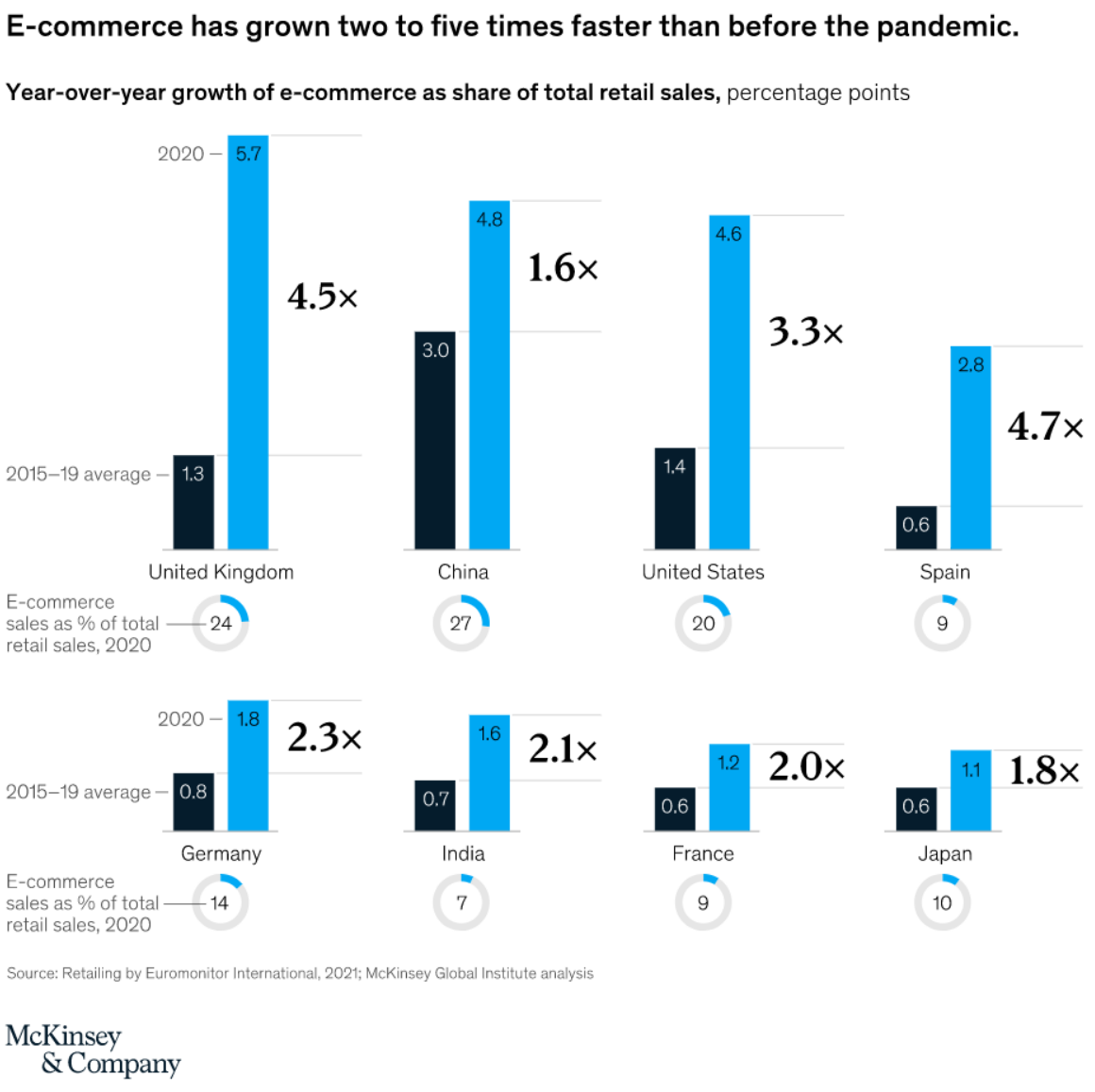
Collaborative filtering system

Data Science - 99222098 - January of 2024

by Farzam Manafzadeh

Introduction

Over the past two decades, there has been a significant shift from purchasing groceries at local hypermarts to online e-commerce stores. The convenience of shopping from home has led many customers to prefer online platforms. The COVID-19 pandemic in 2020 further accelerated this trend, causing a decline in physical hypermart sales and a surge in e-commerce demand. McKinsey has a detailed report on this shift in here.



> The vast range of products available on e-commerce platforms, from toothbrushes to cars, highlights the competitiveness of the space. To thrive, it's crucial for these platforms to understand customer preferences and keep them engaged. Recommendation systems play a key role in achieving this by suggesting complementary items. An example is if you're buying bread, the system might recommend butter or milk.

Importance of Recommendation Systems

Netflix: According to a recent study by McKinsey, Netflix uses personalized recommendations, and it is responsible for 80 percent of the content streamed. This has allowed Netflix to earn 1 billion dollars in a single year. Netflix doesn't spend much on marketing but on recommendations to improve customer retention.

Amazon: Similarly, 35% of the Amazon website's sales come from personalized recommendations. So, even Amazon does not spend much on marketing. However, it is still able to retain customers using personalized recommendations.

Spotify: Every week, Spotify generates a new customized playlist for each subscriber called "Discover Weekly" which is a personalized list of 30 songs based on users' unique music tastes. Their acquisition of Echo Nest, a music intelligence and data-analytics startup, enable them to create a music recommendation engine that uses three different types of recommendation models:

- **Collaborative filtering:** Filtering songs by comparing users' historical listening data with other users' listening history.
- **Natural language processing:** Scraping the internet for information about specific artists and songs. Each artist or song is then assigned a dynamic list of top terms that changes daily and is weighted by relevance. The engine then determines whether two pieces of music or artists are similar.
- **Audio file analysis:** The algorithm each individual audio file's characteristics, including tempo, loudness, key, and time signature, and makes recommendations accordingly.

Type of Recommendation System

Popularity Based: Recommends the most popular or most selling products. E.g. Trending videos on Youtube.

Advantages:

- We just need the product information. We don't need customer preference data.
- We can use this approach for the cold start problem.

Disadvantage: The recommendations are similar for all customers and not personalized.

Content-based: It works on similarities between products. First, we must create a vector representing all the product features. Then, we calculate the similarity between those vectors using methods like:

> 1. Euclidean Distance
> 2. Manhattan Distance
> 3. Jaccard Distance
> 4. Cosine Distance (or cosine similarity – we will be using this metric in our article)

For example, if a set of users likes action movies by Jackie Chan, this algorithm may recommend the movies having the below characteristics.

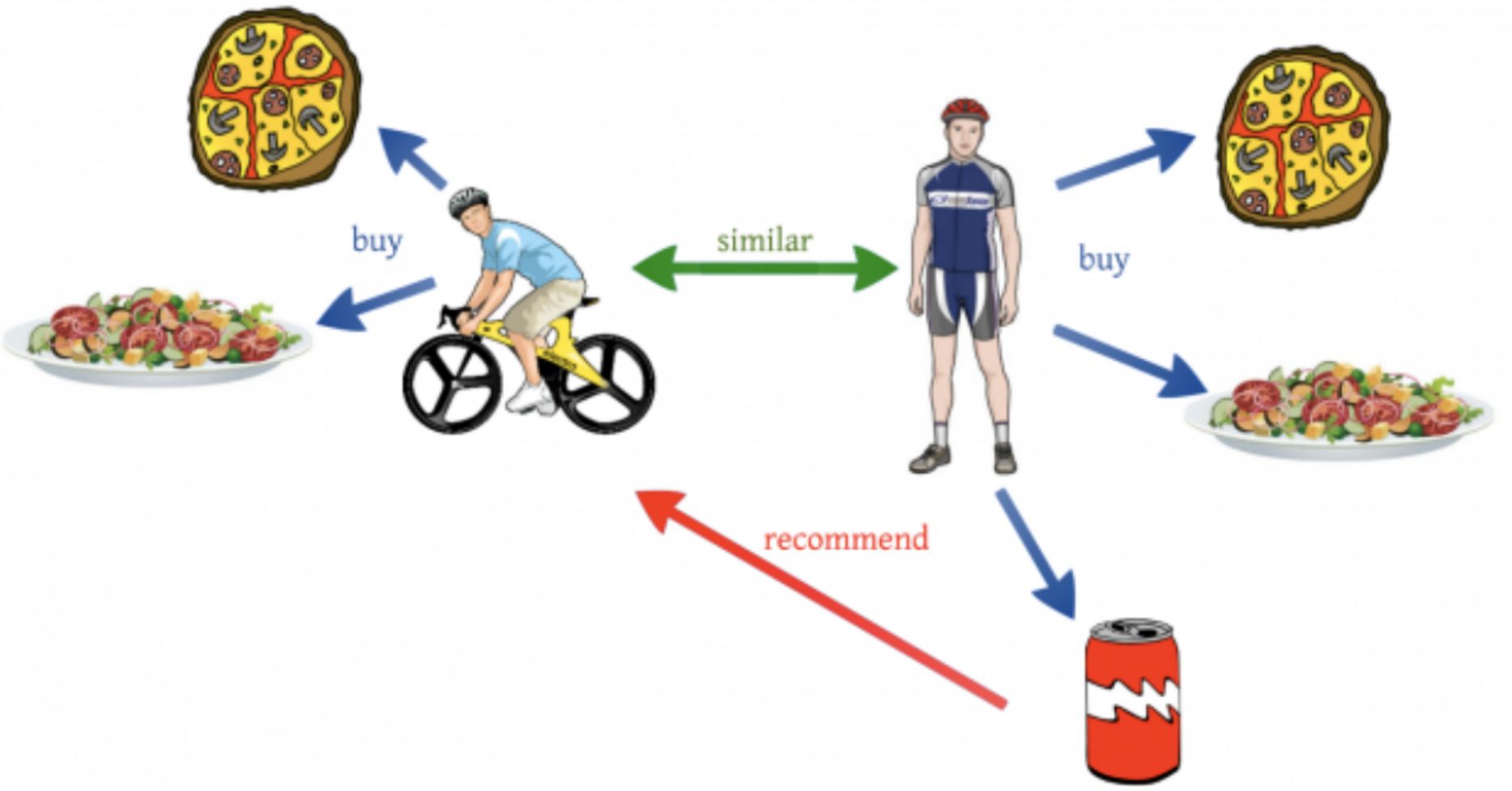
- Having a genre of action
- Having Jackie Chan in the cast.

Advantage: Quick to implement

Disadvantages:

- The recommendations are similar for a set of customers.
- They are slightly personalized based on content.

Collaborative-based: It works on similarities between users. If there are 2 users.

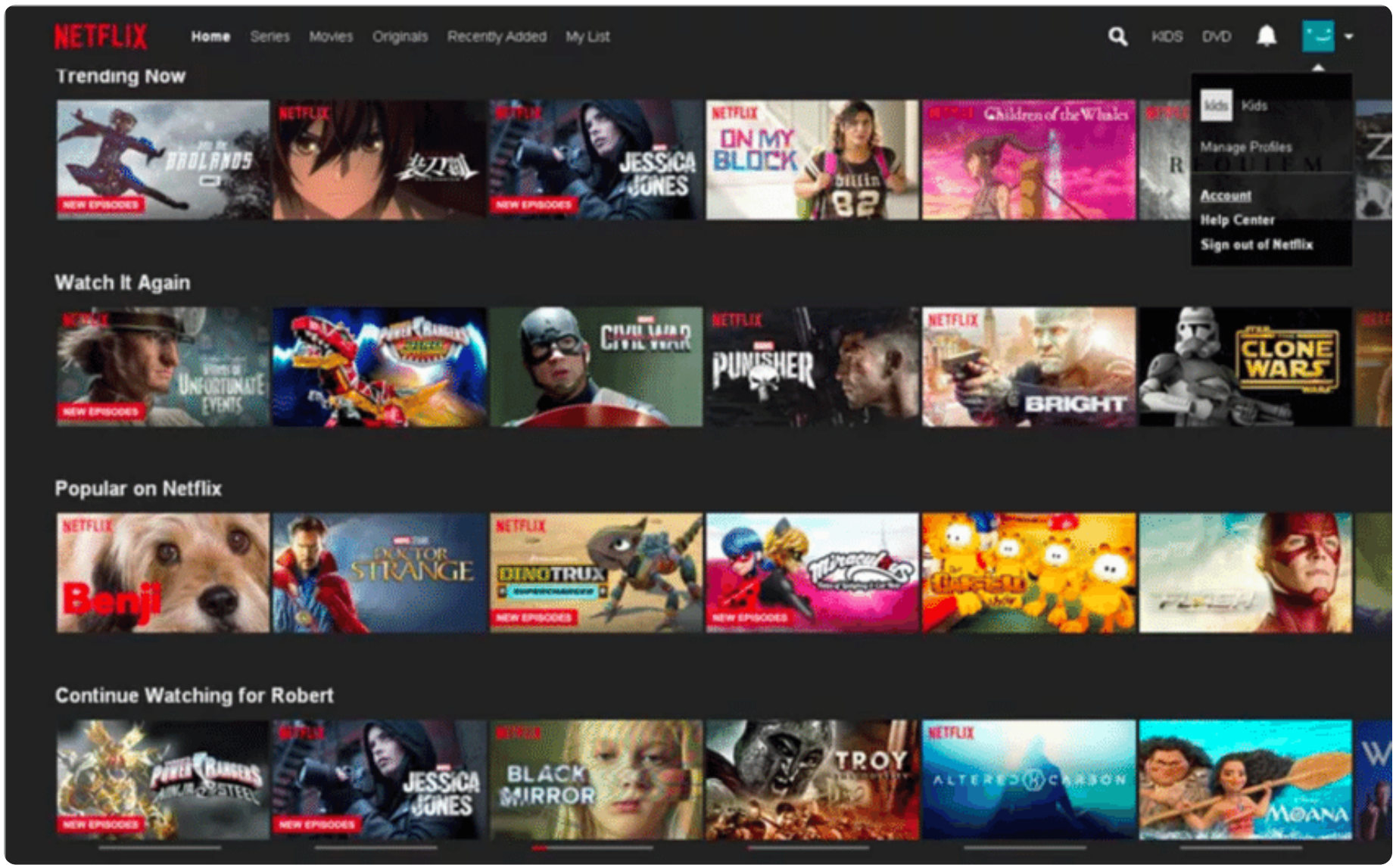


In the illustration above, there are 2 users with similar taste preferences. Both of them liked pie and protein salad and looked like fitness enthusiasts, so they are similar. Now, the user on the right liked a can of energy drink, so we recommend the same energy drink to the user on the left.

Advantages: It gives personalized recommendations for each user basis their historic preferences.

Disadvantage: The algorithm needs a lot of historical data to train, and the model's accuracy increases with increased data.

Hybrid: Here, we combine all the above 3 recommender systems



For example, the Netflix homepage has several strips for recommending content to its subscriber.

- It has strips like
- Trending now
- Popular on Netflix
- Watch it again
- Personalized recommendation for the subscriber

Amazon

Amazon.com is one of the largest electronic commerce and cloud computing companies.

Just a few **Amazon** related facts

- They lost $4.8 million in August 2013, when their website went down for 40 mins.
- They hold the patent on 1-Click buying, and licenses it to Apple.
- Their Phoenix fulfilment centre is a massive 1.2 million square feet.

Amazon relies heavily on a Recommendation engine that reviews customer ratings and purchase history to recommend items and improve sales.

Data for Recommendation System

using a publicly available dataset from Kaggle. It can be downloaded from here :

www.kaggle.com

Amazon - Ratings (Beauty Products)

2+ Million data points related to Beauty products sold on Amaon.com

About this file. This dataset contains the below 10 columns:

1. **UserId (object):** Unique identifier for users who provided ratings.
2. **ProductId (object):** Unique identifier for the products being rated.
3. **Rating (float64):** The rating given by users for the corresponding product. It's a numerical value, potentially ranging from a minimum to a maximum value (e.g., 1 to 5).
4. **Timestamp (int64):** The timestamp when the rating was provided, represented in integer format.

EDA (Exploratory Data Analysis) and Data Cleaning

Check for Missing data: If the data is null or missing, cannot be use for data analysis. So, let's look at the data distribution.

Missing count:

```
UserId      0
ProductId   0
Rating      0
Timestamp   0
dtype: int64
```

Findings from the EDA:

There are no missing values in the dataset.

The absence of missing values in the columns (`UserId`, `ProductId`, `Rating`, `Timestamp`) indicates a clean dataset. This completeness is crucial for accurate analysis and ensures that every entry is considered in the development of the recommendation system.

Understanding Data Types of Columns

```
UserId     object
ProductId  object
Rating     float64
Timestamp  int64
```

Findings from this step:

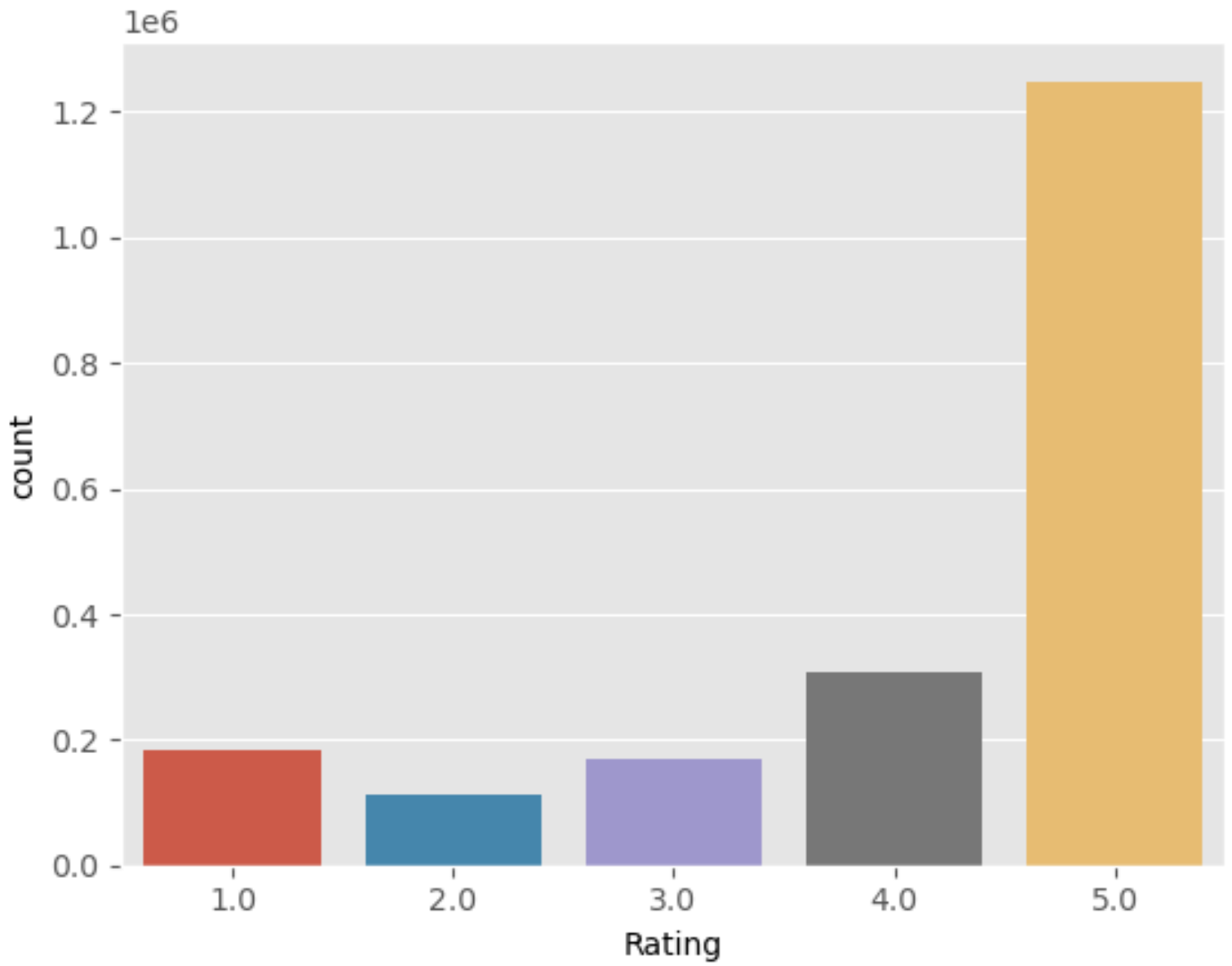
The dataset consists of the following columns with their respective data types:

- `UserId`: Object
- `ProductId`: Object
- `Rating`: Float64
- `Timestamp`: Int64

The absence of missing values, along with the understanding of column data types, ensures a robust foundation for building and analyzing a recommendation system.

Ratings Analysis

Since this is a numeric column, Let’s look at the Plot of this column.



Most of the products have a higher rating.

The distribution of ratings is as follows:

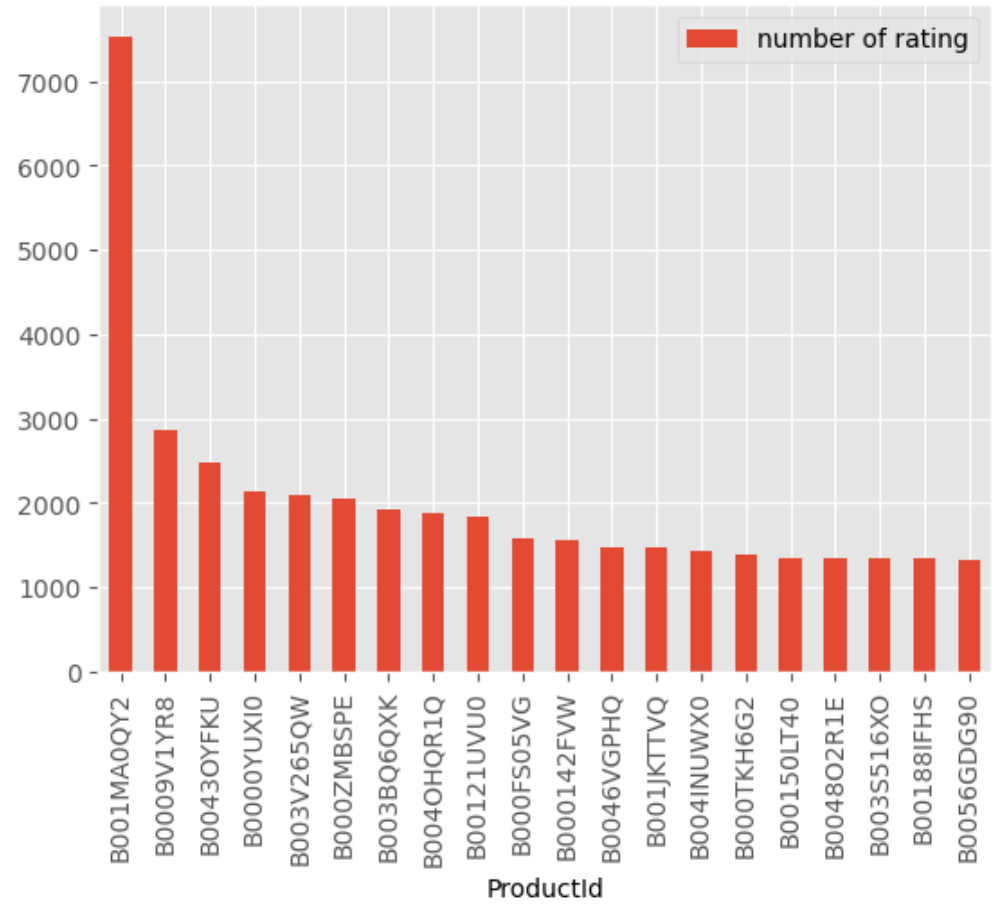
- Rating 1.0: 183,784 entries
- Rating 2.0: 113,034 entries
- Rating 3.0: 169,791 entries
- Rating 4.0: 307,740 entries
- Rating 5.0: 1,248,721 entries

A new DataFrame named rating has been created, providing detailed insights into product ratings:

ProductId	Average Rating	Number of Ratings
B001MA0QY2	4.321386	7533
B0009V1YR8	3.568839	2869
B0043OYFKU	4.310456	2477
B0000YUXI0	4.405040	2143
B003V265QW	4.365421	2088
...	...	...
B0013H228W	5.000000	1
B0013GNAIE	5.000000	1
B0013GNAG6	5.000000	1
B0013GMDT6	5.000000	1
B00740KNT2	1.000000	1

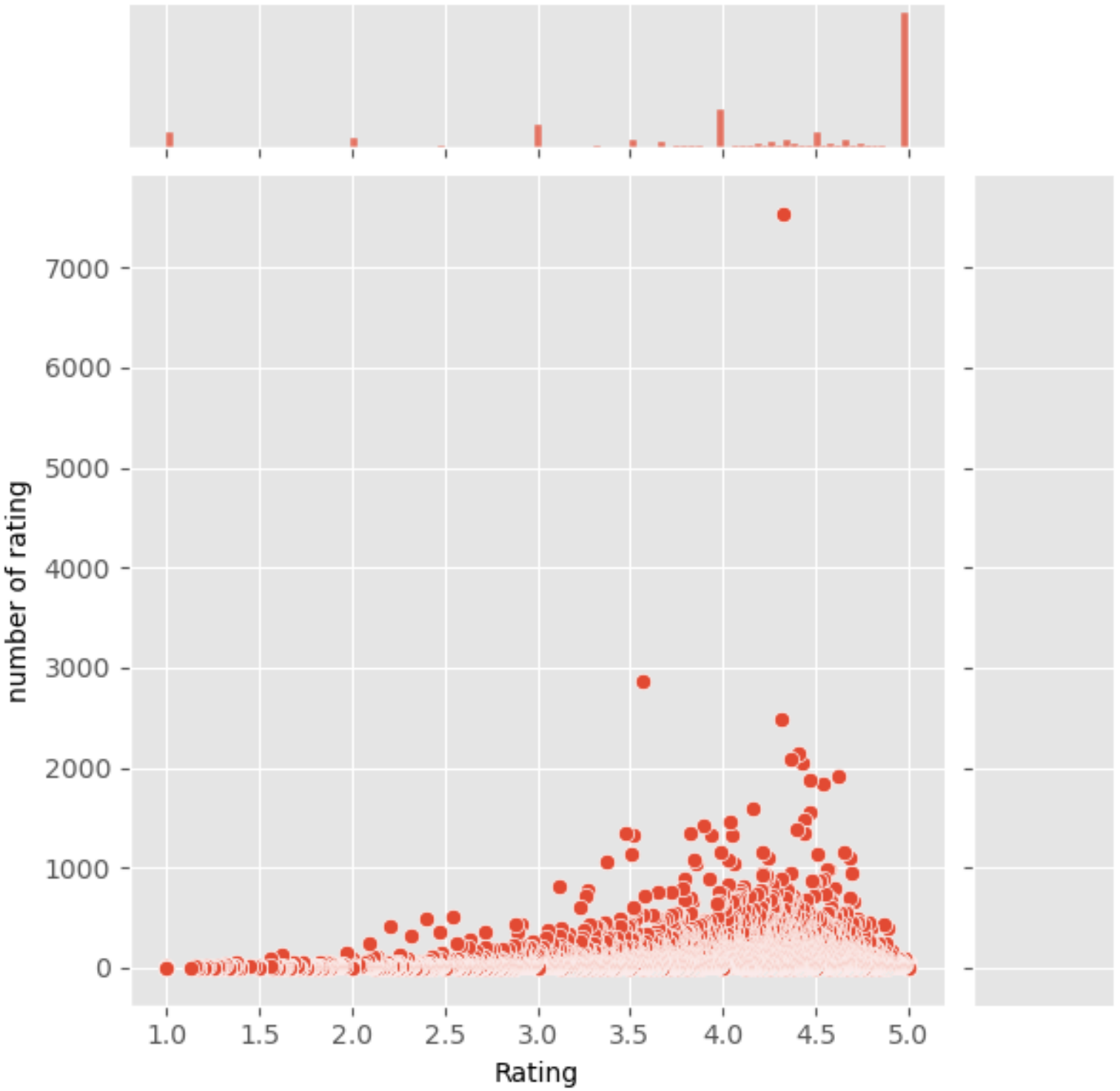
This information allows for a more nuanced understanding of product ratings, considering both the average rating and the number of ratings.

Top 20 products sell by ecommerce website



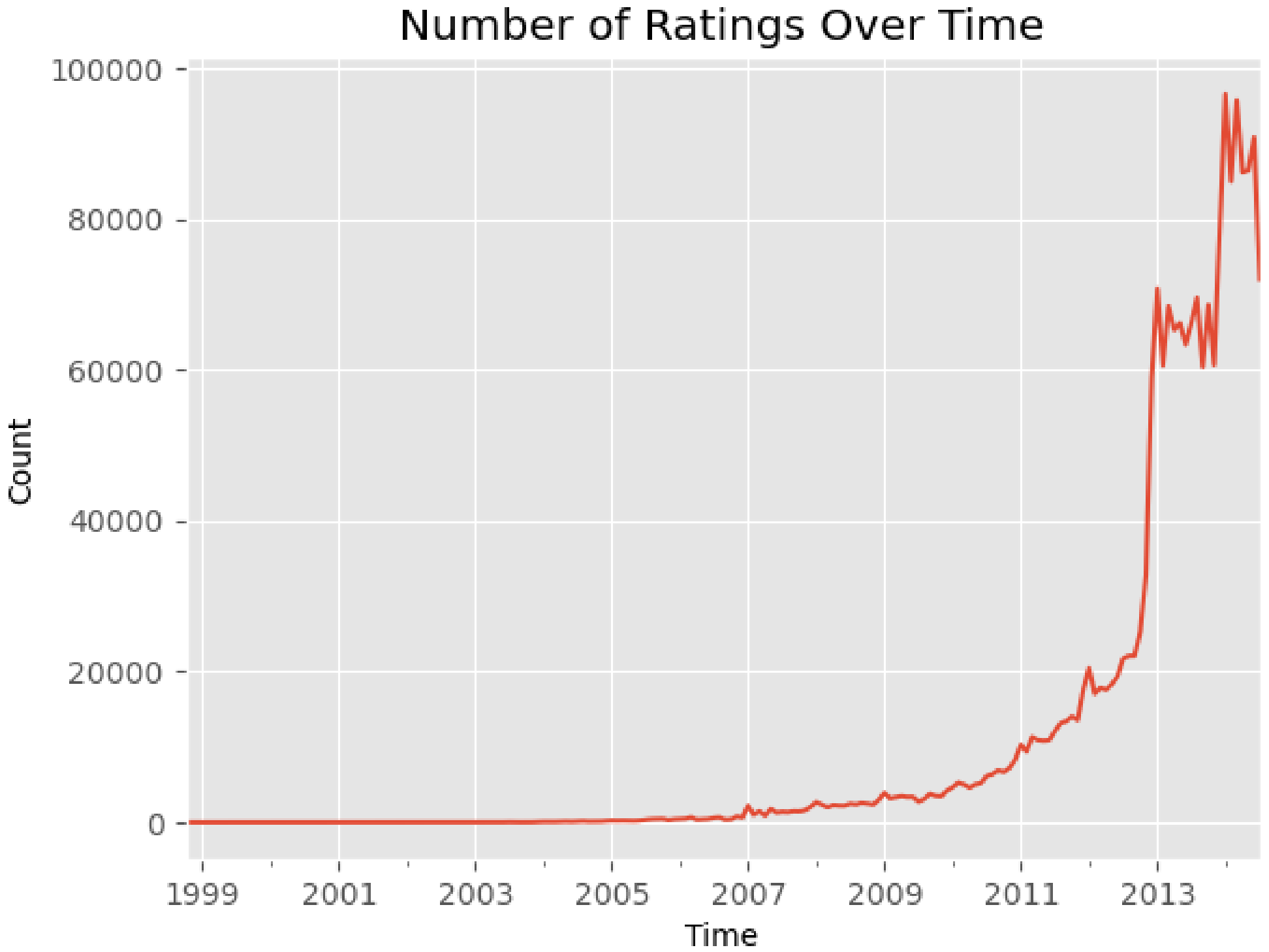
Analysis:

- The above graph gives us the most popular products (arranged in descending order) sold by the business.
- For eaxmple, product, ID # B001MA0QY2 has sales of over 7000, the next most popular product, ID # B0009V1YR8 has sales of 3000, etc.

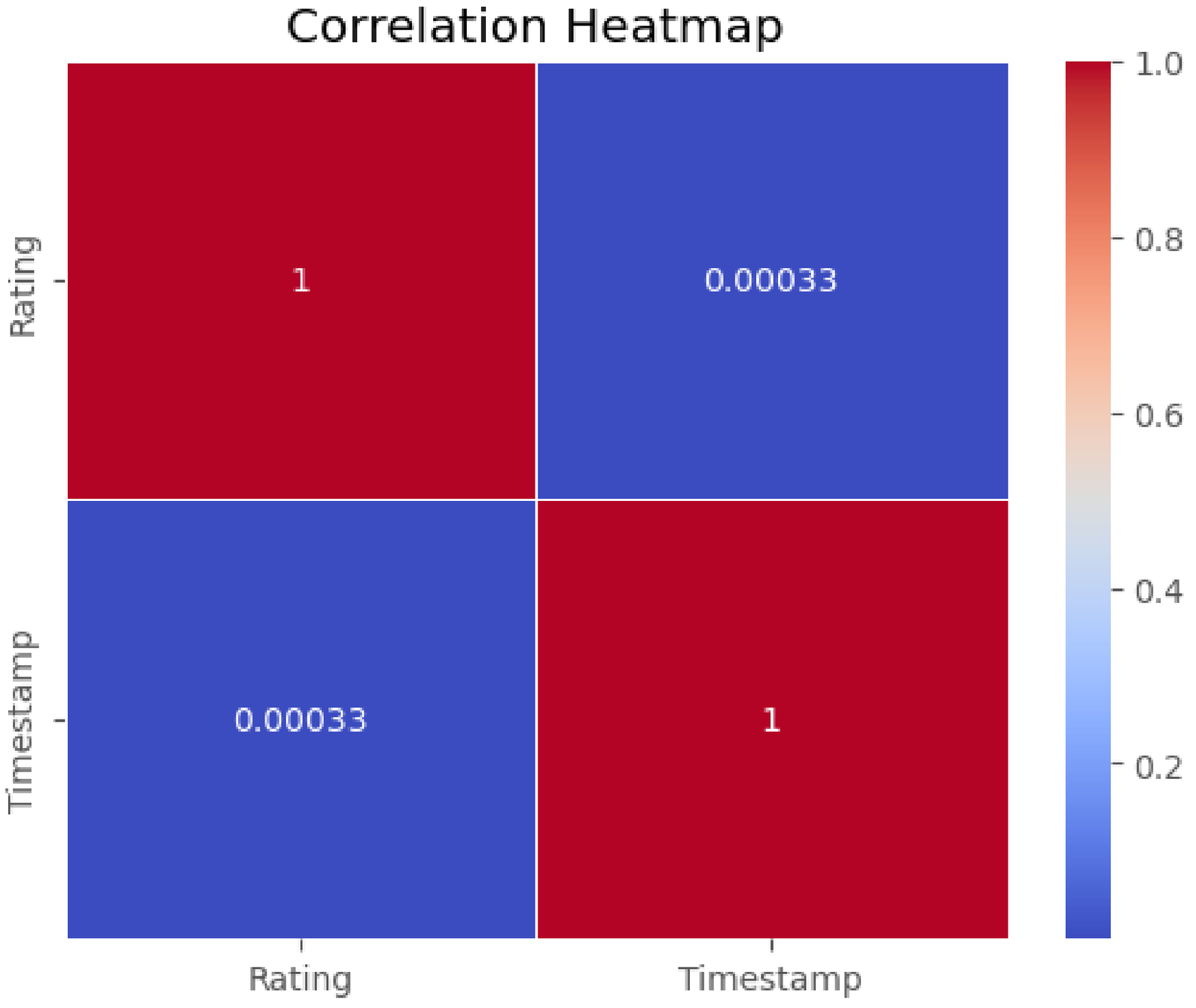


> This graph doesn't add much insight but we can figure out that most of the 5 rating given by very few users.

Time-based Analysis:



It shows that with the time moves on, people have voted for more products than before. its like a culture that people learned to vote for products that have been bought.



This heatmap shows the same conclusion of the previews Plot.

Model-based collaborative filtering system

Recommend items to users based on purchase history and similarity of ratings provided by other users who bought items to that of a particular customer. A model based collaborative filtering technique is chosen here as it helps in making predicting products for a particular user by identifying patterns based on preferences from multiple user data. Utility Matrix based on products sold and user reviews.

Utility Matrix : An utility matrix is consists of all possible user-item preferences (ratings) details represented as a matrix. The utility matrix is sparce as none of the users would buy all the items in the list, hence, most of the values are unknown.

To handle the large dataset, a fraction of the data (first 10,000 rows) has been taken to create a user-item matrix for further analysis.

The user-item matrix (`ratings_utility_matrix`) has been formed with users on one axis, products on the other, and their corresponding ratings in the matrix cells.

Here's a small portion of the table with 2 rows and 2 columns:

ProductId	0205616461	0558925278
A00205921JHJK5X9LNP42	NaN	NaN
A024581134CV80ZBLIZTZ	NaN	NaN

This table represents the user-item matrix (`ratings_utility_matrix`)

This matrix is consists of all possible user-item preferences (ratings) details represented as a matrix.

As you can see, Most of the values in utility matrix is not filled, which is expected because every user can not give rating to all products. That's why fill NaN values with 0.

Decomposing the matrix using TruncatedSVD, It is a dimensionality reduction technique.

The user-item matrix has undergone dimensionality reduction using TruncatedSVD, a technique to capture essential features while reducing the overall dimensionality. The resulting decomposed matrix (`decomposed_matrix`) has been obtained with 10 desired components.

Correlation_matrix

The correlation matrix (correlation_matrix) has been calculated using np.corrcoef() on the decomposed matrix. This matrix contains correlation coefficients between the different components, providing insights into the relationships among the reduced dimensions.

array([[1. , 0.93685339, 0.31608315, ..., 0.74189242, 0.67358848, 0.87510795], [0.93685339, 1. , 0.49189412, ..., 0.64670719, 0.37067888, 0.95627097], [0.31608315, 0.49189412, 1. , ..., 0.28410511, 0.8924298 , 0.5012165], ..., [0.74189242, 0.64670719, 0.28410511, ..., 1. , 0.22253368, 0.57716699], [0.67358848, 0.37067888, 0.8924298 , ..., 0.22253368, 1. , 0.38704044], [0.87510795, 0.95627097, 0.5012165 , ..., 0.57716699, 0.38704044, 1.]])

Random index

i = 493

The Product ID of the product the customer purchased :

```
'9790789890'
```

Top products to be displayed by the recommendation system to the above customer based on the purchase history of other customers on website

Top 10 Products recommended to the user based on the purchase done by user

```
['0205616461',
 '1304139212',
 '1304139220',
 '130414643X',
 '130414674X',
 '1304174778',
 '1304174867',
 '1304174905',
 '1304196046',
 '1304196062']
```

Conclusion

In conclusion, this collaborative recommender system project is not just about algorithms and data; it's about understanding users, optimizing data quality, and strategically implementing collaborative filtering to create a personalized and impactful recommendation system for the Amazon Beauty dataset. The journey undertaken sets the stage for the continuous refinement and improvement of the recommendation system's performance.