Capstone Project 1

Milestone Report

Rate A Read With goodreads

goodreads

Introduction:

Data Source:

Database Design and Data Wrangling:

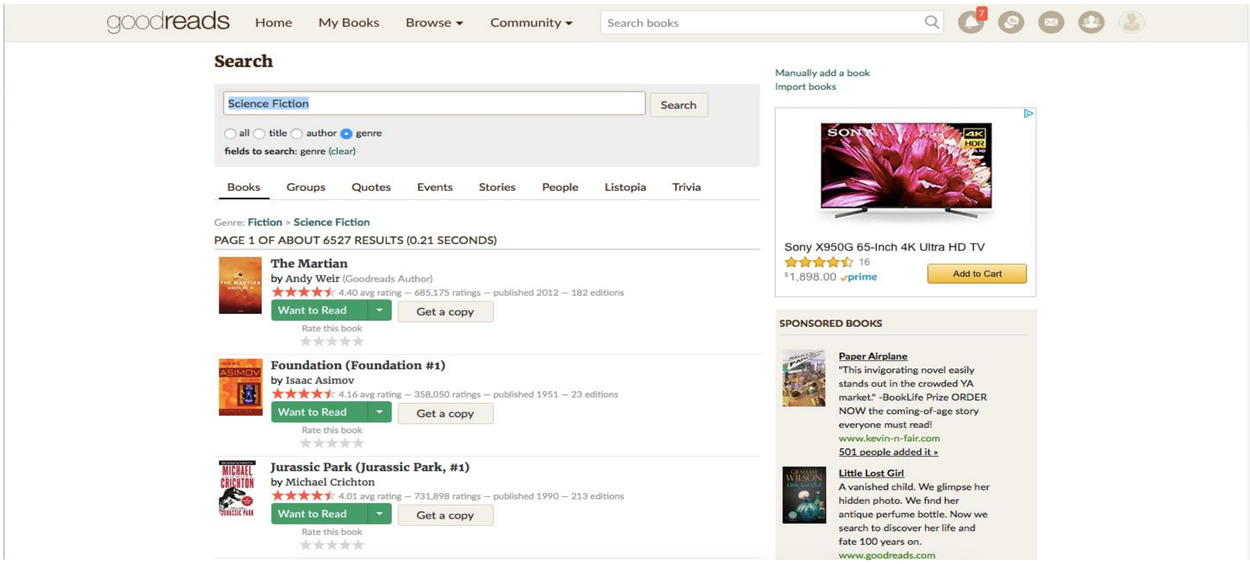
- Database Design
- Fetch Data from Database
- Data Wrangling
- Feature Extraction

Exploratory Data Analysis:

Scope Of Further Development:

Conclusion:

Introduction:



Search

Science Fiction

○ all ○ title ○ author ● genre

fields to search: genre (clear)

Genre: Fiction > Science Fiction

PAGE 1 OF ABOUT 6527 RESULTS (0.21 SECONDS)

The Martian
by Andy Weir (Goodreads Author)
4.40 avg rating — 685,175 ratings — published 2012 — 182 editions

Foundation (Foundation #1)
by Isaac Asimov
4.16 avg rating — 358,050 ratings — published 1951 — 23 editions

Jurassic Park (Jurassic Park, #1)
by Michael Crichton
4.01 avg rating — 731,898 ratings — published 1990 — 213 editions

Goodreads is a social cataloging website for people who love Books.Users can just sign up and then create a reading list or update the books they have read or currently reading or even write a review. They can also form their own groups of book suggestions, surveys, polls, blogs, and discussions.

In this project, I have explored the different features extracted from Books and Authors to determine what makes a book popular or what are the determinants in a book which earns a good rating?

As a user, we can login to the site and search for books of a particular genre.
In this project, we are extracting books details for the **"Science Fiction"** Genre. We have used the below tags to get a respectable amount of data:

1. science fiction
2. science-fiction-fantasy
3. science-fiction-romance
4. Apocalyptic
5. Space
6. Dystopia
7. Aliens
8. Fantasy

Data Source:

To access the GoodReads Data, we can use the GoodReads API. But to use the GoodReads API, we need to register for a developer key. The key can be registered on https://www.goodreads.com/api/keys.

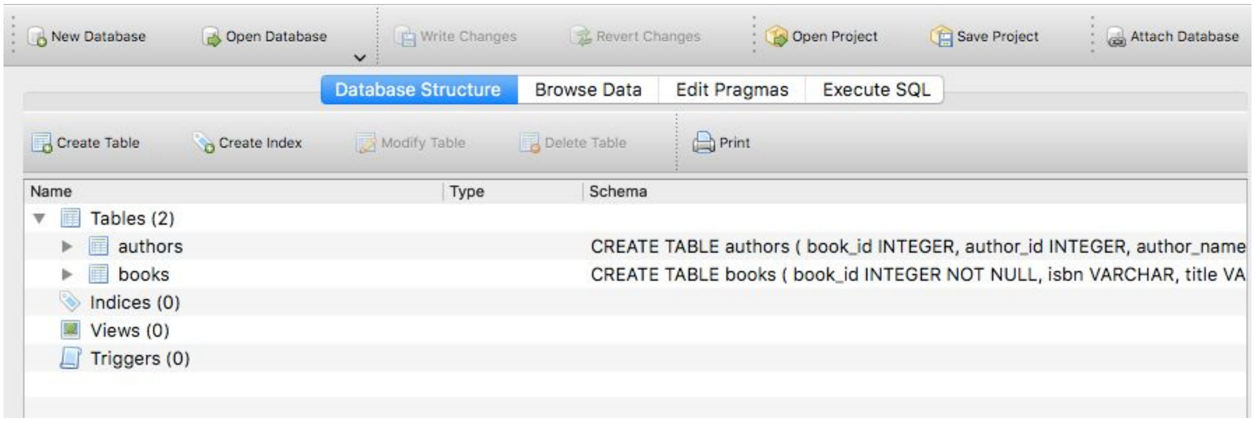
The credentials are secret information and thus can be stored in a pkl file and loaded when required.

Database Design and Data Wrangling:

Database Design

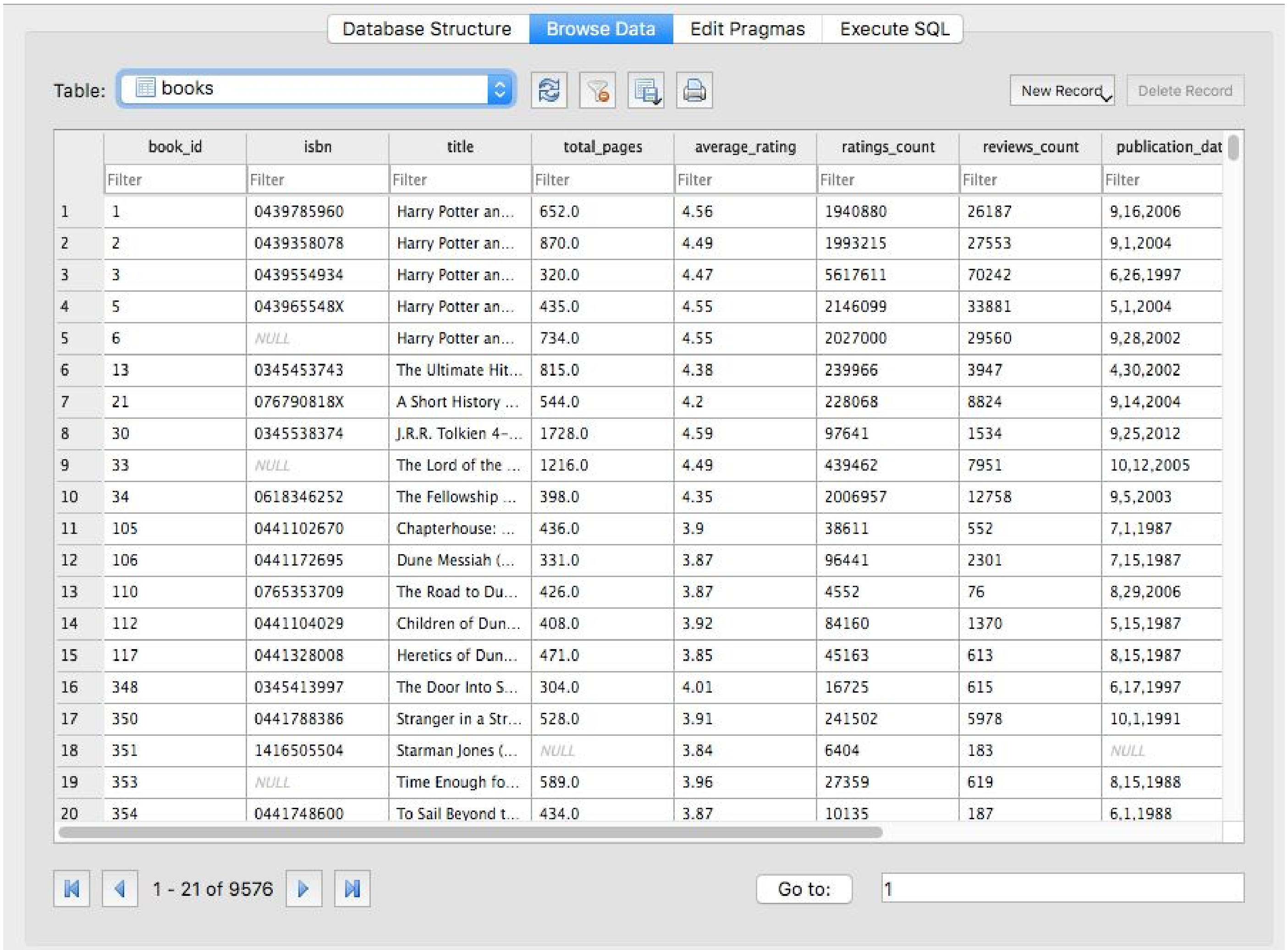
In this project I have used a SQLite3 Database to load the data. I have extracted both the Books details and the Author information from goodreads for a particular genre.

Thus we need to create 2 tables as below:



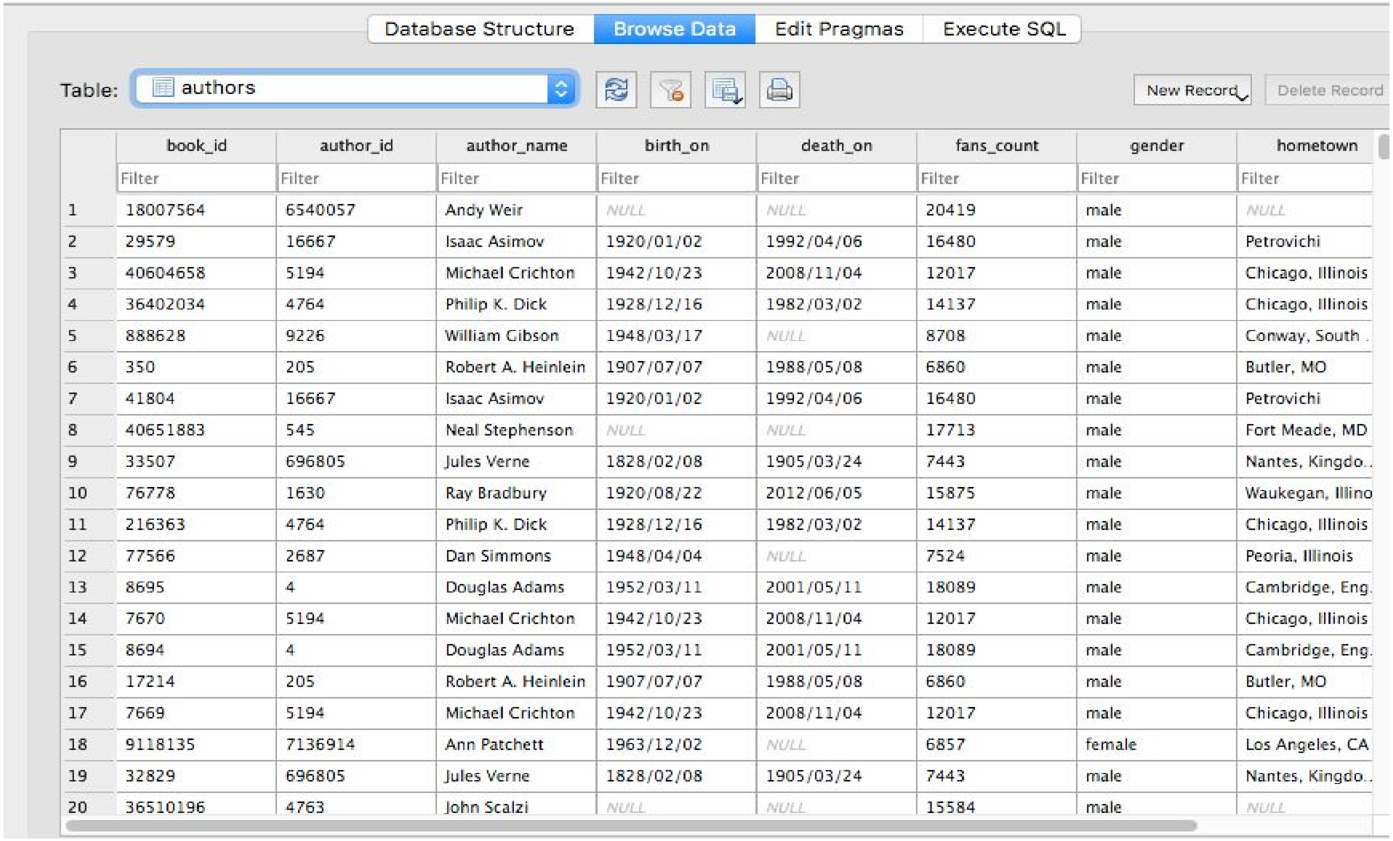
Name	Type	Schema
Tables (2)		
authors		CREATE TABLE authors (book_id INTEGER, author_id INTEGER, author_name
books		CREATE TABLE books (book_id INTEGER NOT NULL, isbn VARCHAR, title VA
Indices (0)		
Views (0)		
Triggers (0)		

1. **Books -** To store the book details where book_id is the Primary key



	book_id	isbn	title	total_pages	average_rating	ratings_count	reviews_count	publication_dat
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	0439785960	Harry Potter an...	652.0	4.56	1940880	26187	9,16,2006
2	2	0439358078	Harry Potter an...	870.0	4.49	1993215	27553	9,1,2004
3	3	0439554934	Harry Potter an...	320.0	4.47	5617611	70242	6,26,1997
4	5	043965548X	Harry Potter an...	435.0	4.55	2146099	33881	5,1,2004
5	6	NULL	Harry Potter an...	734.0	4.55	2027000	29560	9,28,2002
6	13	0345453743	The Ultimate Hit...	815.0	4.38	239966	3947	4,30,2002
7	21	076790818X	A Short History ...	544.0	4.2	228068	8824	9,14,2004
8	30	0345538374	J.R.R. Tolkien 4-...	1728.0	4.59	97641	1534	9,25,2012
9	33	NULL	The Lord of the ...	1216.0	4.49	439462	7951	10,12,2005
10	34	0618346252	The Fellowship ...	398.0	4.35	2006957	12758	9,5,2003
11	105	0441102670	Chapterhouse: ...	436.0	3.9	38611	552	7,1,1987
12	106	0441172695	Dune Messiah (...	331.0	3.87	96441	2301	7,15,1987
13	110	0765353709	The Road to Du...	426.0	3.87	4552	76	8,29,2006
14	112	0441104029	Children of Dun...	408.0	3.92	84160	1370	5,15,1987
15	117	0441328008	Heretics of Dun...	471.0	3.85	45163	613	8,15,1987
16	348	0345413997	The Door Into S...	304.0	4.01	16725	615	6,17,1997
17	350	0441788386	Stranger in a Str...	528.0	3.91	241502	5978	10,1,1991
18	351	1416505504	Starman Jones (...	NULL	3.84	6404	183	NULL
19	353	NULL	Time Enough fo...	589.0	3.96	27359	619	8,15,1988
20	354	0441748600	To Sail Beyond t...	434.0	3.87	10135	187	6,1,1988

2. **Authors** - To store the author details where book_id from Books table is the Foreign key

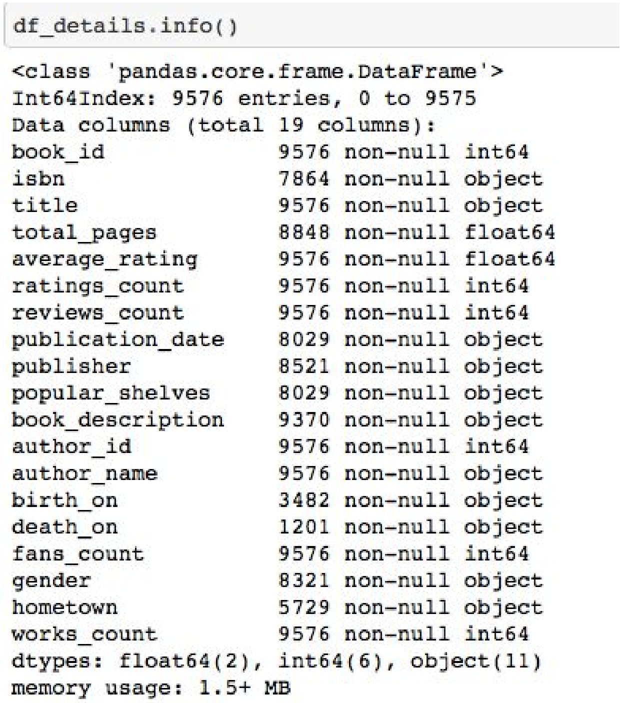


	book_id	author_id	author_name	birth_on	death_on	fans_count	gender	hometown
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	18007564	6540057	Andy Weir	NULL	NULL	20419	male	NULL
2	29579	16667	Isaac Asimov	1920/01/02	1992/04/06	16480	male	Petrovichi
3	40604658	5194	Michael Crichton	1942/10/23	2008/11/04	12017	male	Chicago, Illinois
4	36402034	4764	Philip K. Dick	1928/12/16	1982/03/02	14137	male	Chicago, Illinois
5	888628	9226	William Gibson	1948/03/17	NULL	8708	male	Conway, South .
6	350	205	Robert A. Heinlein	1907/07/07	1988/05/08	6860	male	Butler, MO
7	41804	16667	Isaac Asimov	1920/01/02	1992/04/06	16480	male	Petrovichi
8	40651883	545	Neal Stephenson	NULL	NULL	17713	male	Fort Meade, MD
9	33507	696805	Jules Verne	1828/02/08	1905/03/24	7443	male	Nantes, Kingdo..
10	76778	1630	Ray Bradbury	1920/08/22	2012/06/05	15875	male	Waukegan, Illino
11	216363	4764	Philip K. Dick	1928/12/16	1982/03/02	14137	male	Chicago, Illinois
12	77566	2687	Dan Simmons	1948/04/04	NULL	7524	male	Peoria, Illinois
13	8695	4	Douglas Adams	1952/03/11	2001/05/11	18089	male	Cambridge, Eng.
14	7670	5194	Michael Crichton	1942/10/23	2008/11/04	12017	male	Chicago, Illinois
15	8694	4	Douglas Adams	1952/03/11	2001/05/11	18089	male	Cambridge, Eng.
16	17214	205	Robert A. Heinlein	1907/07/07	1988/05/08	6860	male	Butler, MO
17	7669	5194	Michael Crichton	1942/10/23	2008/11/04	12017	male	Chicago, Illinois
18	9118135	7136914	Ann Patchett	1963/12/02	NULL	6857	female	Los Angeles, CA
19	32829	696805	Jules Verne	1828/02/08	1905/03/24	7443	male	Nantes, Kingdo..
20	36510196	4763	John Scalzi	NULL	NULL	15584	male	NULL

Fetch Data from Database

I have used “**read_sql_query**” from **Pandas** Library to read Data from Database .

Books Details are fetched into a Dataframe **df_books**. Author Details are fetched into a Dataframe **df_authors**. Books and Author details Dataframe are merged into another Dataframe **df_details** and below is the final result.



```
df_details.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9576 entries, 0 to 9575
Data columns (total 19 columns):
book_id             9576 non-null int64
isbn                7864 non-null object
title               9576 non-null object
total_pages         8848 non-null float64
average_rating      9576 non-null float64
ratings_count       9576 non-null int64
reviews_count       9576 non-null int64
publication_date    8029 non-null object
publisher           8521 non-null object
popular_shelves     8029 non-null object
book_description    9370 non-null object
author_id           9576 non-null int64
author_name         9576 non-null object
birth_on            3482 non-null object
death_on            1201 non-null object
fans_count          9576 non-null int64
gender              8321 non-null object
hometown            5729 non-null object
works_count         9576 non-null int64
dtypes: float64(2), int64(6), object(11)
memory usage: 1.5+ MB
```

Data Wrangling

- ❖ Convert the Gender column to category
- ❖ Convert the Date columns to Dates
- ❖ Handling Missing Data
  - For **total_pages** column, the missing values are filled with the MEAN of the total pages of the other records.
  - For **fans_count** column, the missing values are filled with the MEAN of the fans count of the other records.
  - For **popular shelves**, the mission value is filled with “No_Tags”.
  - **Gender** Missing Values are filled with the Forward fill method.
  - Missing **Book_description** column is filled with a constant value “No_Description”.

Feature Extraction

Tags:

Popular_shelves column of the books are used to fetch tags of each book by using the below steps:

- Join shelves of each record to get all the shelves.
- Exclude not so important Tags
- Fetched the Most Common Tags Value
- Create new Tags like “classics”, “thriller”, “romance” ,”paranormal” , “humour”, “dystopian”, “historical”, “comics” and put True/False for each record

Bag of Words from Book Description:

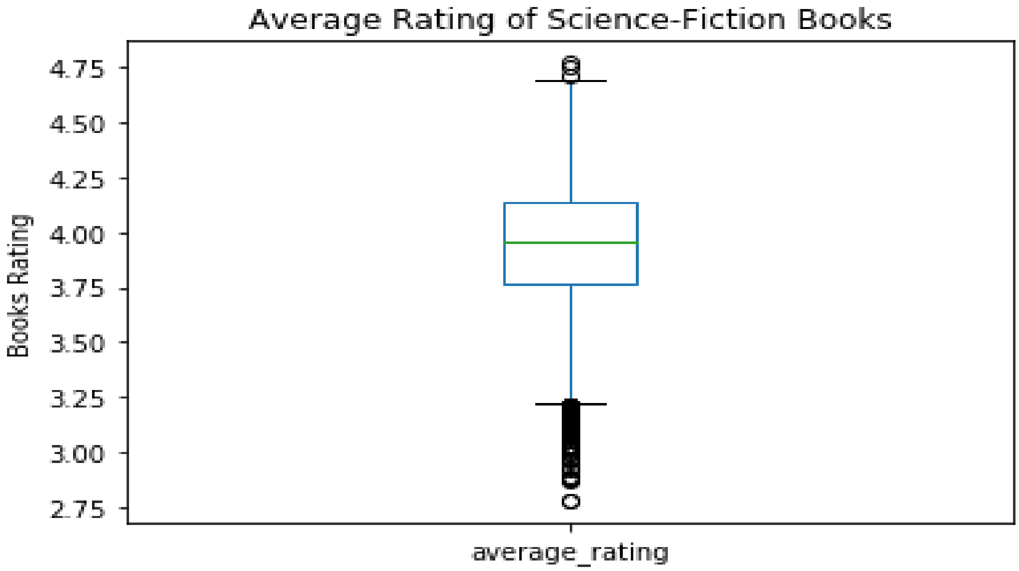
Bag Of Words is used to extract features from Book_Description column suing the below steps:

1. The records are converted to lowercase
2. HTML tags are removed from the records
3. Punctuations are removed from the records
4. Trailing spaces are removed from the records
5. Spaces in between words are removed from the records
6. Numbers are removed from the records
7. English stop words are removed
8. Tokenization, Stemming and Lemmatization process are used to clean the data
9. CountVectorizer method is used to get counts of each words

Exploratory Data Analysis:

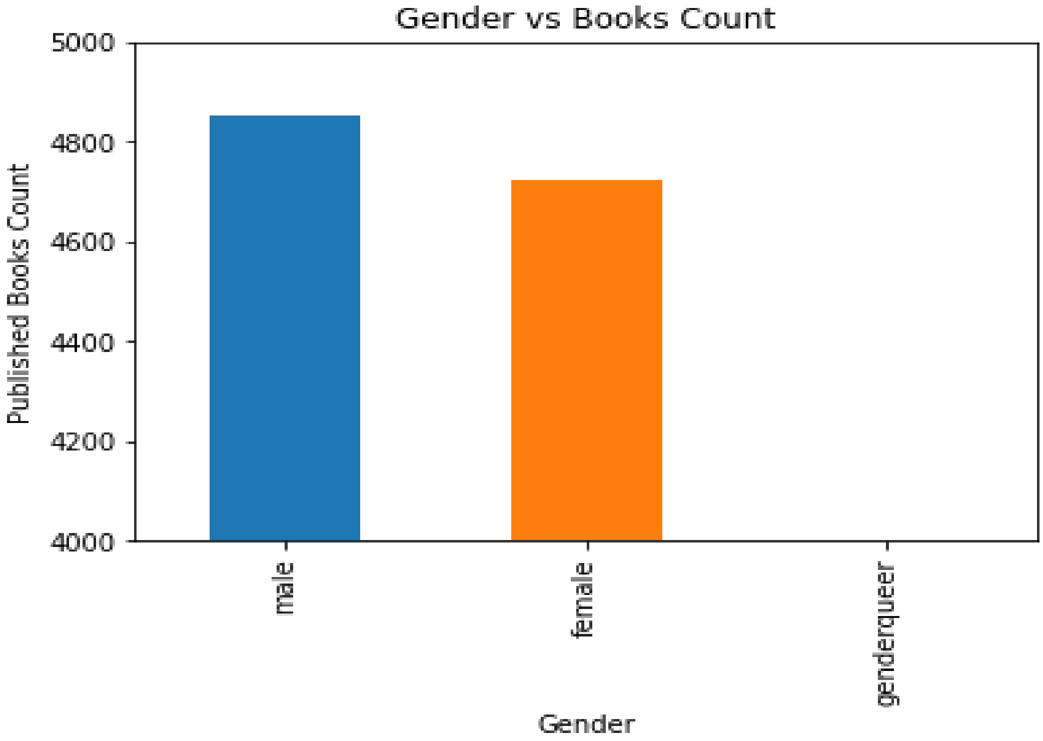
In this project, we are predicting the average rating of a book in Science Fiction Genre.
In the dataset, most of the features are categorical features.

Average Rating:



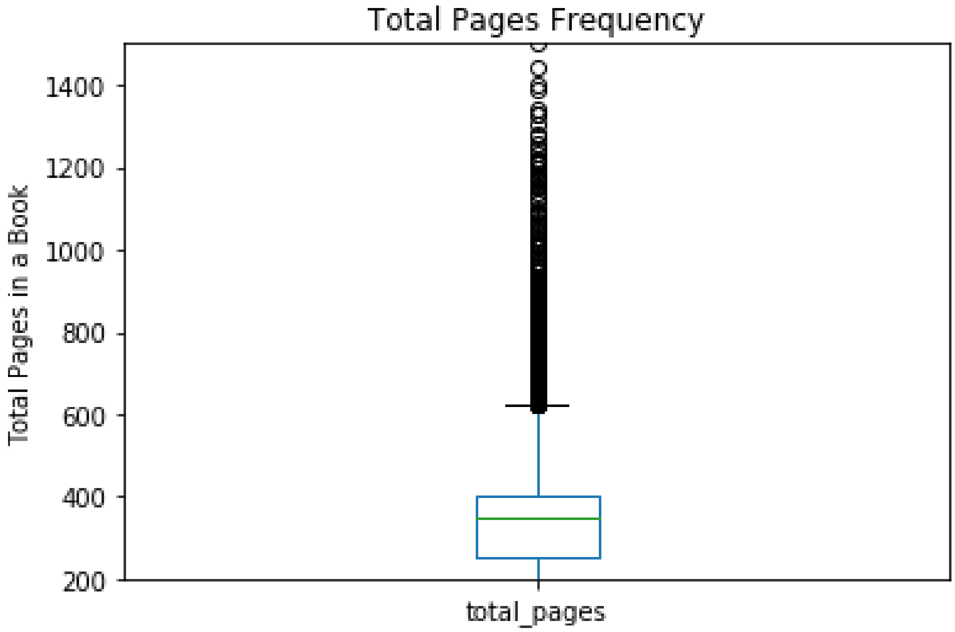
In this project, we are predicting the Average Rating of the books. The rating of the book varies from 2.75 to 4.75 with a mean value around 4.

Gender:



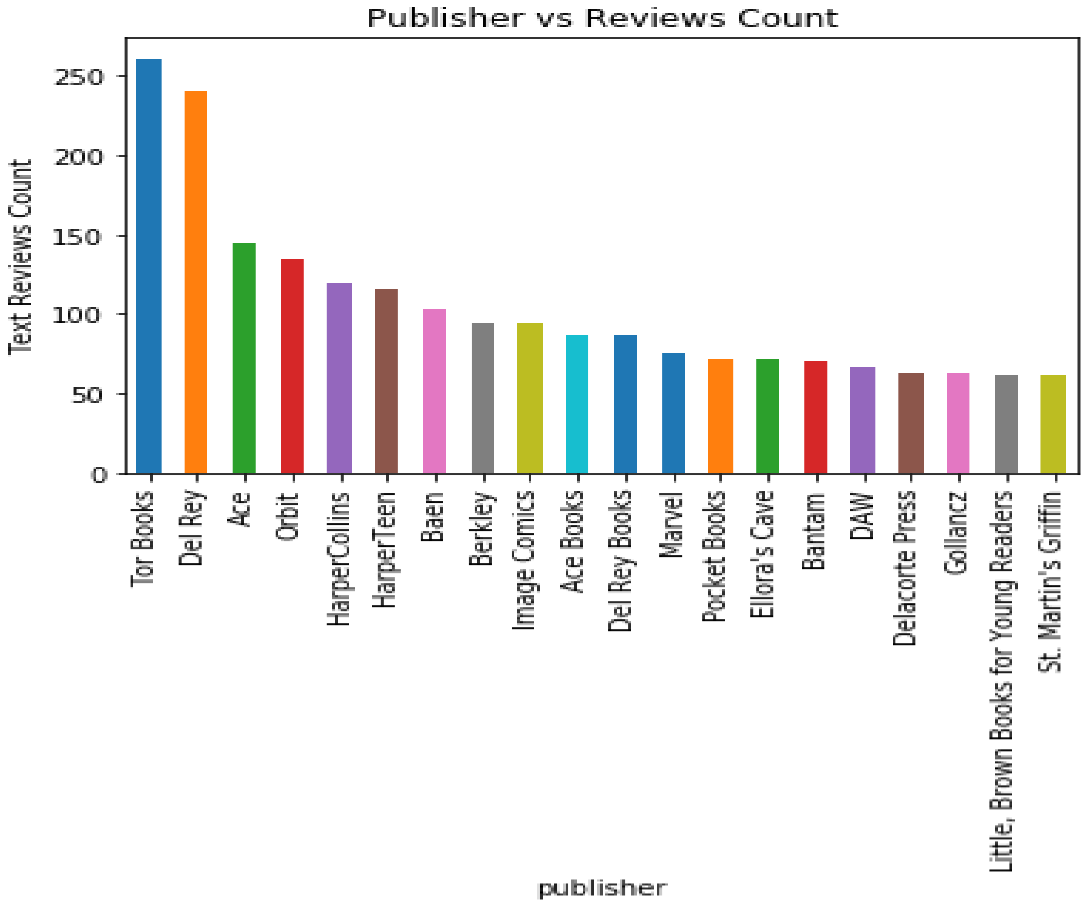
In this project, the Gender of the Author is playing an important role and it seems that there are more Male author than Female authors in the world of Science Fiction.

Total Pages:



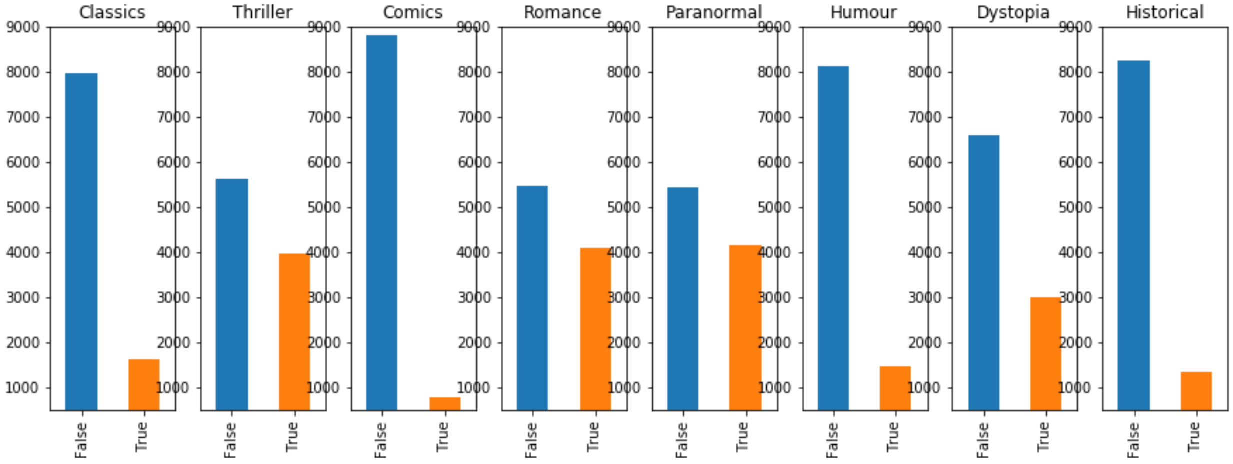
In Science Fiction, there are a couple of books which consist of many pages. But, for most of the books, the page count is at a mean of around 350.

Publishers:



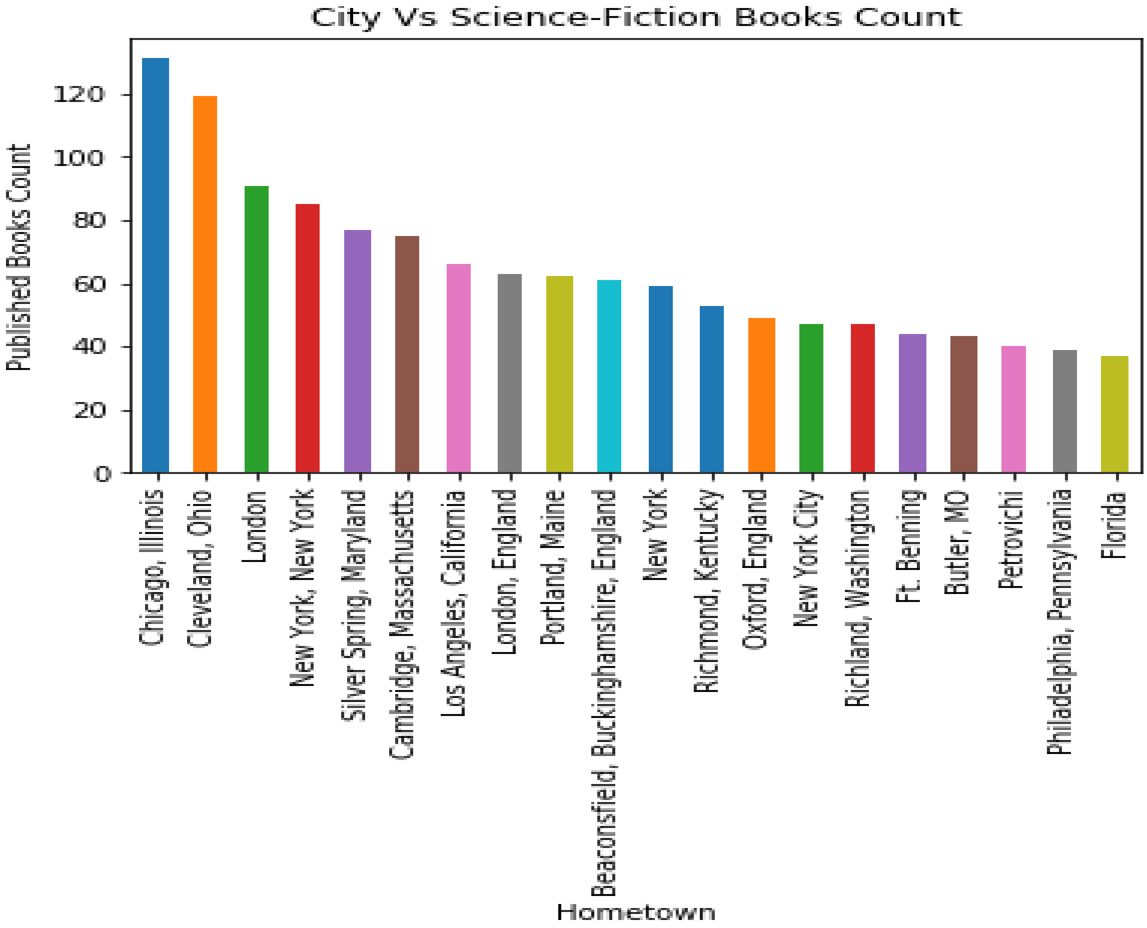
“Tor Books” is the most popular publisher in the world of Science Fiction.

Genre:



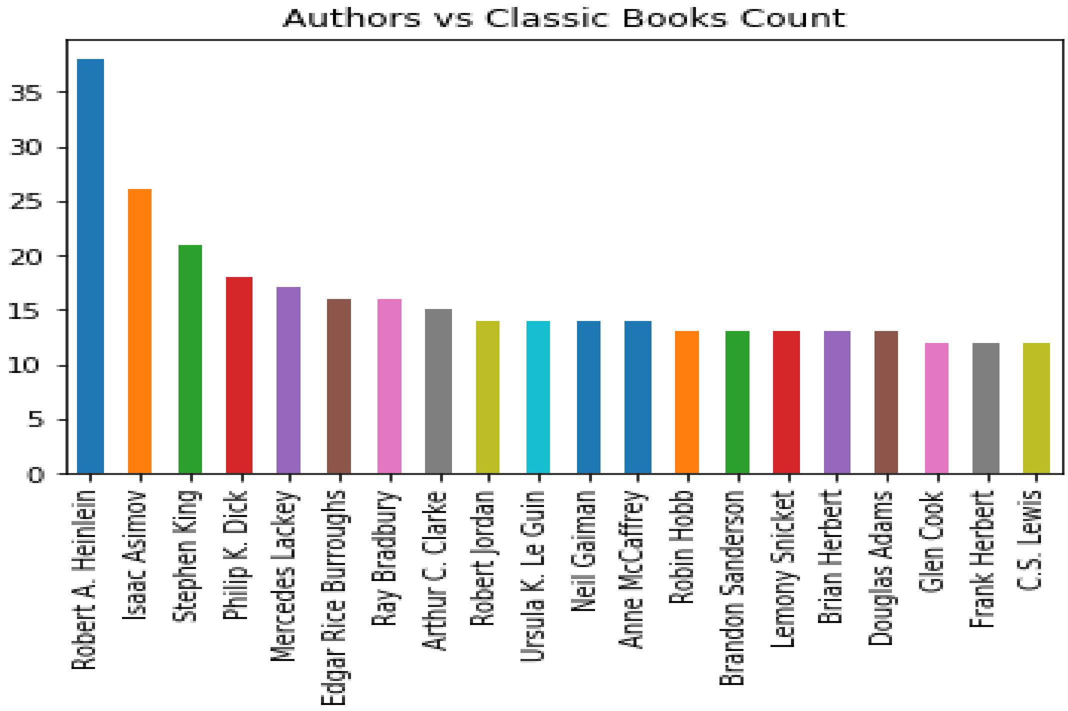
Science Fiction is a broad Genre. Under Science Fiction, there are some sub-categories and the above plot show the distribution.

City:



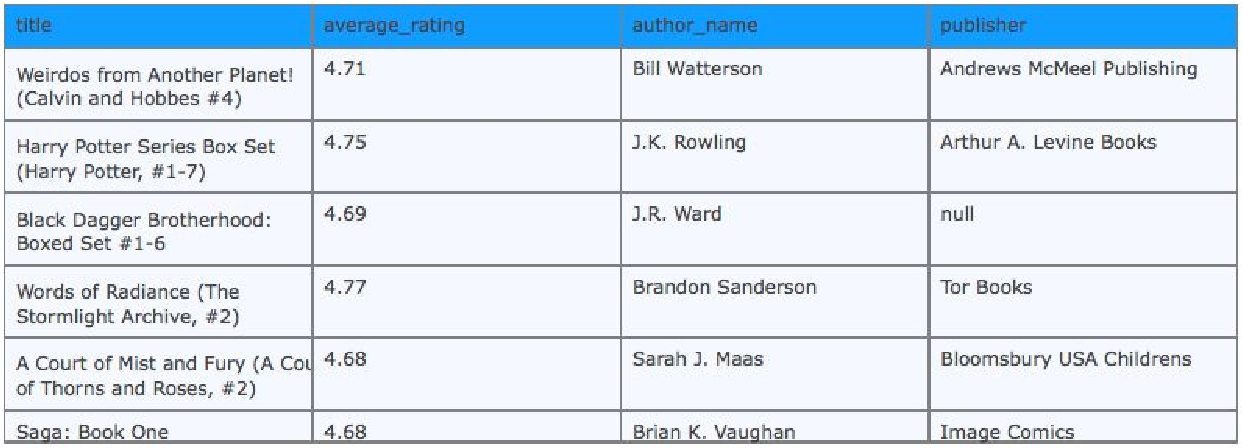
Can a City influence a creation? The above plot shows that it can! Chicago and Cleveland have given birth to most of the Science Fiction Creations.

Authors:



Authors who created most of the "Classics" in Science Fiction is "Robert A. Heinlein".

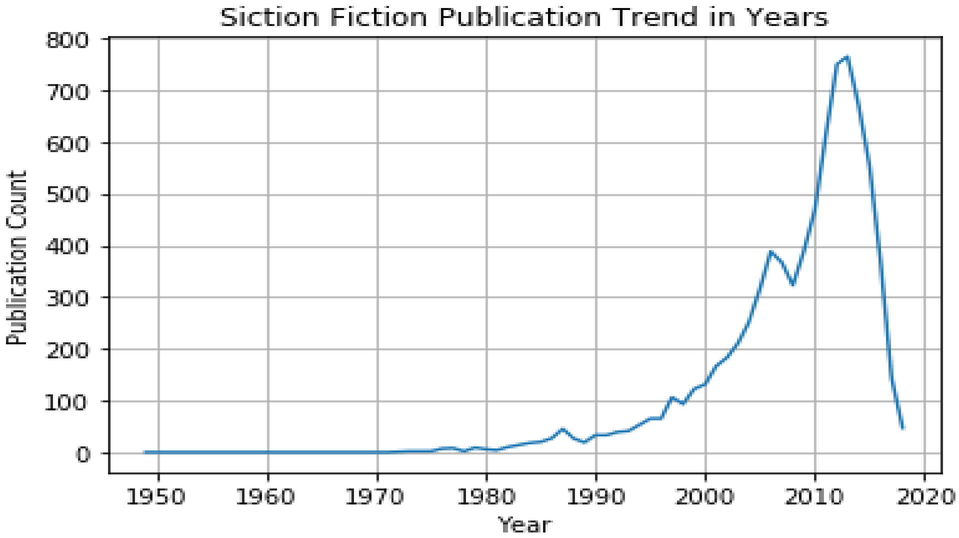
Best Reads in Science Fiction:



title	average_rating	author_name	publisher
Weirdos from Another Planet! (Calvin and Hobbes #4)	4.71	Bill Watterson	Andrews McMeel Publishing
Harry Potter Series Box Set (Harry Potter, #1-7)	4.75	J.K. Rowling	Arthur A. Levine Books
Black Dagger Brotherhood: Boxed Set #1-6	4.69	J.R. Ward	null
Words of Radiance (The Stormlight Archive, #2)	4.77	Brandon Sanderson	Tor Books
A Court of Mist and Fury (A Cou of Thorns and Roses, #2)	4.68	Sarah J. Maas	Bloomsbury USA Childrens
Saga: Book One	4.68	Brian K. Vaughan	Image Comics

The top 5 Best Reads in Science Fiction are displayed in the above chart.

Publication Year:



The above plot shows the count of Science Fiction Publication as per year. It seems that this particular genre started gaining popularity in the 21st century.

Scope Of Further Development:

In this project I have limited the genre as "Science Fiction/Fantasy". I would like to address this issue and accept the genre as a parameter from user and provide some analysis and visualizations that may help the authors to gain more ppopularity among the readers.

Conclusion:

The objective of this project is to understand and utilise the **ETL**(Extract Transfer Load) process and then finally apply some Machine Learning Algorithms for Prediction.

I would like to address this project as a Regression Problem and predict the average rating of a book.