Amazon - Ratings (Beauty Products) & BigBasket Entire Product List Recommendation Systems

Mohammad Rouintan

Computer Science, Shahid Beheshti University

Abstract

This report presents two recommendation systems employed by prominent e-commerce platforms, namely Amazon and BigBasket. The focus is on the ratings of beauty products on Amazon and the entire product list recommendation system of BigBasket. The report begins by providing an overview of Amazon's recommendation engine, which heavily relies on customer ratings and purchase history to offer personalized recommendations. The dataset used for analysis consists of over 2 million customer reviews and ratings of beauty products on Amazon's website. Additionally, the report discusses the BigBasket platform, the largest online grocery supermarket in India, and its comprehensive product dataset. The dataset contains attributes such as product title, category, subcategory, brand, price, rating, and description. Two methods employed in the recommendation systems are explored: content-based methods, utilizing techniques like TfidfVectorizer and CountVectorizer, and collaborative filtering methods, specifically Singular Value Decomposition (SVD). The report concludes by highlighting the effectiveness and adaptability of these methods in enhancing the recommendation systems of Amazon and BigBasket.

Introduction

The rise of e-commerce has revolutionized the way businesses operate, with companies like Amazon and BigBasket leading the way in providing convenient online shopping experiences. These platforms employ sophisticated recommendation systems to personalize the shopping journey for their customers. In this report, we delve into the specifics of the recommendation systems used by Amazon and BigBasket, focusing on the ratings of beauty products on Amazon and the entire product list recommendation system of BigBasket.

Amazon, a global e-commerce and cloud computing giant, relies heavily on its recommendation engine to enhance sales and customer satisfaction. The engine analyzes customer ratings and purchase history to generate personalized recommendations. The dataset used for analysis comprises over 2 million customer reviews and ratings of beauty products sold on Amazon's website. This dataset includes unique identifiers for customers (UserId), products (ASIN), ratings on a scale of 1 to 5, and timestamps.

On the other hand, BigBasket, the largest online grocery supermarket in India, offers a wide range of products for customers to choose from. Their recommendation system aims to guide customers through their extensive product list. The dataset used for analysis contains attributes such as product title, category, subcategory, brand, price, rating, and description. Notably, the creation of a "discount" feature enables a better understanding of consumer preferences.

To improve the recommendation systems, two methods are employed: content-based methods and collaborative filtering methods. Content-based methods leverage user likes and product details to provide tailored suggestions. Techniques like TfidfVectorizer and CountVectorizer are utilized to analyze product descriptions and categories. The cosine similarity measure is then applied to recommend items aligned with individual preferences. Collaborative filtering methods, on the other hand, consider the preferences of users with similar tastes. Patterns are identified based on what similar users have bought or liked, generating recommendations accordingly. Singular Value Decomposition (SVD) is employed as a collaborative filtering technique.

This report aims to shed light on the effectiveness and adaptability of these methods in improving the recommendation systems of Amazon and BigBasket. By understanding the intricacies of these systems, we can gain insights into how these e-commerce platforms optimize customer experiences and drive sales through personalized recommendations.

Methodologies

Exploratory Data Analysis (EDA) for BigBasket

In this section, we will analyze the data a bit to have a better view of it. For this purpose, we have used bar plot, box plot and violin plot.

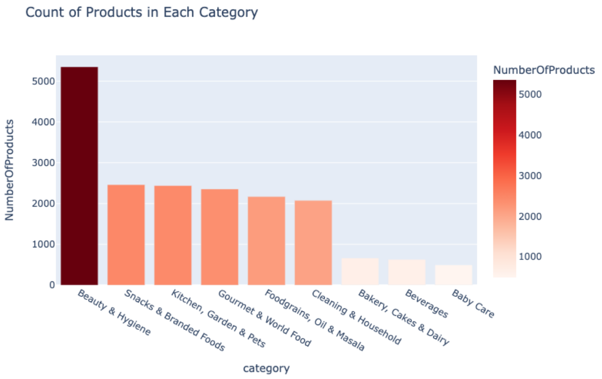


Figure 1: *Count of products in each category*

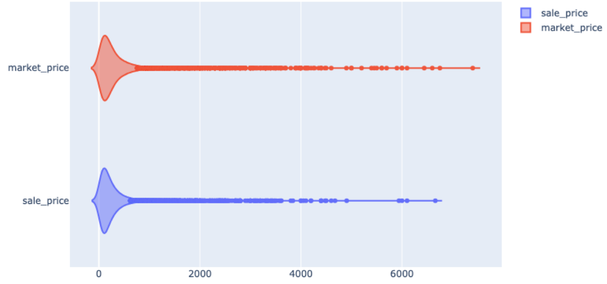


Figure 5: *Violinplot of sale_price and market_price*

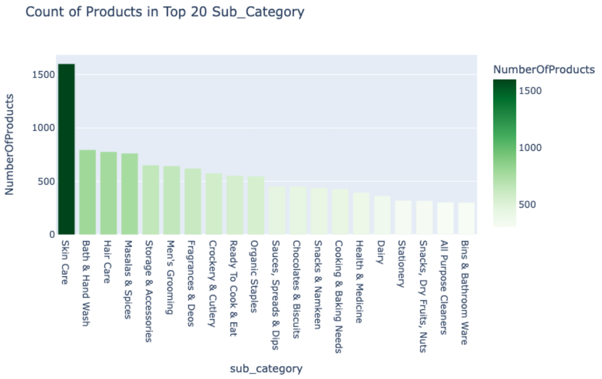


Figure 2: *Count of products in top 20 sub_category*

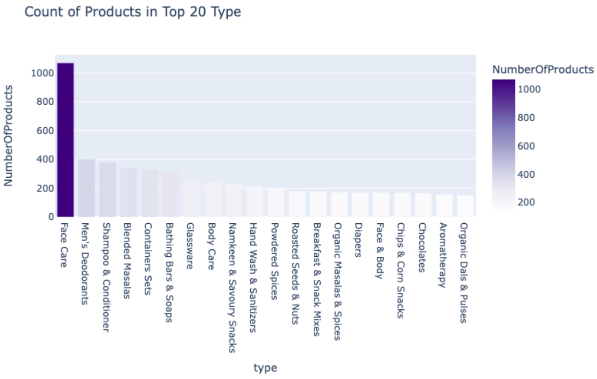


Figure 6: *Count of products in top 20 type*

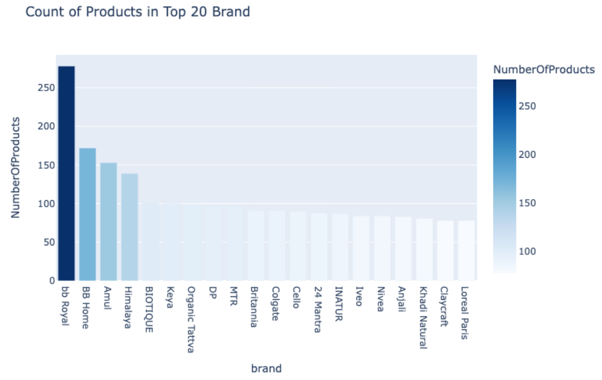


Figure 3: *Count of products in top 20 brand*

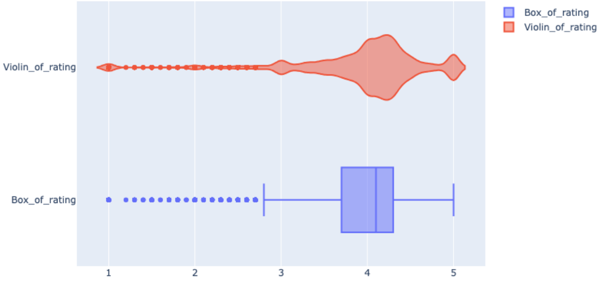


Figure 7: *Boxplot & Violinplot of ratings*

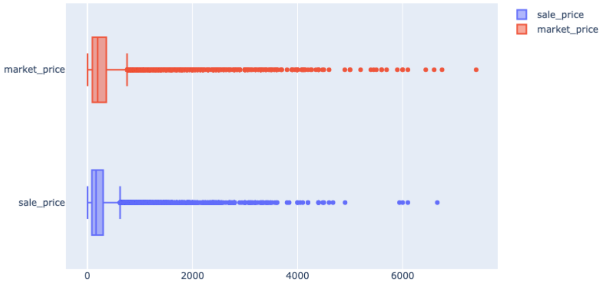


Figure 4: *Boxplot of sale_price and market_price*

Content-Based Method

These methods create personalized shopping guides on Amazon by analyzing user likes and product details to provide tailored suggestions. User and product profiles are formed, considering elements like descriptions and categories. Amazon utilizes this information to recommend items aligning with individual preferences. The discussion explores how content-based methods operate in Amazon's recommendation system and their effectiveness in handling the extensive product range.

TF-IDFVectorizer & CountVectorizer: TF-IDF Vectorizer and Count Vectorizer are both methods used in natural language processing to vectorize text. However, there is a fundamental difference between the two methods.

CountVectorizer simply counts the number of times a word appears in a document (using a bag-of-words approach), while TF-IDF Vectorizer takes into account not only how many times a word appears in a document but also how important that word is to the whole corpus.

This is done by penalizing words that often appear across all documents, reducing the count of these as these words are likely to be less important.

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

Figure 8: *Formula of TF-IDF*

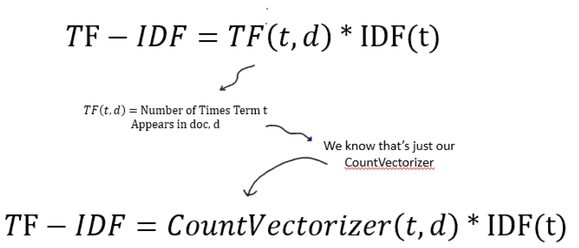
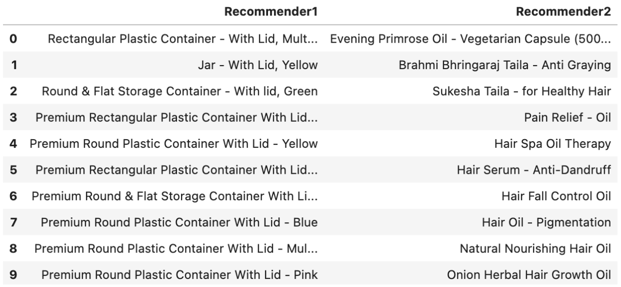

TF − IDF = TF(t, d) * IDF(t)
TF(t, d) = Number of Times Term t Appears in doc. d
We know that's just our CountVectorizer
TF − IDF = CountVectorizer(t, d) * IDF(t)

Figure 9: *Difference between TF-IDF and Count vectorizer*

Before doing anything, we have to say that first we have dropped all the data points with missing value and the data size has reached from 27201 to 18650. We have implemented the recommender system by using two TFIDF vectorizer and Countvectorizer methods. To implement the first method, we have combined several features. The features we have combined are "description", "category", "subcategory", "brand" and "type". Then we fit TFIDF on this composite feature, which finally gives us a matrix where each row represents a product, and this row is a vector of numbers obtained by TFIDFvectorizer. Finally, we obtained the cosine similarity between each of these products, which becomes an 18650 by 18650 matrix. Now, we have designed a function for the recommender that checks for each product and finds the first N product that has the most similarity with the original product (except the original product itself) according to the cosine similarity matrix and recommends them as suggestions.

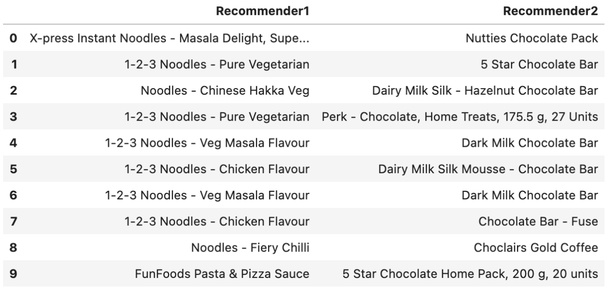
In order to implement the second method, we first listed the features of "category", "subcategory", and "type", because they were initially separated by commas or & as strings. Then we have combined the features of "category", "subcategory", "brand" and "type" and put them in one feature. Finally, we have used Count Vectorizer to convert this feature into a vector. Then, as before, we have calculated the cosine similarity matrix and implemented the recommender function in the same way as before.

We show the comparison of these two methods for two different examples.



	Recommender1	Recommender2
0	Rectangular Plastic Container - With Lid, Mult...	Evening Primrose Oil - Vegetarian Capsule (500...
1	Jar - With Lid, Yellow	Brahmi Bhringaraj Taila - Anti Graying
2	Round & Flat Storage Container - With lid, Green	Sukesha Taila - for Healthy Hair
3	Premium Rectangular Plastic Container With Lid...	Pain Relief - Oil
4	Premium Round Plastic Container With Lid - Yellow	Hair Spa Oil Therapy
5	Premium Rectangular Plastic Container With Lid...	Hair Serum - Anti-Dandruff
6	Premium Round & Flat Storage Container With Li...	Hair Fall Control Oil
7	Premium Round Plastic Container With Lid - Blue	Hair Oil - Pigmentation
8	Premium Round Plastic Container With Lid - Mul...	Natural Nourishing Hair Oil
9	Premium Round Plastic Container With Lid - Pink	Onion Herbal Hair Growth Oil

Figure 10: *Example: Garlic Oil - Vegetarian Capsule 500 mg*



	Recommender1	Recommender2
0	X-press Instant Noodles - Masala Delight, Supe...	Nutties Chocolate Pack
1	1-2-3 Noodles - Pure Vegetarian	5 Star Chocolate Bar
2	Noodles - Chinese Hakka Veg	Dairy Milk Silk - Hazelnut Chocolate Bar
3	1-2-3 Noodles - Pure Vegetarian	Perk - Chocolate, Home Treats, 175.5 g, 27 Units
4	1-2-3 Noodles - Veg Masala Flavour	Dark Milk Chocolate Bar
5	1-2-3 Noodles - Chicken Flavour	Dairy Milk Silk Mousse - Chocolate Bar
6	1-2-3 Noodles - Veg Masala Flavour	Dark Milk Chocolate Bar
7	1-2-3 Noodles - Chicken Flavour	Chocolate Bar - Fuse
8	Noodles - Fiery Chilli	Choclairs Gold Coffee
9	FunFoods Pasta & Pizza Sauce	5 Star Chocolate Home Pack, 200 g, 20 units

Figure 11: *Example: Cadbury Perk - Chocolate Bar*

Collaborative Filtering Method

Collaborative filtering acts as a team effort in Amazon's recommendations, considering the preferences of users with similar tastes. It examines what similar users have bought or liked, generating suggestions based on those patterns. The discussion explains how collaborative filtering works, its value in Amazon's recommendation system, and its adaptability to the dynamic online store with a diverse product range.

TruncatedSVD Dimensionality reduction using truncated SVD (aka LSA). This transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD). Contrary to PCA, this estimator does not center the data before computing the singular value decomposition. This means it can work with sparse matrices efficiently.

Due to the large amount of data, we have first separated a sample of 25,000 data so that we can implement the methods on it. Then we have to convert the main data matrix into a matrix whose rows represent users and whose columns represent products. We have used the truncated SVD method to decompose the output of the matrix obtained from the previous step and we have decomposed the matrix into 20 components. Then we have calculated the correlation matrix so that we can have the relationship between each of the products. Finally, we have randomly selected a product for testing and we have made a chance that the user has purchased this product. Then, according to the correlation matrix, we have displayed the products that we can recommend to the desired user according to the purchases of other users. An example is in the notebook.

Results

The main goal of this project was to be able to implement the simplest methods for the implementation of two methods, Collaborative Filtering(such as SVD) and Content Based(such as TFIDF_Vectorizer & Count_Vectorizer). These methods may give good or bad results according to the volume of the data and the cleanliness of the data. According to the observations in the content-based section, we saw that the Recommender of the second system has achieved much better results and can still be improved.