Data Science

Assignment 5 – Recommendation System

Farimah Rashidi – 99222040

Amazon - Ratings (Beauty Products)

1. Introduction

This is a dataset related to over 2 million customer reviews and ratings of Beauty related products sold on their website. We want to create a recommendation system which suggests similar products to users based on their ratings.

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)

2. EDA and preprocessing

At the first let's see columns:

```
Column names: Index(['UserId', 'ProductId', 'Rating', 'Timestamp'],
dtype='object')
```

Now let's check data types:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2023070 entries, 0 to 2023069
Data columns (total 4 columns):
 #   Column     Dtype  
---  ------     -----  
 0   UserId     object 
 1   ProductId  object 
 2   Rating     float64
 3   Timestamp  int64  
dtypes: float64(1), int64(1), object(2)
memory usage: 61.7+ MB
```

As you can see, 'UserId' and 'ProductId' are object, 'Rating' is float and 'Timestamp' is integer.

Describe:

	Rating	Timestamp
count	2.023070e+06	2.023070e+06
mean	4.149036e+00	1.360389e+09
std	1.311505e+00	4.611860e+07
min	1.000000e+00	9.087552e+08
25%	4.000000e+00	1.350259e+09
50%	5.000000e+00	1.372810e+09
75%	5.000000e+00	1.391472e+09
max	5.000000e+00	1.406074e+09

In terms of 'Rating,' the data spans from a minimum of 1 to a maximum of 5, with a mean of approximately 4.15. The ratings exhibit a standard deviation of around 1.31, indicating a moderate level of variability. The timestamp values, representing time in seconds, range from a minimum of approximately 908,755,200 seconds to a maximum of about 1.406074e+09 seconds. The mean timestamp is roughly 1.36e+09 seconds, with a standard deviation of approximately 46,118,600 seconds. These descriptive statistics offer valuable insights into the central tendency, spread, and distribution of both 'Rating' and 'Timestamp,' providing a foundation for further exploration and interpretation of the dataset in my analysis.

We can see some first few rows:

	UserId	ProductId	Rating	Timestamp
0	A39HTATAQ9V7YF	0205616461	5.0	1369699200
1	A3JM6GV9MNOF9X	0558925278	3.0	1355443200
2	A1Z513UWSAAO0F	0558925278	5.0	1404691200
3	A1WMRR494NWEWV	0733001998	4.0	1382572800
4	A3IAAVS479H7M7	0737104473	1.0	1274227200
5	AKJHHD5VEH7VG	0762451459	5.0	1404518400
6	A1BG8QW55XHN6U	1304139212	5.0	1371945600
7	A22VW0P4VZHDE3	1304139220	5.0	1373068800
8	A3V3RE4132GKRO	130414089X	5.0	1401840000
9	A327B0I7CYTEJC	130414643X	4.0	1389052800
10	A1BG8QW55XHN6U	130414643X	5.0	1372032000
11	AIFAAVTUYHEHB	130414643X	4.0	1378252800
12	AVOGV98AYOFG2	1304146537	5.0	1372118400
13	A22VW0P4VZHDE3	130414674X	5.0	1371686400
14	AVOGV98AYOFG2	1304168522	5.0	1372118400
15	A6R426V4J7AOM	1304168522	5.0	1373414400

Now if we check data shape, we can see that this dataset has 2023070 rows.

Unique UserID and ProductID count:

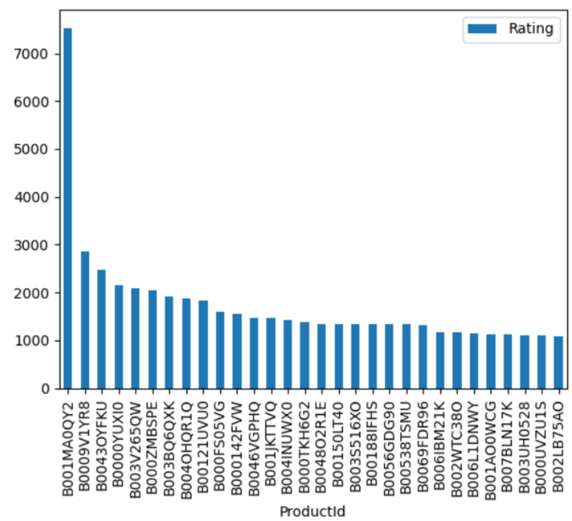
```
Unique UserID count: 1210271
Unique ProductID count: 249274
```

The dataset contains a total of 1,210,271 unique UserIDs, indicating the presence of a vast and diverse user base. Each UserID represents a distinct user who has engaged with the system in various ways. Additionally, the dataset encompasses 249,274 unique ProductIDs, showcasing a rich and varied assortment of products within the platform. These unique counts provide valuable insights into the scale and diversity of user-product interactions.

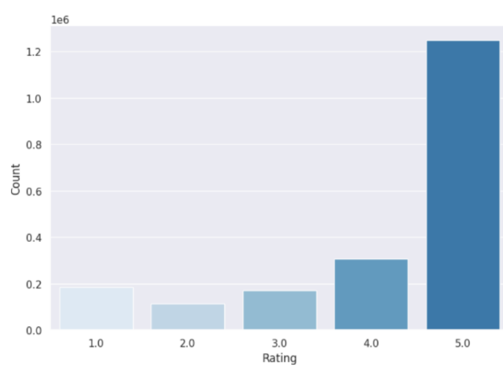
Most popular products:

ProductId	Rating
B001MA0QY2	7533
B0009V1YR8	2869
B0043OYFKU	2477
B0000YUXI0	2143
B003V265QW	2088
B000ZMBSPE	2041
B003BQ6QXK	1918
B004OHQR1Q	1885
B00121UVU0	1838
B000FS05VG	1589

distribution of the Ratings:



In this plot we can see distribution of ratings.



As we can see, 5 and 4 are the most common ratings that users have given to the products.

Users prefer to give higher scores compared to other scores. 2 and 3 are the least common which shows people are most likely to give the highest scores to their favorite product

3. Collaborative Filtering Method

At the first we create a utility matrix where each row represents a unique user (UserId), each column represents a unique product (ProductId), and the values are the corresponding ratings given by users to the products. If a user has not rated a product, the fill value is set to 0.

shape (number of rows and columns) of the ratings_utility_matrix:

```
(108983, 8137)
```

top 20 rows of the ratings_utility_matrix:

ProductId	0205616461	0558925278	0733001998	0737104473	0762451459	1304139212	1304139220	130414089X	130414643X	1304146537	...	B0006Q05XU	B0006Q06JS	B0006Q0HEC	B0006Q0ILY	B0006Q1KRK	B0006
UserId																	
A00205921JHJK5X9LNP42	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A004205218STRNUW6PPPA	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A00473363TJ8YSZ3YAGG9	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A00667432UL1ZRFLQA836	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A00700212KB3K0MVESPIY	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A0081289HG0BXFQJQUWW	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A01247753D6GFZD87MUV8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A01362343O2D2DRZLC42E	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A01379141PEJ6FIH7UH38	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A0143622X8ZC66HZXLUP	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A01437583CZ7V02UKZQ5S	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A01456542S5QPYUEGJXR8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A01884683H3F0505B7RAB	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A01907982I6OHXDYN5HD6	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A020135981U0UNEAE4JV	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A02027553MVF3OPLWDYPS	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A02278831YTIM059V25A7	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A02354170VQ79DHUZH39	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A024581134CV80ZBLIZTZ	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
A0254327142R60GSJIKIP	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

Then we transpose the matrix, swapping rows and columns using `.T`. This new matrix, `X`, has products as rows and users as columns. Let's look at this transposed matrix:

UserId	A00205921JHJK5X9LNP42	A004205218STRNUW6PPPA	A00473363TJ8YSZ3YAGG9	A00667432UL1ZRFLQA836	A00700212KB3K0MVESPIY	A0081289HG0BXFQJQUWW	A01247753D6GFZD87MUV8	A
ProductId								
0205616461	0	0	0	0	0	0	0	
0558925278	0	0	0	0	0	0	0	
0733001998	0	0	0	0	0	0	0	
0737104473	0	0	0	0	0	0	0	
0762451459	0	0	0	0	0	0	0	

5 rows × 108983 columns

After that we apply truncated Singular Value Decomposition (SVD) with 10 dimensions to reduce the dimensionality of the transposed matrix `X` to 10 components.

Now we calculate the correlation matrix based on the decomposed matrix. It measures the correlation between the different products based on user ratings.

I create a recommendor function which get an index `i`, the transposed matrix `X`, and the correlation matrix as inputs. it suggests the most similar products.

Let's try it:

```
recommended products list: B000141KTK
['1412759676', '6041134473', '604113449X', '6041134511', '8096399322', '8901110814', '9511181564', '9601403787', '9605406446', '974935706X', '9788071511', '9788072208', '9788073239', '978
8073409', '9788073417', '9788075622', '978807894X', '9788079970', '9788080669', '9788080928']
```

As you can see, our recommender system recommended some similar products.

Amazon - Ratings (Beauty Products)

1. Introduction

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.

We are tasked to build a content-based recommendation system for this dataset.

2. EDA and Preprocessing

Let's start with a quick look at our dataset:

```
<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
dtypes: float64(3), int64(1), object(6)
memory usage: 2.1+ MB
```

Columns are index, product, category, sub_category, brand, sale_price, market_price, type, rating and description. Each column's type can be seen in photo.

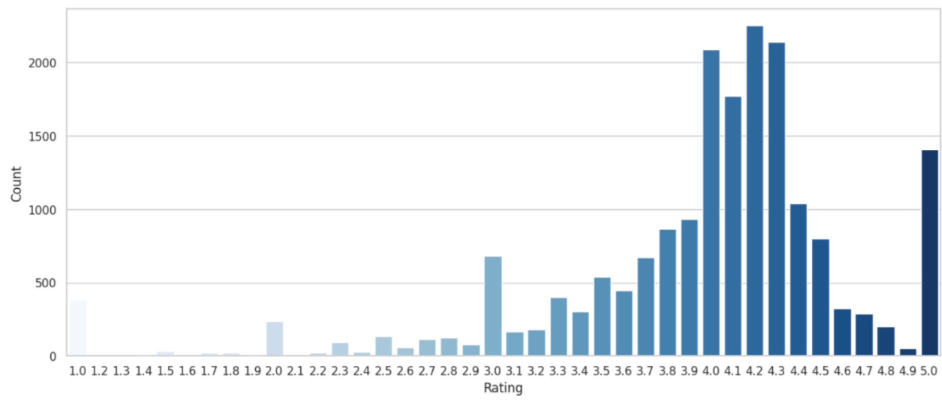
- Index: an index for each row
- Product: the name of the product
- Category: the category which the product falls in
- Sub_Category: the sub category which the product falls in
- Brand: the brand of the product
- Sale_Price: the price of the product in company
- Market_Price: the price of the product in Store
- Type: the type which the product falls in
- Rating: the rating given to the product
- Description: a short summary about the product

```
df.shape
```

```
(27555, 10)
```

It also shows that our dataset has 27555 rows.

Rating distribution:



It shows that most people rate products between 3.5 and 4.5. So, they prefer to give high rates around 4.

Now let’s check null values:

```
index              0
product            1
category           0
sub_category       0
brand              1
sale_price         0
market_price       0
type               0
rating          8626
description      115
dtype: int64
```

	column_name	percent_missing
index	index	0.000000
product	product	0.003629
category	category	0.000000
sub_category	sub_category	0.000000
brand	brand	0.003629
sale_price	sale_price	0.000000
market_price	market_price	0.000000
type	type	0.000000
rating	rating	31.304663
description	description	0.417347

We have around 9000 null values which will be handle after some visualizations to better understanding of dataset. You can see percentage of missing values in each column.

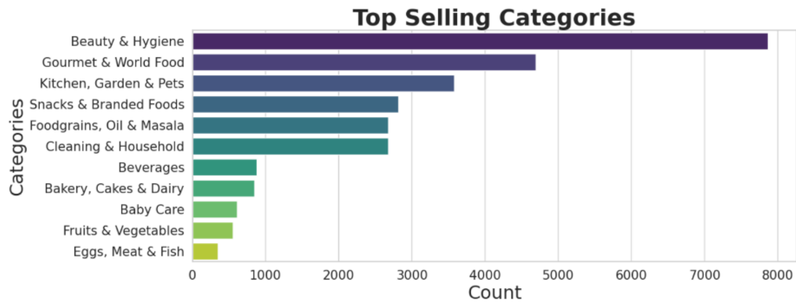
Now, let’s see about the categories.

Unique categories:

```
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
```

Top selling Categories:

The below image shows the top selling categories.

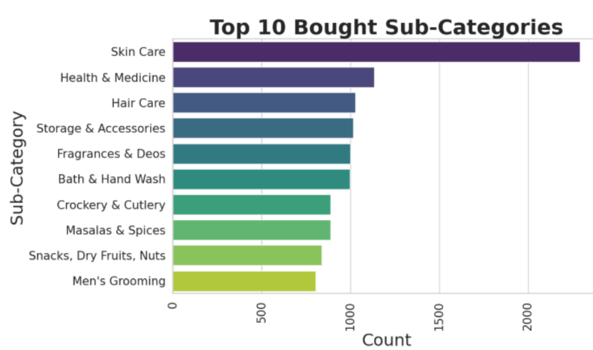


As we see the most sold products from **Beauty & Hygiene** category and that seems the interest of Indians. **Eggs, Meat & Fish** category has the least sold products that may indicate to low quality of these products.

Unique sub_categories:

```
Hair Care
Storage & Accessories
Pooja Needs
Bins & Bathroom Ware
Bath & Hand Wash
All Purpose Cleaners
Skin Care
Mops, Brushes & Scrubs
Cooking & Baking Needs
Chocolates & Biscuits
Fresheners & Repellents
Snacks, Dry Fruits, Nuts
Dairy & Cheese
Pasta, Soup & Noodles
Dry Fruits
Drinks & Beverages
Kitchen Accessories
Flask & Casserole
Breakfast Cereals
Frozen Veggies & Snacks
Fruit Juices & Drinks
Cookies, Rusk & Khari
Fragrances & Deos
Tea
Masalas & Spices
Men's Grooming
Chocolates & Candies
Steel Utensils
Tinned & Processed Food
Organic Staples
Sauces, Spreads & Dips
Pickles & Chutney
Ready To Cook & Eat
Baby Bath & Hygiene
Stationery
Pet Food & Accessories
Biscuits & Cookies
Oral Care
Snacks & Namkeen
Detergents & Dishwash
Crockery & Cutlery
Cuts & Sprouts
Health & Medicine
Cookware & Non Stick
Dairy
Feminine Hygiene
Diapers & Wipes
Edible Oils & Ghee
Baby Food & Formula
Fresh Fruits
Fresh Vegetables
Herbs & Seasonings
Breads & Buns
Oils & Vinegar
Feeding & Nursing
Energy & Soft Drinks
Appliances & Electricals
Salt, Sugar & Jaggery
Gourmet Breads
Organic Fruits & Vegetables
Indian Mithai
Fish & Seafood
Sausages, Bacon & Salami
Disposables, Garbage Bag
Dals & Pulses
Noodle, Pasta, Vermicelli
Rice & Rice Products
Cakes & Pastries
Spreads, Sauces, Ketchup
Cereals & Breakfast
Party & Festive Needs
Eggs
Health Drink, Supplement
Non Dairy
Bakeware
Exotic Fruits & Veggies
Baby Accessories
Coffee
Makeup
Atta, Flours & Sooji
Car & Shoe Care
Mutton & Lamb
Gardening
Ice Creams & Desserts
Bakery Snacks
Water
Mothers & Maternity
Marinades
Pork & Other Meats
Flower Bouquets, Bunches
```

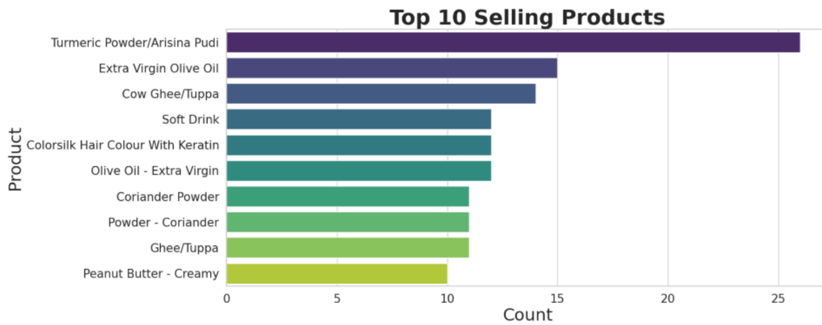
Top sub_categories:



Its clear that **Skin Care** and **Health & Medicine** are likely to being bought by Indian people.

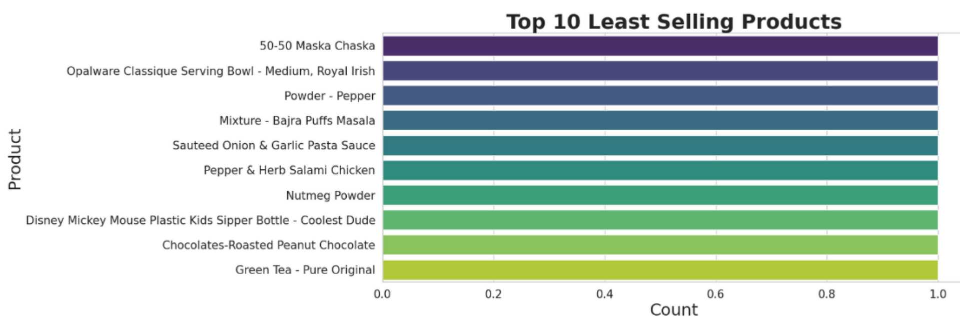
Also, there are a lot of unique data types which you can see in my notebook.

Top 10 Selling Products:



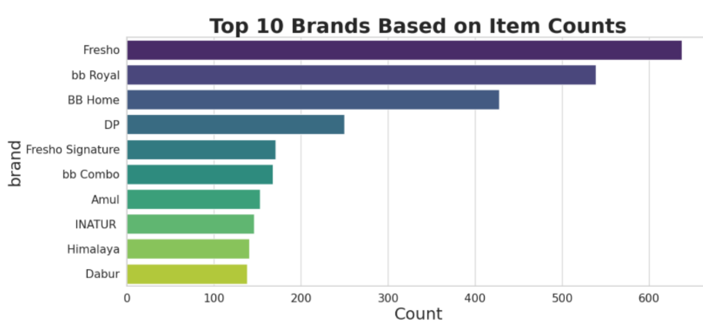
As we see from above analysis that bigbasket supermarket customers have interest in **Foodgrains, Oil & Masala** category. **Turmeric** is most sold product as it's on **Foodgrains, Oil & Masala** we can expect that Indian is interest with haircare.

Top 10 Least Selling Products:



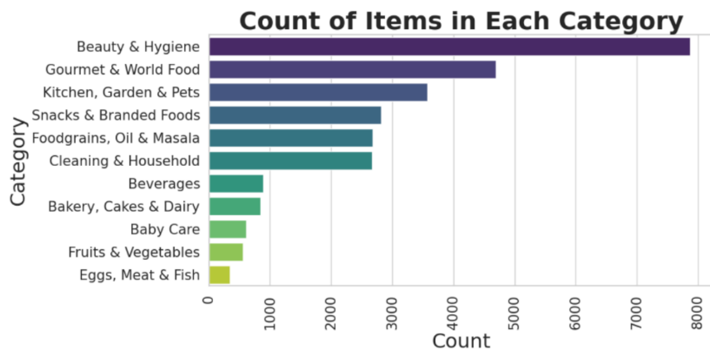
As we see, least selling products are **Green Tea, Chocolates** and…

Top 10 Brands Based on Item Counts



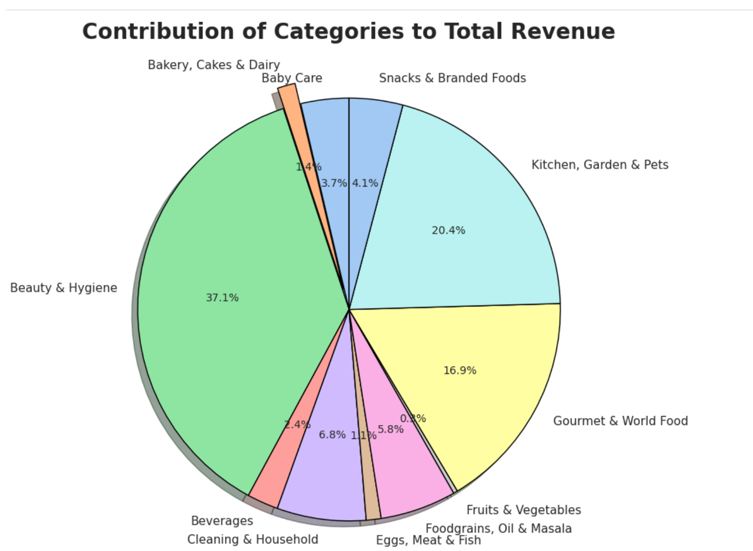
Brand **Fresho** has the highest number of Product Types, followed by **bb Royal** and **BB Home**.

Count of Items in Each Category



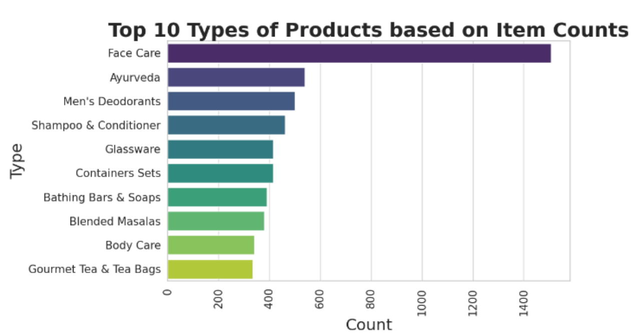
Beauty & Hygiene is the category which has the most items in.

Contribution of Categories to Total Revenue



We can see which category makes the most money. **Beauty & Hygiene** is the most money maker category. After that, it seems that **Kitchen, garden & Pets** category makes a lot of money too.

Top 10 Types of Products based on Item Counts



Face Care product type has the most count of items.

After analyzing all these visualizations, now it's time to drop all rows containing null values.

3. Content based recommender

At the first, we use some text cleaning functions to remove leading and trailing whitespaces from a string and split a string using specific delimiters and remove leading/trailing whitespaces. These functions will return a list of cleaned items.

Then we initialize a text cleaning function that takes either a list or a string as input. If it's a list, it converts each element to a lowercase string and removes spaces. If it's a string, it converts it to a lowercase string and removes spaces. We apply it to the 'category', 'sub_category', 'type', and 'brand' columns in the dataset.

After that we make a function which combines multiple columns ('category', 'sub_category', 'brand', 'type') into a single column called 'keywords' by joining their values.

Then we convert the 'keywords' into a sparse matrix of token counts called `count_matrix`.

`cosine_similarity` computes the cosine similarity between vectors in `count_matrix`.

The result is a similarity matrix (`cosine_sim`), where each entry (i, j) represents the cosine similarity between product i and product j.

Then we reset DataFrame index, and we create a Series called 'indices' with the product names as the index and their corresponding indices as values.

After that we create recommender function which takes a product title and cosine similarity matrix as input. It retrieves the index of the input product, computes the cosine similarity scores, sorts them, and returns the top 10 recommended products.

Let's test our model with some examples:

Product: Brass Angle Deep - Plain

```
['Brass Kachua Stand Deepam - No.1' 'Brass Angle Deep Stand - Plain, No.2'
 'Brass Lakshmi Deepam - Plain, No.2' 'Brass Kuber Deepam - No.1'
 'Brass Deepa Matki - Round, No.3' 'Brass Kuber Deepam - No.2'
 'Brass Deepa Matki - Round, No.1' 'Brass Angle Deep Stand - Plain, No.3'
 'Brass Angle Deep Stand - Plain, No.1' 'Brass Kachua Stand Deepam - No.2']
```

Product: Water Bottle - Orange

```
['Glass Water Bottle - Aquaria Organic Purple'
 'Glass Water Bottle With Round Base - Transparent, B1364'
 'H2O Unbreakable Water Bottle - Pink' 'Water Bottle H2O Purple'
 'H2O Unbreakable Water Bottle - Green'
 'Regel Tritan Plastic Sports Water Bottle - Black'
 'Apsara 1 Water Bottle - Assorted Colour'
 'Glass Water Bottle With Round Base - Yellow, B1363'
 'Trendy Stainless Steel Bottle With Steel Cap - Steel Matt Finish, PXP 1002 CV'
 'Penta Plastic Pet Water Bottle - Violet, Wide Mouth']
```

Product: Powder - Pepper

```
['Powder - Coriander' 'Turmeric Powder/Arisina Pudi' 'Hing'
 'Powder - Chilly' 'White Pepper Powder' 'Powder - Cumin'
 'Masala - Paneer Butter' 'Turmeric Powder/Arisina Pudi'
 'Masala - Brahim Sambar' 'Asafoetida Powder']
```

Product: Peri-Peri Sweet Potato Chips

```
['High Protein Soya Chips' 'Chia Seeds Chips'
 'Peri-Peri Sweet Potato Chips' 'Sour Cream & Onion'
 'Nacho Chips - Cheese With Herbs, No Onion, No Garlic'
 'Nacho Crisps - Cheese & Herbs' 'Shells - Taco'
 '6 Corn Wraps Try It With Prawns & Avocado'
 'On The Go - Peri Peri Nachos & Salsa Dip'
 'Chips - Keralas Nendran Banana']
```