Data Science 5th Assignment

Dataset: Amazon - Ratings (Beauty Products)

Mahan Madani – 99222092

1. Overview

This dataset is comprised of over **2 million records** of beauty product sales from Amazon, containing information in 4 columns.

The Goal of this project is to design and implement a recommender system based on the given dataset. I used two different methods to build the recommender system using **Collaborative Filtering**: Matrix Factorization and DBSCAN Clustering.

2. Dataset Exploration & Analysis

The dataset stores 4 different attributes. Each record in the dataset belongs to one specific purchase of an item.

Below you can find a list of all the attributes:

- **UserId:** A unique value used for customer identification.
- **ProductID:** The product ASIN (Amazon's unique product identification code for each product.
- **Rating:** Ranging from 1-5 based on customer satisfaction (higher is better).
- **Timestamp:** the timestamp of the rating in UNIX time.

3. Data Preprocessing

3.1 Check for Duplicate Records:

To ensure the dataset contains no duplicate records, a combination of the 'ProductID', 'UserID', and 'Timestamp' features can be used. If two or more records share the same value for these columns, that means they are duplicates. **After testing the dataset, it seems that there are no duplicate records.**

3.2 Handle Null Values:

As seen in this table, out of the 4 features in the dataset none contain null values. No further action is required here.

	Null Count
UserID	0
ProductID	0
Rating	0
Timestamp	0

3.3 Feature Generation:

Based on the available data, the following features can be added to further improve the dataset:

- **Datetime:** The datetime value of the purchase. Extracted from 'Timestamp'
- **Year:** The year of the purchase. Extracted from 'Datetime'
- **Month:** The month of the purchase. Extracted from 'Datetime'

These attributes can be used to create a more advanced recommendation system. For example, users may be more likely to purchase a group of items in a specific month, like Christmas decorations. However, for a simple collaborative filtering system, the 'Rating' column should be the focus.

3.4 Additional Notes:

The dataset contains more than 2 million records. About 1.2 million users and 250k unique products. Clustering all of the data at once presents a computational challenge, while also requiring large amounts of memory.

	UserID	ProductID
count	2023070	2023070
unique	1210271	249274

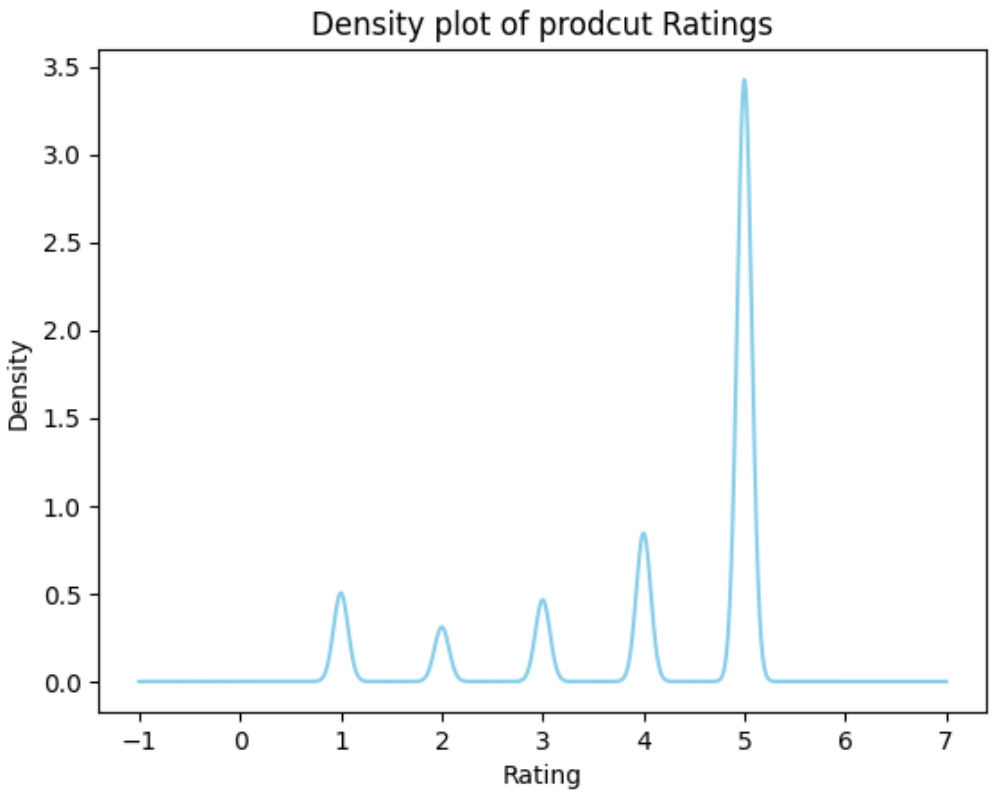
One approach could be to only cluster part of the data as a proof that the model can successfully find the underlying patterns between the different records.

A more advanced (but time-consuming) approach would be to process the data in chunks and then concatenate it to build the recommender system.

4. Visualization

Figure 4.1: Density plot of product Ratings

This plot displays the density of the various ratings given by various customers.



It is evident that the majority of the ratings given were positive, with most users giving out either a 5 or 4 star rating. This can be helpful for recommending similar products to customers as it is more likely to find two users who share interests.

Figure 4.2: Bar plot of product sales by month

This plot demonstrates the distribution of product sales by month. It appears that the period of time from August to November has the lowest number of sales.

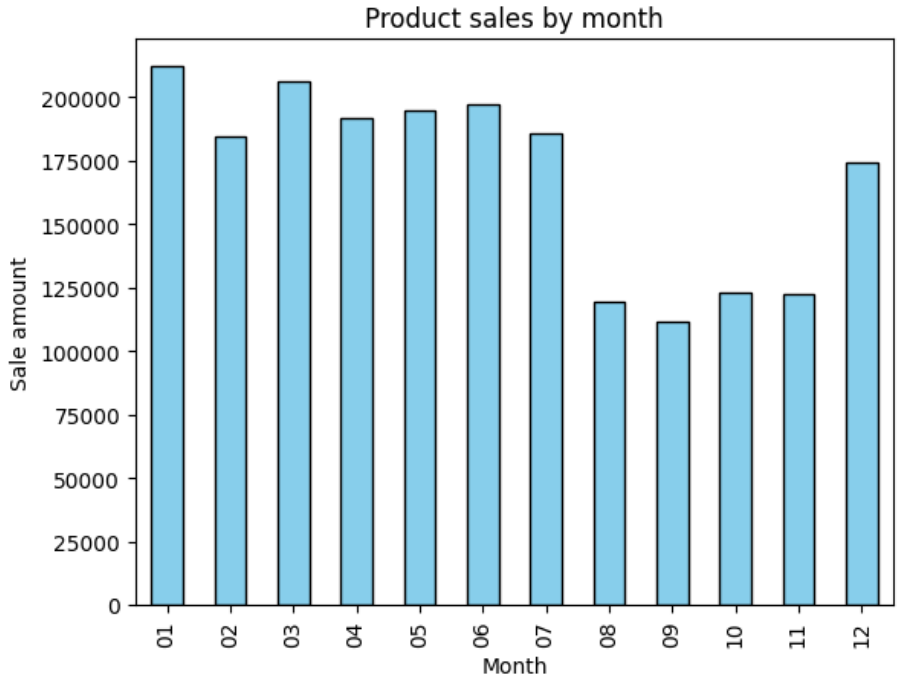
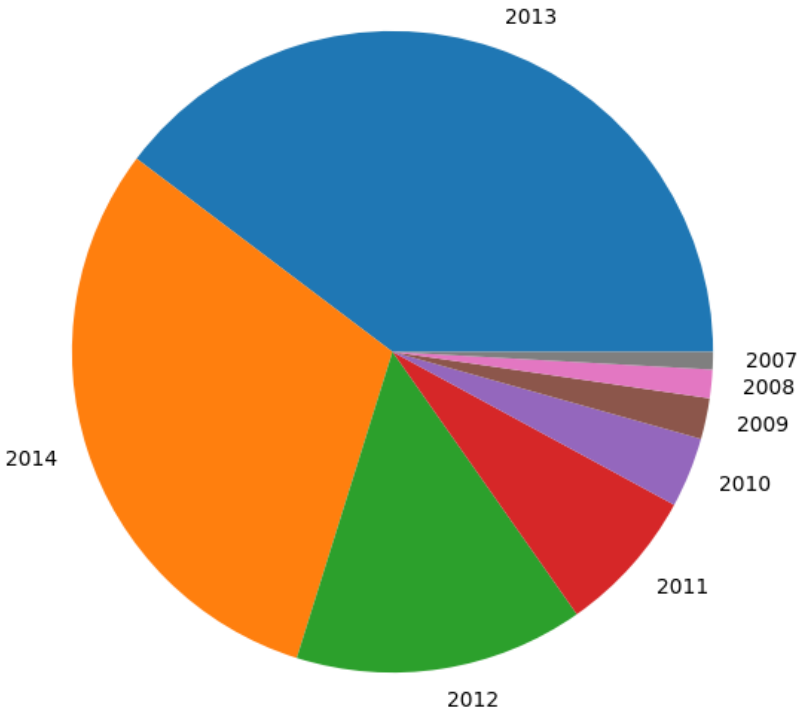


Figure 4.3: Pie chart of product sales by year

This pie chart displays the relative amount of product sales based on year. We can see the majority of the records in this dataset belong to 2013 and 2014.

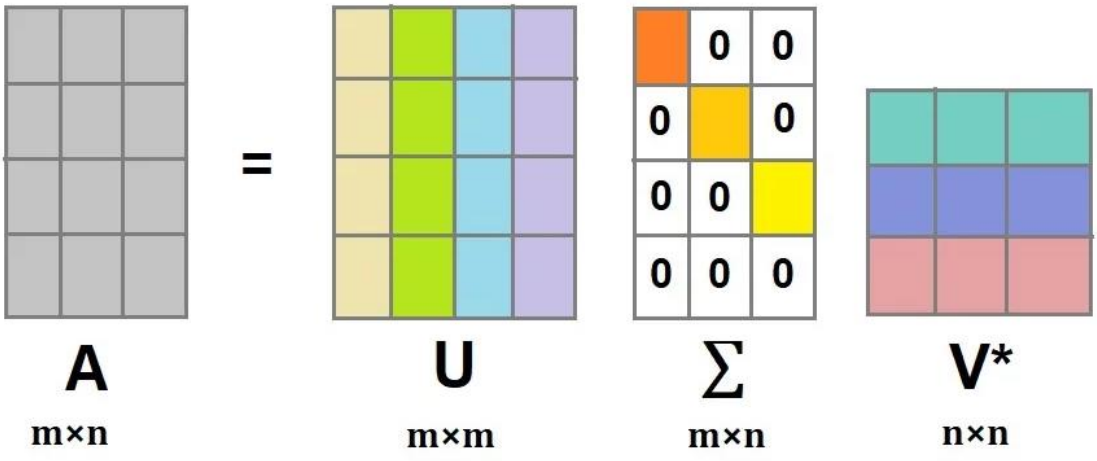
Each year has had more sales compared to the previous one, displaying the growth of the business.



5. Matrix Factorization

One method of creating a recommendation system is using matrix factorization to predict what rating each user would give to the available items. First, a sparse matrix (the user-item interaction matrix) is created where each row represents a user, and each column represents a product. Each element in the matrix should be the rating the user has given to a product, and if they haven't purchased the product, the value should be 0.

This matrix can take up a large amount of memory, so using SVD (Singular Value Decomposition) we can factorize it to become the product of several smaller matrices. In doing so, a number of features will be naturally discovered, which can then help predict a user's rating to a product they haven't purchased yet.



The two resulting matrices (the "Users" matrix and the "Products" matrix) can be used to recommend products. First a row must be selected from the "Users" matrix, and since each row corresponds to one of the users, we're basically choosing a user. Then the product of this row and the "Products" matrix is calculated, which results in a vector with n elements where each element corresponds to the predicted rating of a product.

We can sort this vector from high to low and start recommending the top results.

6. DBSCAN Clustering

Another method for creating the recommendation system is to use a clustering method to create clusters of users (or products) and start recommending similar products to users in the same cluster.

I chose the DBSCAN clustering method (Density-Based Spatial Clustering of Applications with Noise) for this project due to the fact that the number of clusters is not pre-determined.

As an example, with the hyperparameters that I adjusted, my model generated 47 different user clusters. We can tag different users in the original dataset with their cluster ID and then create a list of recommended products for each cluster.