Implementing a content-based recommendation system

Sarah Hosseini 400222026

Shahid Beheshti university

1. Exploratory data analysis

1.1. Data Exploration

In this report, we present the findings obtained from the dataset at

'https://www.kaggle.com/datasets/surajjha101/bigbasket-entire-product-list-28k-datapoints/data'.

This dataset has 27555 records with 10 fields:

- index - this is just the index so we drop it later.
- product - Title of the product. Not sorted.
- 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 customers
- description - Detailed description of the data

We have 354 duplicated records to remove. The count of unique values in each column is listed below:
product = 23541
category = 11
sub_category = 90
brand = 2314
sale_price = 3256
market_price = 1348
type = 426
rating = 41
description = 21945

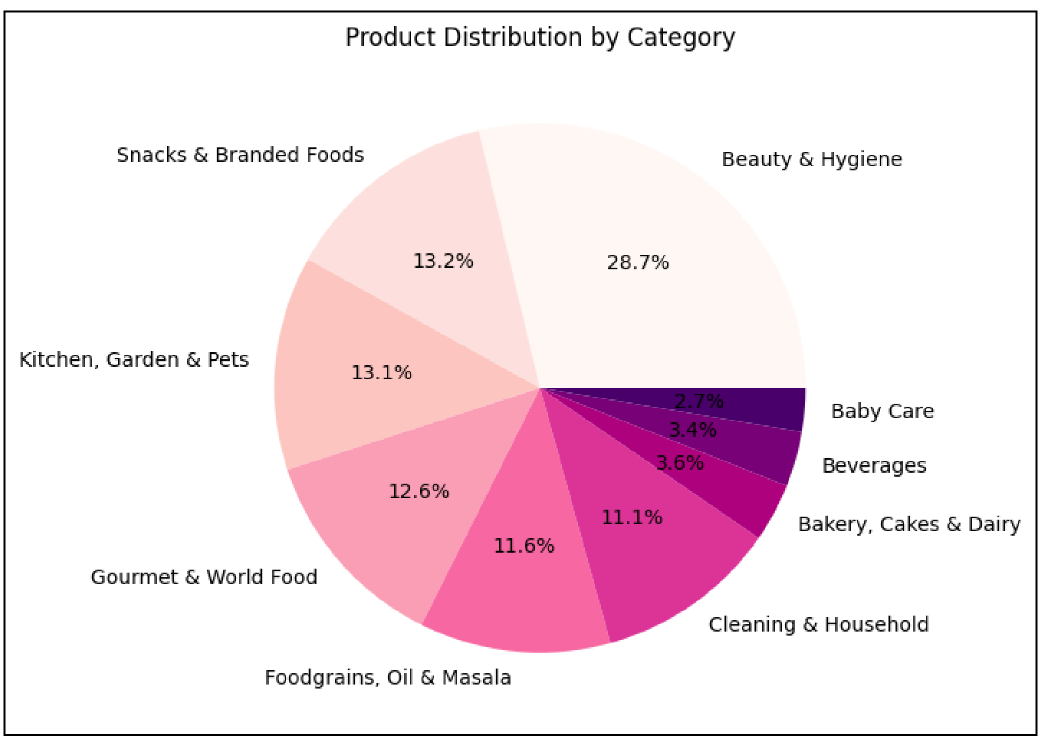
Count of null values in the columns are:

product	1
category	0
sub_category	0
brand	1
sale_price	0
market_price	0
type	0
rating	8463
description	113

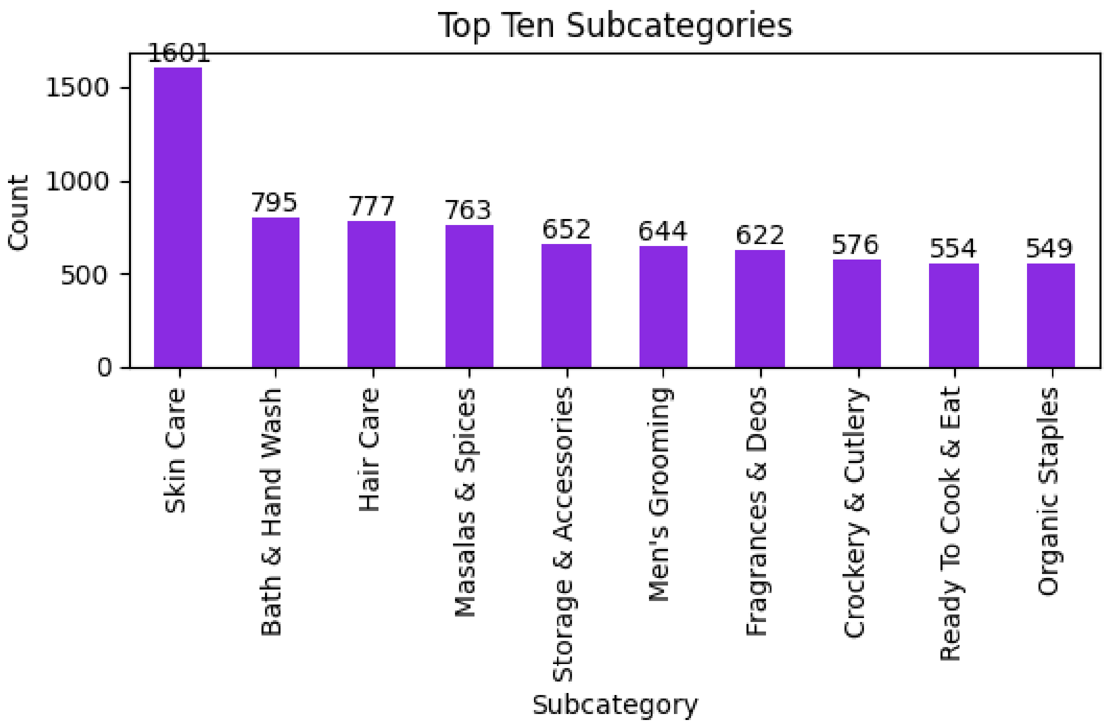
This suggests that product 31.11 percent of ratings are missing. Now we have the option to impute them or simply drop those rows. Dropping missing values allows us to focus on the available data without the complexities of imputation methods. This simplicity can be advantageous when dealing with large datasets or when time is a constraint. Also, removing them ensures that our analysis is based on the most accurate and complete information available. After dropping them, our dataset is now of shape (18650, 9).

1.2. Univariate data analysis

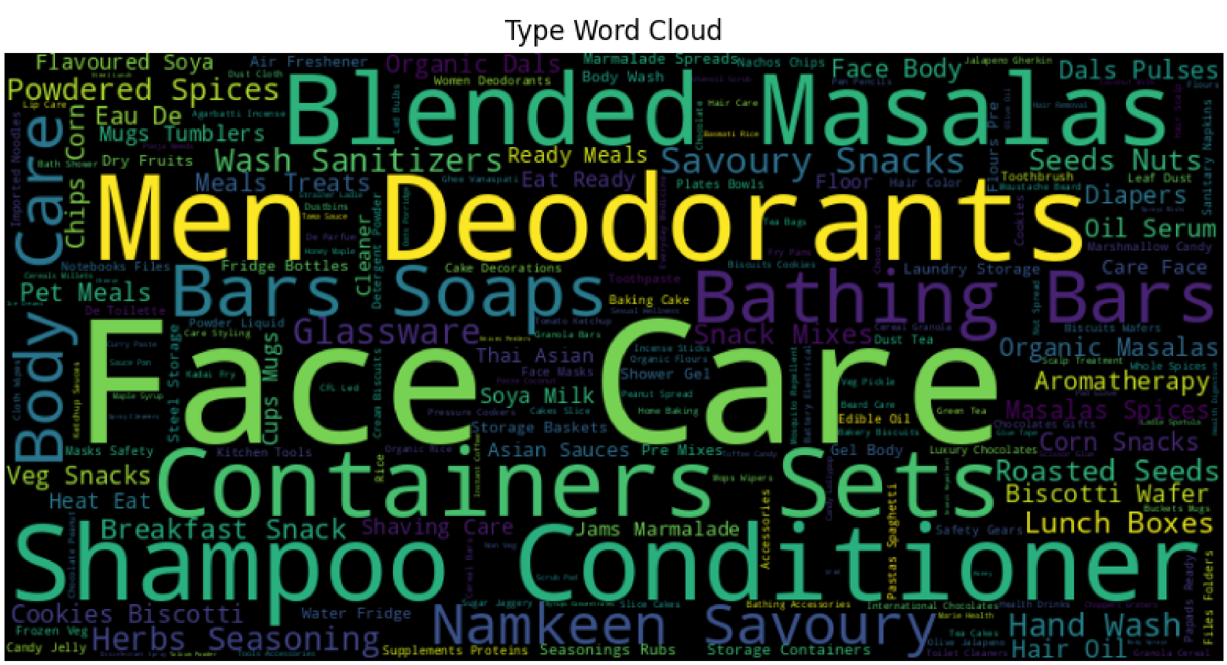
The distribution of the data in categories looks like this:



This plot shows the subcategories with the highest number of products:

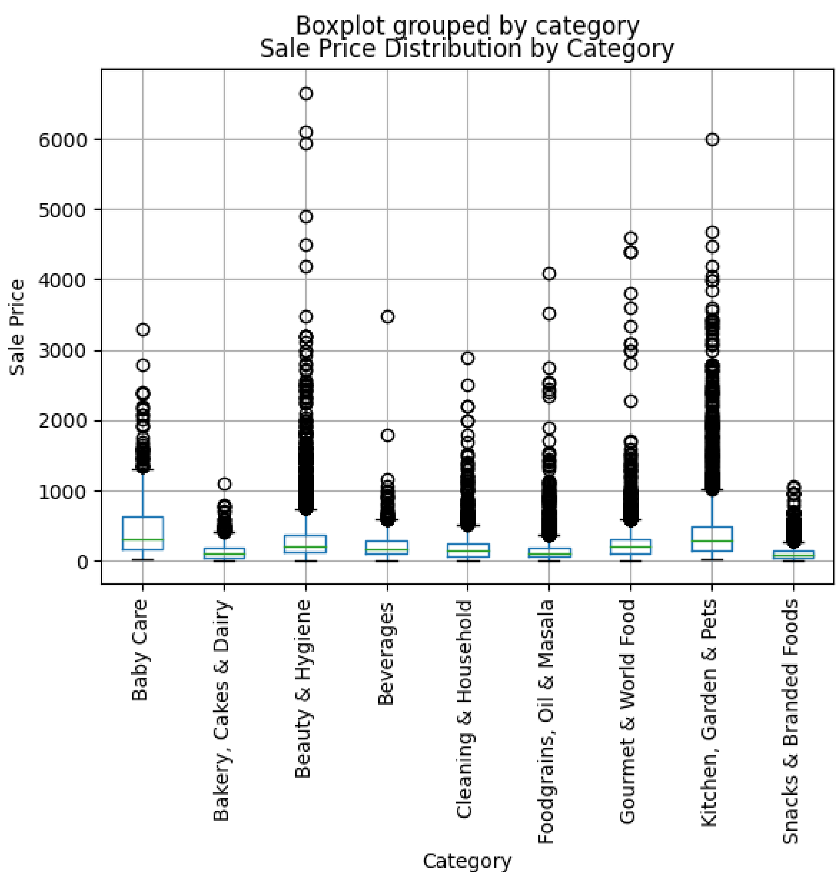


Since the unique values in Type column are too many, we demonstrate them with a wordcloud:



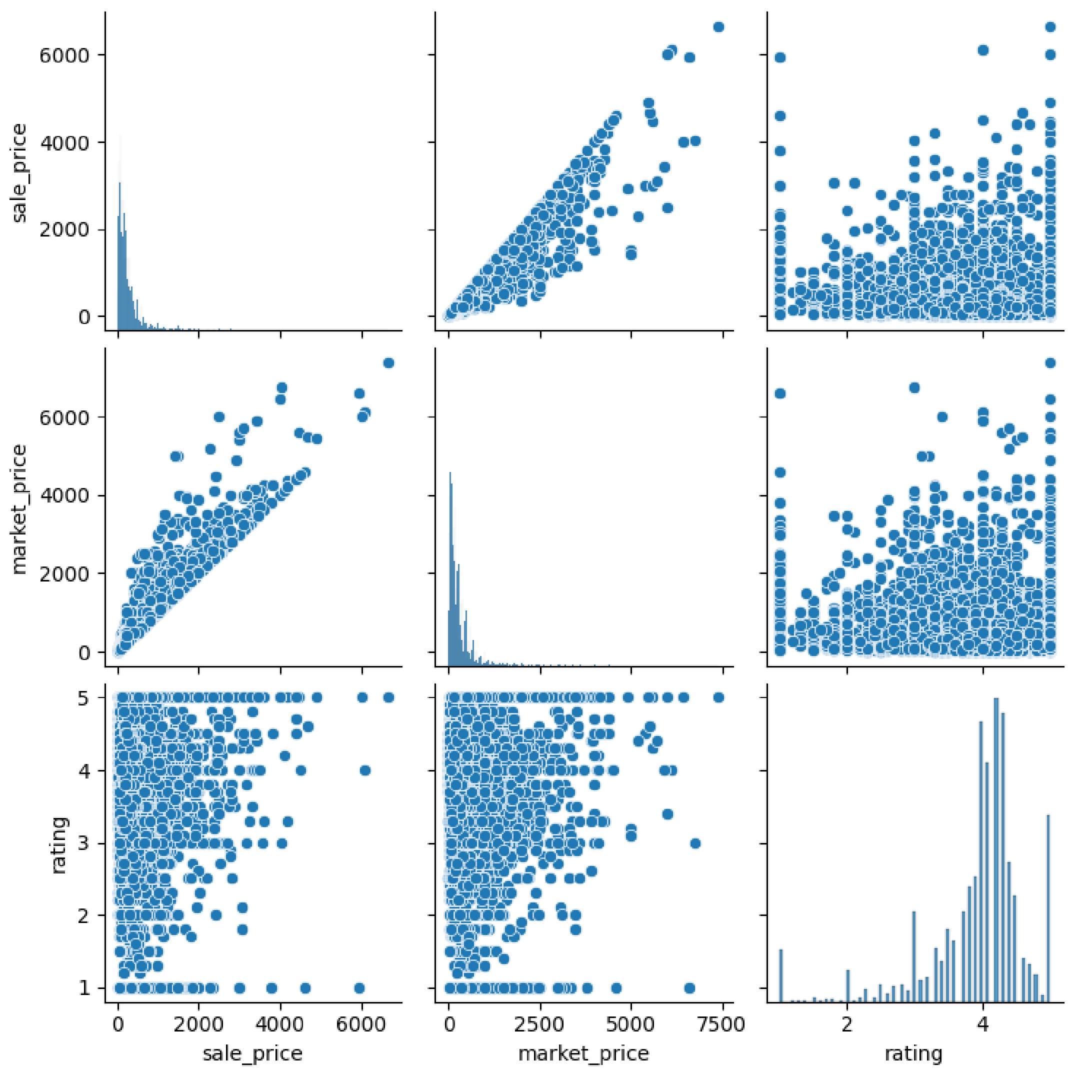
The distribution of products's prices:

1.3. Multivariate data analysis



This plot shows that despite a wide range of sale prices in each of the categories, all categories have mean prices of approximately the same values.

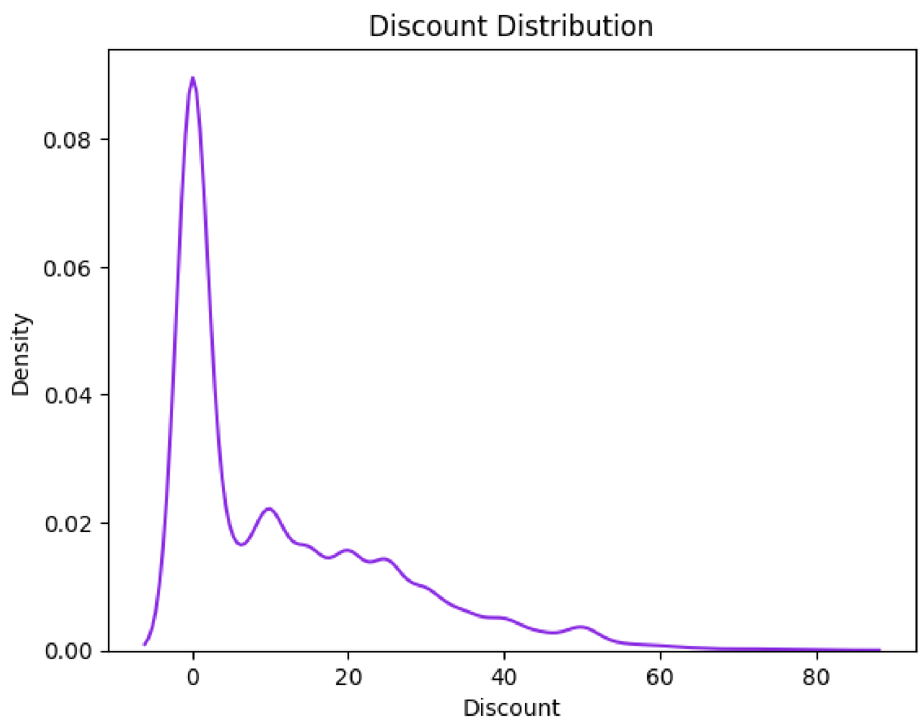
The pairplot of numerical variables along with the correlation plot show that ratings have no significant correlation to prices:



2. Feature engineering

We create a new feature 'discount' which is (('market_price'-'sale_price')/'market_price')*100.

This is the plot of the density of it:



We also created a new feature 'tags' that is the concatation of 'category' , 'subcategory', 'brand', and 'type'. This feature is used in the recommender to calculate the similarity of the items.

3. Model

In this section, we used two methods to build our recommender engine. In the first method, we used TF-IDF vectorizer and in the second CountVectorizer.

TF-IDF (Term Frequency-Inverse Document Frequency) and CountVectorizer are both commonly used techniques in natural language processing (NLP) for transforming text data into numerical representations. However, they have different approaches and serve different purposes. CountVectorizer focuses on word frequencies, while TF-IDF considers both term frequency and inverse document frequency to capture the importance of words.

At first, we created a feature matrix from the 'description' column. Then, we used linear kernel to calculate the similarities of items.

For the product ‘Water Bottle - Orange’, our recommender system results are as listed below:

11320	Rectangular Plastic Container - With Lid, Mult...
11642	Jar - With Lid, Yellow
26451	Round & Flat Storage Container - With lid, Green
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
26067	Premium Round Plastic Container With Lid - Mul...
26074	Premium Round Plastic Container With Lid - Pink
8588	Plastic Container - Square, Pink
10707	Plastic Round Glass With Lid - Yellow
13533	Plastic Round Glass With Lid - Pink
15863	Container - Square, Tower Shape, Blue

In the second method, we used CountVectorizer to convert ‘tags’ feature to numerical values.

Then, calculated the cosine similarity matrix. The results for the same product are:

139	Glass Water Bottle - Aquaria Organic Purple
1038	Glass Water Bottle With Round Base - Transpare...
1701	H2O Unbreakable Water Bottle - Pink
2209	Water Bottle H2O Purple
2704	H2O Unbreakable Water Bottle - Green
2908	Regel Tritan Plastic Sports Water Bottle - Black
3225	Apsara 1 Water Bottle - Assorted Colour
3481	Glass Water Bottle With Round Base - Yellow, B...
3669	Trendy Stainless Steel Bottle With Steel Cap -...

3708	Penta Plastic Pet Water Bottle - Violet, Wide ...
3834	Glass Water Bottle With Maroon Cap - BB1245MRN
3930	Loopy Pet water Bottle - Violet
3935	Ivory Premium Glass Bottle - With Yellow Floral
3976	Double Walled Glass Bottle With Cream Cap - BB...

Since we had used more features, the second method's results seem to be more accurate.

In conclusion, this report has explored various aspects of text processing and similarity measures in natural language processing (NLP). We discussed the importance of transforming text data into numerical representations for various NLP tasks. CountVectorizer, a simple technique, captures the frequency of words in a document, while TF-IDF considers both term frequency and inverse document frequency to capture word importance. The choice between these techniques depends on the specific task and data characteristics. Additionally, we examined cosine similarity as a measure of similarity between vectors, particularly suited for TF-IDF representation. However, other similarity measures can also be used based on the requirements of the task. Understanding these techniques and their applications provides a foundation for effective text analysis, document retrieval, and recommendation systems in the field of NLP.