CSE508 Information Retrieval
Winter 2024
Assignment - 3

Siddharth
2021424

This project aims to develop a predictive model based on Amazon review data to recommend relevant items to users. The system aims to enhance user engagement by leveraging collaborative filtering techniques by providing personalized recommendations tailored to individual preferences and interests.

Approach:

I have taken the Electronics product “Headphone” from the metadata and extracted the reviews and other details from the “Electronics_5.json”

Methodologies:

Data Collection: The project begins with the collection of Amazon review data, specifically focusing on headphone products. The dataset includes details such as product ID (ASIN), review text, overall rating, and review metadata.

Data Preprocessing: Raw review data undergoes preprocessing to clean and prepare it for analysis. This includes steps such as removing HTML tags, handling accented characters, expanding acronyms, removing special characters, lemmatization, and text normalization.

Descriptive Statistics: Descriptive statistics are calculated to gain insights into the dataset. This includes calculating the total number of reviews, average rating score, number of unique products, and categorizing reviews into Good and Bad based on a defined threshold.

Feature Engineering: Feature engineering techniques are applied to extract meaningful features from the review text. TF-IDF (Term Frequency-Inverse Document Frequency) is utilized to represent the importance of words in reviews.

Model Training: Machine learning models are trained using the TF-IDF features to classify reviews into Good, Average, and Bad categories. Models include Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN).

Model Evaluation: The performance of each trained model is evaluated using metrics such as precision, recall, F1-score, and support for each target class. This allows for the comparison of model performance and the selection of the best-performing model.

Product Recommendation: Once the model is trained and evaluated, it can be deployed to provide personalized product recommendations to users based on their preferences and past interactions with the platform.

Assumptions:

Before preprocessing the dataset of headphones' reviews for NA values and duplicates, the total no. of entries were 4,11,201

But after preprocessing, the total no. of entries were 4,01,595

Rating Class Definition: Ratings above a certain threshold are considered Good, while those below are considered Bad. The threshold for categorization may vary based on specific business requirements.

Feature Representation: TF-IDF features are assumed to adequately capture the significance of words in reviews for classification purposes.

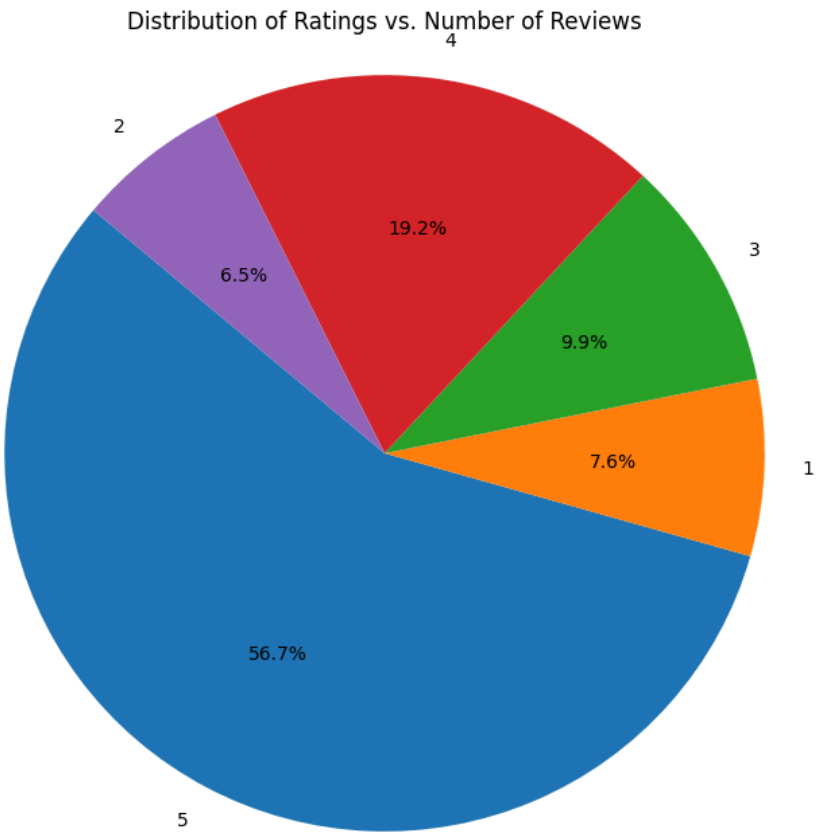
Results:

WordClouds:

Percentage of words in Good Reviews Word Cloud: 86.21%

Percentage of words in Bad Reviews Word Cloud: 13.79%

Piechart:



Q10) I have run the model for 1lakh entries.

```
Evaluating Logistic Regression...
c:\Users\DELL\AppData\Local\Programs\Python\Python39\lib\site-
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver
    https://scikit-learn.org/stable/modules/linear_model.html#
  n_iter_i = _check_optimize_result(
              precision    recall  f1-score   support

         Bad       0.72      0.64      0.68      3433
     Average       0.47      0.17      0.25      2499
        Good       0.88      0.97      0.92     19068

    accuracy                           0.84     25000
   macro avg       0.69      0.59      0.62     25000
weighted avg       0.81      0.84      0.82     25000
```

```
============================================================
Evaluating Decision Tree...
              precision    recall  f1-score   support

         Bad       0.45      0.45      0.45      3433
     Average       0.24      0.22      0.23      2499
        Good       0.86      0.86      0.86     19068

    accuracy                           0.74     25000
   macro avg       0.51      0.51      0.51     25000
weighted avg       0.74      0.74      0.74     25000

============================================================
Evaluating Random Forest...
              precision    recall  f1-score   support

         Bad       0.84      0.22      0.35      3433
     Average       0.75      0.03      0.05      2499
        Good       0.79      1.00      0.88     19068

    accuracy                           0.79     25000
   macro avg       0.79      0.42      0.43     25000
weighted avg       0.80      0.79      0.73     25000

============================================================
```

```
============================================================
Evaluating Support Vector Machine...
              precision    recall  f1-score   support

         Bad       0.74      0.65      0.69      3433
     Average       0.56      0.12      0.19      2499
        Good       0.87      0.98      0.92     19068

    accuracy                           0.85     25000
   macro avg       0.73      0.58      0.60     25000
weighted avg       0.82      0.85      0.82     25000

============================================================
Evaluating K-Nearest Neighbors...
              precision    recall  f1-score   support

         Bad       0.69      0.08      0.14      3433
     Average       0.24      0.04      0.07      2499
        Good       0.78      0.98      0.87     19068

    accuracy                           0.77     25000
   macro avg       0.57      0.37      0.36     25000
weighted avg       0.71      0.77      0.69     25000

============================================================
```

```
  n_iter_i = _check_optimize_result(
              precision    recall  f1-score   support

         Bad       0.73      0.67      0.70     14299
     Average       0.45      0.18      0.26      9919
        Good       0.88      0.97      0.92     76102

    accuracy                           0.85    100320
   macro avg       0.69      0.60      0.63    100320
weighted avg       0.82      0.85      0.82    100320

============================================================
Evaluating Decision Tree...
              precision    recall  f1-score   support

         Bad       0.50      0.48      0.49     14299
     Average       0.26      0.23      0.24      9919
        Good       0.86      0.88      0.87     76102

    accuracy                           0.76    100320
   macro avg       0.54      0.53      0.53    100320
weighted avg       0.75      0.76      0.75    100320
```

Interpretation: Logistic Regression Model

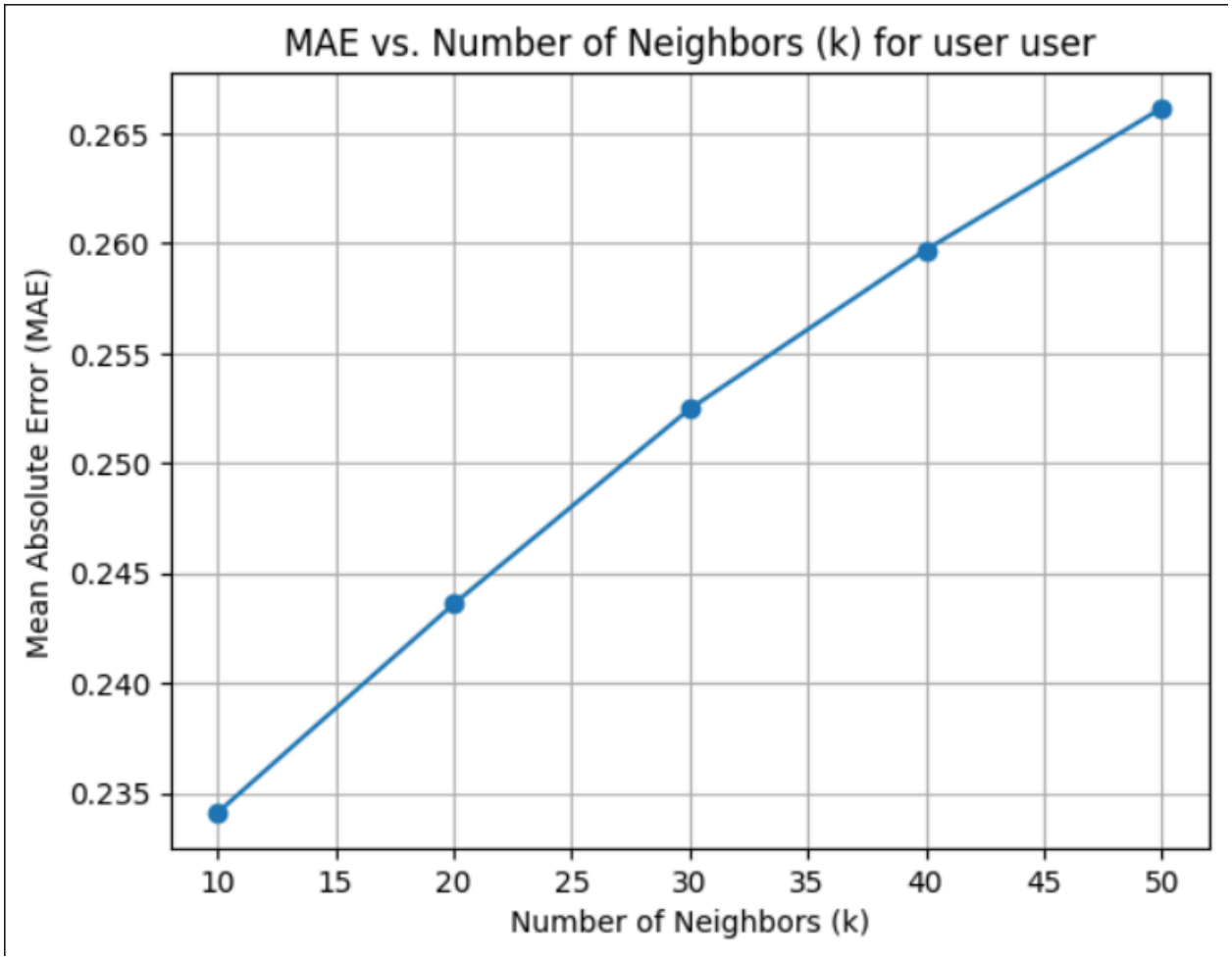
- The overall accuracy of the model is 85%, indicating that it correctly predicts the rating class for 85% of the instances.
- The precision for each class indicates the percentage of correct predictions among the instances predicted as that class. For example, among instances predicted as "Good," 88% were actually "Good" ratings.
- Recall represents the percentage of correctly predicted instances of each class out of all instances of that class. For example, among all actual "Good" ratings, 97% were correctly predicted as "Good."
- The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics.
- The support indicates the number of instances for each class.

Interpretation: Decision Tree Model

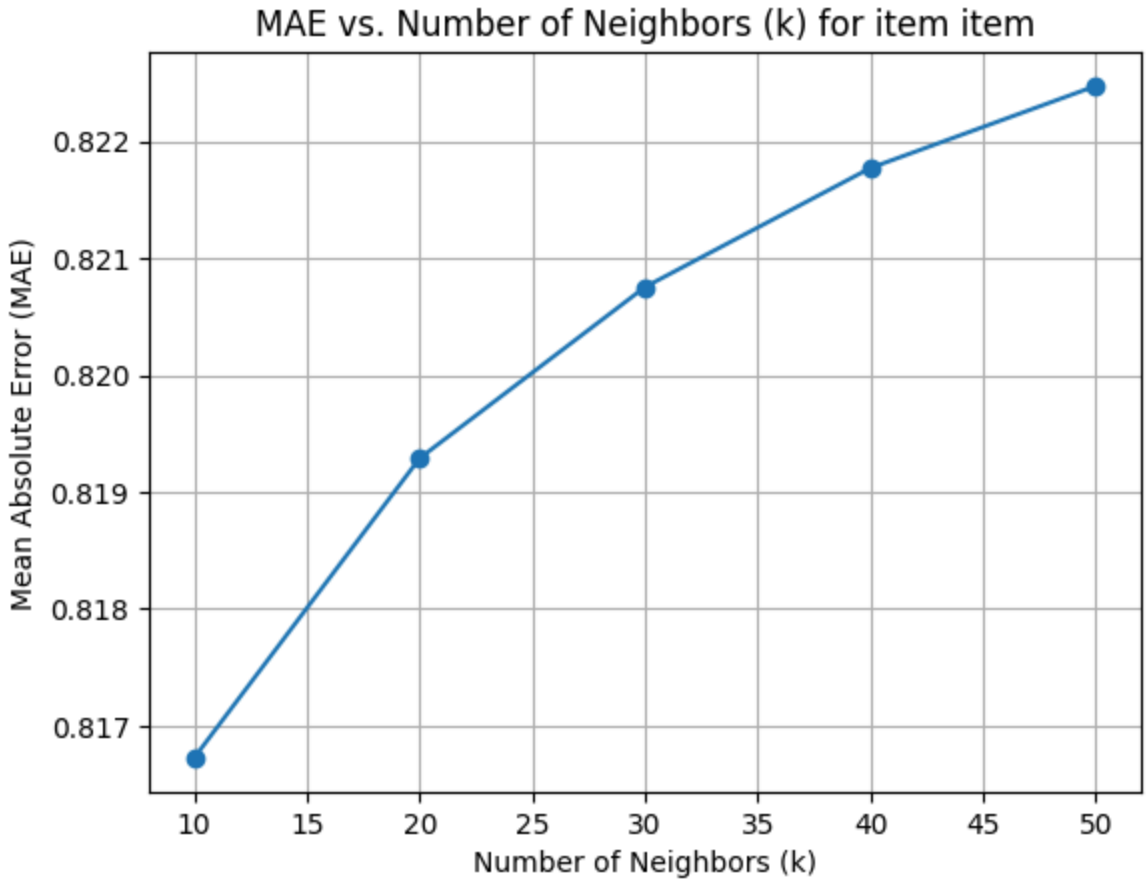
- The Decision Tree model has an overall accuracy of 76%, which is lower than the first model.
- Precision, recall, and F1-score for each class indicate the performance of the model in predicting that class. Compared to the first model, the Decision Tree model generally performs worse, especially in terms of precision and recall for the "Bad" and "Average" classes.
- Support values remain the same, indicating the number of instances for each class.

Overall, while both models have their strengths and weaknesses, the first model outperforms the Decision Tree model in terms of overall accuracy and class-specific metrics.

Q11)
The output is:



N=10: MAE=0.2341
N=20: MAE=0.2436
N=30: MAE=0.2525
N=40: MAE=0.2597
N=50: MAE=0.2661



k=10: MAE=0.8167
k=20: MAE=0.8193
k=30: MAE=0.8208
k=40: MAE=0.8218
k=50: MAE=0.8225

Note: I have run the above part for 50,000 entries from the final_final_tfidf.json

Descriptive Statistics: Descriptive statistics provide insights into the dataset, such as the total number of reviews, average rating score, and distribution of reviews over time.

Model Comparison: The performance of different machine learning models is compared based on precision, recall, F1-score, and support for each target class. This helps identify the most suitable model for the recommendation system.

Q12)
Outputs:

reviewerID row_sum
asin
B000067RC4 8242.000000
B00004T8R2 7375.000000
B0002H02ZY 7050.000000
B000ULAP4U 6130.000000
B000WL6YY8 5715.000000
B00001P4ZH 5140.000000
B0007NWL70 5090.333333
B00001WRSJ 5089.000000
B0007XJSQC 4823.000000
B0001FTVEK 3745.000000

asin
4126895493 144.0
B000001OMI 146.0
B000001OMR 21.0
B00000DMA3 19.0
B00000J1EJ 140.0
...
B0015AE4C4 53.0
B0015AE4CE 8.0
B0015AFOL4 29.0
B0015AFONW 7.0
B0015AFWC0 74.0
Name: row_sum, Length: 651, dtype: float64

Conclusion:

The project successfully develops a product recommendation system based on Amazon review data. By leveraging machine learning techniques and collaborative filtering, the system offers personalized recommendations to users, thereby enhancing user engagement and satisfaction. The project highlights the importance of data preprocessing, feature engineering, and model evaluation in building effective recommendation systems. Further improvements and optimizations can be made based on feedback and user interactions to enhance the recommendation experience continuously.