IR ASSIGNMENT 3

Samanyu Kamra 2021487

Product Recommendation System based on Amazon Review:

Category: Headphones

Creating Final Dataframe:

1. Removed duplicates from the meta dataframe.
2. Merged the review dataset and meta dataset on the basis of asin.
3. Removed duplicate rows from merged dataframe.

Descriptive Statistics Results:

```
Descriptive Statistics for the Dataset:

a. Number of Reviews: 193841
b. Average Rating Score: 4.06
c. Number of Unique Products: 3887
d. Number of Good Ratings (≥ 3): 164542
e. Number of Bad Ratings (< 3): 29299
f. Number of Reviews corresponding to each Rating:
   - 1 star: 15311
   - 2 star: 13988
   - 3 star: 20516
   - 4 star: 38892
   - 5 star: 105134
```

Preprocess the Review Text:

1. Removing the HTML Tags.
2. Removing accented characters.
3. Expanding Acronyms.
4. Removing Special Characters
5. Lemmatization

EDA:

All outputs and explanations are done in the ipynb file.

Features Extracted and Model Training:

1. Calculated TF-IDF scores of Review Text for every row.
2. Ratings were divided into 3 classes:
    a. Good (Ratings more than 3)
    b. Average (Ratings equal to 3)
    c. Bad (Ratings less than 3)
3. Dataset was divided into training and validation sets as 75:25.
4. Machine Learning Models used for Classification:
    a. Logistic Regression
    b. Naive Bayes
    c. Random Forest
    d. SVM
    e. Decision Tree

```
Model: Logistic Regression
              precision    recall  f1-score   support

     Average      0.467     0.146     0.222      5163
         Bad      0.701     0.634     0.666      7227
        Good      0.863     0.964     0.911     36071

    accuracy                          0.828     48461
   macro avg      0.677     0.581     0.600     48461
weighted avg      0.797     0.828     0.801     48461

------------------------------------------------------------
Model: Naive Bayes
              precision    recall  f1-score   support

     Average      1.000     0.001     0.002      5163
         Bad      0.948     0.055     0.105      7227
        Good      0.751     1.000     0.858     36071

    accuracy                          0.753     48461
   macro avg      0.900     0.352     0.321     48461
weighted avg      0.807     0.753     0.654     48461

------------------------------------------------------------
Model: Random Forest
              precision    recall  f1-score   support

     Average      0.769     0.073     0.133      5163
         Bad      0.809     0.386     0.523      7227
        Good      0.803     0.991     0.887     36071

    accuracy                          0.803     48461
   macro avg      0.794     0.483     0.514     48461
weighted avg      0.800     0.803     0.753     48461

------------------------------------------------------------
```

```
------------------------------------------------------------
Model: Support Vector Machine
              precision    recall  f1-score   support

     Average      0.455     0.119     0.188      5163
         Bad      0.680     0.640     0.659      7227
        Good      0.861     0.962     0.908     36071

    accuracy                          0.824     48461
   macro avg      0.665     0.573     0.585     48461
weighted avg      0.790     0.824     0.795     48461

------------------------------------------------------------
Model: Decision Tree
              precision    recall  f1-score   support

     Average      0.260     0.222     0.240      5163
         Bad      0.509     0.502     0.505      7227
        Good      0.854     0.874     0.864     36071

    accuracy                          0.749     48461
   macro avg      0.541     0.533     0.536     48461
weighted avg      0.739     0.749     0.744     48461

------------------------------------------------------------
```

Collaborative Filtering:

Create Item-User Rating Matrix:

Pivot the DataFrame to create an item-user rating matrix (item_user_matrix) instead of a user-item rating matrix. This matrix represents ratings given by users to different items.

Calculate Item Similarity Matrix:

Use the cosine similarity between items to calculate the item-item similarity matrix (item_cosine_sim_df). This matrix represents the similarity between different items based on the ratings given by users.

Define Functions for Item-Item Recommendation:

Create functions to find the top similar items (find_top_n_similar_items) and predict ratings based on item similarity (predict_rating).

Revised MAE Calculation:

Modify the calculate_mae function to iterate over items in the test set, rather than users. This ensures that predictions are made for each item in the test set.

Compute MAE for Different Numbers of Nearest Neighbors:

Calculate the mean absolute error (MAE) for different numbers of nearest neighbors in the item-item recommender system.
Plot MAE vs. Number of Nearest Neighbors:

Plot the MAE values against the number of nearest neighbors to visualize how the MAE varies with the number of neighbors in the item-item recommendation model.