000000   0.818482     0.241543     0.067319
1            0.000000   0.229923     0.028415     0.011841
2            0.000000   0.297030     0.018560     0.014135
3            0.090909   0.198020     0.050262     0.029493
4            0.136364   0.386139     0.006355     0.010985
..                ...        ...          ...          ...
105          0.909091   0.024202     0.020941     0.028130
106          0.909091   0.326733     0.026173     0.037741
107          0.954545   0.344334     0.591536     0.885121
108          1.000000   0.354235     0.050518     0.075012
109          1.000000   0.304730     0.018932     0.037610
```

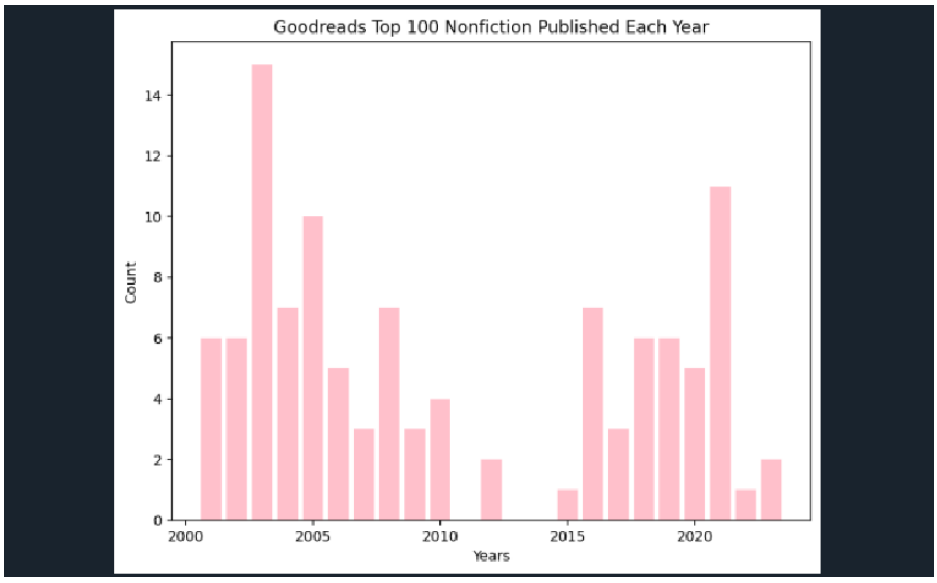
A new data frame was created with only unlabeled and quantitative data for KMeans,andsaved as Kmeans_Goodreads.csv; the index column was removed.

Below are images of the cleaned dataset.

	A	B	C	D	E
1	publication_date	num_pages	num_ratings	num_reviews	
2	2001	751	359949	7166	
3	2001	216	42378	1267	
4	2001	277	27693	1511	
5	2003	187	74931	3144	
6	2004	358	9507	1176	
7	2001	772	49178	1310	
8	2001	800	2032	60	
9	2001	269	16367	441	
10	2002	340	72035	7526	
11	2002	256	51184	7510	
12	2002	533	11595	908	
13	2002	560	24341	1119	
14	2002	240	16410	1259	
15	2003	544	382792	15633	
16	2003	317	130277	5920	
17	2003	303	216965	17385	
18	2003	400	205452	14241	
19	2003	341	179213	11894	
20	2003	408	20555	1241	
21	2003	375	138934	2743	
22	2003	328	3948	321	
23	2003	677	11560	971	
24	2003	409	12287	1491	

```
     publication_date  num_pages  num_ratings  num_reviews
0                2001        751       359949         7166
1                2001        216        42378         1267
2                2001        277        27693         1511
3                2003        187        74931         3144
4                2004        358         9507         1176
..                ...        ...          ...          ...
105              2021         29        31241         2999
106              2021        304        39037         4021
107              2022        320       881457        94122
108              2023        329        75313         7984
109              2023        284        28248         4007

[110 rows x 4 columns]
```



New York Times Nonfiction Bestsellers Dataset Before Cleaning

This image shows the top eight lines and all variables in the “Before” version of the dataset.

	A	B	C	D	E	F	G	H	I
1	list_publish_date	amazon_product_u	author	description	primary_isbn13	rank	rank_last_week	title	weeks_on_list
2	2015-01-04	http://www.amazon.com/l	Bill O'Reilly and Martin D	The host of “The O'Reilly	9780805096682	1	1	KILLING PATTON	13
3	2015-01-04	http://www.amazon.com/	George W. Bush	The former president's pe	9780553447781	2	2	41	6
4	2015-01-04	http://www.amazon.com/	Randall Munroe		9780544272996	3	4	WHAT IF?	16
5	2015-01-04	http://www.amazon.com/	Amy Poehler	A humorous miscellany f	9780062268341	4	3	YES PLEASE	8
6	2015-01-04	http://www.amazon.com/l	Brandon Stanton	Four hundred color photo	9781250038821	5	5	HUMANS OF NEW YOR	25
7	2015-01-04	http://www.amazon.com/l	Laura Hillenbrand	An Olympic runner's stor	9781400064168	6	12	UNBROKEN	187
8	2015-01-04	http://www.amazon.com/	Cary Elwes with Joe Lay	The making of the movie	9781476764023	7	9	AS YOU WISH	9

Variable and Exploratory Data Analysis

Before cleaning, the New York Times Nonfiction Bestsellers 2015-2019 dataset contained nine variables. They included list_publish_date, amazon_product_url, author, description, primary_isbn13, rank, rank_last_week, title, and weeks_on_list. The three columns that do not add value for analysis include the object datatype variable amazon_product_url, the int64 rank_last_week, and the flat64 primary_isbn13. As such, they were removed leaving six columns.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4490 entries, 0 to 4489
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   list_publish_date  4490 non-null   object
 1   author             4490 non-null   object
 2   description        3679 non-null   object
 3   rank               4490 non-null   int64 
 4   title              4490 non-null   object
 5   weeks_on_list      4490 non-null   int64 
dtypes: int64(2), object(4)
memory usage: 210.6+ KB
None
```

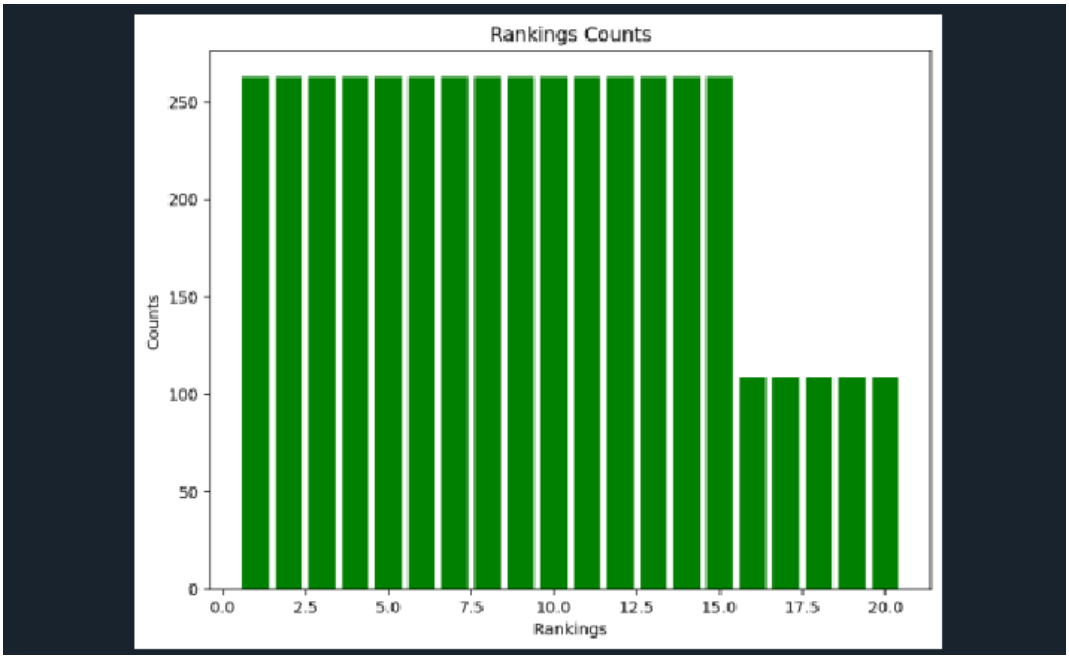
Duplicates are the main concern for this set as books can be on the Bestseller’s list multiple weeks in a row, as seen in the weeks_on_list columns. The. drop_duplicates() method looks for rows that are identical in every way unless only specific columns are checked but doesn’t allow for keeping the first and last book instance to later combine. Instead, the .groupby() method was used.

The list_publish_date is an object datatype in a string format but needs to be in datetime format. There are 4490 values in the 4490 rows in each column showing no missing values. Duplicates with be addressed elsewhere in the cleaning process, as described above. The API call allowed for a specific range of dates so date outliers are not a concern.

The title and authors variables are object datatypes in string formats. The object datatype is the correct choice as text strings should be the only values in these columns. There are 4490 values in the 4490 rows in each column showing no missing values; this is correct as each book should have a title and an author(s) value. Duplicate removal is the only step needed to clean these columns.

The description variable has an object datatype. The object datatype is the correct choice as text strings should be the only value in this column and outliers as a result are not possible. There are 3679 values in the 4400 rows in each column showing over 700 missing values. No cleaning is necessary in this column; it will be used to review which books may pique my interest once selected. If a description is missing, there is no accurate way to replace it without manually addressing each with a web search. It will be helpful to have it if available.

The rank column is an int64 datatype, which is correct. Visualizing the values in a bar graph shows approximately 100 rankings from 16 to 20, and over 250 for each 1 -15 ranking. The New York Times changed their list to only include 15 books. For lists from 1/4/15 through 1/29/17, there were 20 books on the Nonfiction Bestseller list, and the list was shortened to 15 books thereafter.



Cleaning is not necessary of books ranked 16 through 20; there were fewer options on the list but the 16 through 20 ranked books were still bestsellers. There are 4490 values in the 4490 rows in each column showing no missing values.

The weeks_on_list column is an int64 datatype and is correct. Applying the unique() method shows only whole numbers from 0 to 191. No cleaning is needed in this column. There are 4490 values in the 4490 rows in each column showing no missing values.

```
[ 13   6  16   8  25 187   9  11   5  10  12  40   0  14
7  17 188  26
   2  15  18 189   3   1  27  41  19   4 190  28  20 191
42  29  21  43
  22  44  23  30  45  24  31  46  47  39  48  49  50  32
33  34  35  36
  37  38  51  52  53  54  55  56  57  58  59  60  61  62
63  64  65  66
  67  68  69  70  71  72  73  74  75  76  77  78  79  80
81  82  83  84
  85  86  87  88  89  90  91  92  93  94  95]
```

Cleaning the New York Times Nonfiction Bestsellers Dataset

Using the .groupby.agg() method, multiple instances of the same book were removed. Each book is a group. Using the aggregate method allowed for keeping the first instance of the list_publish_date, the minimum rank value (highest rank being 1 or first), and the maximum value of the weeks_on_list. The as_index parameter for groupby was used so that the title, author, and description would still be visible. This took my initial count of books from 4490 to 701. The index was reset, and the updated data frame was written into a csv file.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 701 entries, 0 to 700
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype         
---  ------             --------------  -----         
 0   title              701 non-null    object        
 1   author             701 non-null    object        
 2   description        701 non-null    object        
 3   list_publish_date  701 non-null    datetime64[ns]
 4   rank               701 non-null    int64         
 5   weeks_on_list      701 non-null    int64         
dtypes: datetime64[ns](1), int64(2), object(3)
memory usage: 33.0+ KB
None
```

To create a normalized dataset from the cleaned data, the scikit-learn's MinMaxScaler was used and the fit_transform() method was applied to the rank and weeks_on_list variables. The before and after outputs are listed below in order:

```
     rank  weeks_on_list
0      17              0
1      10              2
2      17              0
3      12             12
4      14              3
..    ...            ...
696    14              1
697    15              5
698    14              1
699    14              2
700    15              4

[701 rows x 2 columns]
         rank  weeks_on_list
0    0.823529       0.000000
1    0.411765       0.021053
2    0.823529       0.000000
3    0.529412       0.126316
4    0.647059       0.031579
..        ...            ...
696  0.647059       0.010526
697  0.705882       0.052632
698  0.647059       0.010526
699  0.647059       0.021053
700  0.705882       0.042105

[701 rows x 2 columns]
```

A new data frame was created with only unlabeled and quantitative data for KMeans modelling and saved it as Kmeans_NYTimes.csv, and the index column was removed.

	A	B	C	D	E
1	rank	weeks_on_list			
2	17	0			
3	10	2			
4	17	0			
5	12	12			
6	14	3			
7	17	0			
8	18	0			
9	7	1			
10	16	0			
11	20	8			
12	10	9			
13	18	0			
14	11	1			
15	9	6			
16	12	4			
17	11	1			
18	10	1			
19	14	1			
20	10	1			

```
     rank  weeks_on_list
0      17              0
1      10              2
2      17              0
3      12             12
4      14              3
..    ...            ...
696    14              1
697    15              5
698    14              1
699    14              2
700    15              4

[701 rows x 2 columns]
```

Goodreads – User Export Dataset Before Cleaning

Before cleaning, this is what the data looked like. Notice multiple images to show all columns.

	A	B	C	D	E	F	G	H	I	J	K
1	Book Id	Title	Author	Author l-f	Additional Authors	ISBN	ISBN13	My Rating	Average Rating	Publisher	Binding
2	39380381	Say What You Mean: A Mindful Appr	Oren Jay Sofer	Sofer, Oren Jay		161180583X	9781611805833	4	4.14	Shambhala	Paperbac
3	215153740	All the Colors of the Dark	Chris Whitaker	Whitaker, Chris		0593798880	9780593798881	0	4.26	Crown Publishing Group	Kindle Edi
4	127279000	Listen for the Lie	Amy Tintera	Tintera, Amy		1250880319	9781250880314	4	4.08	Celadon Books	Hardcove
5	123136728	Orbital	Samantha Harvey	Harvey, Samantha		0802161545	9780802161543	5	3.52	Atlantic Monthly Press	Hardcove
6	171681821	The Anxious Generation: How the G	Jonathan Haidt	Haidt, Jonathan		0593655036	9780593655030	5	4.32	Penguin Press	Hardcove
7	48989372	A Very Punchable Face	Colin Jost	Jost, Colin		1101906324	9781101906323	5	4.2	Crown	Hardcove
8	63024287	The First Ladies	Marie Benedict	Benedict, Marie	Victoria Christopher Mur	0593440285	9780593440285	0	4.04	Berkley	Hardcove
9	8215796	Taking Charge of Adult ADHD	Russell A. Barkley	Barkley, Russell A.		1606233386	9781606233382	5	3.85	The Guilford Press	Paperbac
10	29010395	I Am Not Your Perfect Mexican Dau	Erika L. SÃ¡nchez	SÃ¡nchez, Erika L.		1524700487	9781524700485	5	4	Knopf Books for Young Rea	Hardcove
11	36692478	There There	Tommy Orange	Orange, Tommy				0	3.97	Knopf	Hardcove

	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Binding	Number of Pages	Year Published	Original Publication Year	Date Read	Date Added	Bookshelves	Bookshelves with positions	Exclusive Shelf	My Review	Spoiler	Private Notes	Read Count	Owned Copies	
2	Paperback	304	2018	2018	10/21/2025	10/21/2025			read				1	0	
3	Kindle Edition	597	2024	2024	10/17/2025	10/17/2025			read				1	0	
4	Hardcover	336	2024	2024	10/10/2025	10/10/2025			read				1	0	
5	Hardcover	224	2023	2023	10/7/2025	10/7/2025			read	My favorite book of the year to dat			1	0	
6	Hardcover	400	2024	2024	10/2/2025	10/2/2025			read				1	0	
7	Hardcover	312	2020	2020	9/22/2025	9/22/2025			read	A few stories were perfect and rea			1	0	
8	Hardcover	389	2023	2023	9/19/2025	9/19/2025			read				1	0	
9	Paperback	294	2010	2000	9/11/2025	9/11/2025			read				1	0	
10	Hardcover	344	2017	2017	9/4/2025	9/4/2025			read				1	0	
11	Hardcover	294	2018	2018	9/3/2025	9/3/2025			read				1	0	

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 24 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Book Id                       246 non-null    int64
 1   Title                         246 non-null    object
 2   Author                        246 non-null    object
 3   Author l-f                    246 non-null    object
 4   Additional Authors            39 non-null     object
 5   ISBN                          246 non-null    object
 6   ISBN13                        246 non-null    object
 7   My Rating                     246 non-null    int64
 8   Average Rating                246 non-null    float64
 9   Publisher                     241 non-null    object
 10  Binding                       246 non-null    object
 11  Number of Pages               246 non-null    int64
 12  Year Published                246 non-null    int64
 13  Original Publication Year     237 non-null    float64
 14  Date Read                     146 non-null    object
 15  Date Added                    246 non-null    object
 16  Bookshelves                   7 non-null      object
 17  Bookshelves with positions    7 non-null      object
 18  Exclusive Shelf               246 non-null    object
 19  My Review                     16 non-null     object
 20  Spoiler                       0 non-null      float64
 21  Private Notes                 0 non-null      float64
 22  Read Count                    246 non-null    int64
 23  Owned Copies                  246 non-null    int64
dtypes: float64(4), int64(6), object(14)
memory usage: 46.3+ KB
None
```

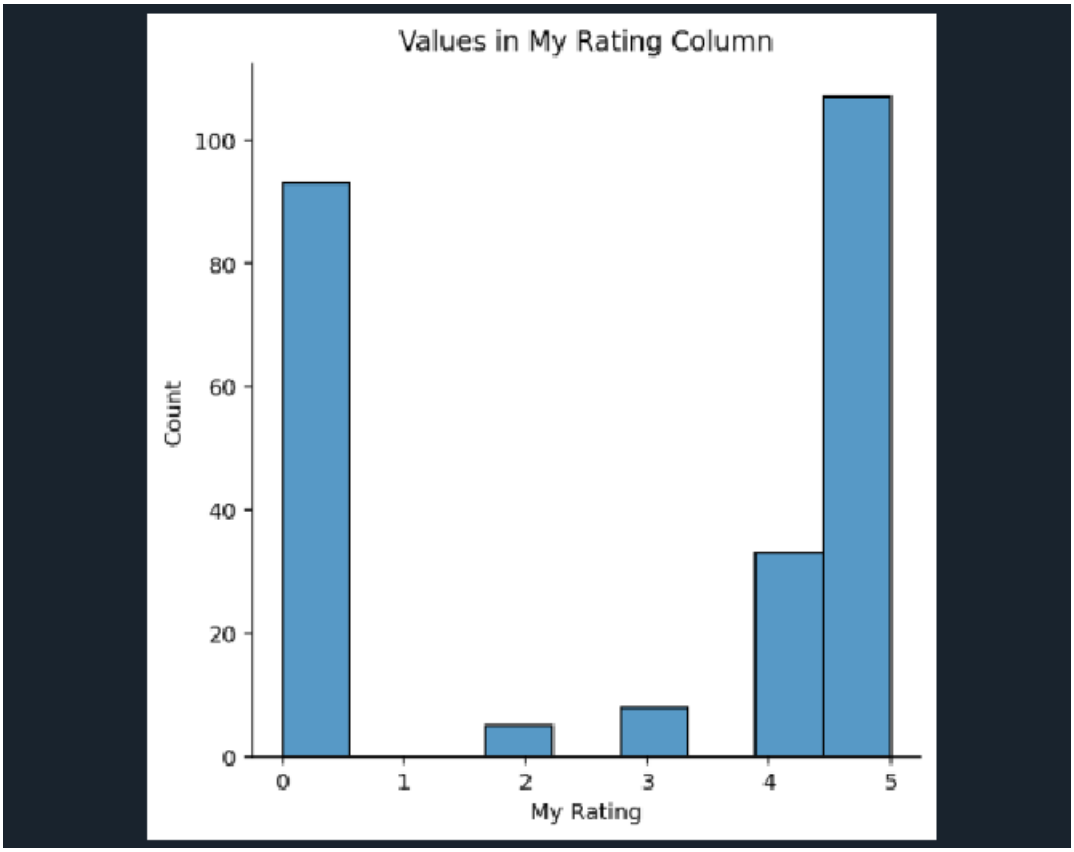
Variable and Exploratory Data Analysis, and Cleaning

Before cleaning, the dataset contained 24 variables as seen in the images above. They include Book Id, Title, Author, Author l-f, Additional Authors, ISBN, ISBN13, My Rating, Average Rating, Publisher, Binding, Number of Pages, Year Published, Original Publication Year, Date Read, Date Added, Bookshelves, Bookshelves with positions, Exclusive Shelf, My Review, Spoiler, Private Notes, Read Count, and Owned Copies.

This dataset will be cleaned for the purpose of KMeans clustering. Labeled and qualitative data must be removed in addition to columns that do not add value. The qualitative data that must be removed includes Title, Author, Author l-f, Additional Authors, Publisher, Binding, Bookshelves, Bookshelves with positions, Exclusive Shelf, and My Review columns. Additional quantitative columns must be removed as the project requirements explicitly require three and only three columns. The additional columns to drop include Book Id, ISBN, ISBN13, Number of Pages, Original Publication Year, Date Read, Date Added, Spoiler, Private Notes, Read Count, and Owned Copies. This leaves the dataset with int64 data type My Rating, float64 data type Average Rating, and int64 data type Year Published. The data types are correct.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   My Rating       246 non-null    int64
 1   Average Rating  246 non-null    float64
 2   Year Published  246 non-null    int64
dtypes: float64(1), int64(2)
memory usage: 5.9 KB
None
        My Rating  Average Rating  Year Published
count  246.000000      246.000000      246.000000
mean     2.849593        4.067195     2015.963415
std      2.301229        0.264510        6.928401
min      0.000000        3.020000     1995.000000
25%      0.000000        3.912500     2012.000000
50%      4.000000        4.070000     2018.000000
75%      5.000000        4.260000     2021.000000
max      5.000000        4.620000     2035.000000
```

The My Rating column shows 246 values in the 246 rows of the data frame; there are no NaN values. This column allows for whole numbers only, as 1,2,3,4, or 5 stars would be selected when rating a book. Many books are not rated. This appears to be captured as a 0.



All rows with a 0 are dropped. This brings the row count down from 246 to 153. The index was reset to 0 to 152. This column is now clean.

	My Rating	Average Rating	Year Published
0	4	4.14	2018
1	4	4.08	2024
2	5	3.52	2023
3	5	4.32	2024
4	5	4.20	2020
..	...	...	...
148	5	4.14	2021
149	5	4.12	2020
150	5	3.95	2013
151	5	4.07	2012
152	5	4.37	2019

[153 rows x 3 columns]

The describe() method seen in the image below shows the minimum value in the Average Rating column is 3.02 and the maximum value is 4.62. This range is appropriate and proves no outliers as all should fall between 0 and 5. The info() method shows the non-null count and the row count match; there are no NaN values in the Average Rating column. This column is clean.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 153 entries, 0 to 152
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   My Rating       153 non-null    int64
 1   Average Rating  153 non-null    float64
 2   Year Published  153 non-null    int64
dtypes: float64(1), int64(2)
memory usage: 3.7 KB
None
        My Rating  Average Rating  Year Published
count  153.000000      153.000000      153.000000
mean     4.581699        4.087908     2015.026144
std      0.739971        0.264345        7.271020
min      2.000000        3.020000     1995.000000
25%      4.000000        3.950000     2012.000000
50%      5.000000        4.080000     2017.000000
75%      5.000000        4.270000     2020.000000
max      5.000000        4.620000     2025.000000
```

The describe() method for the Year Published column shows a minimum value of 1995 and a maximum of 2025. These match the allowed years of 1995 to 2025; there are no outliers. The info() method shows the non-null count and the row count match; there are no NaN values in the Year Published column. This column is clean.

Unsupervised Learning with KMeans Clustering

Preparing the Data

The author's Goodreads Books Read Export cleaned dataset was used for KMeans modelling.

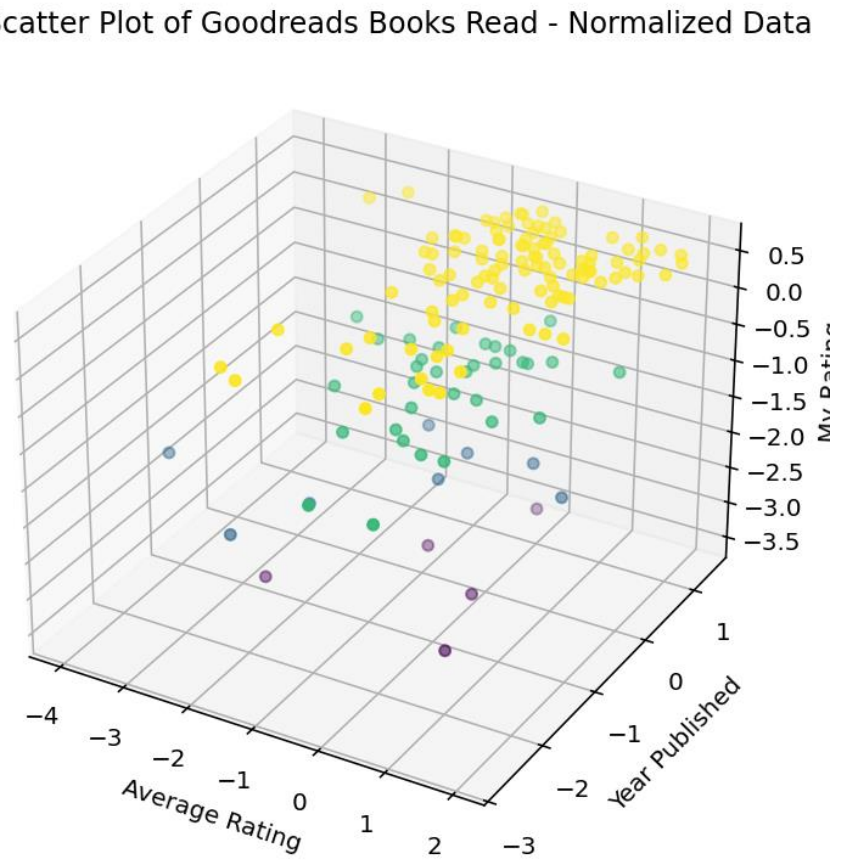
	My Rating	Average Rating	Year Published
0	4	4.14	2018
1	4	4.08	2024
2	5	3.52	2023
3	5	4.32	2024
4	5	4.20	2020
..	...	...	...
148	5	4.14	2021
149	5	4.12	2020
150	5	3.95	2013
151	5	4.07	2012
152	5	4.37	2019

There is a range of 3 for the My Rating column, a range of less than 2 for the Average Rating column, and a range of 30 for the Year Published column. To account for the variation in ranges, normalization of the data is needed. After applying the StandardScaler from the sklearn preprocessing module, this is the clean and normalized data.

	My Rating	Average Rating	Year Published
0	-0.788692	0.197706	0.410344
1	-0.788692	-0.030016	1.238248
2	0.567149	-2.155413	1.100264
3	0.567149	0.880869	1.238248
4	0.567149	0.425427	0.686312
..	...	...	...
148	0.567149	0.197706	0.824296
149	0.567149	0.121799	0.686312
150	0.567149	-0.523411	-0.279575
151	0.567149	-0.067969	-0.417559
152	0.567149	1.070637	0.548328

[153 rows x 3 columns]

Visualization of the Data

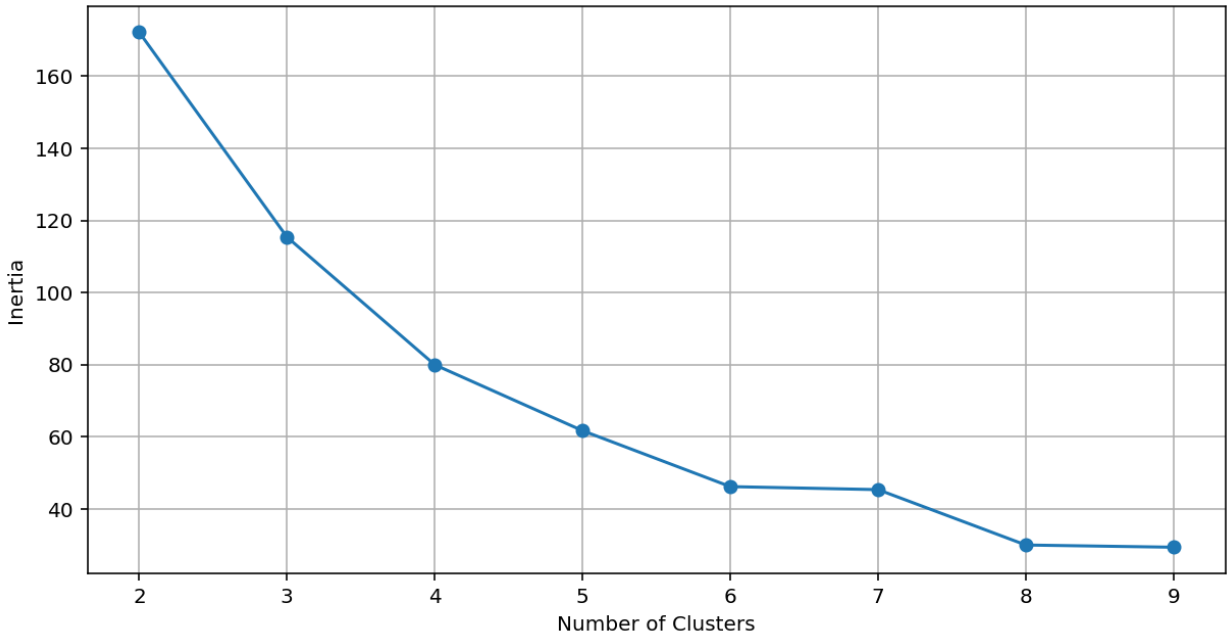


The scatter plot above shows a correlation between user ratings and the average rating. There is one main cluster of data points – the user rating is close to the Average rating. Virtually all the Average Rating books are rated around 4.0 or higher, and most of My Rating points are high as well. This visual shows the user tends to like books that have high Average Ratings on Goodreads.

There does not appear to be any relation between the year a book was published and the ratings – mine or the Goodreads average. The cluster runs evenly from mid-range to top along the year y axis. There are fewer books at the low end, which matches how I started reading and tracking what I read in 2021.

Applying KMeans Using Sklearn in Python

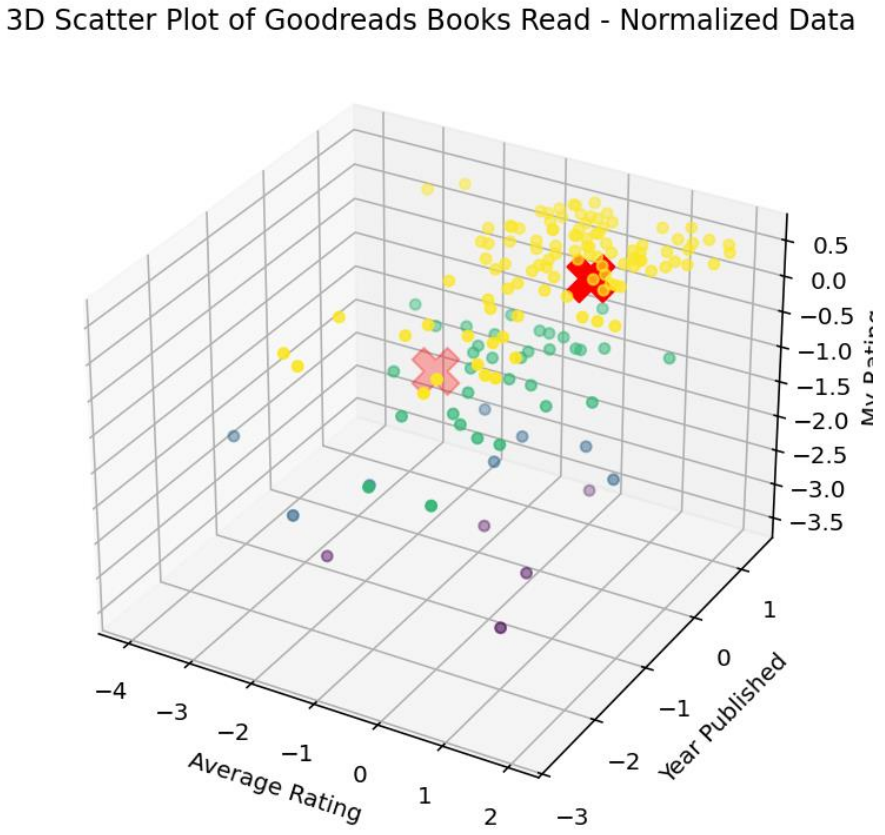
To determine the optimal number of clusters, the elbow method is applied. Eight KMeans applications were ran on the data that is correlated visually – the My Rating and Average Rating columns. The eight inertias were plotted from two clusters to nine; inertias are the average distance the data points in a cluster are to the centroid/center of the cluster for each KMeans run. There is a sharp drop from 2 to 3 and from 3 to 4. The rate of inertia reduction slows after four, which is the elbow point. Four clusters appear to be the optimal number of clusters for this data. See inertia plotting below:



When implementing KMeans with a cluster count of two on the data, the centroids are located at [0.47761278, 0.32588831, 0.31141254] and [-1.07716924, -0.73498214, -0.70233467]. This is the console output showing the centroid locations, the cluster assignments, and a count for each cluster.

```
[0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1
 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0
 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1
 1 1 1 1 0 1 1 1 0 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 1 0 1 1 0 0 0 0 0 1
 0 0 0 0 0]
[[ 0.47761278  0.32588831  0.31141254]
 [-1.07716924 -0.73498214 -0.70233467]]
0    106
1     47
```

The first centroid holds 106 data points and the second holds 47. If a new book is entered with normalized data of .5 My Rating, 1 Average Rating, and .6 Year Published, the KMeans prediction is that the book would fall in the first cluster, which appears accurate based on the visual of the two-cluster model below.

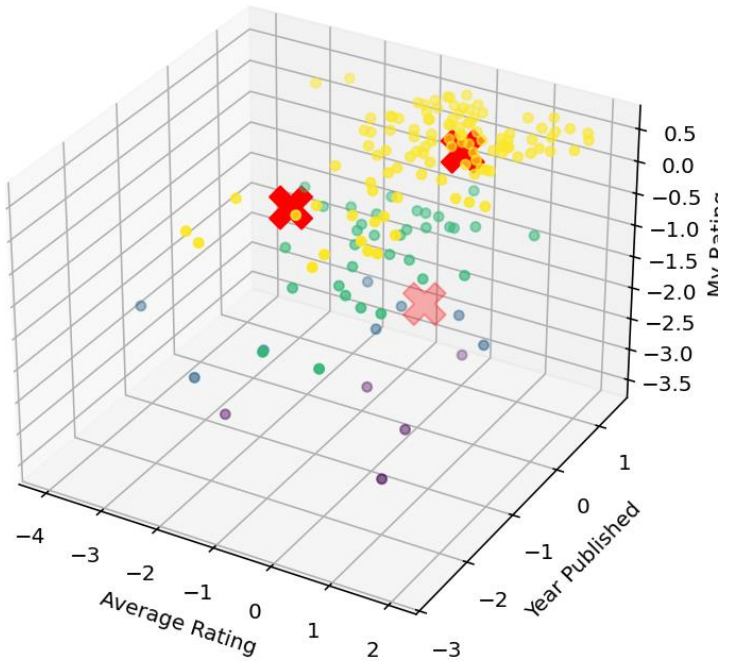


When implementing KMeans with a cluster count of three on the data, the centroids are located at [0.48151739, 0.30037992, 0.51492174], [0.14996742, -0.06213006, -1.71248559], and [-1.55135326, -0.84127221, -0.13727938]. This is the console output showing the centroid locations, the cluster assignments, and a count for each cluster.

```
[0 0 0 0 0 1 0 1 2 0 0 1 2 0 2 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 1 0 0 0 1
 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 1 0 0 0 0 0 0
 0 2 0 0 0 1 0 0 0 2 2 0 0 2 1 2 0 0 0 2 0 0 0 0 2 0 2 0 1 1 1 1 1 2 1 2 2
 2 2 2 2 2 2 2 2 0 0 0 0 2 0 0 0 0 1 2 2 2 0 1 2 1 1 1 1 1 2 2 1 0 0 0 0 2
 0 0 0 0 0]
[[ 0.48151739  0.30037992  0.51492174]
 [ 0.14996742 -0.06213006 -1.71248559]
 [-1.55135326 -0.84127221 -0.13727938]]
0    95
2    32
1    26
```

The first centroid holds 95 data points, the second holds 26, and the third holds 32. If a new book is entered with normalized data of .5 My Rating, 1 Average Rating, and .6 Year Published, the KMeans prediction is that the book would fall in the first cluster, which appears accurate based on the visual of the three-cluster model below.

3D Scatter Plot of Goodreads Books Read - Normalized Data

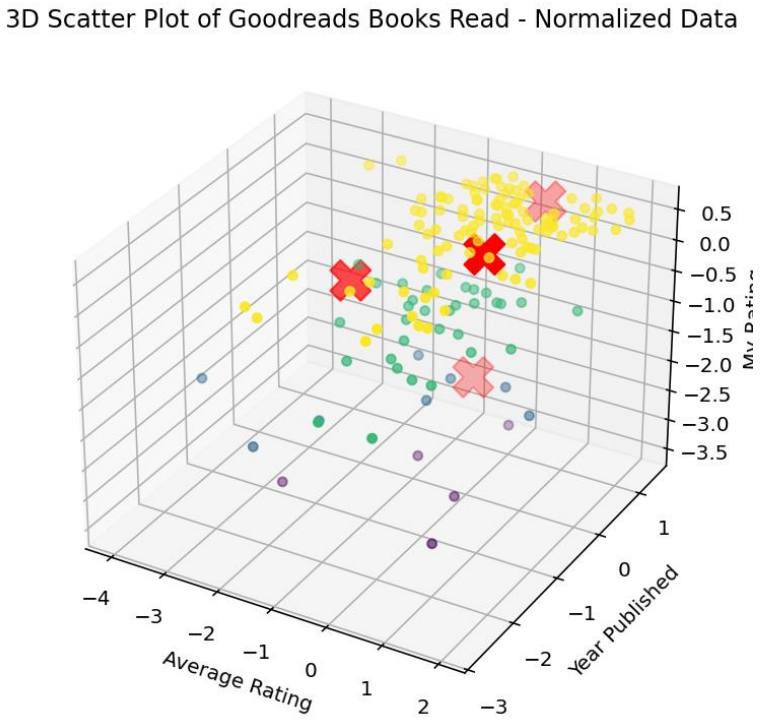


When implementing Kmeans with a cluster count of four on the data, the centroids are located [0.46285398, 1.1660072, 0.46341518], [-1.50649085, -0.77792331, -0.10506621],

[0.54249783, -0.3329537, 0.53327559], and [0.13328013, -0.02849739, -1.75324393]. This is the console output showing the centroid locations, the cluster assignments, and a count for each cluster.

```
[1 2 2 0 2 2 2 3 1 2 0 3 1 0 1 0 0 2 2 2 0 2 1 2 0 0 0 0 0 2 2 0 3 2 0 0 3
 0 2 0 2 3 0 2 3 3 0 2 2 2 2 2 0 2 2 0 2 2 2 1 3 2 2 0 2 2 0 3 2 2 2 0 2 2
 2 1 0 0 0 3 2 2 2 1 1 2 0 1 3 1 0 2 2 1 2 2 0 0 1 2 1 2 3 3 3 3 3 1 3 1 1
 1 1 1 1 1 1 1 1 1 0 0 0 1 2 2 2 0 3 1 1 1 2 3 1 3 3 3 3 3 1 1 3 0 0 0 0 1
 2 2 2 2 0]
[[ 0.46285398  1.1660072   0.46341518]
 [-1.50649085 -0.77792331 -0.10506621]
 [ 0.54249783 -0.3329537   0.53327559]
 [ 0.13328013 -0.02849739 -1.75324393]]
2    55
0    39
1    34
3    25
```

The first centroid holds 39 data points, the second holds 34, the third holds 55, and the fourth holds 25. If a new book is entered with normalized data of .5 My Rating, 1 Average Rating, and .6 Year Published, the KMeans prediction is that the book would fall in the first cluster, which appears accurate based on the visual of the four-cluster model below.



Conclusions From Kmeans Clustering

KMeans clustering helps organize data into clusters. The data must be unlabeled, quantitative, and ideally normalized. The closer the points are on a graph, the better in theory the KMeans algorithm will cluster. While the correlation found between user book ratings and average Goodreads reader's ratings is present, there is not a correlation between either of the two ratings and the year the book was published.

After visualizing my data, it is evident how influential a few points outside the typical range can be. For example, look at any of the graphs and you will find 10ish books published in the first 15 years of the data (1995 to 2010). Those books significantly influence the average distance to its cluster center. This suggests going back and cleaning the data again by either dropping the older published books or removing the Year Published column altogether for KMeans clustering. While they don't appear to be outliers, they behave like them when running KMeans making the predictions and optimal cluster count decisions less accurate. KMeans has helped find the natural groups in the user data.

Supervised Learning with Decision Trees

Formatting the Data

The cleaned version of the Goodreads Top100 dataset was utilized for Decision Tree modelling. The 110 books listed are less than 1000 pages long, rated higher than 4.0, nonfiction genres, and were published after 2000. This is a screenshot of the data before formatting it.

	A	B	C	D	E	F	G	H	I
1	Title	Authors	Description	Page Count	Genres	Publica	Star Sc	Rating	Reviews
2	John Adams	David McCullough	The enthralling, often	751	['History', 'Biography', 'Non	2001	4.07	359949	7166
3	The Universe in	Stephen Hawking	Stephen Hawkingâ€™s	216	['Science', 'Nonfiction', 'Phy	2001	4.18	42378	1267
4	A Cook's Tour: I	Anthony Bourdain	From the star of	277	['Nonfiction', 'Food', 'Trave	2001	4.08	27693	1511
5	Persepolis 2: Tl	Marjane Satrapi	In	187	['Graphic Novels', 'Nonficti	2003	4.21	74931	3144
6	Animals in Trar	Temple Grandin	Why would a cow lick a	358	['Nonfiction', 'Animals', 'Sc	2004	4.15	9507	1176
7	Theodore Rex	Edmund Morris	Theodore Roosevelt and	772	['Biography', 'History', 'Non	2001	4.18	49178	1310
8	New and Collec	CzesÅ‚aw MiÅ‚osz	New and Collected	800	['Poetry', 'Polish Literature',	2001	4.34	2032	60
9	Napalm & Silly	George Carlin	I THINK I AM,	269	['Humor', 'Nonfiction', 'Con	2001	4.09	16367	441
10	The Midwife: A	Jennifer Worth	At the age of twenty-two,	340	['Nonfiction', 'Memoir', 'His	2002	4.18	72035	7526

Decision Tree Modeling using Sklearn's Decision Tree Classification in Python requires the data to be labeled and quantitative. The Title, Authors, Description, Page Count, and Genres columns are removed as they are qualitative variables.

```
     Publication Year  Star Score  Rating Count  Reviews Count
0                2001        4.07        359949           7166
1                2001        4.18         42378           1267
2                2001        4.08         27693           1511
3                2003        4.21         74931           3144
4                2004        4.15          9507           1176
..                ...         ...           ...            ...
105              2021        4.56         31241           2999
106              2021        4.34         39037           4021
107              2022        4.47        881457          94122
108              2023        4.23         75313           7984
109              2023        4.30         28248           4007

[110 rows x 4 columns]
```

Then, a label column is created based on the Star Score column. Values between 4.01 and 4.10 will be labelled Good, 4.11 to 4.24 will be Better, and 4.25 and higher will be Best. The Star Score column is removed as this is what the Published Year, Rating Count, and Review Count is looking to classify/predict; the groupings based on the Star Score are the label. Below is the data frame after addressing these points. The label choice counts are roughly even.

```
Label
Better    39
Best      39
Good      32
Name: count, dtype: int64
     Publication Year  Rating Count  Reviews Count   Label
0                2001        359949           7166    Good
1                2001         42378           1267  Better
2                2001         27693           1511    Good
3                2003         74931           3144  Better
4                2004          9507           1176  Better
..                ...           ...            ...     ...
105              2021         31241           2999    Best
106              2021         39037           4021    Best
107              2022        881457          94122    Best
108              2023         75313           7984  Better
109              2023         28248           4007    Best

[110 rows x 4 columns]
```

To ensure that enough data about a book exists, rows with less than a 500 Rating Count or 200 Review Counts are removed. This leaves the dataset with 107 books. The columns that are not labels are all quantitative and int64 data types.

```
     Publication Year  Rating Count  Reviews Count   Label
0                2001        359949           7166    Good
1                2001         42378           1267  Better
2                2001         27693           1511    Good
3                2003         74931           3144  Better
4                2004          9507           1176  Better
..                ...           ...            ...     ...
105              2021         31241           2999    Best
106              2021         39037           4021    Best
107              2022        881457          94122    Best
108              2023         75313           7984  Better
109              2023         28248           4007    Best

[107 rows x 4 columns]
<class 'pandas.core.frame.DataFrame'>
Index: 107 entries, 0 to 109
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Publication Year  107 non-null    int64
 1   Rating Count      107 non-null    int64
 2   Reviews Count     107 non-null    int64
 3   Label             107 non-null    object
dtypes: int64(3), object(1)
memory usage: 4.2+ KB
None
```

Last, the formatted dataset is split using the train_test_split() method. This divides the vectors into 4 disjoint groups: training data, training data labels, testing data, and testing data labels. Below is the first 5 rows of data from X_train, Y_train, X_test, and Y_test in that order,

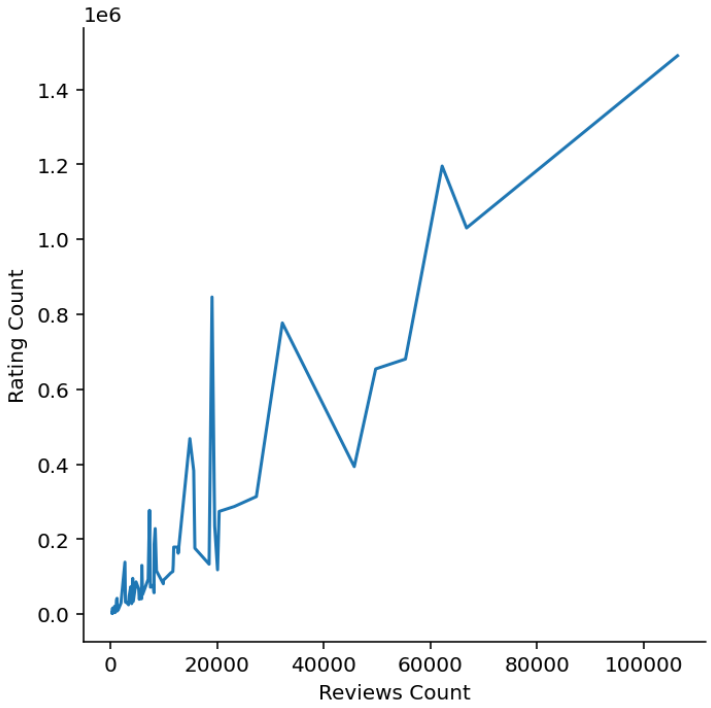
followed by the shape of each subset of data. The test size is 30% of the data and random state 3 is used to ensure rerunning the program splits the data the same way each time.

```
    Publication Year  Rating Count  Reviews Count
92              2020        114058          11731
71              2016         95603          11547
88              2019        287417          23255
86              2019         96138          10360
33              2004         21189            810
92      Best
71      Best
88      Good
86      Best
33    Better
Name: Label, dtype: object
     Publication Year  Rating Count  Reviews Count
74               2016         25140           3412
102              2021         93243           7101
7                2001         16367            441
5                2001         49178           1310
93               2020        120344          11949
74       Best
102    Better
7        Good
5      Better
93     Better
Name: Label, dtype: object
(74, 3)
(74,)
(33, 3)
(33,)
```

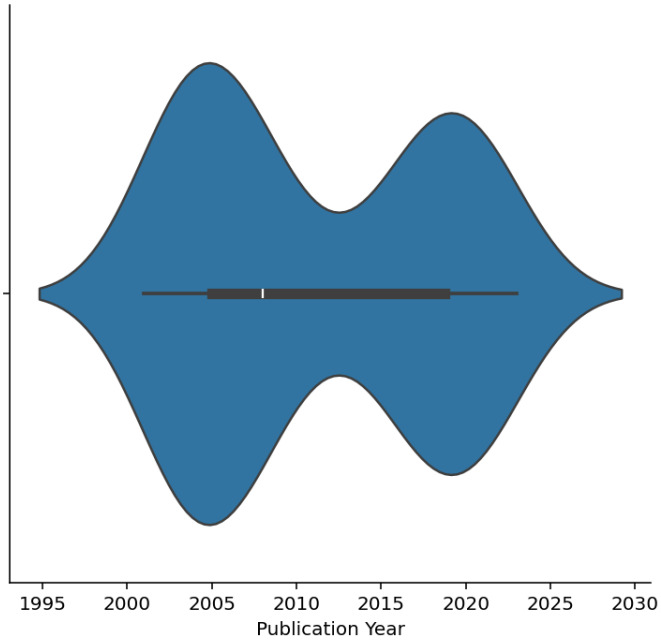
Visualizing the Data

Using the Seaborn data visualization library based on the matplotlib library, some information can be viewed and gleaned from the training dataset. This first image, below, is a bar chart and proves the testing data set labels are nearly the same count for each option. This proves the dataset is balanced. While normalization is not typical for decision tree modeling, sufficient vectors for each label will assist the model in accurate classification or prediction.

This next image is a line plot. It shows the relationship between the Rating Count and the Reviews Count. While there is variation present (the wiggling up and down of the line as it climbs to the top right corner), there is a clear direct relationship between the two variables for a given book. As the Review Count increases, the Rating Count also increases.



The last visualization is a violin plot, see below. It shows the distribution of books by the year they were published. The most nonfiction books making the Top 100 Goodreads list appear to have been published around 2005 and 2020, with fewer nonfiction books published in 2012 making the list around 2012.

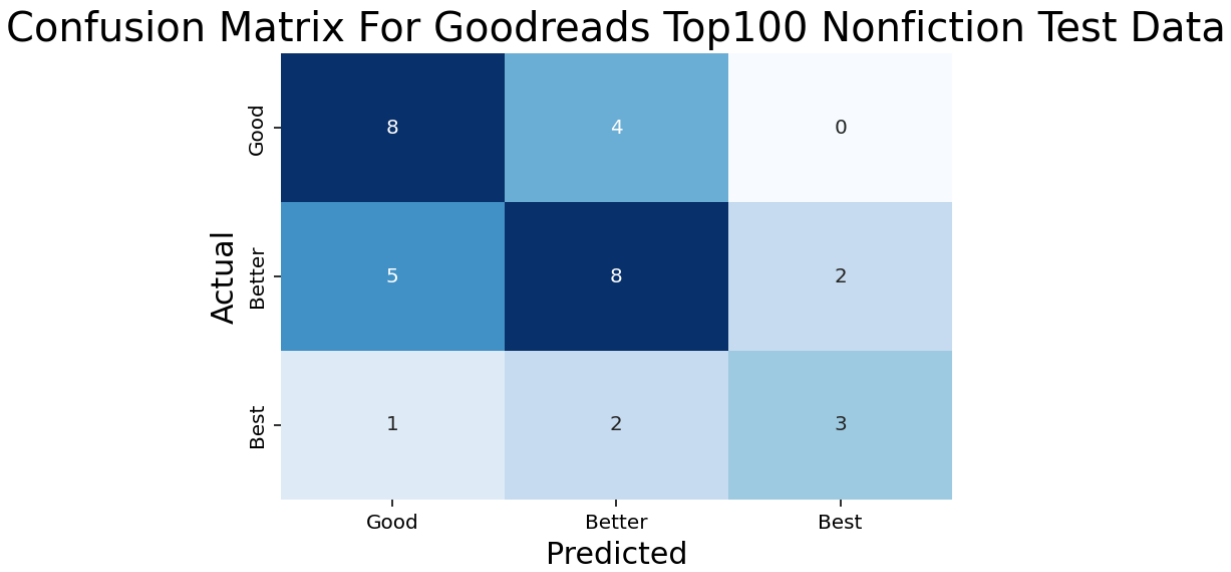


Decision Tree Modeling and Analysis

The Decision Tree Classifier was used to instantiate a Decision Tree model and the fit() method was applied to build the classifier from the training set. The tree accurately classified a book 58% of the time. More precisely, accurate classification of good or 4.0 to 4.10 star-rated books occurs 50% of the time, better or 4.11 to 4.24 star-rated books 53% of the time, and best or 4.25 or higher star-rated books 67% of the time. The image below is the classification report of the model.

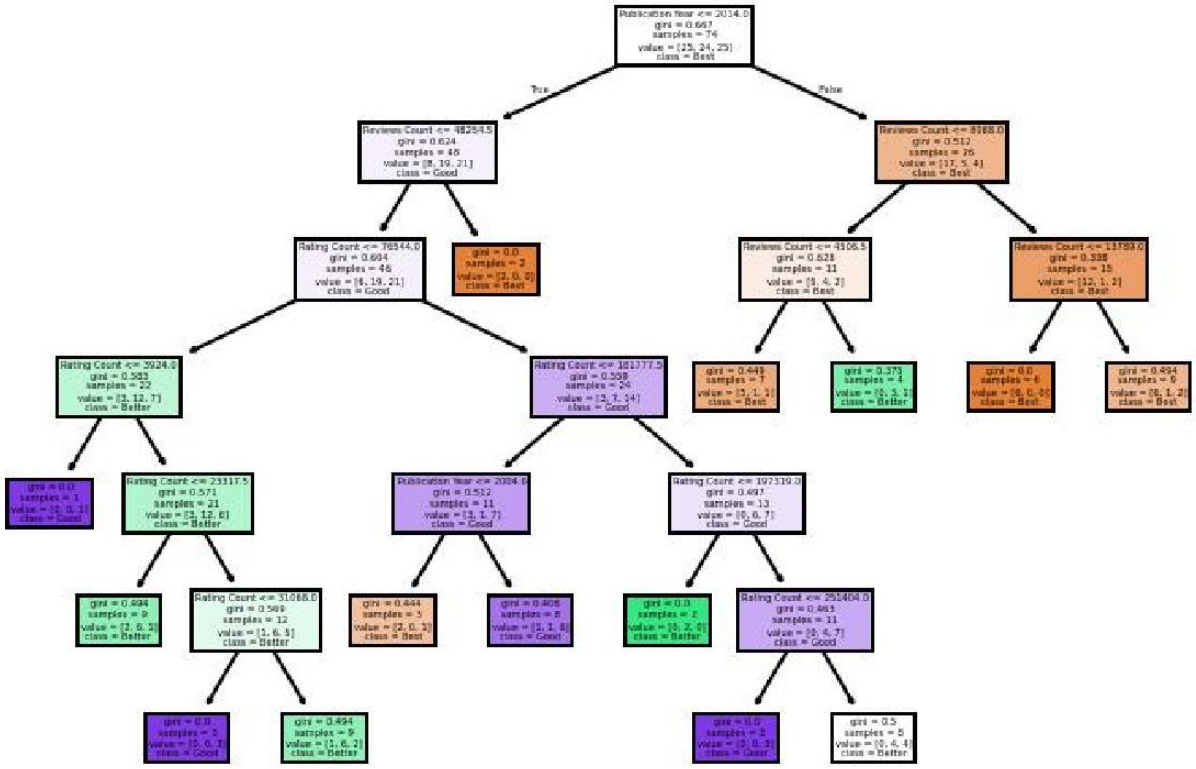
	precision	recall	f1-score	support
Best	0.67	0.57	0.62	14
Better	0.53	0.57	0.55	14
Good	0.50	0.60	0.55	5
accuracy			0.58	33
macro avg	0.57	0.58	0.57	33
weighted avg	0.58	0.58	0.58	33

Using Seaborn again, the confusion matrix is visualized below. Fourteen of the 33 books were incorrectly classified; 19 were accurate.



Decision Tree Visualization

The image of the decision tree was created using the plot_tree() method. When instantiated, the only added parameter utilized was the minimum samples for a split. This was specified at 10. The root node asks if the Publication Year is less than or equal to 2014 or not. It makes it way through the tree, using 5 branches and 14 leaves.



Conclusions from Decision Tree Modeling

The label on the data includes 3 options: Good, Better, and Best. This means that if I made a random guess of where to classify a non-fiction book with a star-rating of 4.0 or higher and I only have the year it was published, the number of ratings, and the number of reviews, there would be a 33.3% chance my random guess is correct. Based on the 58% accuracy of my model, I nearly double the chance of correctly predicting a book's label by building and utilizing a decision tree. Further, the confusion matrix demonstrates the errors are nearly all to a neighboring label. Though eight books were accurately labeled as good, four were incorrectly labeled as better but none were incorrectly labeled as best. There were eight accurately better-labeled books, but five were incorrectly predicted as good and two were incorrectly predicted as best. There were three properly best-labeled books, but one was incorrectly classified as good and two were wrongly labeled as better. To improve the accuracy results, I would adjust the number of labels up and down, increase the sample size of the data, and potentially add more variables.

Non-Technical Conclusion

Finding interesting, well-written books was the focus of this data science project. An analytical framework was used to identify compelling books based on specific criteria and personal preferences. The project used the author as a primary user, leveraging their preference for nonfiction books that have less than 1000 pages and haven't been read previously. The goal of this analysis was to answer two specific questions: Which books will appear in both the Top 100 Goodreads and the NY Times datasets but not in the Goodreads – User Export dataset, and will distinct patterns emerge?

To answer the first question, another program was written that found books listed in both the Top100 Goodreads list and the NY Times bestsellers list but not in the user's export list. There were 12 books meeting the criteria, see below.

	A	B	C	D	E	F	G	H
1	Title	Authors	Descriptio	Page Coun	Genres	Publicatio	Star Score	Rating Cou
2	Unbroken: A World War II Story of Survival, Resilience and Redemption	Laura Hillenbrand	On a May	475	['Nonfictio	2010	4.38	928426
3	Hamilton: The Revolution	Lin-Manuel Miranda	Lin-	285	['Nonfictio	2015	4.45	52649
4	Evicted: Poverty and Profit in the American City	Matthew Desmond	In	418	['Nonfictio	2016	4.47	95603
5	Born a Crime: Stories From a South African Childhood	Trevor Noah	The	289	['Nonfictio	2016	4.49	680766
6	A Mother's Reckoning: Living in the Aftermath of Tragedy	Sue Klebold	On April	336	['Nonfictio	2016	4.13	39906
7	Dear Ijeawele, or A Feminist Manifesto in Fifteen Suggestions	Chimamanda Ngozi Adichie	From the	63	['Nonfictio	2017	4.51	81243
8	Educated	Tara Westover	Tara	352	['Nonfictio	2018	4.47	1490090
9	Calypso	David Sedaris	David	273	['Nonfictio	2018	4.11	131019
10	The Sun Does Shine: How I Found Life and Freedom on Death Row	Anthony Ray Hinton	A	272	['Nonfictio	2018	4.64	57253
11	Catch and Kill: Lies, Spies, and a Conspiracy to Protect Predators	Ronan Farrow	In 2017, a	608	['Nonfictio	2019	4.41	96138
12	The Moment of Lift: How Empowering Women Changes the World	Melinda French Gates	A debut	273	['Nonfictio	2019	4.28	60066
13	Know My Name	Chanel Miller	She was	384	['Nonfictio	2019	4.71	192990
14								

To answer the second question, there are several insights to be gleaned. First, the original Goodreads Top100 books 1980-2023 dataset included 4400. After removing all fiction, books published before 2000, books with 1000 or more pages, and books with a rating score less than 4.0, only 110 books or 2.5% of the dataset remained. Goodreads has proven to be a better source for fiction readers, as less than 5 books a year on average is listed for the 23-year range. Second, a direct relationship between the user's book rating score and the Goodreads rating score was found. If Goodreads readers rated a book high, the user tended to as well. Lastly, the New York Times list alone is not an exact measurement of books our user would be

interested in. From just eyeing the list, there are several books that made it to the #1 ranking and were on the list for multiple weeks that the test subject would not want to read.

After applying the methodology of overlapping the datasets, it is clear this technique is the best-known option for finding a user's next great reads. While only twelve books were located, they each appear interesting and appropriate for the user's tastes.

References

[1] Nuvance Health, "A must-read: Physical and mental health benefits of reading books," Nuvance Health based in Western Connecticut and the Hudson Valley (NY). [Online]. Available: https://www.nuvancehealth.org/health-tips-and-news/physical-and-mental-health-benefits-of-reading-books. (Accessed: Dec. 8, 2025).

[2] L. Visser, "Goodreads Top 100 from 1980 to 2023," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/laravisser/goodreads-top100-from1980to2023-csv. (Accessed: Dec. 8, 2025).

[3] The New York Times, "Books API," The New York Times Developer Network. [Online]. Available: https://developer.nytimes.com/docs/books-product/1/overview. (Accessed: Dec. 8, 2025).

[4] Goodreads, "Library Export," [CSV Data file]. Available: https://www.goodreads.com/review/import. (Accessed: Dec. 8, 2025).

[5] Scikit-learn developers, "sklearn.cluster.KMeans," Scikit-learn Documentation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. (Accessed: Dec. 8, 2025).

[6] Scikit-learn developers, "1.10. Decision Trees," Scikit-learn Documentation. [Online]. Available: https://scikit-learn.org/stable/modules/tree.html. (Accessed: Dec. 8, 2025).