Documentation - exercise 5
Dataset1 : BigBasket Entire Product List (~28K datapoints)
Dataset2 : Amazon - Ratings (Beauty Products)

Professor : Dr. Kheradpisheh

Teacher Assistant : MohammadReza Khanmohammadi

By: Katayoun Kobraei

1.1 Introduction to Dataset1 : BigBasket Entire Product List (~28K datapoints)

E-commerce (electronic commerce) is the activity of electronically buying or selling of products on online services or over the Internet. E-commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. E-commerce is in turn driven by the technological advances of the semiconductor industry, and is the largest sector of the electronics industry. Bigbasket is the largest online grocery supermarket in India. Was launched somewhere around in 2011 since then they've been expanding their business. This dataset contains 10 attributes with simple meaning and which are described as follows:

- index - Simply the Index!
- product - Title of the product (as they're listed)
- category - Category into which product has been classified
- sub_category - Subcategory into which product has been kept
- brand - Brand of the product
- sale_price - Price at which product is being sold on the site
- market_price - Market price of the product
- type - Type into which product falls
- rating - Rating the product has got from its consumers
- description - Description of the dataset (in detail)

1.2 Introduction to Dataset2 : Amazon - Ratings (Beauty Products)

This is a dataset related to over 2 Million customer reviews and ratings of Beauty related products sold on their website.

It contains:

- the unique UserId (Customer Identification),
- the product ASIN (Amazon's unique product identification code for each product),
- Ratings (ranging from 1-5 based on customer satisfaction) and
- the Timestamp of the rating (in UNIX time)

This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs).

2. Abstract

In this assignment we investigate recommendation systems, focusing on two key methods: Content-Based and Collaborative Filtering. Content-Based Methods create personalized shopping guides by analyzing user preferences and product details. Collaborative Filtering considers the preferences of users with similar tastes to generate suggestions. We are tasked with implementing one model for each method, emphasizing preprocessing steps and feature engineering too. In this report evaluates these methods' effectiveness, encouraging exploration of advanced algorithms and innovative findings. This documentation comes with comprehensive figures, pictures, and tables.

PART 1 : Content-Based Method

Overview of the dataset

BigBasket Entire Product List dataset consists of 27555 rows in 10 columns. First we check columns of the dataset to see some samples and find out the type of each feature.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27555 entries, 0 to 27554
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   index         27555 non-null  int64
 1   product       27554 non-null  object
 2   category      27555 non-null  object
 3   sub_category  27555 non-null  object
 4   brand         27554 non-null  object
 5   sale_price    27555 non-null  float64
 6   market_price  27555 non-null  float64
 7   type          27555 non-null  object
 8   rating        18929 non-null  float64
 9   description   27440 non-null  object
```

Then we see some statistics features of each column:

	index	sale_price	market_price	rating
count	27555.00000	27555.000000	27555.000000	18929.000000
mean	13778.00000	322.514808	382.056664	3.943410
std	7954.58767	486.263116	581.730717	0.739063
min	1.00000	2.450000	3.000000	1.000000
25%	6889.50000	95.000000	100.000000	3.700000
50%	13778.00000	190.000000	220.000000	4.100000
75%	20666.50000	359.000000	425.000000	4.300000
max	27555.00000	12500.000000	12500.000000	5.000000

We see this dataset contains these 10 columns:

```
'index', 'product', 'category', 'sub_category', 'brand', 'sale_price',
'market_price', 'type', 'rating', 'description'],
```

And for the categorical and numerical feature we found out this:

```
Categorical Features: ['product', 'category', 'sub_category', 'brand', 'type', 'description']
Numerical Features: ['index', 'sale_price', 'market_price', 'rating']
```

As an example we can see category column has this unique values:

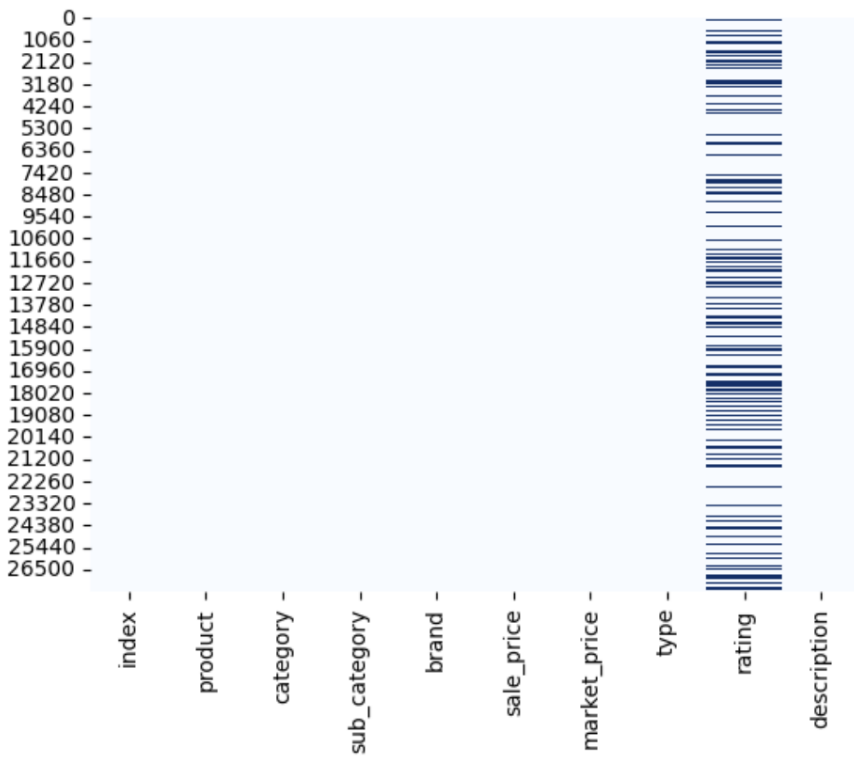
```
['Beauty & Hygiene', 'Kitchen, Garden & Pets',
 'Cleaning & Household', 'Gourmet & World Food',
 'Foodgrains, Oil & Masala', 'Snacks & Branded Foods', 'Beverages',
 'Bakery, Cakes & Dairy', 'Baby Care', 'Fruits & Vegetables',
 'Eggs, Meat & Fish'], dtype=object)
```

And we can count number of products in each unique value in brand column:

```
Fresho              638
bb Royal            539
BB Home             428
DP                  250
Fresho Signature    171
bb Combo            168
Amul                153
INATUR              146
Himalaya            141
Dabur               138
GoodDiet            134
Nike                124
Cello               124
Iveo                118
BIOTIQUE            117
Aroma Magic         109
Colgate             107
Organic Tattva      106
Loreal Paris        104
Britannia           104
Nakoda              103
Soulflower          102
Keya                101
NUTRIWISH           101
MTR                 100
True Elements        99
Mamaearth            97
Graminway            96
HappyChef            96
Dettol               95
```

Then we check if these is any null values in features to handle in this step:

```
index             0
product           1
category          0
sub_category      0
brand             1
sale_price        0
market_price      0
type              0
rating         8626
description     115
```

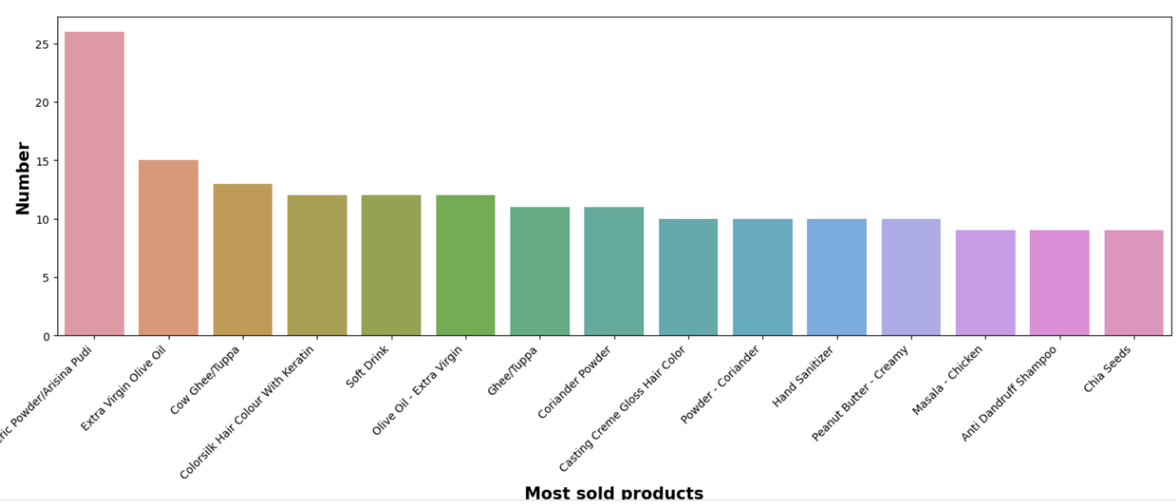


Rating is a crucial feature in this task. So we decided not to remove rows with null values in the rating feature. We instead fill null values with the mean value of the rating column.

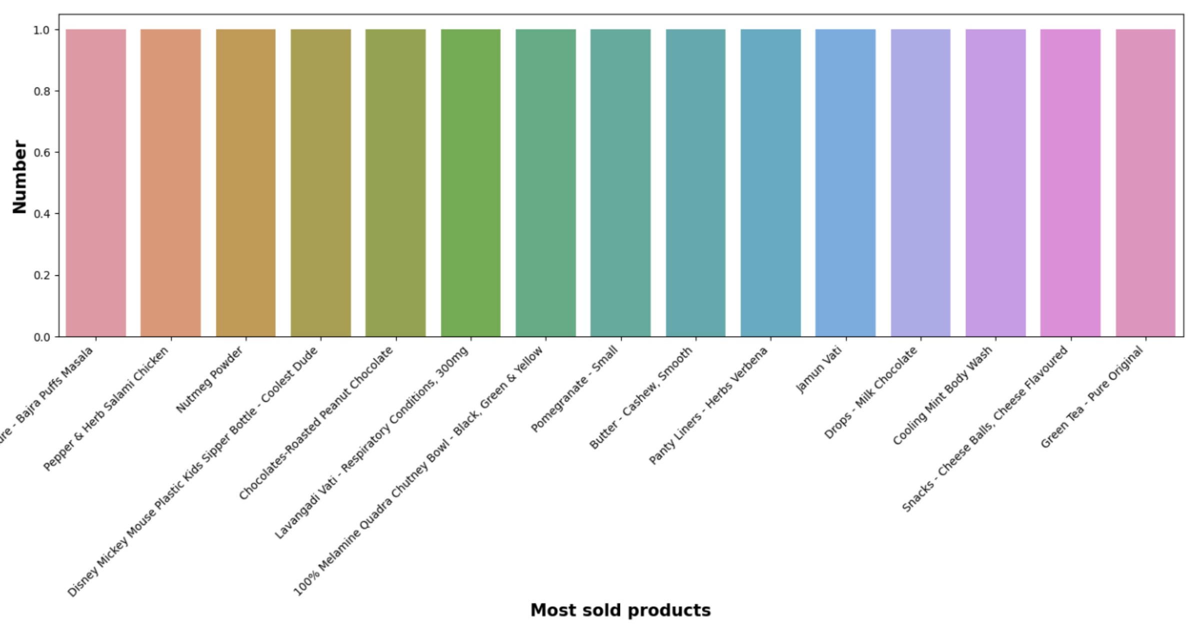
EDA

Top & least sold products

```
Most Sold products:
product
Turmeric Powder/Arisina Pudi          26
Extra Virgin Olive Oil                15
Cow Ghee/Tuppa                        13
Colorsilk Hair Colour With Keratin    12
Soft Drink                            12
Olive Oil - Extra Virgin              12
Ghee/Tuppa                            11
Coriander Powder                      11
Casting Creme Gloss Hair Color        10
Powder - Coriander                    10
Hand Sanitizer                        10
Peanut Butter - Creamy                10
Masala - Chicken                       9
Anti Dandruff Shampoo                  9
Chia Seeds                             9
```



```
Least Sold products:
product
Mixture - Bajra Puffs Masala                                            1
Pepper & Herb Salami Chicken                                            1
Nutmeg Powder                                                           1
Disney Mickey Mouse Plastic Kids Sipper Bottle - Coolest Dude           1
Chocolates-Roasted Peanut Chocolate                                     1
Lavangadi Vati - Respiratory Conditions, 300mg                          1
100% Melamine Quadra Chutney Bowl - Black, Green & Yellow               1
Pomegranate - Small                                                     1
Butter - Cashew, Smooth                                                 1
Panty Liners - Herbs Verbena                                            1
Jamun Vati                                                              1
Drops - Milk Chocolate                                                  1
Cooling Mint Body Wash                                                  1
Snacks - Cheese Balls, Cheese Flavoured                                 1
Green Tea - Pure Original                                               1
```



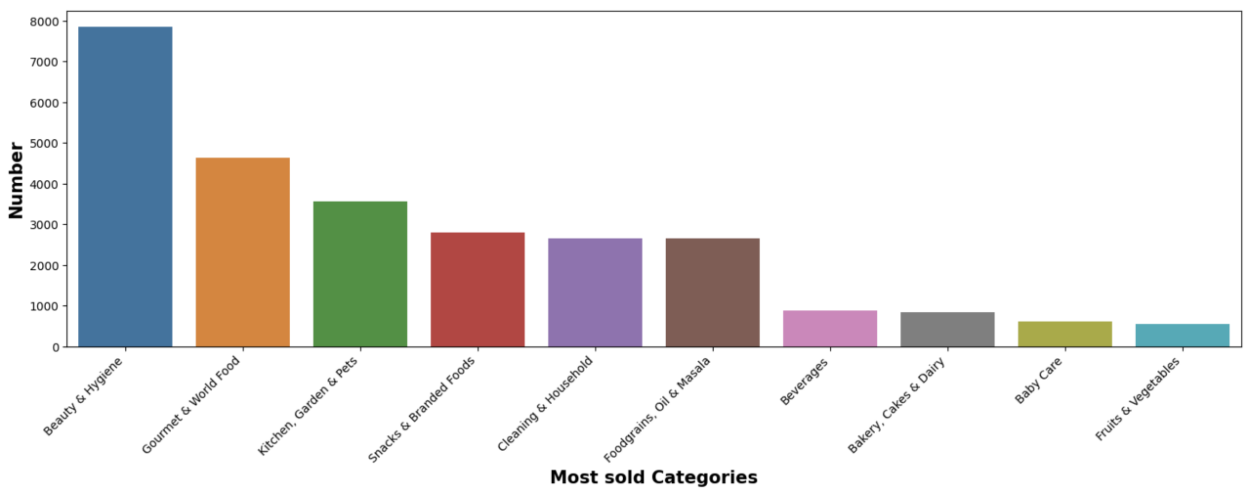
Through the plots of most and least sold products, we get to know that:

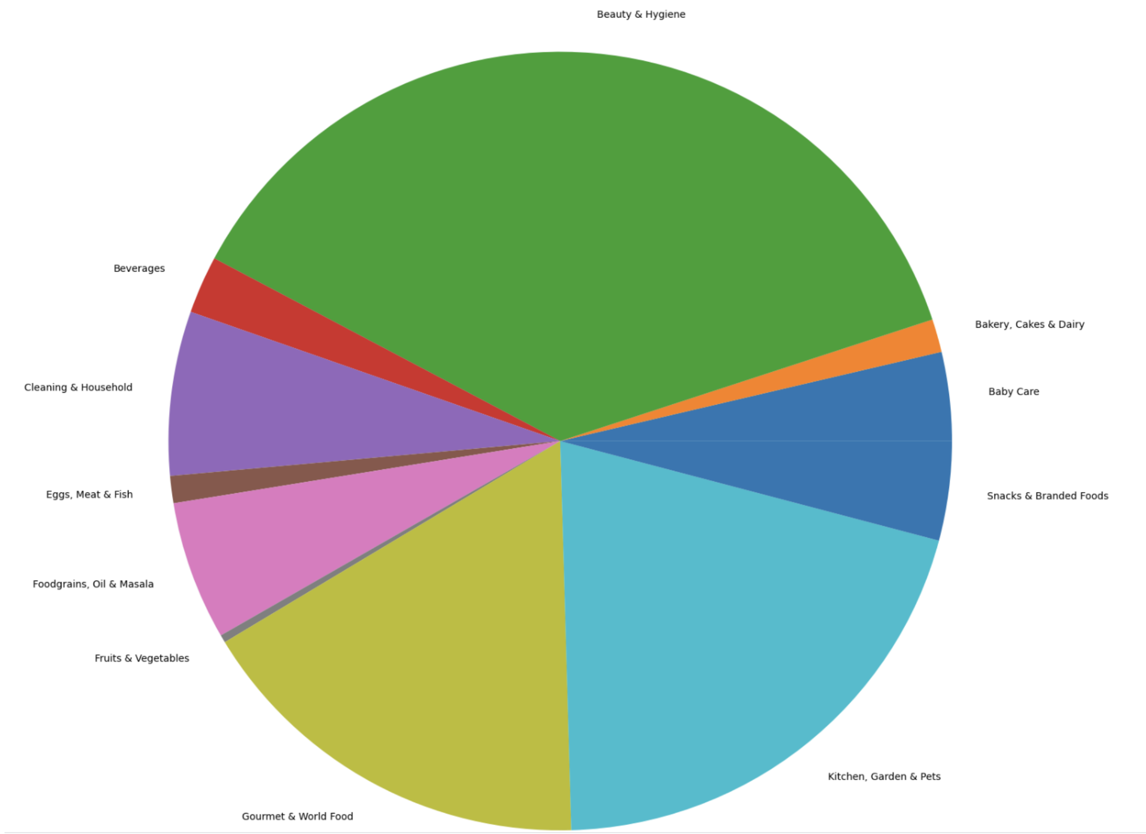
Indians have a trend to purchase different types of oils and ghee. Turmeric and coriander are the most used spices in Indian cuisine. On the other hand, Nutmeg and pepper powder are used in scarce amounts. Hand sanitizer is one of the highest selling products most probably due to covid-19 pandemic. Indians are very keen about haircare as we see hair color, shampoo and oils in the highest selling products. Raw spices like turmeric are preferred on masala mixtures.

Top and least sold Categories

Number of products in each category:

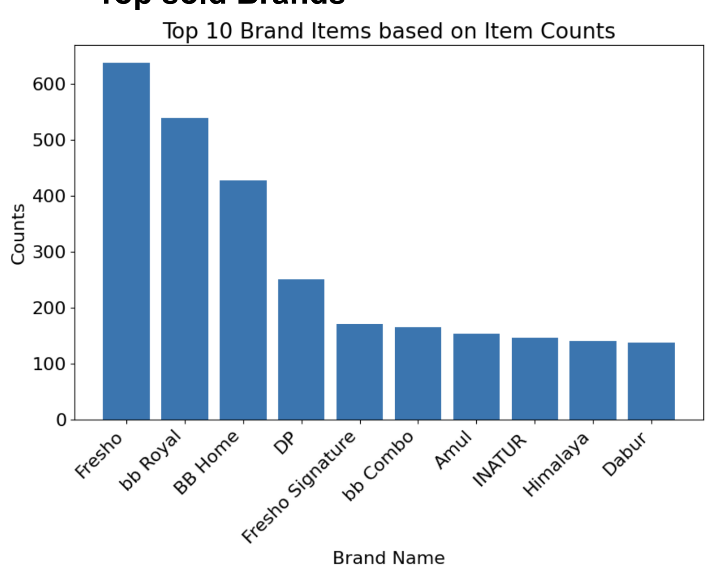
```
category
Beauty & Hygiene            7856
Gourmet & World Food        4647
Kitchen, Garden & Pets      3562
Snacks & Branded Foods      2813
Cleaning & Household        2665
Foodgrains, Oil & Masala    2655
Beverages                    881
Bakery, Cakes & Dairy        851
Baby Care                    609
Fruits & Vegetables          556
Eggs, Meat & Fish            344
```



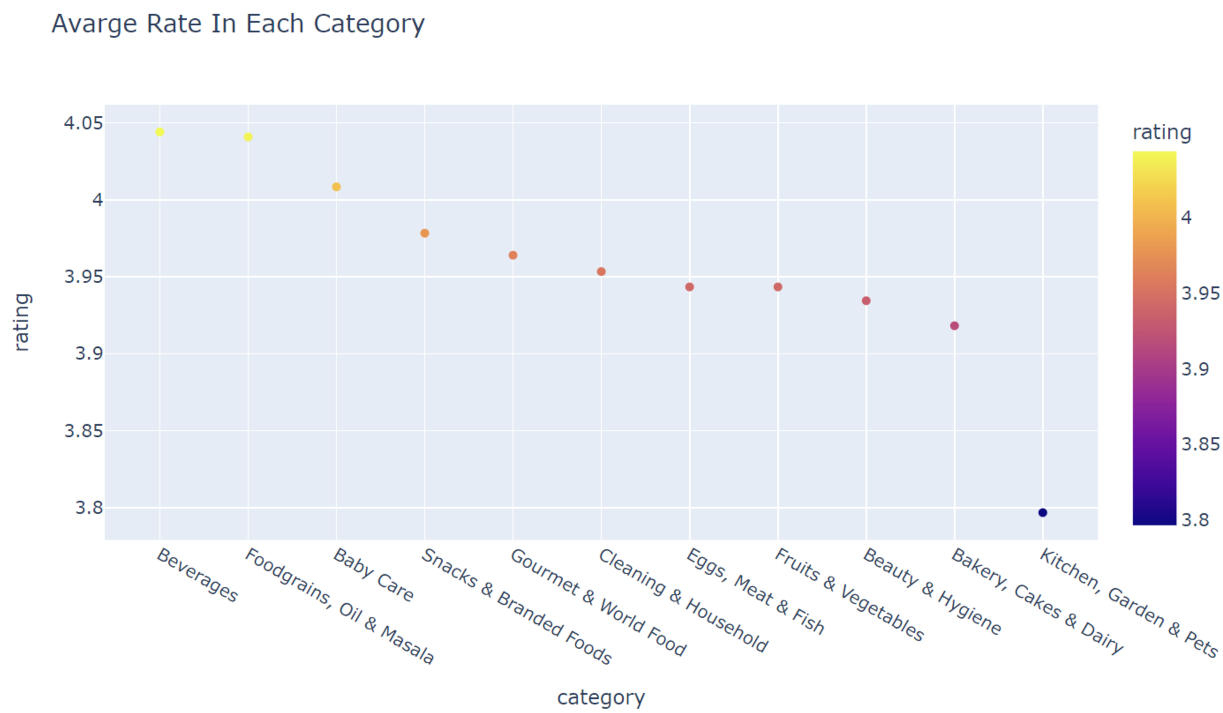


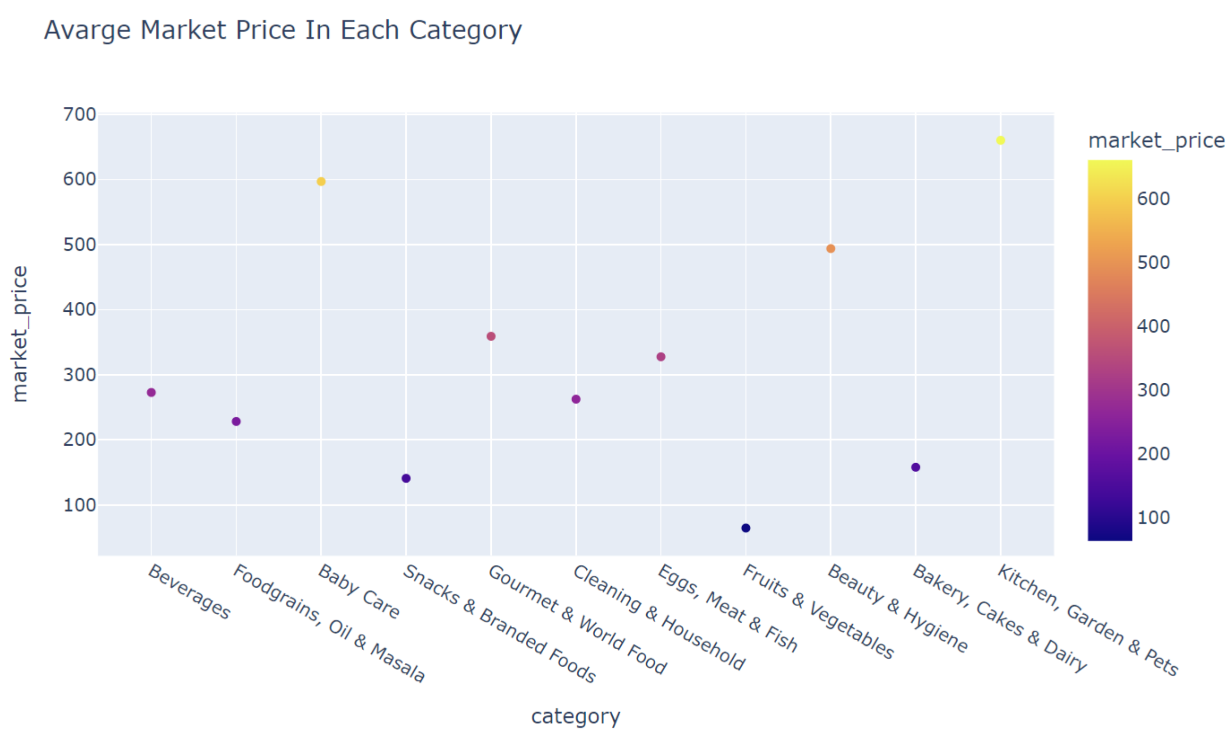
Based on the provided results, the "Beauty & Hygiene" category appears to be the most popular among people, as it has the highest product count with 7856 items.

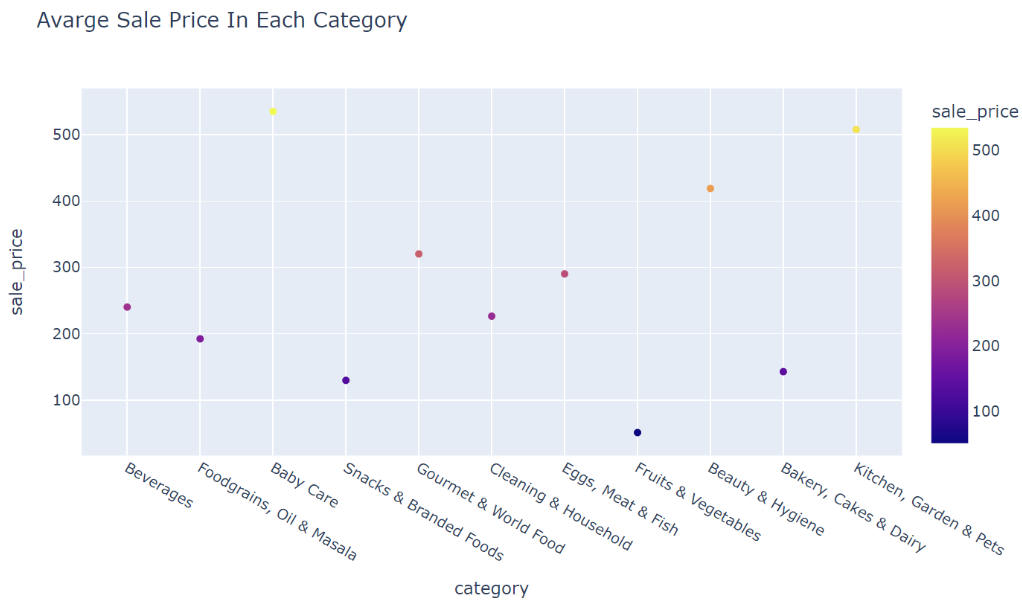
Top sold Brands



Avarge Rate, Market Price & Sale Price In Each Category







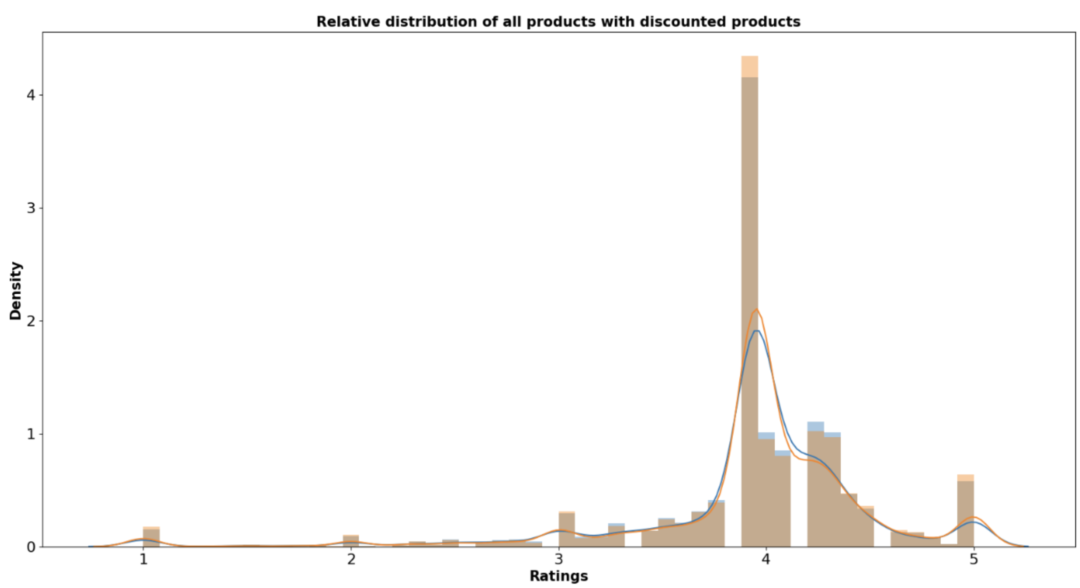
Effect of discounted prices

Lets check which items can be bought at discounted price from Big basket. For this, we will add another feature "diff_in_prices" measuring discount on a certain item.

First we made a column named discount to calculate discount using this formula:

data1["diff_in_prices"] = data1["market_price"] - data1["sale_price"]

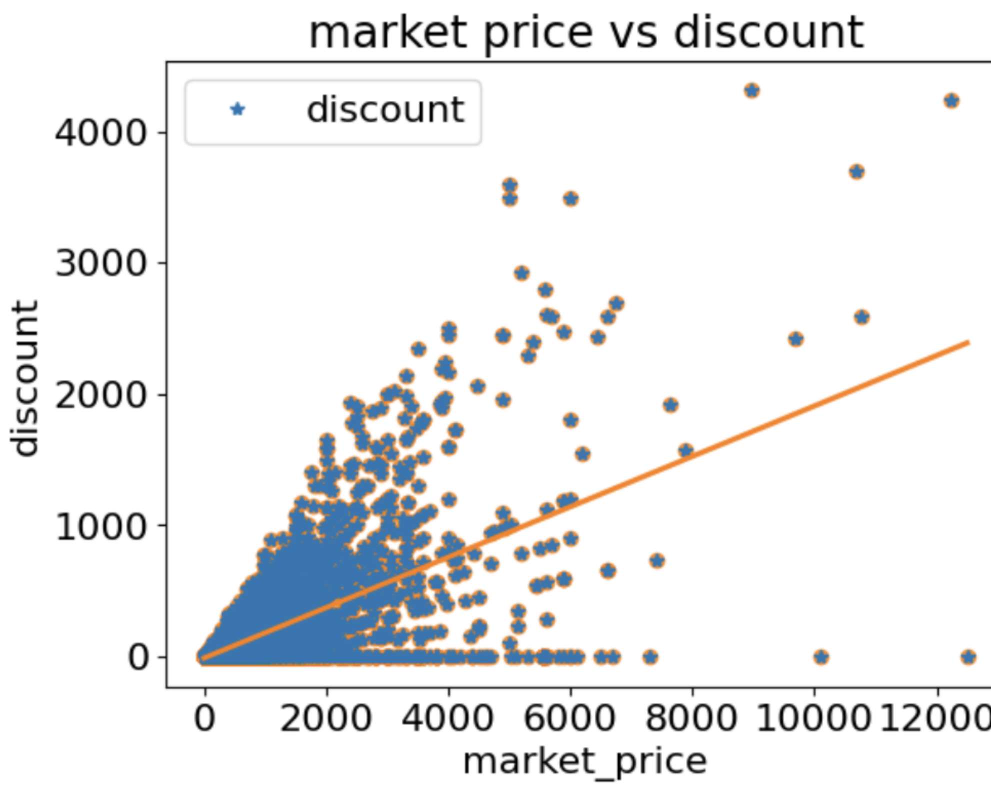
Then we plot Relative distribution of all products with discounted products:



In the above graph, yellow color specifies rating of all the items, whereas blue color denotes the ratings of the items on which some discount has been offered. As we see,

The offered discounts showed a little increase in purchase of items with 3.0 to 4.2 ratings. Otherwise, discounts helped no increase in purchase. Another interesting observation was that the highest rated products (4.5 to 5) with no discount exceeded the rate of purchase of discounted products. It means the customers, if provided with high quality products which satisfy them, will buy the products no matter if a discount is offered or not.

Then we plot market_price vs. discount to compare and see their relation:



In the above charts we try to compare the relationship between the market price with the discount price and discount percentage. From the above charts we notice that the items with higher market price have higher discount price, However, if we see it in terms of percentage, the items with lower market price have higher discount percentage.

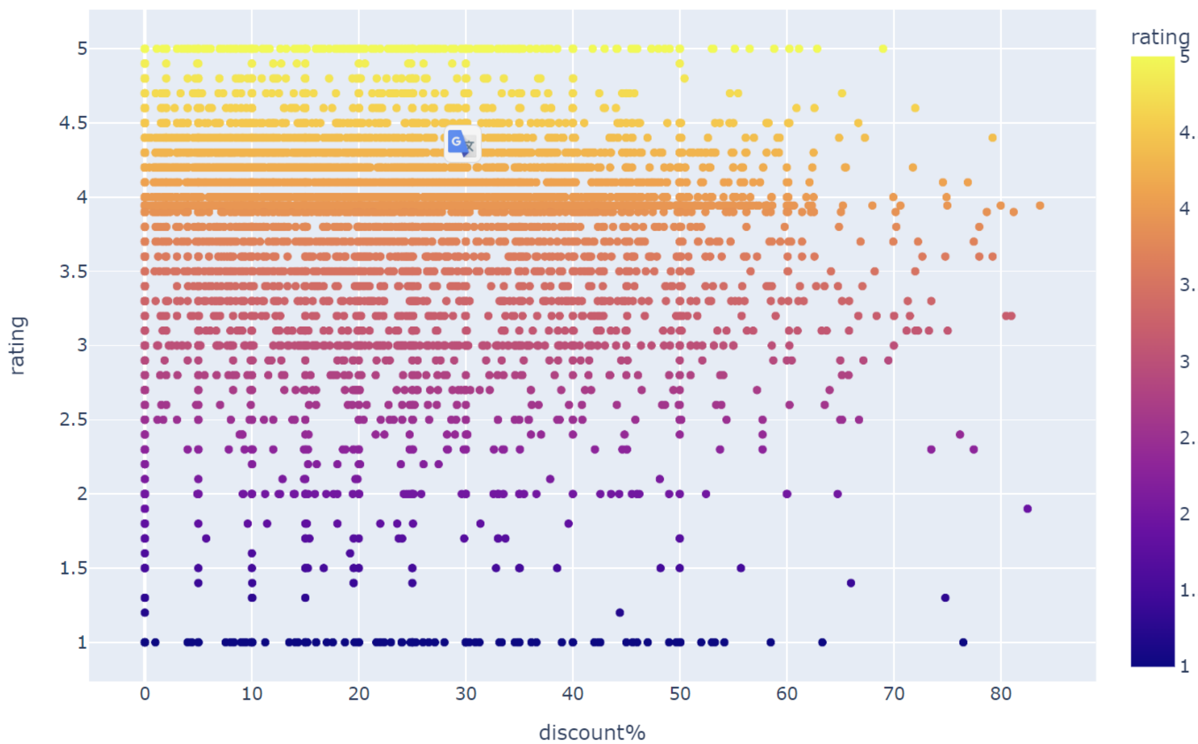
Correlations

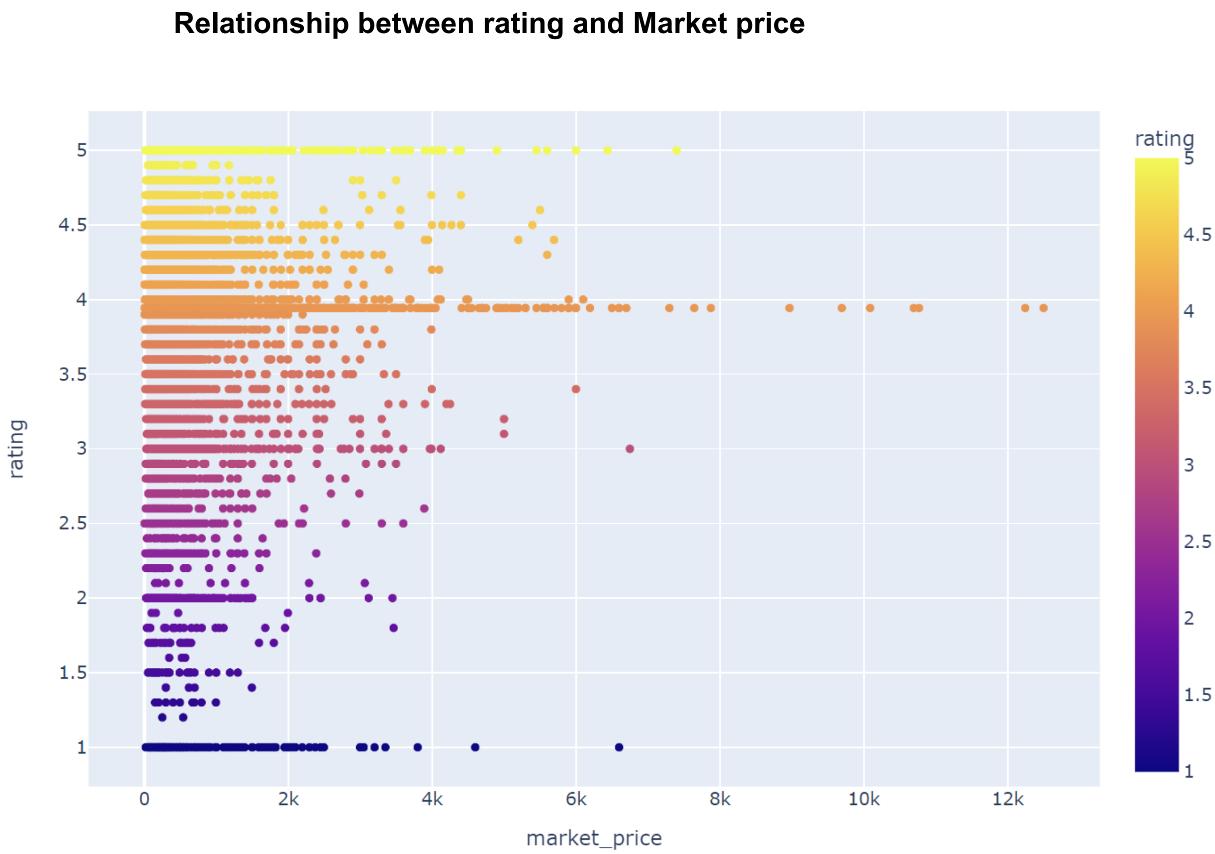
Checking for correlation between market price and discount

```
Coefficient of Correlation
PearsonRResult(statistic=0.6611903567911539, pvalue=0.0)
```

The calculated Pearson correlation coefficient is approximately 0.66, indicating a moderately positive correlation between market price and discount. The p-value is 0.0, suggesting that the correlation is statistically significant. This implies that as the market price increases, there is a tendency for the discount to also increase, and vice versa.

Relationship between rating and discount %



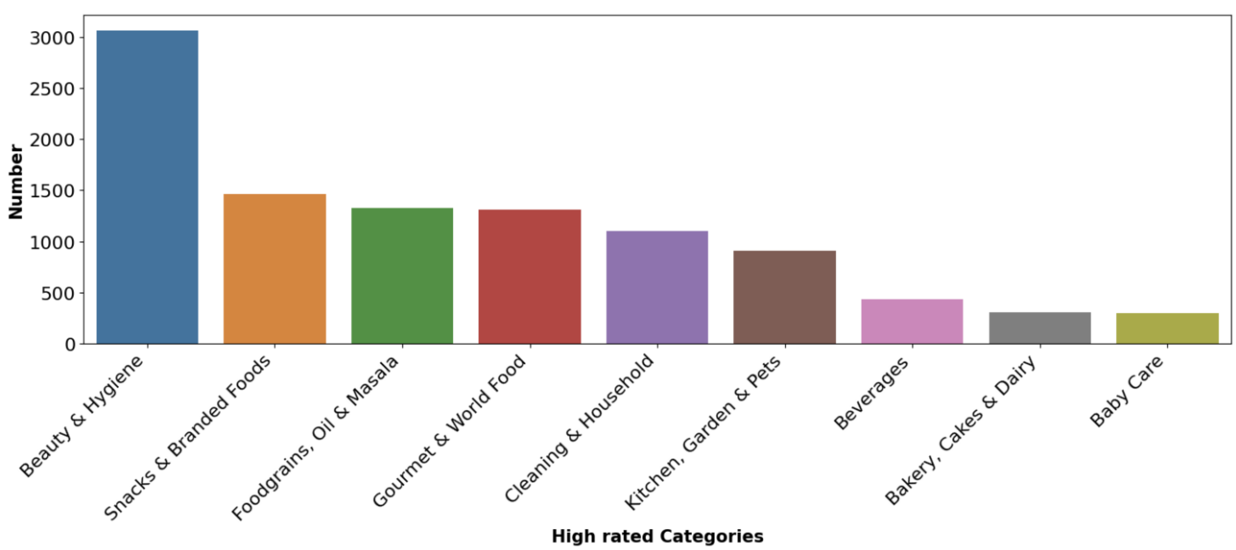


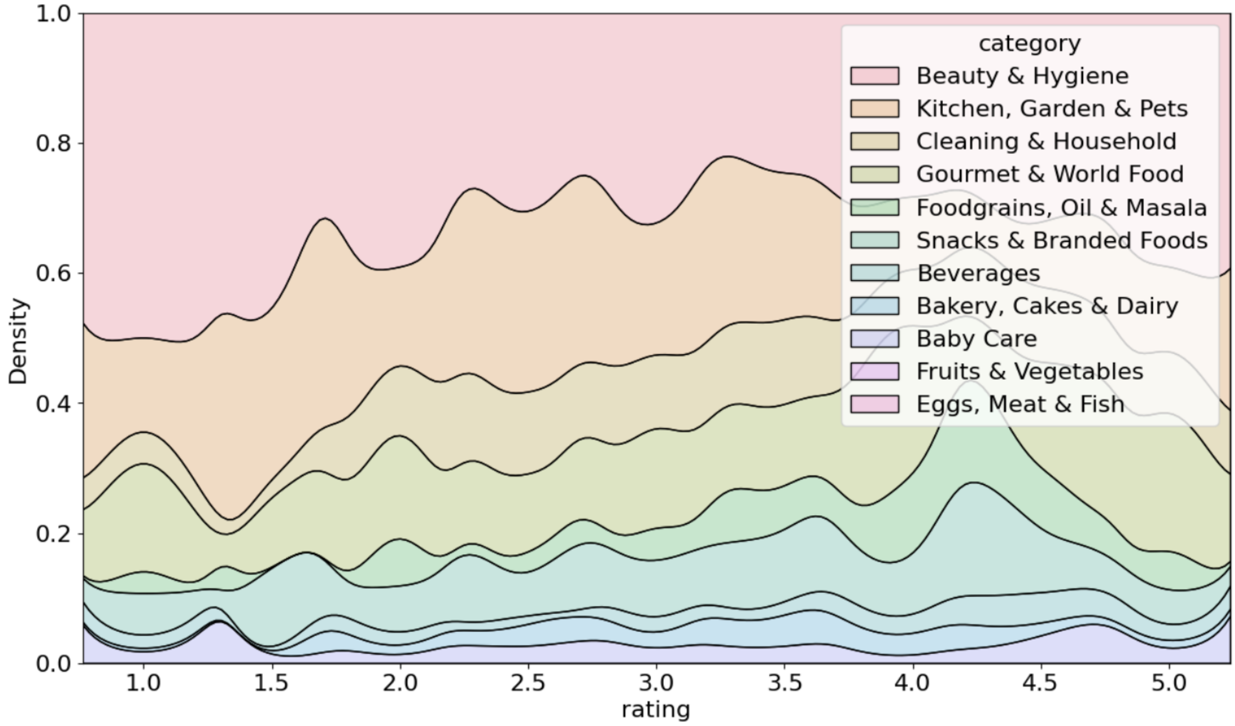
Relationship between rating and Market price
rating
5
4.5
4
3.5
3
2.5
2
1.5
1
0
2k
4k
6k
8k
10k
12k
market_price

high rated products (more than 4 score)

```
Number of products with more than 4 rating is 10212
```

	index	product	category	sub_category	brand	sale_price	market_price	type	rating	description	diff_in_prices	discount%	discou
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.0	220.0	Hair Oil & Serum	4.1	This Product contains Garlic Oil that is known...	0.0	0.0	
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.0	162.0	Bathing Bars & Soaps	4.4	Nivea Creme Soft Soap gives your skin the best...	0.0	0.0	
9	10	Scrub Pad - Anti-Bacterial, Regular	Cleaning & Household	Mops, Brushes & Scrubs	Scotch brite	20.0	20.0	Utensil Scrub-Pad, Glove	4.3	Scotch Brite Anti-Bacterial Scrub Pad thoroug...	0.0	0.0	
12	13	Face Wash - Oil Control, Active	Beauty & Hygiene	Skin Care	Oxy	110.0	110.0	Face Care	5.0	This face wash deeply cleanses dirt and impuri...	0.0	0.0	
14	15	Just Spray - Mosquito Repellent Room Spray	Cleaning & Household	Fresheners & Repellents	Herbal Strategi	200.0	200.0	Mosquito Repellent	4.2	Strategi Just Spray is a very effective 100% H...	0.0	0.0	





This above plot is a Kernel Density Estimate (KDE) plot using the seaborn library. The KDE plot visualizes the distribution of the "rating" variable in the dataset, with different colors representing different categories.

Recommender system:

A Content-Based Recommender System is a type of recommendation system that suggests items to users based on the features or content of the items and the preferences expressed by the user. Here's how it typically works:

User Profile Creation: The system creates a profile for each user based on their historical preferences or explicit feedback.

Item Profile Creation: Similarly, a profile is created for each item based on its features, such as keywords, genres, or other relevant attributes.

Matching User and Item Profiles: The system then matches user profiles with item profiles to determine recommendations.

Recommendation Generation: Recommendations are generated by suggesting items that are similar to the ones the user has shown interest in before, based on the content features.

Personalization: The system provides personalized recommendations for each user, enhancing the user experience by offering items that align with their tastes.

Advantages:
Content-based systems can provide recommendations for new or niche items without relying on user history.
They are less dependent on the availability of user-item interaction data.

Challenges:
Limited diversity in recommendations since it relies on the user's past preferences.
Difficulty capturing complex user tastes that may not be fully expressed by content features.

In the first step of this task we compute the similarity score, let's use Linear_Kernel. Linear Kernel which Calculates the Dot Product of the tfidf_matrix and returns an aggregate value depicting the Similarity score.

```
array([[1.        , 0.01577945, 0.00997173, ..., 0.01074711, 0.01118616,
        0.        ],
       [0.01577945, 1.        , 0.00697308, ..., 0.        , 0.        ,
        0.        ],
       [0.00997173, 0.00697308, 1.        , ..., 0.0062593 , 0.        ,
        0.        ],
       ...,
       [0.01074711, 0.        , 0.0062593 , ..., 1.        , 0.        ,
        0.        ],
       [0.01118616, 0.        , 0.        , ..., 0.        , 1.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        1.        ]])
```

so we will be recommending items based on similarity score.
But our problem is that we will be getting back the similarity scores so we will be sorting the scores.
Now we need a reverse-map to get the title and that is what indices are for.

So as an example we can recommend these products for 'Water Bottle - Orange':

```
11320    Rectangular Plastic Container - With Lid, Mult...
11642                                   Jar - With Lid, Yellow
14551        Rectangular Container - With lid, Multicolour
26451     Round & Flat Storage Container - With lid, Green
26460              Round Plastic Container - With Lid, Pink
6163     Premium Rectangular Plastic Container With Lid...
9546     Premium Round Plastic Container With Lid - Yellow
13959    Premium Rectangular Plastic Container With Lid...
19381    Premium Round & Flat Storage Container With Li...
24255      Premium Round Plastic Container With Lid - Blue
```

PART 2 : Collaborative Filtering Method

Overview of the dataset

This dataset contains 2023070 datapoints in 5 columns:

	UserId	ProductId	Rating	Timestamp	user_id
0	A39HTATAQ9V7YF	0205616461	5.0	1369699200	0
1	A3JM6GV9MNOF9X	0558925278	3.0	1355443200	1
2	A1Z513UWSAAO0F	0558925278	5.0	1404691200	2
3	A1WMRR494NWEWV	0733001998	4.0	1382572800	3
4	A3IAAVS479H7M7	0737104473	1.0	1274227200	4
...	...	...	...	...	...
2023065	A3DEHKPFANB8VA	B00LORWRJA	5.0	1405296000	1207977
2023066	A3DEHKPFANB8VA	B00LOS7MEE	5.0	1405296000	1207977
2023067	AG9TJLJUN5OM3	B00LP2YB8E	5.0	1405382400	1210242
2023068	AYBIB14QOI9PC	B00LPVG6V0	5.0	1405555200	1209896

We check some statistics features of each column:

	Rating	Timestamp	user_id
count	2.023070e+06	2.023070e+06	2.023070e+06
mean	4.149036e+00	1.360389e+09	5.036093e+05
std	1.311505e+00	4.611860e+07	3.535750e+05
min	1.000000e+00	9.087552e+08	0.000000e+00
25%	4.000000e+00	1.350259e+09	1.895360e+05
50%	5.000000e+00	1.372810e+09	4.557920e+05
75%	5.000000e+00	1.391472e+09	7.950578e+05
max	5.000000e+00	1.406074e+09	1.210270e+06

```
unique users: 1210271
unique products: 249274
total ratings:  2023070
```

Fortunately there is not any null value in the dataset:

```
index            0
product          0
category         0
sub_category     0
brand            0
sale_price       0
market_price     0
type             0
rating           0
description      0
diff_in_prices   0
discount%        0
discount         0
dtype: int64
```

EDA

Number of rated products per user

```
UserId
A3KEZLJ59C1JVH    389
A281NPSIMI1C2R    336
A3M174IC0VXOS2    326
A2V5R832QCSOMX    278
A3LJLRIZL38GG3    276
                 ...
A2G8M8PDTN09UZ      1
A2G8MAFIIQSJ42      1
A2G8MTKRE6MV52      1
A2G8MWBXG6JIY6      1
AZZZU2TD7Q3ET       1
```

Number of ratings per product

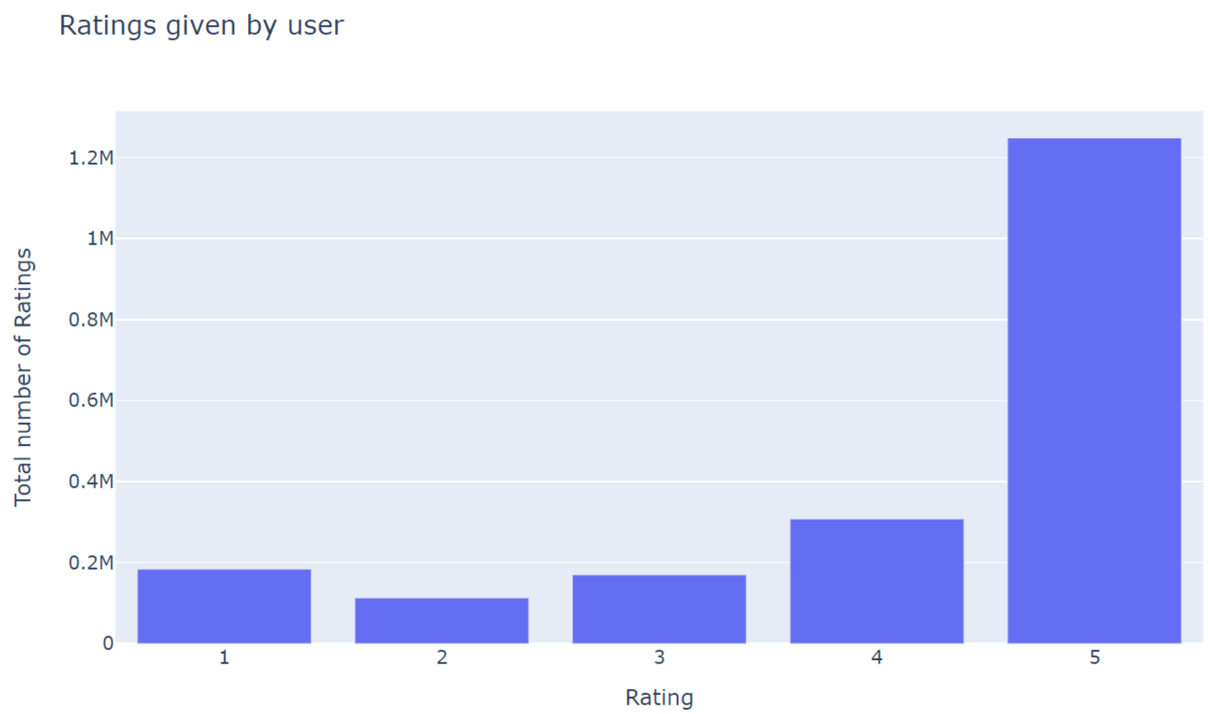
```
ProductId
B001MA0QY2    7533
B0009V1YR8    2869
B0043OYFKU    2477
B0000YUXI0    2143
B003V265QW    2088
              ...
B004U81OBC       1
B004U7R0EI       1
B004U7Q2O2       1
B004U7NKRE       1
B00LU0LTOU       1
```

Number of products rated by each user

```
ProductId
B001MA0QY2    7533
B0009V1YR8    2869
B0043OYFKU    2477
B0000YUXI0    2143
B003V265QW    2088
              ...
B004U81OBC       1
B004U7R0EI       1
B004U7Q2O2       1
B004U7NKRE       1
B00LU0LTOU       1
```

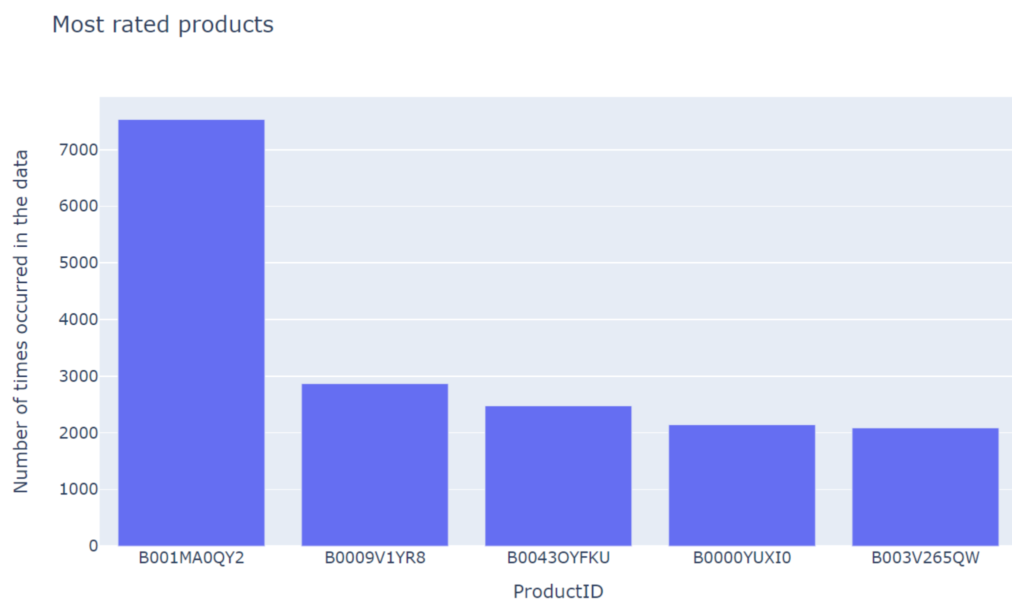
The ratings given by users

```
Range of Ratings:  Rating
5.0    1248721
4.0     307740
1.0     183784
3.0     169791
2.0     113034
Name: count, dtype: int64
[1248721, 307740, 183784, 169791, 113034]
```



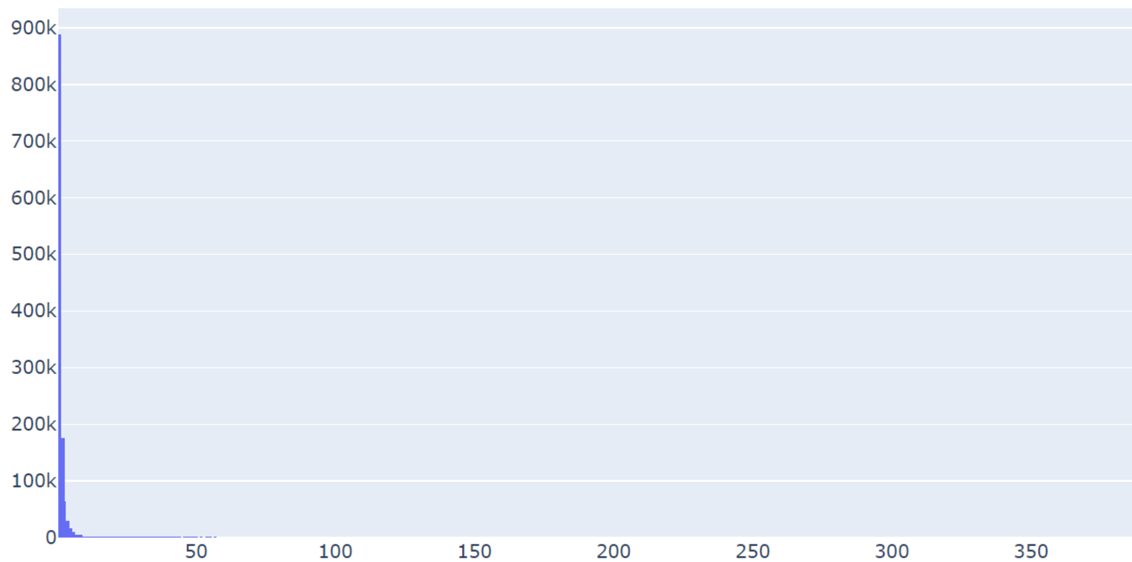
Products which are most popular

```
Products with occurred the most:
 ProductId
B001MA0QY2    7533
B0009V1YR8    2869
B0043OYFKU    2477
B0000YUXI0    2143
B003V265QW    2088
Name: count, dtype: int64
```



Average rating given by each user

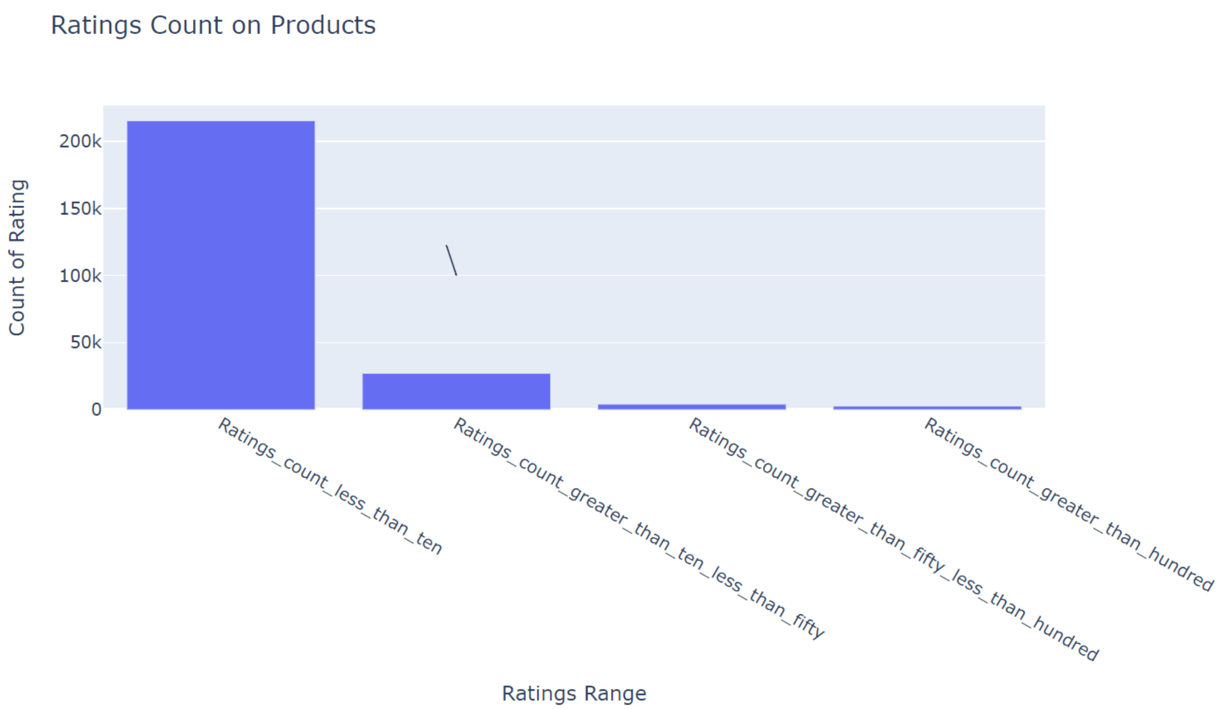
```
Average rating given by each user:  UserId
A3KEZLJ59C1JVH    389
A281NPSIMI1C2R    336
A3M174IC0VXOS2    326
A2V5R832QCSOMX    278
A3LJLRIZL38GG3    276
Name: Rating, dtype: int64
```



Products with very less ratings

```
Products with ratings given by users:
            Rating
ProductId
0205616461       1
0558925278       2
0733001998       1
0737104473       1
0762451459       1
```

```
Ratings_count_less_than_ten:  215395
Ratings_count_greater_than_ten_less_than_fifty:  27082
Ratings_count_greater_than_fifty_less_than_hundred:  4110
Ratings_count_greater_than_hundred:  2687
Average number of products rated by users:  8.115848423822781
```



Recommender System:

```
Average rating given by users:
   user  Rating
0     0     5.0
1     1     5.0
2     2     3.0
3     3     5.0
4     4     5.0
----------------------------------------------------------

Modified dataset:
            UserId   ProductId  Rating_x   Timestamp  user_id    user  product  \
0  A39HTATAQ9V7YF  0205616461       5.0  1369699200        0  725046        0
1  A39HTATAQ9V7YF  B002OVV7F0       3.0  1369699200        0  725046    81854
2  A39HTATAQ9V7YF  B0031IH5FQ       5.0  1369699200        0  725046    89013
3  A39HTATAQ9V7YF  B006GQPZ8E       4.0  1369699200        0  725046   154092
4  A3JM6GV9MNOF9X  0558925278       3.0  1355443200        1  814606        1

   Rating_y
0      4.25
1      4.25
2      4.25
3      4.25
4      3.50
----------------------------------------------------------

Dataset:
            UserId   ProductId  real_rating   Timestamp  user_id    user  \
0  A39HTATAQ9V7YF  0205616461          5.0  1369699200        0  725046
1  A39HTATAQ9V7YF  B002OVV7F0          3.0  1369699200        0  725046
2  A39HTATAQ9V7YF  B0031IH5FQ          5.0  1369699200        0  725046
3  A39HTATAQ9V7YF  B006GQPZ8E          4.0  1369699200        0  725046
4  A3JM6GV9MNOF9X  0558925278          3.0  1355443200        1  814606

   product  average_rating
0        0            4.25
1    81854            4.25
2    89013            4.25
3   154092            4.25
4        1            3.50
----------------------------------------------------------
```

Certain users tend to give higher ratings while others tend to gibve lower ratings. To negate this bias, we normalise the ratings given by the users.

```
Data with adjusted rating:
           UserId   ProductId  real_rating   Timestamp  user_id    user  \
0  A39HTATAQ9V7YF  0205616461          5.0  1369699200        0  725046
1  A39HTATAQ9V7YF  B002OVV7F0          3.0  1369699200        0  725046
2  A39HTATAQ9V7YF  B0031IH5FQ          5.0  1369699200        0  725046
3  A39HTATAQ9V7YF  B006GQPZ8E          4.0  1369699200        0  725046
4  A3JM6GV9MNOF9X  0558925278          3.0  1355443200        1  814606

   product  average_rating  normalized_rating
0        0            4.25               0.75
1    81854            4.25              -1.25
2    89013            4.25               0.75
3   154092            4.25              -0.25
4        1            3.50              -0.50
```

Cosine Similarity

We use a distance based metric - cosine similarity to identify similar users. It is important first, to remove products that have a very low number of ratings.

```
Real ratings:
          real_rating
product
0                   1
1                   2
2                   1
3                   1
4                   1
         real_rating
product
704              558
719              377
754              288
834              412
843              313
```

```
Popular product count which have ratings over average rating count:  934
-------------------------------------------------------------------------------
Filtered rated product in the dataset:
            UserId   ProductId  real_rating   Timestamp  user_id     user  \
1   A39HTATAQ9V7YF  B002OVV7F0          3.0  1369699200        0   725046
18   AKJHHD5VEH7VG  B0000UTUVU          5.0  1232323200        5  1073169
20   AKJHHD5VEH7VG  B000F8HWXU          5.0  1379721600        5  1073169
45   AKJHHD5VEH7VG  B001LF4I8I          4.0  1232841600        5  1073169
47   AKJHHD5VEH7VG  B001OMI93S          5.0  1236643200        5  1073169

    product  average_rating  normalized_rating
1     81854        4.250000          -1.250000
18     2237        4.222222           0.777778
20    16510        4.222222           0.777778
45    65074        4.222222          -0.222222
47    67333        4.222222           0.777778
--------------------------------------------------------------------------------
The size of dataset has changed from  2023070  to  370511
--------------------------------------------------------------------------------
```

Creating the User-item matrix

```
Updated Dataset:
 product                 704     719     754     834     843     858     861     \
UserId
A0010876CNE3ILIM9HV0      0.0     0.0     0.0     0.0     0.0     0.0     0.0
A0011102257KBXODKL24I     0.0     0.0     0.0     0.0     0.0     0.0     0.0
A00120381FL204MYH7G3B     0.0     0.0     0.0     0.0     0.0     0.0     0.0
A00126503SUWI86KZBMIN     0.0     0.0     0.0     0.0     0.0     0.0     0.0
A001573229XK5T8PI0OKA     0.0     0.0     0.0     0.0     0.0     0.0     0.0

product                 873     944     981     ...  241604  242018  242048  \
UserId                                          ...
A0010876CNE3ILIM9HV0      0.0     0.0     0.0  ...     0.0     0.0     0.0
A0011102257KBXODKL24I     0.0     0.0     0.0  ...     0.0     0.0     0.0
A00120381FL204MYH7G3B     0.0     0.0     0.0  ...     0.0     0.0     0.0
A00126503SUWI86KZBMIN     0.0     0.0     0.0  ...     0.0     0.0     0.0
A001573229XK5T8PI0OKA     0.0     0.0     0.0  ...     0.0     0.0     0.0

product                 243416  244376  244448  245600  247603  249109  249211
UserId
A0010876CNE3ILIM9HV0      0.0     0.0     0.0     0.0     0.0     0.0     0.0
A0011102257KBXODKL24I     0.0     0.0     0.0     0.0     0.0     0.0     0.0
A00120381FL204MYH7G3B     0.0     0.0     0.0     0.0     0.0     0.0     0.0
A00126503SUWI86KZBMIN     0.0     0.0     0.0     0.0     0.0     0.0     0.0
A001573229XK5T8PI0OKA     0.0     0.0     0.0     0.0     0.0     0.0     0.0
```

Now using this recommender system we can find similar users to a specific user:

```
Top 5 similar users for user_id: A0010876CNE3ILIM9HV0  are:  ['AXNF1BLDR4P47', 'ARTHT19OB79VZ', 'ARQ9I3Y0VPB6N', 'AOXEXSN7M
9ENJ', 'AN0AO97264HP4']
```

```
Top 5 productID recommended are:  [704, 122630, 119407, 119506, 119742]
```

	UserId	ProductId	real_rating	Timestamp	user_id	user	product	average_rating	normalized_rating
1160176	A0010876CNE3ILIM9HV0	B0055MYJ0U	1.0	1390521600	547427	11	136012	2.5	-1.5

After that we can recommend product using these similar users:

```
recommend_products_for_user("A2XVNI270N97GL", similarity)
```

```
[30773, 27327, 704, 119742, 120416]
```