# Recommendation System

Reza Mosavi 400222100

4 januari 2024

## 1 Abstract

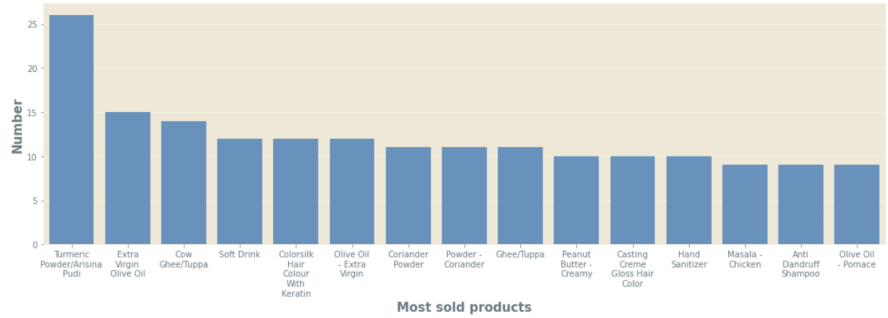
This data science exercise focuses on building a recommender system for beauty products using a dataset of over 2 million customer reviews from Amazon. The dataset includes essential information such as unique customer IDs, product codes, ratings, and timestamps. Inspired by Amazon's reliance on recommendation engines, the goal is to create an effective system that suggests beauty products based on customer preferences and historical data. The dataset spans from May 1996 to July 2014, offering a rich source for exploring trends in customer behavior. The exercise challenges participants to transform raw data, handle sparse matrices, and implement algorithms to capture user-product interactions. The ultimate question is whether a robust recommendation engine can be developed from this relatively minimal dataset. Participants are encouraged to share their findings, code implementations, and any additional insights or queries related to Amazon.

## 2 Introduction

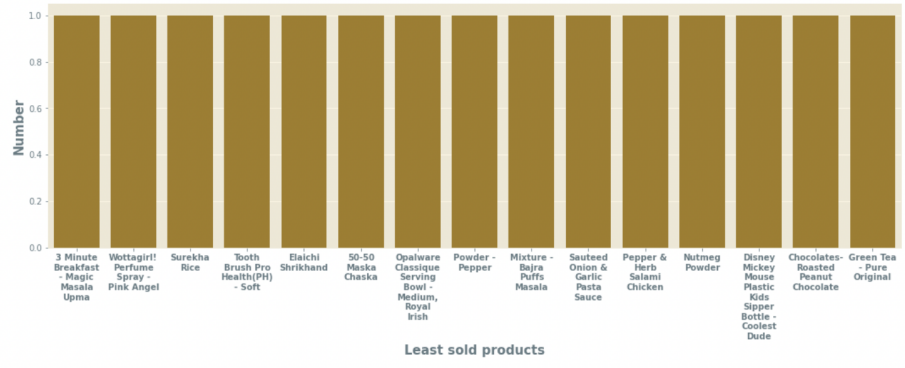
This data science exercise revolves around the construction of a recommender system, mirroring the strategies implemented by Amazon in its vast online marketplace. Two key methodologies are explored: Content-Based Methods and Collaborative Filtering Methods.

**Content-Based Methods** involve the analysis of user likes and product details to create personalized shopping guides. Amazon forms user and product profiles by considering elements like descriptions and categories. The discussion investigates how these methods function within Amazon's recommendation system and their effectiveness in managing the extensive range of products.

**Collaborative Filtering** Methods operate as a collaborative effort, considering the preferences of users with similar tastes. By examining patterns in similar users' behavior, it generates suggestions aligned with collective preferences. The discussion delves into the mechanics of collaborative filtering, emphasizing its value in Amazon's recommendation system and its adaptability to the dynamic online store.




Figur 1: Top sold products




Figur 2: least sold products

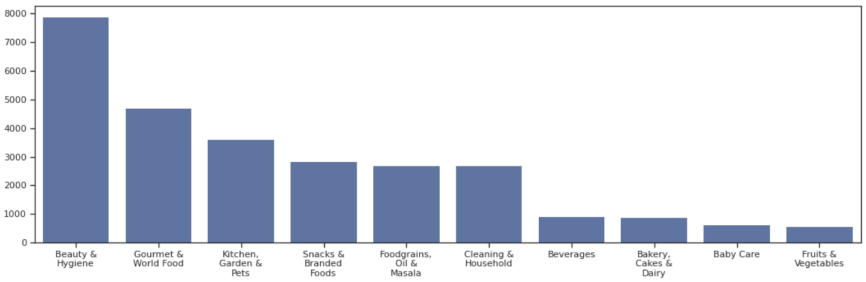
# 3 Data Analysis

The dataset under consideration pertains to BigBasket, the leading online grocery supermarket in India, and provides valuable insights into the realm of e-commerce. Launched around 2011, BigBasket has maintained its market dominance through continuous expansion and a strategic shift to online purchasing. This summary outlines the key features present in the dataset, shedding light on the nature and scope of the digital marketplace.

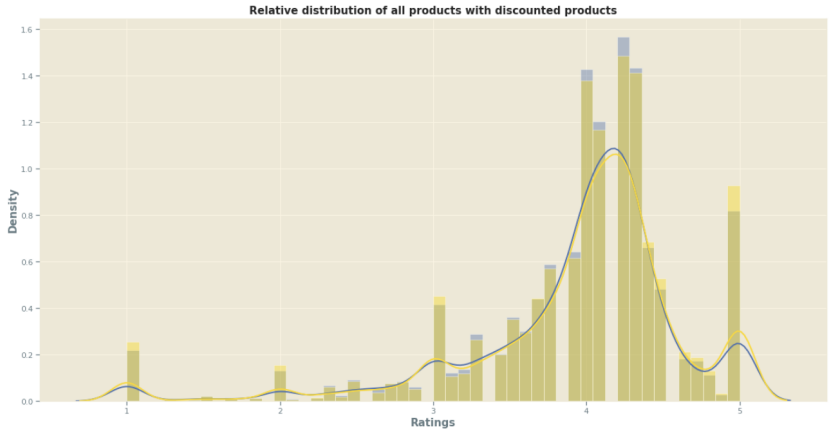
### 3.0.1 Top and least sold products

In this section, we review the best-selling and low-selling products:

It is quite clear and obvious, except for the first case, which has a large difference in the number of measurements, the rest of the cases are almost in the same range.




Figur 3: Top and least sold Categories




Figur 4: Discount effect

### 3.0.2 Top and least sold Categories

In this section, we review the Top and least sold Categories:

### 3.0.3 Discount effect

In this section, we examine the effect of discounts on sales:

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 discount is offered or not.

# 4 Methodology

In this section, we will examine the methods mentioned in the introduction section and examine each method on the related data.

## 4.1 Collaborative Filtering

In the context of recommendation systems, model-based collaborative filtering is a technique employed to provide personalized suggestions to users based on their purchase history and the similarity of their preferences to those of other users. This method focuses on identifying patterns in user preferences from multiple sources of user data, allowing for the prediction of products that a particular user might be interested in.

### 4.1.1 Utility Matrix :

- The first step involves creating a utility matrix. This matrix represents all possible user-item preferences (ratings) and is constructed from the purchase history and user reviews. In this matrix, each row corresponds to a user, each column corresponds to a product, and the entries represent user ratings. Since users typically do not rate all items, the utility matrix is sparse, with many unknown values.

### 4.1.2 Matrix Decomposition :

- The utility matrix is then decomposed using a technique called Truncated Singular Value Decomposition (SVD). This process reduces the dimensionality of the original matrix while retaining the essential patterns in the data. The result is a decomposed matrix that captures latent features and relationships between users and items.

### 4.1.3 Correlation Matrix :

- The next step involves calculating the correlation matrix from the decomposed matrix. This matrix measures the similarity between the preferences of different items based on user interactions. It provides a basis for identifying items that are likely to be preferred by a user given their history.

### 4.1.4 Recommendation :

- To make recommendations for a specific user, the system selects a target user (in this case, identified by the UserId "6117043058") and retrieves the row corresponding to that user in the correlation matrix. - Items with a correlation coefficient above a certain threshold (in this case, 0.90) are considered similar to the items the user has interacted with positively. - The system then compiles a list of recommended items, excluding those that the user has already purchased.

#### 4.1.5 Purpose :

Model-based collaborative filtering is chosen for its ability to predict products for a specific user by uncovering hidden patterns in the preferences of multiple users. By utilizing a utility matrix and matrix decomposition, the method efficiently captures the underlying structure of user-item interactions. This approach is valuable in scenarios where explicit user-item ratings are not available for all products, enabling the system to make accurate predictions even with sparse data. The ultimate goal is to enhance user experience by offering personalized recommendations based on patterns derived from the broader user community.

### 4.2 Content-Based Methods

#### 4.2.1

Introduction to Recommender Systems:

Recommender systems, a type of information filtering system, aim to enhance search results by providing more relevant items based on user preferences and search history. These systems analyze user interactions to find patterns and recommend items with sustainable similarities.

#### 4.2.2 Sort Recommendor Function:

The notebook presents a function named sort recommendor, which recommends products based on sorting them by a specified feature (e.g., rating, sale price). The function allows ascending or descending ordering of the products.

#### 4.2.3 TF-IDF (Term Frequency-Inverse Document Frequency):

TF-IDF is introduced as a technique to extract useful information from features, especially descriptions. It stands for term frequency-inverse document frequency and is used to measure the importance of terms in a document relative to a corpus.

#### 4.2.4 TF-IDF Workflow:

The TF-IDF process involves transforming the item descriptions into a matrix representation, calculating cosine similarity, and creating a recommendation function (get recommendations1) that suggests items based on the similarity of their descriptions.

#### 4.2.5 Count Vectorization and Cosine Similarity:

An alternative approach using count vectorization and cosine similarity is presented (get recommendations2). This method processes item descriptions, calculates cosine similarity, and recommends items based on the similarity scores.

#### 4.2.6 Comparison of Recommendation Results:

The notebook concludes with a comparison of recommendations generated by the original recommender (get recommendations1) and the new one (get recommendations2), highlighting the differences in their outputs.

#### 4.2.7 Purpose:

Content-Based Methods are chosen to enhance the recommendation system by focusing on the inherent features of items. This approach is particularly valuable when user-item interaction data is sparse or when there is a need to provide recommendations for new items without extensive user history. By analyzing item features and descriptions, content-based methods contribute to delivering personalized and relevant recommendations, thereby improving the overall user experience.