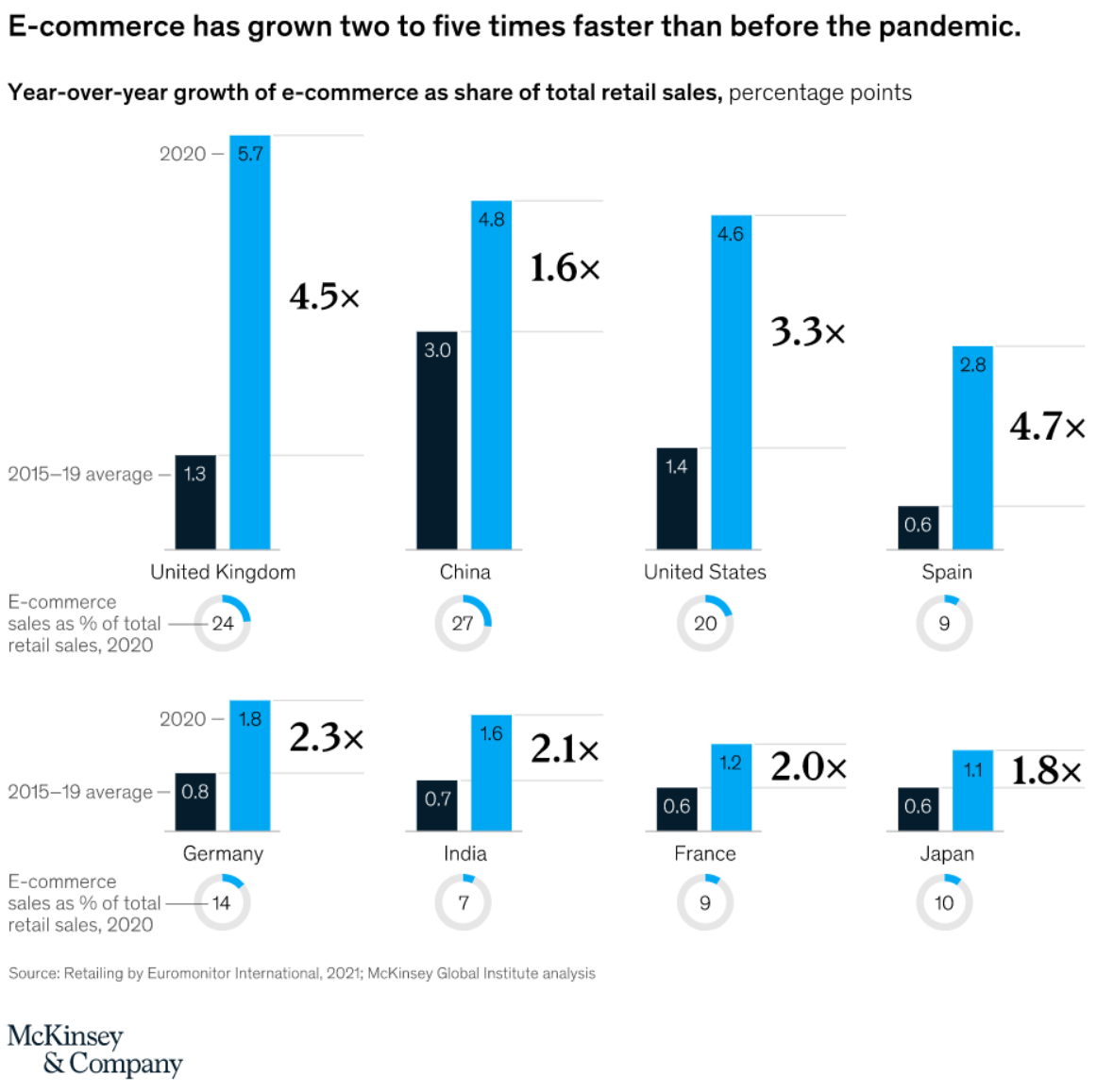
Content-Based Recommendation System

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.

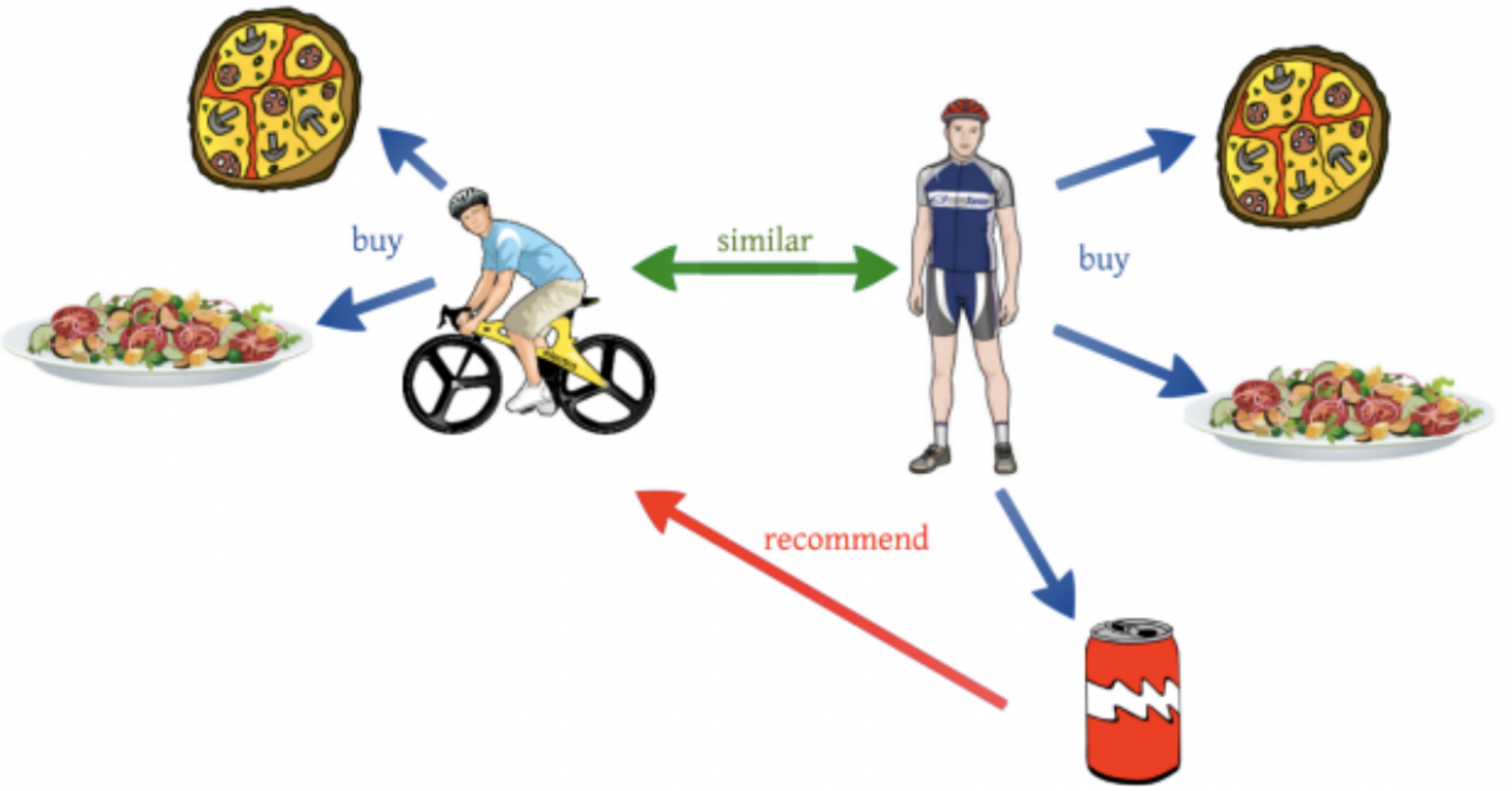
- Having a genre of action
- Having Jackie Chan in the cast.

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.

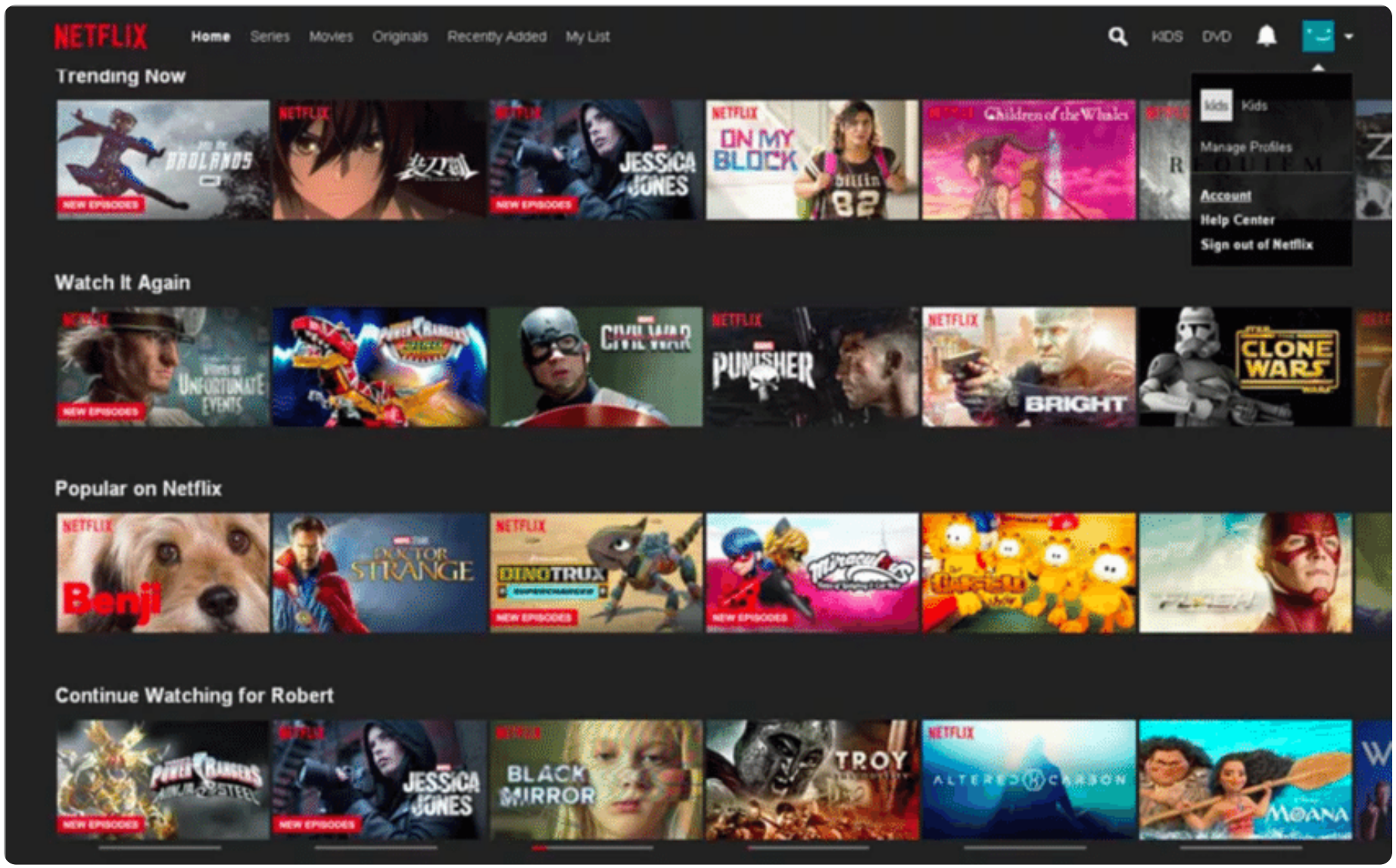


In the illustration above, there are 2 users with similar taste preferences. Both of them liked pie and protein salad and looked like fitness enthusiasts, so they are similar. Now, the user on the right liked a can of energy drink, so we recommend the same energy drink to the user on the left.

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

What is Bigbasket?

Bigbasket is one of the biggest online grocery stores in India. It was launched in 2011, and several competitors have challenged it for market share. However, it can still retain its fair share in the market.

Data for Recommendation System

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

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

1. index – the serial number
2. product – Title (or name) of the product
3. category – Category of the product
4. sub_category – Subcategory of the product
5. brand – Brand of the product
6. sale_price – Price at which product is being sold on the site
7. market_price – The market price of the product
8. type – Type into which product falls
9. rating – aggregate product rating (out of 5) by customers
10. description – Description of the product

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:

```
Null Data Count In Each Column
------------------------------
product            1
category           0
sub_category       0
brand              1
sale_price         0
market_price       0
type               0
rating          8626
description      115
dtype: int64
------------------------------
Null Data % In Each Column
------------------------------
product : 0.00
category : 0.00
sub_category : 0.00
brand : 0.00
sale_price : 0.00
market_price : 0.00
type : 0.00
rating : 31.30
description : 0.42
```

Findings from the EDA:

- There is a product without a name.
- There is a product without a brand.
- 115 products do not have a description.
- 8626 products do not have ratings.

The above features are important for building a recommendation system. So rows from data that contain missing values has been dropped.

Even after dropping nulls, we have a good data size of 18840 records.

Understanding Data Types of Columns

```
product        object
category       object
sub_category   object
brand          object
sale_price     float64
market_price   float64
type           object
rating         float64
description    object
```

Findings from this step:

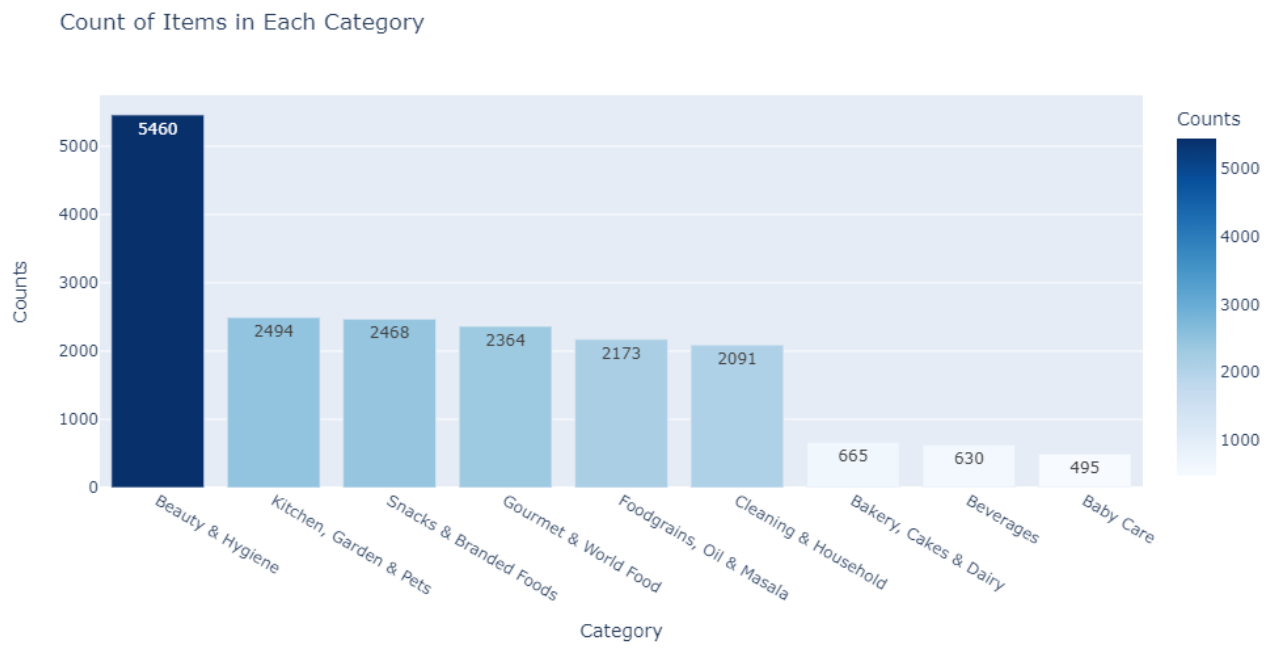
The features sale_price, market_price, and rating are numeric (as they are represented using float64). The rest are all string features (represented as objects).

Univariate Analysis in Recommendation System

understanding the data distribution of several columns. A look at category column distribution:

A look at category column distribution

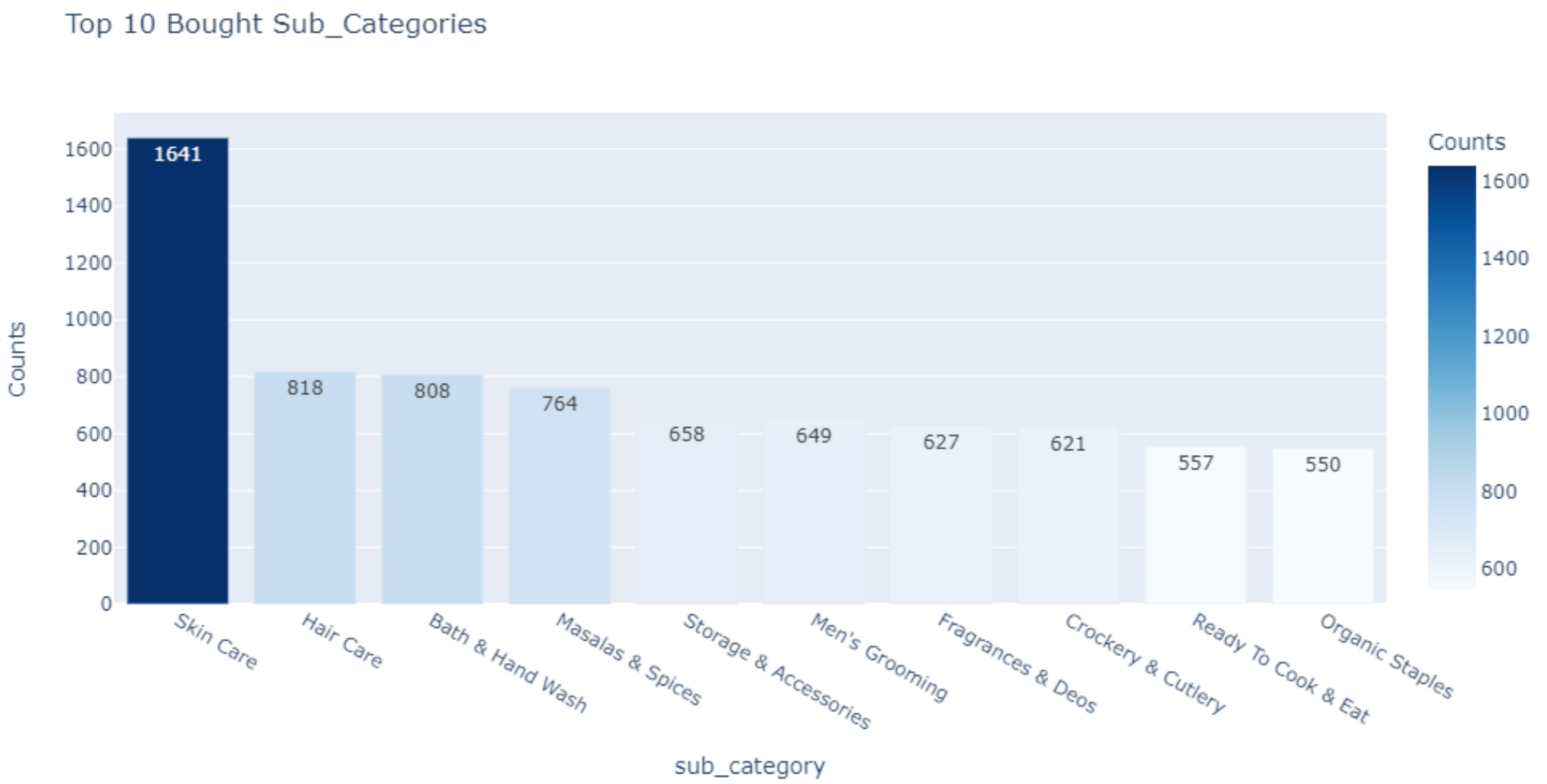
	Category	Counts	Percent
0	Beauty & Hygiene	5460	28.98
1	Kitchen, Garden & Pets	2494	13.24
2	Snacks & Branded Foods	2468	13.10
3	Gourmet & World Food	2364	12.55
4	Foodgrains, Oil & Masala	2173	11.53
5	Cleaning & Household	2091	11.10
6	Bakery, Cakes & Dairy	665	3.53
7	Beverages	630	3.34
8	Baby Care	495	2.63

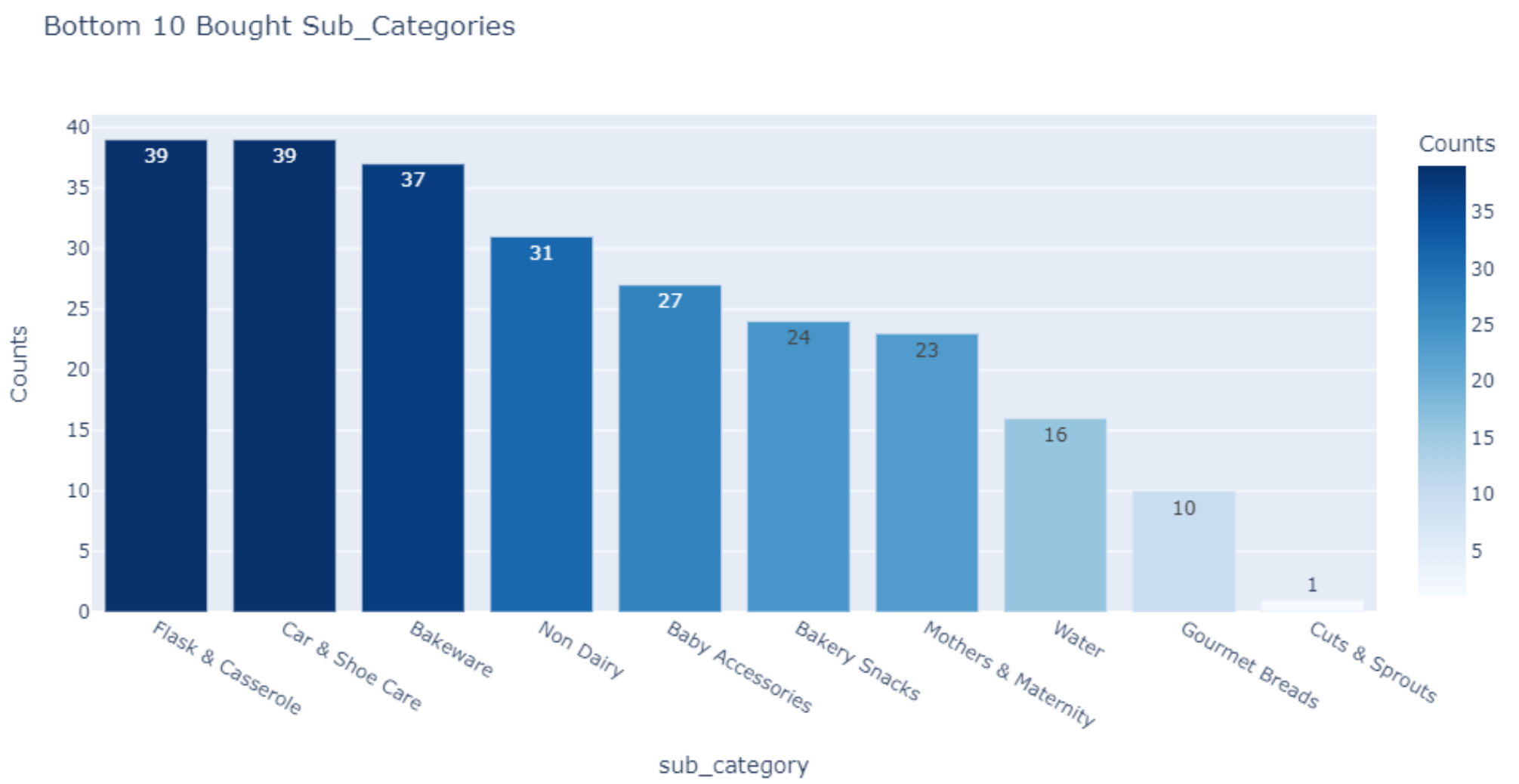


Findings:

- Beauty and Hygiene have a total of 5460 products. It covers 28.98% of the total product portfolio.
- Next, the best category is Kitchen, Garden, and Pets, which has 2494 products. It covers 13.24% of the total product portfolio.
- Baby care has the lowest product count of 495 products. It covers 2.63% of the total product portfolio.

sub_category column distribution

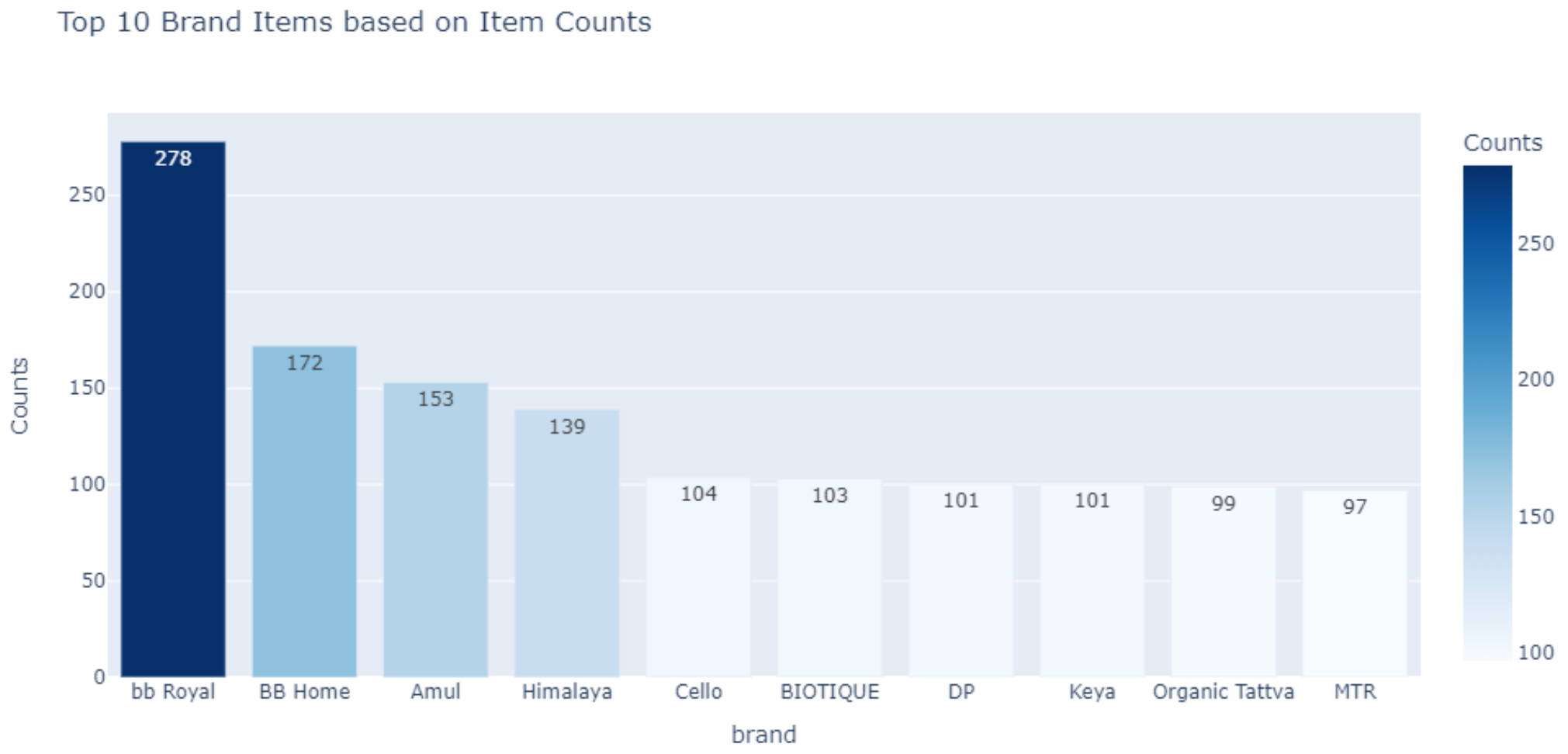




Findings:

- There are 77 unique sub_category values.
- Skin Care has a total of 1641 products. It covers 8.71% of the total product portfolio.
- The next best sub_category is Hair Care which has a total of 818 products. It covers 4.34% of the total product portfolio.
- Cuts & Sprouts has the lowest product count of 1 product. It barely covers 0.01% of the total product portfolio.

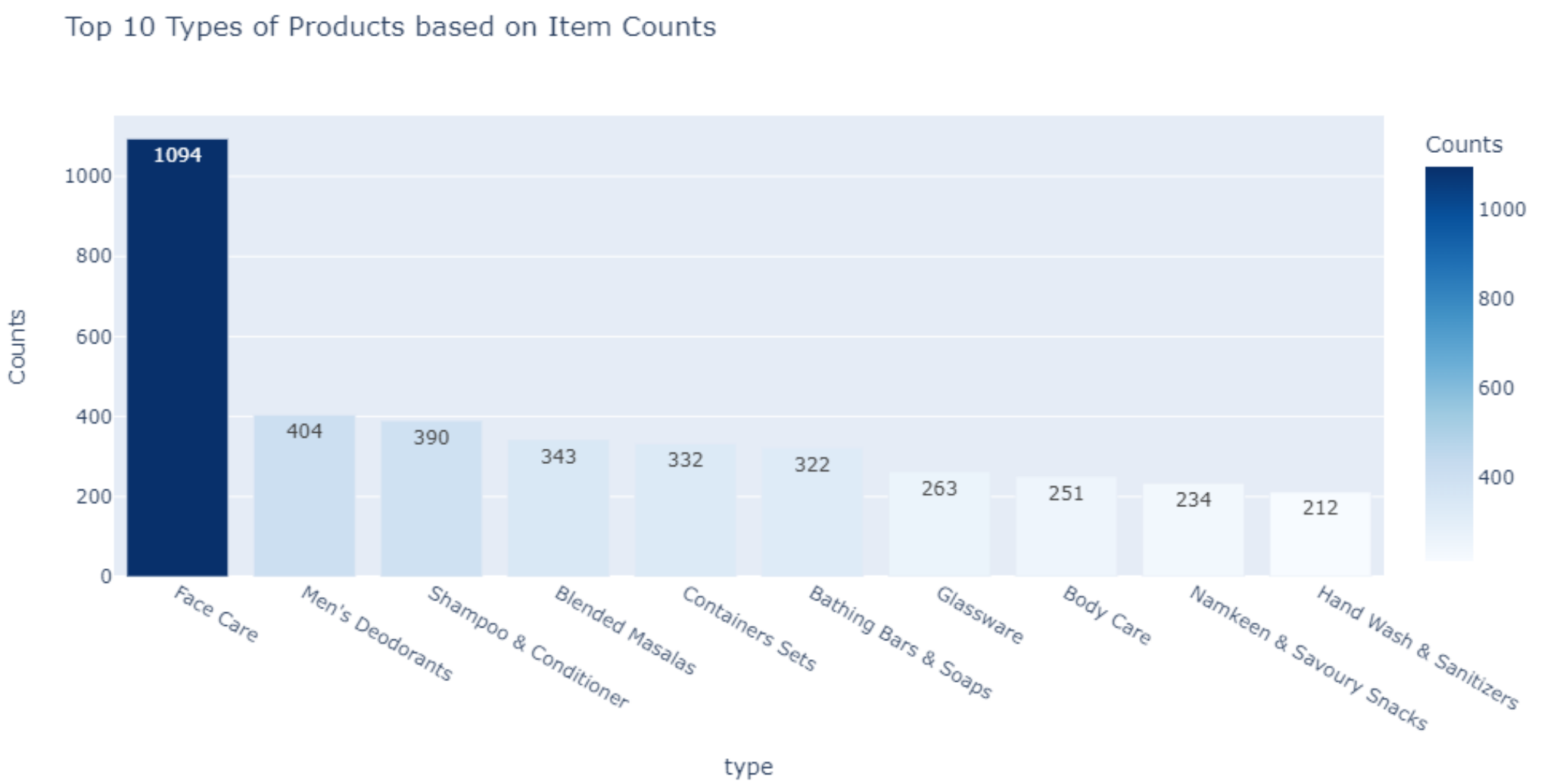
Brand column distribution



Findings:

- There are 1933 unique values for brands.
- There are 494 brands having a single product.
- The top 2 brands are of BigBasket.
- bb Royal has a total of 278 products. It covers 1.48% of the total product portfolio.
- The next best brand is BB Home, which has 172 products. It covers 0.91% of the total product portfolio.

Type column distribution



Findings:

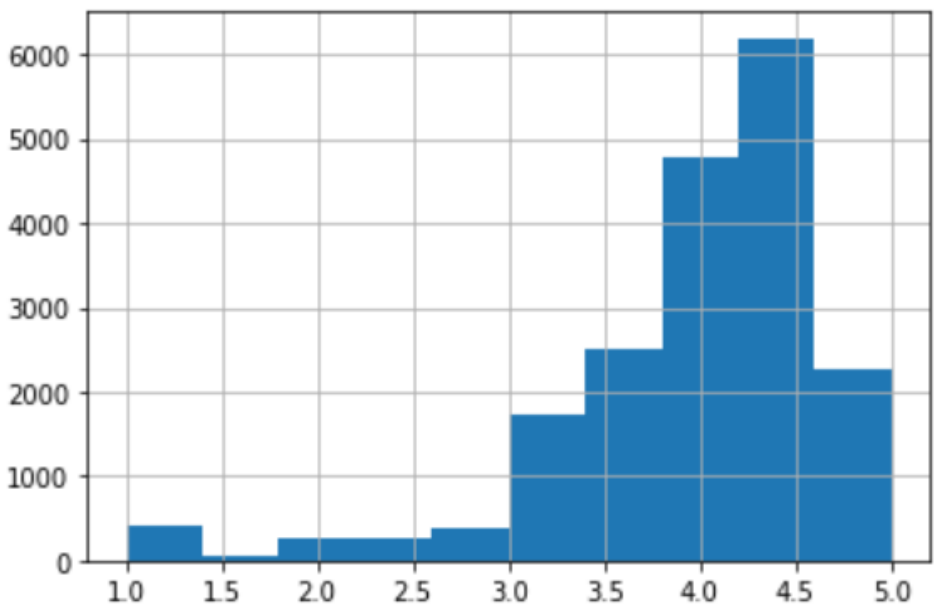
- There are 358 unique values for type.
- There are 9 types having a single product.
- Face Care has a total of 1094 products. It covers 5.81% of the total product portfolio.
- Next, the best type is Men’s Deodorant, which has 404 products. It covers 2.14% of the total product portfolio.

Ratings Analysis

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

```
count    18840.000000
mean         3.943063
std          0.739646
min          1.000000
25%          3.700000
50%          4.100000
75%          4.300000
max          5.000000
Name: rating, dtype: float64

<AxesSubplot:>
```



It is clear that histogram is skewed towards the right, which means that most of the products have a higher rating.

How many Ratings are between the interval of 0 to 1, 1 to 2, and so on?

```
rating
(0, 1]     387
(1, 2]     335
(2, 3]    1347
(3, 4]    6559
(4, 5]   10212
```

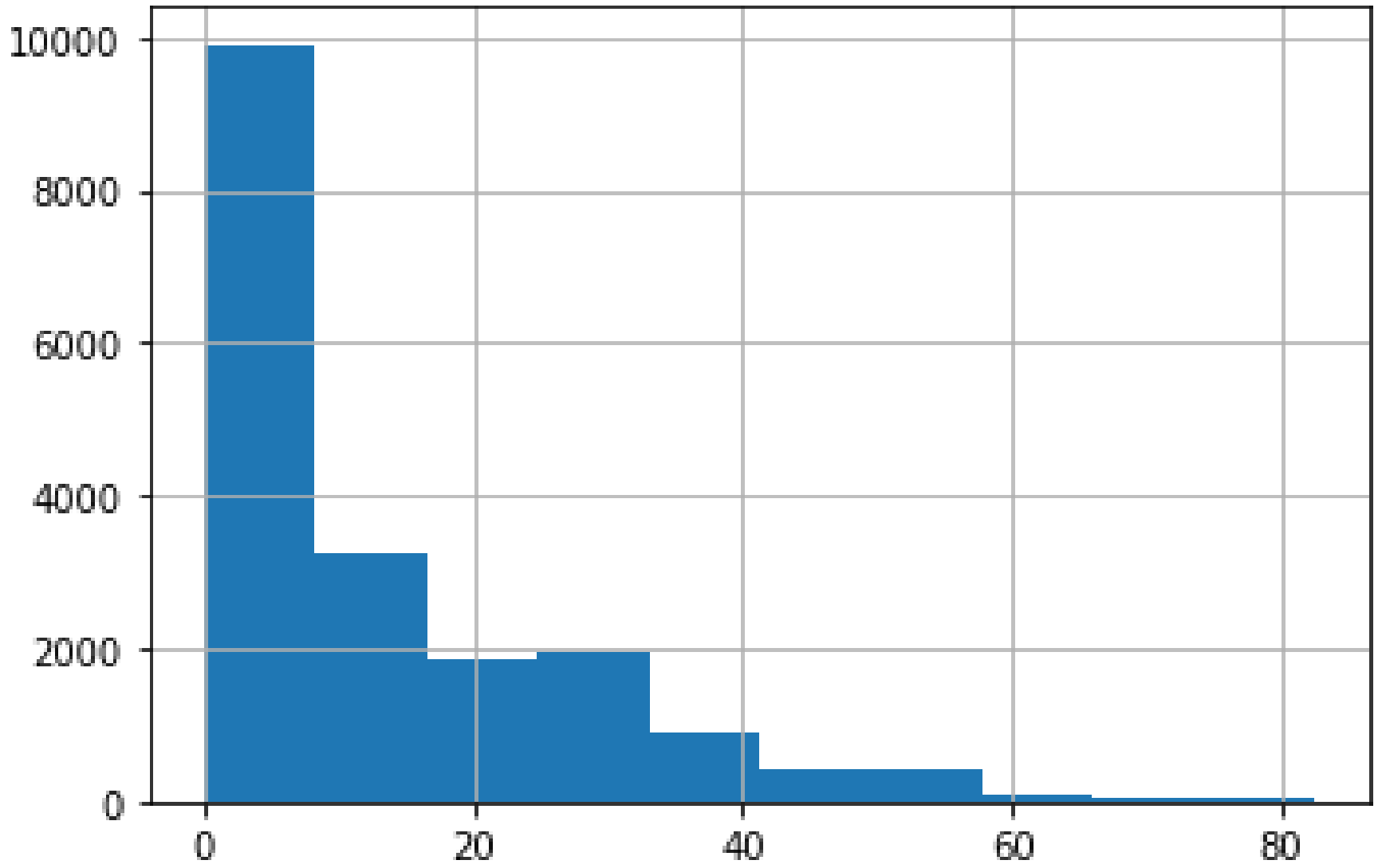
Findings:

- 10212 products had a rating between 4 and 5.
- 387 products had a rating between 0 and 1.

Feature Engineering

Here, created new features which can improve the recommendations. the sale_price and market_price are given. Let’s create a feature discount%
Formula = [(market_price – sale_price)/sale_price] * 100

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

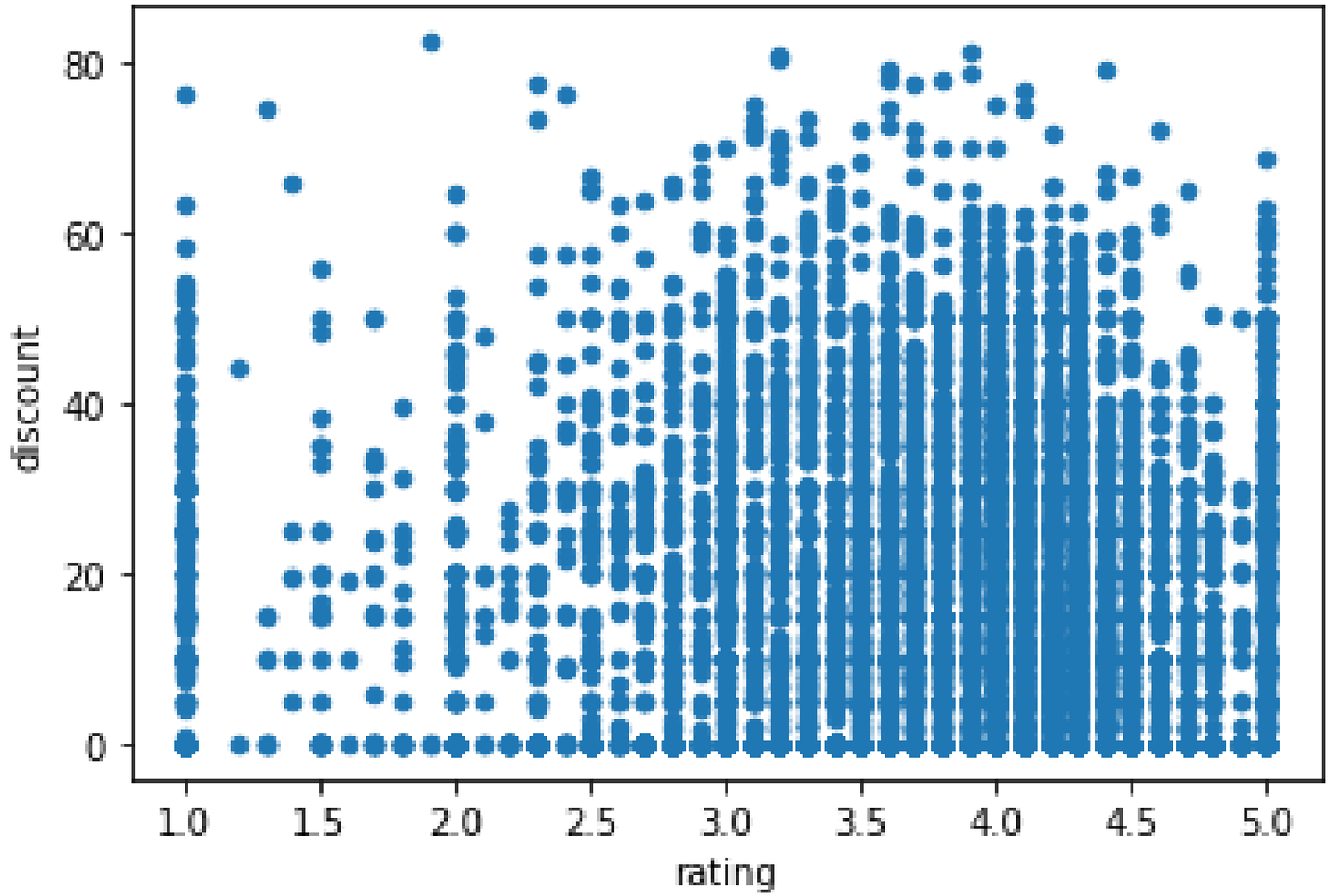


Findings:

- 8157 products do not have any discount.
- At least 4 products have above 80% discount.

Bivariate Analysis

Is there any relation between rating and discount?



Findings: Looking at the scatter plot, we do not see any association between rating and discount.

Let’s Clean a Few String Columns

The category column contains ‘Kitchen, Garden & Pets.’ We will clean it, so it contains [kitchen, garden, pets]. A similar transformation will be applied to sub_category, type, and brand features. Now, let’s create one feature (product_classification_features) which appends all the above 4 cleaned columns.

Now, we have the data ready. Let us build a simple logic for recommendations based on popularity. We will use the recommender feed – type, category, or sub_category. It will return the most popular product or the product having the highest rating.

Let’s look at the most popular products for the category = Beauty & Hygiene

---

index	product	category	rating
20164	Supreme Scalp Rejuvenation Shampoo	Beauty & Hygiene	5.0
5472	Vitamin C Brightening Day Cream With SPF 30 UV...	Beauty & Hygiene	5.0
20936	Exfoliating Face Scrub	Beauty & Hygiene	5.0
5499	Prickly Heat Powder - Cool Chandan With Sandal...	Beauty & Hygiene	5.0
20860	Wheat Grass Powder	Beauty & Hygiene	5.0

Let’s look at the most popular products for sub_category = Hair Care

index	product	sub_category	rating
14617	Black Pearl Shampoo - Hair Loss & Dandruff Con...	Hair Care	5.0
8275	De-Tangling Comb - 1 Row, 1266	Hair Care	5.0
14556	Argan Oil Shampoo	Hair Care	5.0
14441	Blooming Colour Shampoo - Murumuru Butter & Ro...	Hair Care	5.0
13070	Hair Repair Conditioner	Hair Care	5.0

Let’s look at the most popular products for the brand = Amul

index	product	brand	rating
16467	Vanilla Milkshake	Amul	5.0
10431	Spray Infant Milk Food/Substitute	Amul	4.5
5857	Almondo - Roasted Almonds Coated With Milk Cho...	Amul	4.5
9400	Amulya Dairy Whitener	Amul	4.4
20842	Peru Dark Amazon, Single Origin Dark Chocolate...	Amul	4.4

Let’s look at the most popular products for type = Face Care

index	product	type	rating
16906	Nutritivo Pomegranate Radiant Glow Firming Serum	Face Care	5.0
25347	Hydro Replenish Refreshing Face Mist	Face Care	5.0
9286	Bio Morning Nectar Flawless Skin Cream	Face Care	5.0
3356	Total Effects Whip - UV SPF 30	Face Care	5.0
3369	Organic Shield - Anti Tan Facial Kit	Face Care	5.0

Content-based Recommendation System

As the name suggests, these algorithms use the data of the product we want to recommend. E.g., Kids like Toy Story 1 movies. Toy Story is an animated movie created by Pixar studios – so the system can recommend other animated movies by Pixar studios like Toy Story 2. For our, e.g., we will use the product metadata like category, sub_category, type, price, etc. We will extract similar products from the product portfolio based on these features. The feature we created earlier (product_classification_features) will be helpful here.
We will use CountVectorizer to create a feature space.

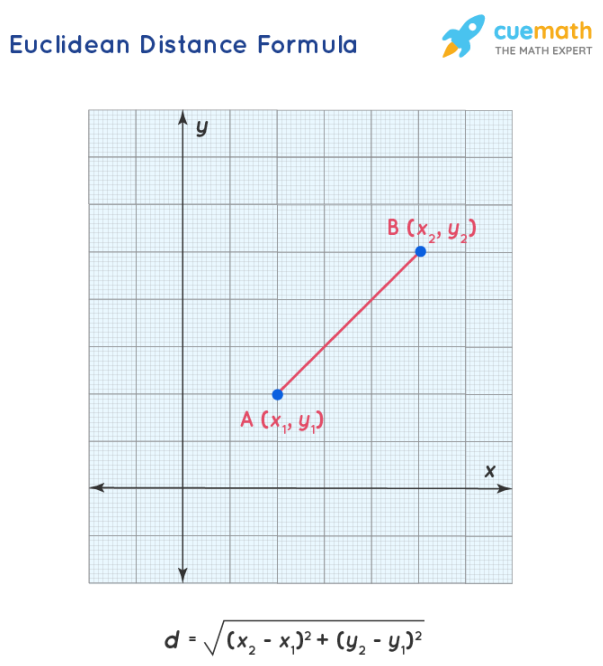
`s1 = ‘Ram is a boy`
`s2 = ‘Ram is good`
`s3 = ‘Good is that boy.’`
`So, we first convert them into lowercase`
`s1 = ‘ram is boy’`
`s2 = ‘ram is good’`
`s3 = ‘good is that boy’`

> So, Vocabulary consists of {‘boy’, ‘good’, ‘is’, ‘ram’, ‘that’} as unique words.
> So, when you use CountVectorizer, it creates 5 features, each representing the count of occurrence of that word.

	Boy	Good	Is	Ram	That
0	1	0	1	1	0
1	0	1	1	1	0
2	1	1	1	0	1

Once we have these vectors, we can use the below similarity metrics for identifying and recommending similar products

1. Euclidean Distance: It measures the straight line distance between the 2 vectors.



For our example, Euclidean Distance looks like the below:

```
array([[0.  , 1.  , 1.73],
       [1.  , 0.  , 1.41],
       [1.73, 1.41, 0.  ]])
```

In the above example, all diagonal entries are 0, which is intuitive – each sentence's Euclidean Distance is 0. Also, Euclidean Distance between

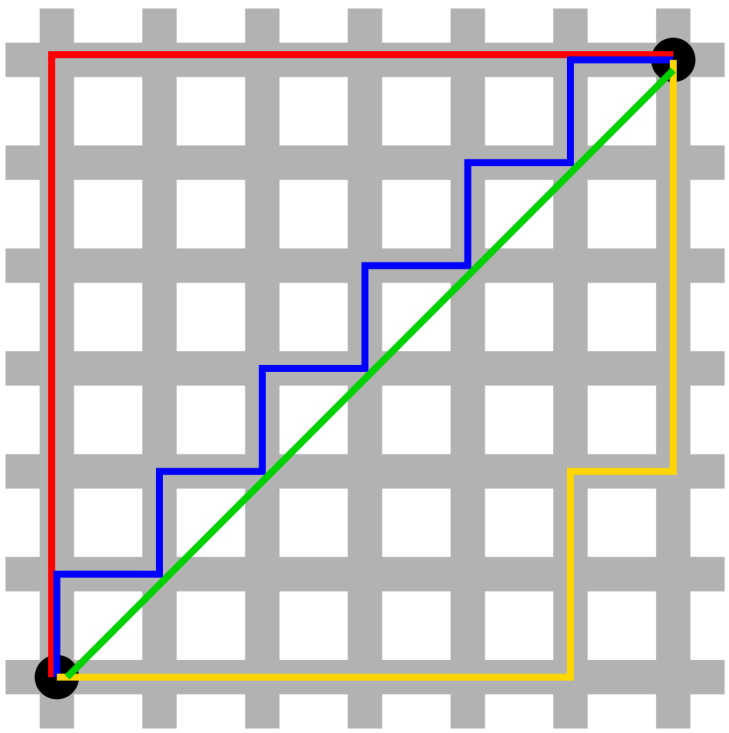
- s1 and s2 = 1
- s2 and s3 = 1.41
- s1 and s3 = 1.73

So, s1 is closer to s2

s2 is closer to s1

s3 is closer to s2

Manhattan Distance: It measures the absolute differences between the two vectors.



- The green line represents Euclidean distance.
- The red, blue, and yellow lines represent Manhattan distance.
- It is a vehicle's travel route to reach from one location to another.

For our example, Manhattan Distance looks like the below:

```
array([[0, 2, 3],
       [2, 0, 3],
       [3, 3, 0]])
```

all diagonal entries are 0, which is intuitive – the Manhattan Distance of each sentence with itself is 0. Also, Manhattan Distance between

- s1 and s2 = 2
- s2 and s3 = 3
- s1 and s3 = 3

So, s1 is closer to s2
s2 is closer to s1
s3 is equidistant from s2 and s1

Jaccard Distance: It measures the ratio of common words between 2 sentences to the overall unique words in those 2 sentences.

For s1 and s2, it is calculated as
s1 and s2 = len({'is','ram'})/len({'boy','is','ram','good'}) = 0.5
For our example, Jaccard Distance looks like the below:

```
array([[0. , 0.5, 0.6],
       [0.5, 0. , 0.6],
       [0.6, 0.6, 0. ]])
```

So, Jaccard's distance between

- s1 and s2 = 0.5
- s2 and s3 = 0.6
- s1 and s3 = 0.6

So, s1 is closer to s2

s2 is closer to s1

s3 is equidistant from s2 and s1

Cosine Distance (or cosine similarity) we will use cosine similarity for identifying and recommending similar products.
Cosine similarity measures the angle between the 2 vectors.
Cosine similarity can give a value between -1 to +1. A value of -1 means that the products are opposite or non-similar and a value of +1 means that the 2 products are the same.
Cosine similarity looks like below:

```
array([[1.  , 0.67, 0.58],
       [0.67, 1.  , 0.58],
       [0.58, 0.58, 1.  ]])
```

In the above example, all diagonal entries are 1, which is intuitive – each sentence's cosine similarity is 1.

Also, cosine similarity between

- s1 and s2 = 0.67
- s2 and s3 = 0.58
- s1 and s3 = 0.58

So, s1 is closer to s2

s2 is closer to s1

s3 is equidistant from s2 and s1

Let’s calculate the cosine similarity of the product_classification_features for all the products.

Let’s see the recommendations for a few products.

```
title = 'Water Bottle - Orange'
content_recommendation_v1(title)
```

Output:

	product	cosine_similarity
109	Glass Water Bottle – Aquaria Organic Purple	0.875
705	Glass Water Bottle With Round Base – Transparent...	0.875
1155	H2O Unbreakable Water Bottle – Pink	0.875
1500	Water Bottle H2O Purple	0.875
1828	H2O Unbreakable Water Bottle – Green	0.875
1976	Regel Tritan Plastic Sports Water Bottle – Black	0.875
2182	Apsara 1 Water Bottle – Assorted Colour	0.875
2361	Glass Water Bottle With Round Base – Yellow, B...	0.875
2485	Trendy Stainless Steel Bottle With Steel Cap -...	0.875

Findings:

- In this example, we can recommend bottles. However, we are recommending all the colors.
- Also, the cosine similarity is the same for all the recommendations.

```
title = 'Dark Chocolate- 55% Rich In Cocoa'
content_recommendation_v1(title)
```

Output:

	product	cosine_similarity
105	I Love You Fruit N Nut Chocolate	1.0
1144	Choco Cracker – Magical Crystal With Milk Choc...	1.0
1718	Dark Chocolate – Single Origin, India	1.0
2517	Fruit N Nut, Dark Chocolate- 55% Rich In Cocoa	1.0
3117	Colombia Classique Black, Single Origin Dark C...	1.0
3167	Dark Chocolate	1.0
4013	Almondo – Roasted Almonds Coated With Milk Cho...	1.0
4358	Sugar-Free Dark Chocolate	1.0
5867	Milk Compound Slab – MCO-11	1.0

Findings:

- In this example, we can recommend all types of Chocolates.
- However, the top recommendation should be dark chocolate.
- Also, the cosine similarity is the same for all the recommendations.

Improving the Model

Let us tweak the algorithm. Let us also use the product column to create another cosine similarity. We will take the average of both the cosine similarity and see if the results are better.

Output:

	product	sim
669	Sante Infuser Water Bottle – Orange	0.824798
2024	H2o Unbreakable Water Bottle – Orange	0.824798
2565	Swat Pet Water Bottle – Orange	0.824798
1912	Glass Water Bottle – Circo Orange & Lemon	0.791053
2084	Spray Glass water Bottle With Cork – Orange	0.791053
1924	Sip-It-Plastic Water Bottle	0.726175
1997	Water Bottle – Twisty, Pink	0.726175
1290	Plastic Water Bottle – Pink	0.726175
195	Water Bottle H2O Purple	0.726175
1863	Water Bottle – Apsara 1 Assorted Colour	0.695699

Findings: The results are much better for this. The orange color bottles have a higher similarity.

```
title = 'Dark Chocolate- 55% Rich In Cocoa'
content_recommendation_v2(title)
```

Output:

	product	sim
437	Fruit N Nut, Dark Chocolate- 55% Rich In Cocoa	0.922577
2340	Sugar-Free Dark Chocolate- 55% Rich In Cocoa	0.922577
2717	Rich Cocoa Dark Chocolate Bar	0.837500
561	Dark Chocolate	0.816228
504	Dlite Rich Cocoa Dark Chocolate Bar	0.802648
3137	Bitter Chocolate- 75% Rich In Cocoa	0.800000
2544	Peru Dark Amazon, Single Origin Dark Chocolate...	0.782843
3148	Bournville Rich Cocoa 70% Dark Chocolate Bar	0.775562
548	Colombia Classique Black, Single Origin Dark C...	0.769680
2941	Tanzania Chocolat Noir, Single Origin Dark Cho...	0.769680

Findings: Even here, we see Dark chocolate products having higher similarities.

```
title = 'Nacho Round Chips'
content_recommendation_v2(title)
```

Output:

	product	sim
3766	Nacho Chips – Crunchy Pizza	0.788675
718	Nacho Chips – Jalapeno	0.770833
3630	Nacho Chips – Cheese	0.770833
1680	Nacho Chips – Salsa	0.770833
464	Nacho Chips – Peri Peri	0.735702
3140	Nacho Chips – Sweet Chilli	0.726175
3912	Nacho Chips – Roasted Masala	0.726175
4478	Nacho Chips – Jalapeno, No Onion, No Garlic	0.695699
1233	Nacho Chips – Peri Peri	0.673202
86	Nacho Chips – Cheese With Herbs, No Onion, No ...	0.673202

```
title = 'Chewy Mints - Lemon'
content_recommendation_v2(title)
```

Output:

	product	sim
452	Chewy Candy Stick – Strawberry Flavour	0.557671
324	Orbit Sugar-Free Chewing Gum – Lemon & Lime	0.537680
1558	Chewy Mints – Spearmint Flavour Candy	0.525460
711	Chewing Gum – Peppermint	0.500000
2676	Chewing Gum – Peppermint	0.500000
1734	Chewing Gum – Peppermint	0.500000
2842	Orange Chewy Dragees	0.433928
877	Sugarfree Strawberry	0.428571
1063	White Xylitol Sugarfree Spearmint Flavour Chew...	0.428571
1543	Mint – Sugarfree, Peppermint Flavour	0.428571

```
title = 'Veggie - Fingers'
content_recommendation_v2(title)
```

Output:

	product	sim
1186	Veggie Fingers – Veggie Delight with Corn, Car...	0.853553
980	Veggie – Nuggets	0.750000
814	Veggie Burger Patty	0.704124
2639	Veggie – Nuggets	0.687500
331	Veggie Stix	0.687500
2656	Veggie Pizza Pocket	0.641624
840	Veggie Burger Patty	0.641624
1231	Crispy Veggie Burger Patty	0.614277
1082	Chicken Fingers – Garlic	0.557678
1121	Quick Snack – Fish Fingers	0.530330

The recommendation results look relevant.

Conclusion

Big companies like Amazon and Netflix leverage recommendation systems to boost revenue by keeping customers engaged with relevant content. The report delves into various recommendation system types, conducts a thorough Exploratory Data Analysis (EDA), and builds a content-based recommendation system. The choice of cosine similarity as a metric helps identify similarities between products for effective recommendations.

Key takeaways:

- For a cold start problem, we should use Popularity Based recommendation system. Recommends the most popular or most selling products. This gives generic recommendations to all users.
- We can use a content-based recommendation system if we want to recommend similar products to a set of users.
- Once we have a good amount of user preference data, we can build Collaborative-based recommendations. Here, we give a personalized recommendation to each user.