Data Science

Assignment 3 – Books

Farimah Rashidi – 99222040

1. Introduction

The dataset is the data of 5699 books. It consists of the following columns:

- Title: book's name
- Author: book's Author
- Edition: the type of cover and the date
- Reviews: the score of the book out of 5
- Ratings: the number of reviews
- Synopsis: a short summary about the book
- Genre: the book's Genre
- BookCategory: the categories of books
- Price: book's price

As you can see, we have 9 columns.

There are no missing values in dataset.

2. Univariate Variable Analysis and Feature Engineering

Now we want to analyze each column.

Author

Let's see is there any author with multiple books?

```
Total Number of Non-Unique Authors: 2261
Total Number of Unique Authors: 3438
```

2261 authors have more than one book in the dataset. You can see the list of Authors with Multiple Rows in notebook.

Reviews and feature engineering

Reviews column shows book's score out of 5. But each row of this column is like:

3.0 out of 5 stars

We need to extract the scores to use this feature. So, we will make a new feature call Reviews_Score.

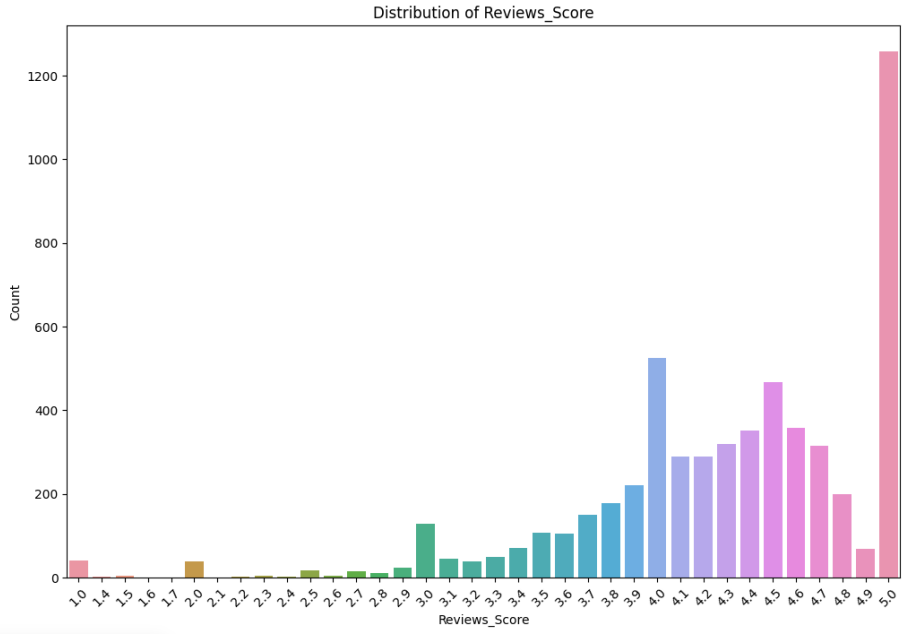
At the first we use the str.extract method with a regular expression (r'(\d+\.\d+)') to extract numeric scores with decimal points from the 'Reviews' column. The regular expression captures one or more digits before and after a decimal point. The extracted scores are then assigned to the new 'Reviews_Score' column.

After extracting the scores, we convert the 'Reviews_Score' column to floating-point numbers using the astype(float) method. This ensures that the scores are stored as numerical values with decimal precision, allowing for mathematical operations or analysis.

	Title	Author	Edition	Reviews	Ratings	Synopsis	Genre	BookCategory	Price	Reviews_Score
0	The Prisoner's Gold (The Hunters 3)	Chris Kuzneski	Paperback,– 10 Mar 2016	4.0 out of 5 stars	8 customer reviews	THE HUNTERS return in their third brilliant no...	Action & Adventure (Books)	Action & Adventure	220.00	4.0
1	Guru Dutt: A Tragedy in Three Acts	Arun Khopkar	Paperback,– 7 Nov 2012	3.9 out of 5 stars	14 customer reviews	A layered portrait of a troubled genius for wh...	Cinema & Broadcast (Books)	Biographies, Diaries & True Accounts	202.93	3.9
2	Leviathan (Penguin Classics)	Thomas Hobbes	Paperback,– 25 Feb 1982	4.8 out of 5 stars	6 customer reviews	"During the time men live without a common Pow...	International Relations	Humour	299.00	4.8
3	A Pocket Full of Rye (Miss Marple)	Agatha Christie	Paperback,– 5 Oct 2017	4.1 out of 5 stars	13 customer reviews	A handful of grain is found in the pocket of a...	Contemporary Fiction (Books)	Crime, Thriller & Mystery	180.00	4.1
4	LIFE 70 Years of Extraordinary Photography	Editors of Life	Hardcover,– 10 Oct 2006	5.0 out of 5 stars	1 customer review	For seven decades, "Life" has been thrilling t...	Photography Textbooks	Arts, Film & Photography	965.62	5.0
5	ChiRunning: A Revolutionary Approach to Effort...	Danny Dreyer	Paperback,– 5 May 2009	4.5 out of 5 stars	8 customer reviews	The revised edition of the bestselling ChiRunn...	Healthy Living & Wellness (Books)	Sports	900.00	4.5
6	Death on the Nile (Poirot)	Agatha Christie	Paperback,– 5 Oct 2017	4.4 out of 5 stars	72 customer reviews	Agatha Christie's most exotic murder mystery\n...	Crime, Thriller & Mystery (Books)	Crime, Thriller & Mystery	224.00	4.4
7	Yoga Your Home Practice Companion: A Complete ...	Sivananda Yoga Vedanta Centre	Hardcover,– Import, 1 Mar 2018	4.7 out of 5 stars	16 customer reviews	Achieve a healthy body, mental alertness, and ...	Sports Training & Coaching (Books)	Sports	836.00	4.7
8	Karmayogi: A Biography of E. Sreedharan	M S Ashokan	Paperback,– 15 Dec 2015	4.2 out of 5 stars	111 customer reviews	Karmayogi is the dramatic and inspiring story ...	Biographies & Autobiographies (Books)	Biographies, Diaries & True Accounts	130.00	4.2
9	The Iron King (The Accursed Kings, Book 1)	Maurice Druon	Paperback,– 26 Mar 2013	4.0 out of 5 stars	1 customer review	'This is the original game of thrones' George ...	Action & Adventure (Books)	Action & Adventure	695.00	4.0

As you can see, now we have a new column which shows us review score as a number.

Here is score's distribution:



We can understand that 5.0 is the most common score for books.

Editions and feature engineering

This feature has two parts. The cover type (paperback - hardcover) and the date of that edition. We can split these two parts for working with them.

So, we define two features:

Edition_Cover: the cover type of the book

Edition_Date: the release date of the edition

As a result, if the 'Edition' column contains values like "Paperback,_10 Mar 2016", this line of code would create a DataFrame where each row has two columns - the first column contains "Paperback" and the second column contains "10 Mar 2016". The column names in the resulting DataFrame would be 0 and 1.

At the first we create a new column in the DataFrame called 'Edition_Date' and assigns the values from the second column (index 1) of the split_data DataFrame to it. This assumes that the second column of split_data contains information about dates.

Then we create a new column in the DataFrame called 'Edition_Cover' and assigns the values from the first column (index 0) of the split_data DataFrame to it. This assumes that the first column of split_data contains information about the cover type or edition details.

Edition_Date

Here we convert the values in the 'Edition_Date' column to datetime objects. The errors='coerce' parameter is used to handle errors by converting problematic values to NaT (Not a Time) values. This ensures that valid dates are represented as datetime objects, while invalid or unconvertible values are replaced with NaT.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5699 entries, 0 to 5698
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Title          5699 non-null   object
 1   Author         5699 non-null   object
 2   Edition        5699 non-null   object
 3   Reviews        5699 non-null   object
 4   Ratings        5699 non-null   object
 5   Synopsis       5699 non-null   object
 6   Genre          5699 non-null   object
 7   BookCategory   5699 non-null   object
 8   Price          5699 non-null   float64
 9   Reviews_Score  5699 non-null   float64
 10  Edition_Date   4608 non-null   datetime64[ns]
 11  Edition_Cover  5699 non-null   object
dtypes: datetime64[ns](1), float64(2), object(9)
memory usage: 534.4+ KB
```

Now we can see our changes in dataset.

let’s look at first rows:

	Title	Author	Edition	Reviews	Ratings	Synopsis	Genre	BookCategory	Price	Reviews_Score	Edition_Date	Edition_Cover
0	The Prisoner's Gold (The Hunters 3)	Chris Kuzneski	Paperback,– 10 Mar 2016	4.0 out of 5 stars	8 customer reviews	THE HUNTERS return in their third brilliant no...	Action & Adventure (Books)	Action & Adventure	220.00	4.0	2016-03-10	Paperback
1	Guru Dutt: A Tragedy in Three Acts	Arun Khopkar	Paperback,– 7 Nov 2012	3.9 out of 5 stars	14 customer reviews	A layered portrait of a troubled genius for wh...	Cinema & Broadcast (Books)	Biographies, Diaries & True Accounts	202.93	3.9	2012-11-07	Paperback
2	Leviathan (Penguin Classics)	Thomas Hobbes	Paperback,– 25 Feb 1982	4.8 out of 5 stars	6 customer reviews	"During the time men live without a common Pow...	International Relations	Humour	299.00	4.8	1982-02-25	Paperback
3	A Pocket Full of Rye (Miss Marple)	Agatha Christie	Paperback,– 5 Oct 2017	4.1 out of 5 stars	13 customer reviews	A handful of grain is found in the pocket of a...	Contemporary Fiction (Books)	Crime, Thriller & Mystery	180.00	4.1	2017-10-05	Paperback
4	LIFE 70 Years of Extraordinary Photography	Editors of Life	Hardcover,– 10 Oct 2006	5.0 out of 5 stars	1 customer review	For seven decades, "Life" has been thrilling t...	Photography Textbooks	Arts, Film & Photography	965.62	5.0	2006-10-10	Hardcover
5	ChiRunning: A Revolutionary Approach to Effort...	Danny Dreyer	Paperback,– 5 May 2009	4.5 out of 5 stars	8 customer reviews	The revised edition of the bestselling ChiRunn...	Healthy Living & Wellness (Books)	Sports	900.00	4.5	2009-05-05	Paperback
6	Death on the Nile (Poirot)	Agatha Christie	Paperback,– 5 Oct 2017	4.4 out of 5 stars	72 customer reviews	Agatha Christie's most exotic murder mystery\n...	Crime, Thriller & Mystery (Books)	Crime, Thriller & Mystery	224.00	4.4	2017-10-05	Paperback
7	Yoga Your Home Practice Companion: A Complete ...	Sivananda Yoga Vedanta Centre	Hardcover,– Import, 1 Mar 2018	4.7 out of 5 stars	16 customer reviews	Achieve a healthy body, mental alertness, and ...	Sports Training & Coaching (Books)	Sports	836.00	4.7	NaT	Hardcover
8	Karmayogi: A Biography of E. Sreedharan	M S Ashokan	Paperback,– 15 Dec 2015	4.2 out of 5 stars	111 customer reviews	Karmayogi is the dramatic and inspiring story ...	Biographies & Autobiographies (Books)	Biographies, Diaries & True Accounts	130.00	4.2	2015-12-15	Paperback
9	The Iron King (The Accursed Kings, Book 1)	Maurice Druon	Paperback,– 26 Mar 2013	4.0 out of 5 stars	1 customer review	'This is the original game of thrones' George ...	Action & Adventure (Books)	Action & Adventure	695.00	4.0	2013-03-26	Paperback

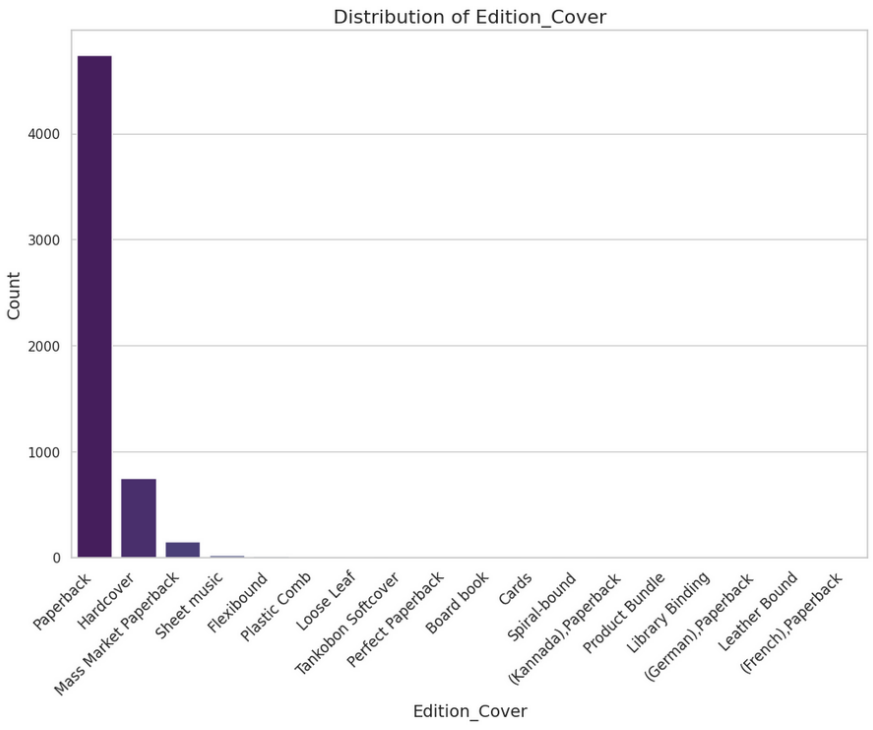
Edition_Cover

Let’s see unique values of this column:

```
Unique Values in Edition_Cover ['Paperback' 'Hardcover' 'Mass Market Paperback' 'Sheet music'
 'Flexibound' 'Plastic Comb' 'Loose Leaf' 'Tankobon Softcover'
 'Perfect Paperback' 'Board book' 'Cards' 'Spiral-bound'
 '(Kannada),Paperback' 'Product Bundle' 'Library Binding'
 '(German),Paperback' 'Leather Bound' '(French),Paperback']
```

As you can see, we have 18 unique values for Edition_Cover.

Now we can see the distribution of this feature:



We can understand from plot that Paperback is the most common Edition Cover according to the dataset. After Paperback, Hardcover is the most common Cover.

Here is a numerical report of Edition_Cover distribution:

```
Edition_Cover
Paperback                4741
Hardcover                 750
Mass Market Paperback     148
Sheet music                22
Flexibound                 14
Cards                       7
Spiral-bound                4
Loose Leaf                  2
Tankobon Softcover          2
Product Bundle              1
Leather Bound               1
(German),Paperback          1
Library Binding             1
Board book                  1
(Kannada),Paperback         1
Perfect Paperback           1
Plastic Comb                1
(French),Paperback          1
Name: count, dtype: int64
```

Edition_Date

This feature shows the date that the book was released.

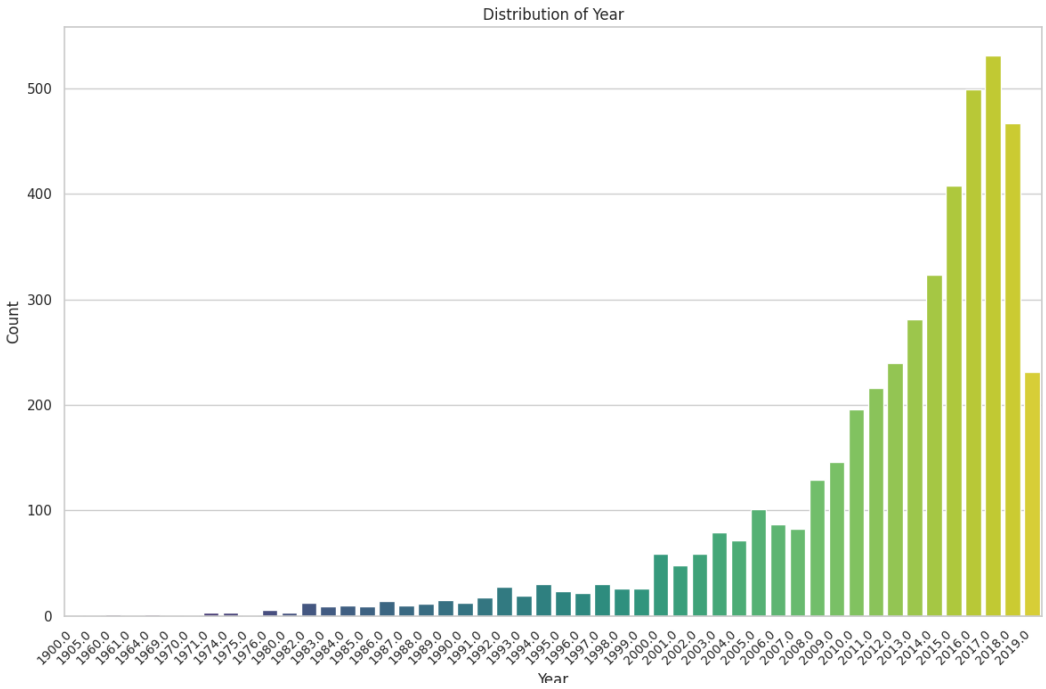
At the first I make a new feature called year, without using to_datetime function, because this function can make a lot of null values which is no good for us because we want to use MSE at the end and MSE cannot be calculated by null values in dataset.

Then, we transform the 'Edition_Date' column into a datetime format and extract the corresponding year and month information, storing it in a new 'Year_Month' column. This transformation can be useful for time-based analysis and visualizations.

Here are our columns until now:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5699 entries, 0 to 5698
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Title          5699 non-null   object
 1   Author         5699 non-null   object
 2   Edition        5699 non-null   object
 3   Reviews        5699 non-null   object
 4   Ratings        5699 non-null   object
 5   Synopsis       5699 non-null   object
 6   Genre          5699 non-null   object
 7   BookCategory   5699 non-null   object
 8   Price          5699 non-null   float64
 9   Reviews_Score  5699 non-null   float64
 10  Edition_Date   4608 non-null   datetime64[ns]
 11  Edition_Cover  5699 non-null   object
 12  Year           5699 non-null   int64
 13  Year_Month     4608 non-null   period[M]
dtypes: datetime64[ns](1), float64(2), int64(1), object(9), period[M](1)
memory usage: 623.5+ KB
```

To better understand the data, we show a plot of Year distribution:



It can be understood that most of the books were released in 2016 and 2017.

Now we do not need Edition column anymore. So, we drop it.

BookCategory and Genre

At the first, let's look at the number of unique values of BookCategory and Genre features.

```
Number of unique values of Genre: 335
Number of unique values of BookCategory: 11
```

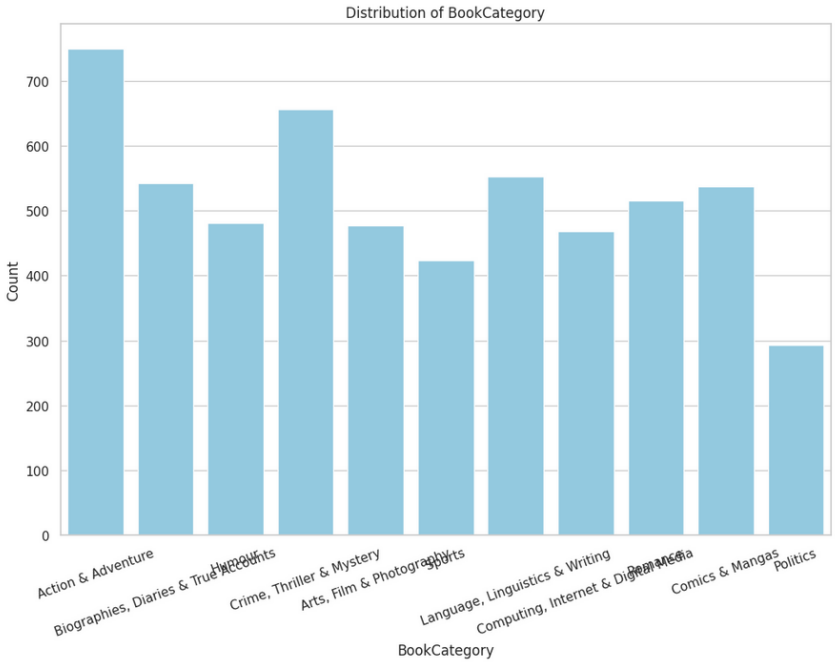
Unique values of BookCategory:

```
['Action & Adventure' 'Biographies, Diaries & True Accounts' 'Humour'
 'Crime, Thriller & Mystery' 'Arts, Film & Photography' 'Sports'
 'Language, Linguistics & Writing' 'Computing, Internet & Digital Media'
 'Romance' 'Comics & Mangas' 'Politics']
```

Also, you can see all unique values of Genre in notebook.

If you see some rows. You can see that both columns are describing one thing. So, we can drop one of them. I decided to drop Genre feature.

Let’s see the difference of BookCategory:



It appears that ‘Action & Adventure’ and ‘Crime, Thriller & Mystery’ are the most common book categories.

Ratings

This feature shows the number of reviews. I decided to extract the number of reviews for easy working. As you can see in dataset, the format of this column is like:

10 customer reviews

We want to extract just the numerical part of that.

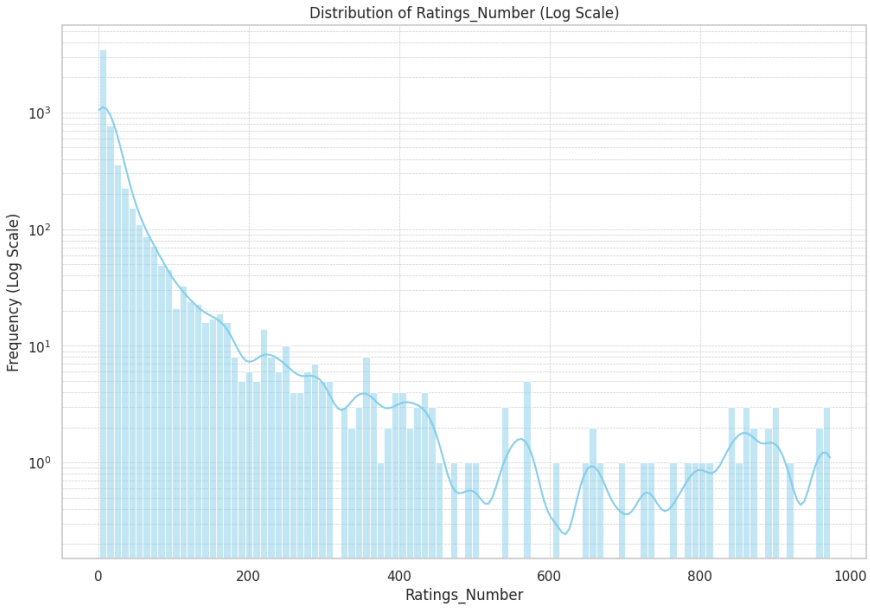
At the first we use the str.extract method with a regular expression (r'(\d+)') to extract the numeric part (digits) from the 'Ratings' column. The regular expression captures one or more digits. The extracted numeric ratings are then assigned to the new 'Ratings_Number' column.

we don't need Ratings column anymore, so we drop it.

You can see changes in columns here:

	Title	Author	Reviews	Synopsis	BookCategory	Price	Reviews_Score	Edition_Date	Edition_Cover	Year_Month	Year	Ratings_Number
0	The Prisoner's Gold (The Hunters 3)	Chris Kuzneski	4.0 out of 5 stars	THE HUNTERS return in their third brilliant no...	Action & Adventure	220.00	4.0	2016-03-10	Paperback	2016-03	2016.0	8
1	Guru Dutt: A Tragedy in Three Acts	Arun Khopkar	3.9 out of 5 stars	A layered portrait of a troubled genius for wh...	Biographies, Diaries & True Accounts	202.93	3.9	2012-11-07	Paperback	2012-11	2012.0	14
2	Leviathan (Penguin Classics)	Thomas Hobbes	4.8 out of 5 stars	"During the time men live without a common Pow...	Humour	299.00	4.8	1982-02-25	Paperback	1982-02	1982.0	6
3	A Pocket Full of Rye (Miss Marple)	Agatha Christie	4.1 out of 5 stars	A handful of grain is found in the pocket of a...	Crime, Thriller & Mystery	180.00	4.1	2017-10-05	Paperback	2017-10	2017.0	13
4	LIFE 70 Years of Extraordinary Photography	Editors of Life	5.0 out of 5 stars	For seven decades, "Life" has been thrilling t...	Arts, Film & Photography	965.62	5.0	2006-10-10	Hardcover	2006-10	2006.0	1

distribution of Ratings_Number:



The plot is scaled on logarithmic scale. Most of the data is between 0 and 300.

New feature with rating and score

We can make a new feature using "Rating Number" and "Reviews_Score". We call it ‘popularuity’

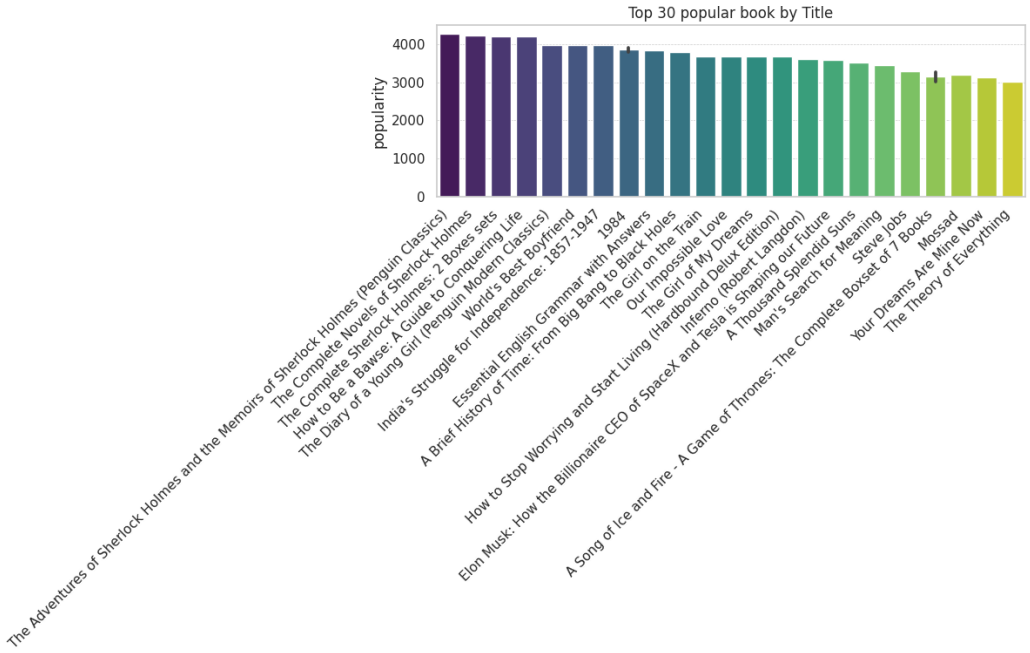
'popularity’= 'Ratings_Number' * 'Reviews_Score'

Here is the result:

	Ratings_Number	Reviews_Score	popularity
0	8	4.0	32.0
1	14	3.9	54.6
2	6	4.8	28.8
3	13	4.1	53.3
4	1	5.0	5.0
...	...	...	...
5694	9	4.9	44.1
5695	2	4.1	8.2
5696	28	4.1	114.8
5697	1	1.0	1.0
5698	7	4.5	31.5

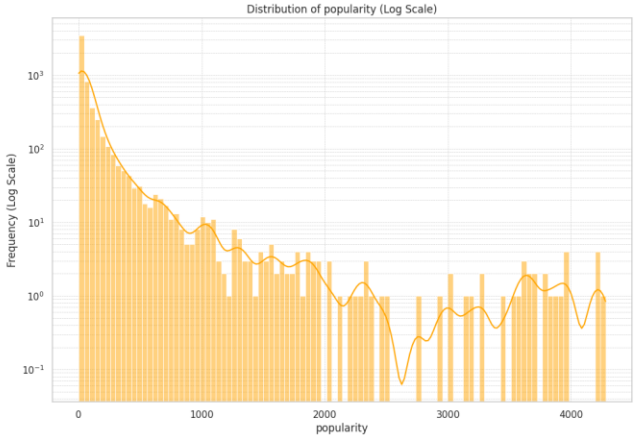
[5699 rows x 3 columns]

Now we can see the most popular books:



The most popular book is The Adventure of Sherlock Holmes.

Now let's look at distribution:



3. Basic Data Analysis

We analyzed each column. Now, let's compare them with each other and use them to make speculations.

the most popular book

let’s see which book is the most popular (has the highest ratings and reviews score).

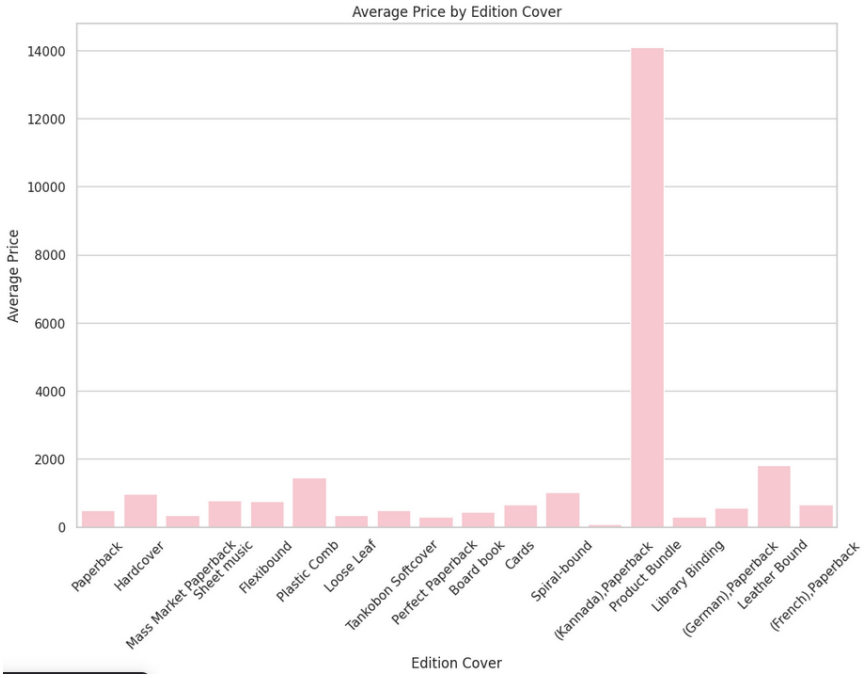
We calculate the product of Reviews_Score and Ratings_Number to find the most popular book (popularity feature that we made before):

```
The most popular book: The Adventures of Sherlock Holmes and the Memoirs of Sherlock Holmes (Penguin Classics), most_popular_book: 4281.200000000001
```

The results show that The Adventures of Sherlock Holmes and the Memoirs of Sherlock Holmes (Penguin Classics) has the highest popularity among all the other books in this dataset.

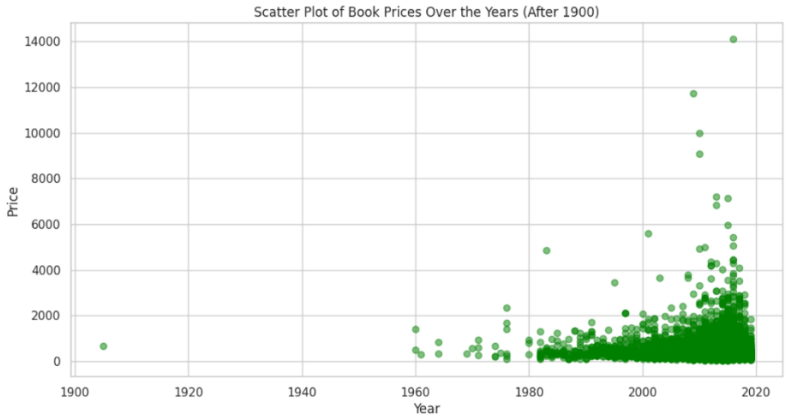
most expensive covers

let’s see which Covers are the most expensive and cheapest.



Product Bundle is the most expensive cover and Paperback is the cheapest cover.

correlation between the prices over the years

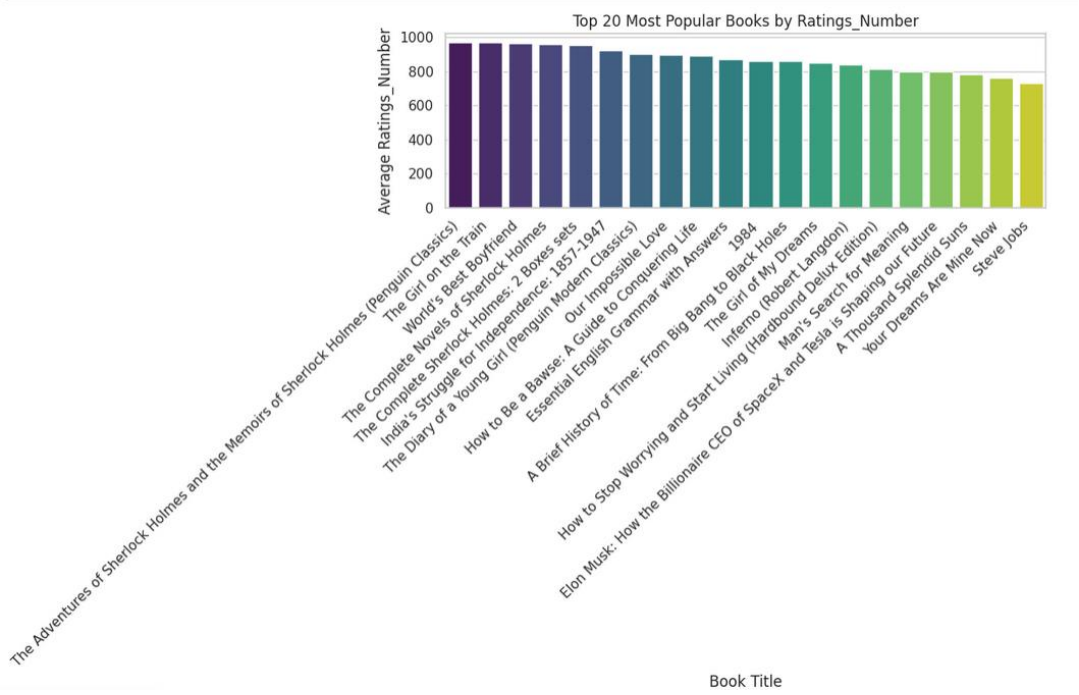


We can see that the prices are increasing. 2020 has the most expensive prices.

Most popular Books

Let's see what the most popular book is according to Ratings_Number:

```
Most Popular Book:
Title            The Adventures of Sherlock Holmes and the Memo...
Ratings_Number                                                 973
Year                                                        2001.0
BookCategory                                    Action & Adventure
Name: 2550, dtype: object
```



You can see top 20 most popular books by Ratings_Number in this plot.

popular authors

For finding the most popular authors, we can use Ratings_Number.

Here are the most popular authors:

```
Author
Dan Brown               5104
Durjoy Datta            4142
Novoneel Chakraborty    2638
Agatha Christie         2637
George Orwell           2606
Sudeep Nagarkar         2118
Sidney Sheldon          2104
Stephen Hawking         2096
Arthur Conan Doyle      2076
Anne Frank              1804
Name: Ratings_Number, dtype: int16
```

As you can see, the most popular author is Dan Brown.

let's see my favorite authors books:

	Author	Title
326	Novoneel Chakraborty	Black Suits You
1250	Novoneel Chakraborty	Marry Me, Stranger
1259	Novoneel Chakraborty	EX...A Twisted love Story
1478	Novoneel Chakraborty	Marry Me, Stranger
1739	Novoneel Chakraborty	Forever is True
2985	Novoneel Chakraborty	Forever is a Lie
3584	Novoneel Chakraborty	Cheaters
3765	Novoneel Chakraborty	Half Torn Hearts
4280	Novoneel Chakraborty	Forget Me Not, Stranger
4510	Novoneel Chakraborty	All Yours, Stranger

4. Feature Transformation

Now we want to do some things to prepare our data for using in machine learning model.

One-hot encoding on Edition_Cover and BookCategory

For Edition_cover and BookCategory columns, we use one-hot encode categorical columns. It creates binary columns for each category in the specified columns ('Edition_Cover' and 'BookCategory'), indicating the presence or absence of each category for each row.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5699 entries, 0 to 5698
Data columns (total 36 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   Title                                         5699 non-null   object
 1   Author                                        5699 non-null   object
 2   Synopsis                                      5699 non-null   object
 3   Price                                         5699 non-null   float64
 4   Reviews_Score                                 5699 non-null   float64
 5   Year                                          4608 non-null   float64
 6   Ratings_Number                                5699 non-null   int16
 7   Edition_(French),Paperback                    5699 non-null   bool
 8   Edition_(German),Paperback                    5699 non-null   bool
 9   Edition_(Kannada),Paperback                   5699 non-null   bool
 10  Edition_Board book                            5699 non-null   bool
 11  Edition_Cards                                 5699 non-null   bool
 12  Edition_Flexibound                            5699 non-null   bool
 13  Edition_Hardcover                             5699 non-null   bool
 14  Edition_Leather Bound                         5699 non-null   bool
 15  Edition_Library Binding                       5699 non-null   bool
 16  Edition_Loose Leaf                            5699 non-null   bool
 17  Edition_Mass Market Paperback                 5699 non-null   bool
 18  Edition_Paperback                             5699 non-null   bool
 19  Edition_Perfect Paperback                     5699 non-null   bool
 20  Edition_Plastic Comb                          5699 non-null   bool
 21  Edition_Product Bundle                        5699 non-null   bool
 22  Edition_Sheet music                           5699 non-null   bool
 23  Edition_Spiral-bound                          5699 non-null   bool
 24  Edition_Tankobon Softcover                    5699 non-null   bool
 25  Category_Action & Adventure                   5699 non-null   bool
 26  Category_Arts, Film & Photography             5699 non-null   bool
 27  Category_Biographies, Diaries & True Accounts 5699 non-null   bool
 28  Category_Comics & Mangas                      5699 non-null   bool
 29  Category_Computing, Internet & Digital Media  5699 non-null   bool
 30  Category_Crime, Thriller & Mystery            5699 non-null   bool
 31  Category_Humour                               5699 non-null   bool
 32  Category_Language, Linguistics & Writing      5699 non-null   bool
 33  Category_Politics                             5699 non-null   bool
 34  Category_Romance                              5699 non-null   bool
 35  Category_Sports                               5699 non-null   bool
dtypes: bool(29), float64(3), int16(1), object(3)
memory usage: 439.8+ KB
```

The columns related to edition covers and book categories have been one-hot encoded, resulting in boolean columns indicating the presence or absence of each category or edition type for each entry.

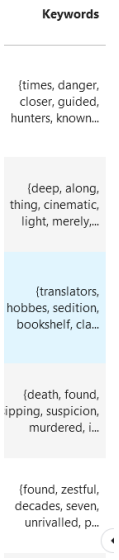
The boolean columns resulting from one-hot encoding have a dtype of bool, which is an efficient representation for binary values.

Extract keywords from Synopsis (tokenize Synopsis)

In this section, I've crafted a function named `extracting_keywords` to process textual synopses. Specifically, it tokenizes the input text, converts the words to lowercase, removes common stop words, and filters out non-alphanumeric words. The function then returns a set of unique words present in the processed synopsis.

Subsequently, I've applied this function to the 'Synopsis' column. It creates a new column called 'Keywords.' Each entry in this 'Keywords' column now contains a set of unique words derived from the corresponding synopsis. This approach aims to condense and highlight essential terms, potentially facilitating further analysis or categorization based on the distinctive words present in each synopsis.

Here you can see the result for first few rows:



Text Vectorization Strategies for Feature Extraction from Keywords in the Analysis (TF-IDF)

In this phase of the analysis, I explored various text vectorization techniques to extract meaningful features from the 'Keywords' column within the dataset. The initial step involved utilizing **TF-IDF** (Term Frequency-Inverse Document Frequency) vectorization, resulting in a new DataFrame featuring TF-IDF features. This approach captures the significance of words in each synopsis while considering their frequency across all synopses.

Subsequently, I employed Count Vectorization, generating a matrix representing the frequency of each word in the 'Keywords' column. This technique provides a straightforward count-based representation, offering insights into the prominence of individual terms.

Lastly, I experimented with Feature Hashing, a method that transforms the 'Keywords' column into a fixed-size numerical representation. This approach is particularly useful for scenarios where the dimensionality of the data needs to be controlled.

Each of these vectorization techniques serves a unique purpose, providing diverse ways to represent textual data.

Moreover, the dataset includes key book information such as title, author, synopsis, price, reviews score, year of publication, and edition details. These features, alongside the newly derived textual representations, lay a solid foundation for subsequent analyses and modeling efforts.

Count Encoding

I applied count encoding to the 'Keywords' feature, where each entry consists of lists of words. The objective was to convert these lists into space-separated strings and then perform count encoding. However, this approach resulted in an undesirable increase in the dataset's dimensionality, like the previous method. Consequently, this method may not be optimal for addressing the challenges associated with dimensionality.

Enhancing Author Representation through Multi-Label Binarization in the Dataset

In this phase of my analysis, I took a closer look at the 'Author' column in the dataset, aiming to enhance its representation for subsequent modeling and analysis. Initially, I employed a lambda function in combination with the `apply` method to split the author names, creating a list of authors for each book by considering comma separation.

Subsequently, to effectively handle the multi-label nature of authorship, I transformed the author lists into a binary matrix, where each author's name is treated as a separate binary feature. The resulting one-hot encoded matrix, stored in the dataframe 'author_one_hot', was then seamlessly integrated with the original dataset. This augmentation not only preserves the integrity of the original author information but also introduces a structured binary representation.

We convert Author to a list and one-hot encode it to convert it to a numerical feature.

Delete some features:

After converting these features to numerical ones, now we can drop the Synopsis, Title and Author from dataset. So, all strings from the model will be removed.

5. Modeling

Preparing dataset for machine learning model

In this section, I prepared the dataset for machine learning model by defining the target column, 'Price,' and splitting the data into training and testing sets. The independent variables (features) are denoted as 'X,' and the dependent variable (target) is represented by 'y.' The dataset was divided into training and testing sets, with a test size of 20% and a random seed of 34 for reproducibility. The resulting shapes of the training and testing sets:

```
X_train shape: (4559, 3778)
y_train shape: (4559,)
X_test shape: (1140, 3778)
y_test shape: (1140,)
```

Now we can predict book prices based on the available features.

Implement model

Now we can use our machine learning model.

After all, I calculated MSE which is:

```
Train mse is: 0.8655436322517444 // Test mse is: 257169.36978007274
```

6. Post Processing

Now after all these steps, I want to do some post processing to get valuable insights and identifying the most important features. I want to do "Feature Importance Analysis" as post processing.

So, I decided to choose some features of my dataset and remove them and test model to see their effects on result and MSE.

At the first I selected these 2 features: 'Price', 'Year'.

MSE is: 345031.

This model with these 2 features leads to higher MSE.

Then I decided to choose 'Price', 'Year', 'Reviews_Score', 'Ratings_Number' as features and run model.

Now the MSE is: 331667. It seems that this is higher than all features model MSE, which is not good.

Then I run model using just these 3 features: 'Price', 'Edition_Cover', 'BookCategory'

In this case, MSE is: 280734. It seems that this one is better than othe models.

So, I think 'Price', 'Edition_Cover', 'BookCategory' are the most important and effective features.