Content Based Recommendation Project

1. ***Data Analysis (Pre-processing/Insights)***
   a. We plotted a bar plot to see the distribution of frequency of ratings in the dataset. We concluded that most of the users are rating the products very highly with a rate of 4 or 5. The distribution is right skewed ,indicating a positive bias towards higher ratings
   b. Positive Bias Towards Higher Ratings: The right-skewed distribution of ratings, with the majority of ratings concentrated at the highest levels (4 or 5), indicates a positive bias towards higher ratings. This suggests that users tend to rate products more favorably overall.
   c. Impact on Product Perception: The skewness towards higher ratings may have implications for how products are perceived by consumers. Products with consistently high ratings may be perceived as more favorable and attract more attention from potential buyers, while those with lower ratings may be viewed less favorably and experience decreased sales or popularity.

2. Feature extraction

1. **About the Data - After removing irrelevant columns from two data sets we got -**
   - **All_Beauty_Review data** -
     COLUMNS:
     ["reviewerID","reviewText",”reviewTime”,"summary","asin","price","description","title","brand"]
     SIZE OF THE DATA:
     282236 rows × 13 columns
   - **meta_All_Beauty_products data**
     COLUMNS:
     ['description','title','brand'])
     SIZE OF THE DATA:
     32892 rows × 19 columns
   - We have merged the two datasets on the basis of asin and this dataset has been taken for further processing.

2. **Dropping the products by classifying the products.**
   - Classification 1 - We have dropped the products that were rated negatively on the basis of negative and positive sentiments by analyzing the ratings < 3 as 0 (negative) and rating > 3(positive) and dropping the products which were rated negatively.
   - Classification 2 - we have further tried to reduce our products on the basis of reviewTime. All products that had been reviewed after 2016 were dropped.

3. **Reduction of irrelevant features** - Data has been processed through Lemmatization, RegEx and StopWords removal models. We used Scikit Learn library for the above three processes on the data to remove irrelevant features.
4. **Final Features** - We have finally taken two main datasets for feature extraction-

- **Data1** represents a **User profile** consists of **review+description+title+brand** as a single document each for each product.
- **Data2** represents a **Item profile** consisting of **description+title+brand** as a single document each for each product.

3. Representing reviews and mapping them with the users.

- Reviews data is represented as User Profile - Data-1
- All products data is represented as Item Profile - Data-2
- Data1 and Data2 are processed using the 2nd and 3rd step mentioned in feature extraction's points.
- We get two processed datasets - **all_beauty** and **all_beauty1** among which **similarity** will be calculated after representing the text matrix using **sklearn's tfidf vectorizer**.

4. Designing the content-based system

1. **SVD Decomposition -** we had taken into consideration **max 45000 features** as parameters while creating the **tfidf** matrix for both the datasets in the **tdfif.vectorization**. Which was reduced to less than 1500 by decomposing using the TruncatedSVD model of **sklearn.decomposition.**

   We also computed **explained_variance_ratio and cumulative_explained_variance** to understand the percentage of variance in the data explained by the reduced no of features. We observed an **important intricacy** with respect to this which we have mentioned in **Novelty**.
2. **Cosine Similarity -** The similarity matrix is computed between the reduced features obtained from SVD decomposition. The cosaine_matrix_1 is calculated between user profile and user profile and cosaine_matrix_2 is calculated between user profile and all products profile and cosaine_matrix_3 is calculated between all products and all products dataset which were reduced using SVD.
3. **Top_K -** top K similar products to a selected item are recommended. The recommendation outputs a dataframe with top k product titles, ratings corresponding to the products and their index in the user profile.
4. **Evaluation -** a very naive approach of evaluation is taken into consideration. Where we take -

   number of successful recommendations = no of the recommended product's ratings that are greater than 3 (either 4 or 5) i.e positive product recommendations.

   Total no of recommendations = top k products recommended

$$Accuracy = \frac{Number of successful Recommendations}{Total Number of Recommendations}$$

5. Novelty and Insights

1. **N_gram and its effect on explained variance ratio -** n-gram is one of the parameters of TfidfVectorizer**,** while working with the data we observed that when we took n-gram=(2,2) that is bigram only 85% and 40% of the user data and all product data was explained by reduced features. But when we took a combination of uni-bi-tri gram by taking n-gram=(1,3), 90% and 50% of our user data and all products data were explained by our reduced features.

2. **User - Product similarity VS Product - Product similarity -** In the end we get recommendations in form of two DataFrames, **User_Product_Reccomended_items** and **Product_product_Reccomended_items** with **[title , rating , index]** columns.

- **User_Product_Reccomended_items** -
  We observed from the **title of** products recommended for **User_Product_Reccomended_items** were more diverse in terms of brand and product nature, only a few products from the same brand were recommended. But it was evident from similarity score that similarity scores were less than 0.45. We also observed from the **ratings** that a lot of products ratings less than 3 which are considered a negative sentiment for our data were also recommended.

- **Product_product_Reccomended_items**
  We observed from the title of products recommended for **Product_product_Reccomended_items** that items were less diverse in terms of brand and nature. Most products from the same brand were recommended. And also it could be observed from **ratings** that all except 1 or 2 were rated 4 or 5 which is a positive sentiment for our data.

  Hence it could be concluded that **Product_product_Reccomended_items** gives **best recommendations for our Amazon product recommendation system and should be preferred over User_Product_Reccomended_items.**

3. **Evaluation -** We have created our own evaluation method for evaluating or comparing the two recommendation systems which are **Product_product_Reccomended_items** and **User_Product_Reccomended_items.**
   Number of successful recommendations = no of the recommended product's ratings that are greater than 3 (either 4 or 5) i.e positive product recommendations.
   Total no of recommendations = top k products recommended
   Formula:

$$Accuracy = \frac{Numberof successful Recommendations}{Total Numberof Recommendations}$$

   Observation:
   We observed that -
   Success rate of recommendations for **Product_product_Reccomended_items** comparison: **0.92**
   Success rate of recommendations for **all_beauty_product and all_beauty_product** comparison: **0.8**
   Hence we were empirically able to prove our claim that **Product_product recommendations provide better results than User_Product recommendations.**