CSE508 Information Retrieval
Winter 2024
Assignment - 3

Dhruv Sood
2021387

The assignment is started with downloading the electronic 5 dataset and product meta data.I used gzip and downloaded in 10,000 sized chunks.

I have saved the final data frames as pickles for df and df1.

Total number of rows on the basis of title are 27412 and we have around 547 duplicates making it 26865 rows. I later checked that around 54 review texts were NA therefore converted them to empty strings.

Now I took the inner product of headphones df and df on the basis of 'asin'.
Then I took the descriptive statistics of the data frame formed.
Number of Reviews: 411201
Average Rating Score: 4.112156828412382
Number of Unique Products: 8064
Number of Good Rating: 353401
Number of Bad Ratings: 57800
Number of Reviews corresponding to each Rating:
overall
1.0 31009
2.0 26791
3.0 40760
4.0 79153

Approach:

I have taken the Electronics product "Headphone" from the metadata and extracted the reviews and other details from the "Electronics_5"

Method

Data Preprocessing: Raw review Review Text which was an essential part of ML model and later parts of the assignment undergoes preprocessing to clean and prepare it for further model training. This includes steps such as removing HTML tags, handling accented characters, expanding contractions, expanding acronyms, removing special characters, removing stop words and doing lemmatization..

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 Classifier, Random Forest classifier, Support Vector Machine (svc) and Multinomial Naive Bayes.

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.Model is trained and evaluated on the first X rows of the data frame.

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.user item matrix is formed and also taken its transpose and shown top 50 products by user sum ratings.

Assumptions:

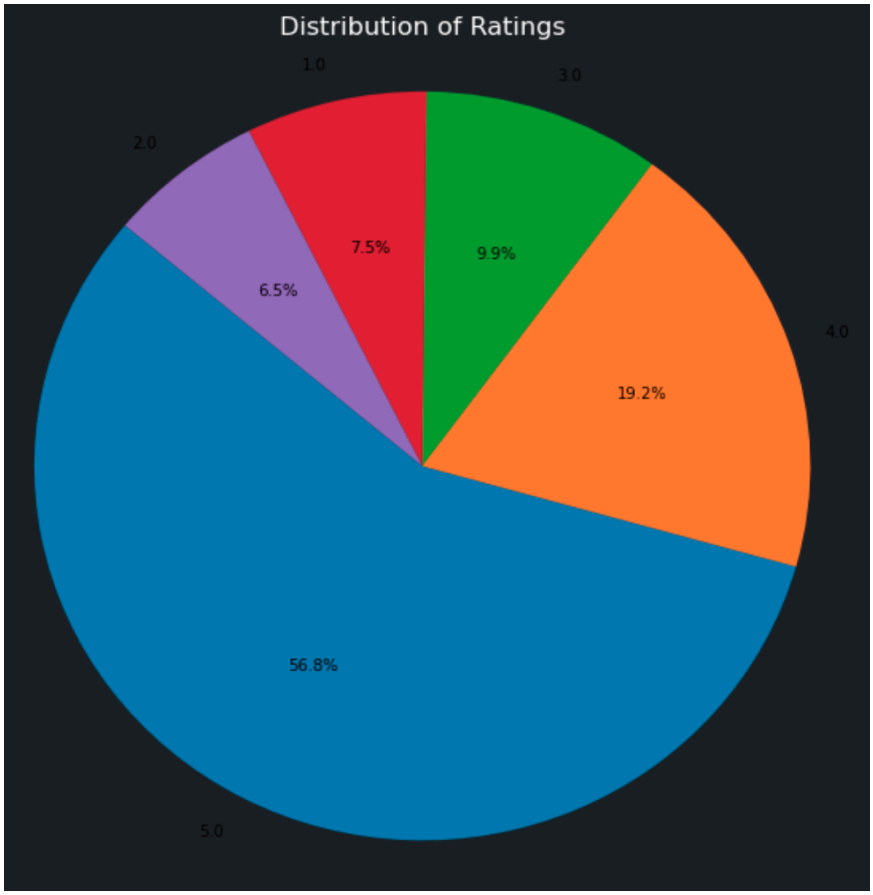
Feature Representation: TF-IDF features are assumed to adequately capture the significance of words in reviews for classification purposes.

The K Folds to find the MAE has been run on randomized data which is a subpart of it due to computational complexity.

Results:

Question6)

Question7)



Question10)

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

         Bad       0.44      0.11      0.18      1307
     Average       0.72      0.56      0.63      1592
        Good       0.86      0.97      0.91      9601

    accuracy                           0.83     12500
   macro avg       0.67      0.55      0.57     12500
weighted avg       0.80      0.83      0.80     12500

================================================================
Classifier: Decision Tree
              precision    recall  f1-score   support

         Bad       0.28      0.24      0.26      1307
     Average       0.46      0.48      0.47      1592
        Good       0.86      0.87      0.86      9601

    accuracy                           0.75     12500
   macro avg       0.53      0.53      0.53     12500
weighted avg       0.74      0.75      0.75     12500

================================================================
Classifier: Random Forest
...
   macro avg       0.56      0.34      0.30     12500
weighted avg       0.71      0.77      0.67     12500
```

```
 Classifier: Multinomial Naive Bayes
               precision    recall  f1-score   support

          Bad       0.26      0.00      0.00     10171
      Average       0.88      0.13      0.23     14456
         Good       0.78      1.00      0.87     78174

     accuracy                           0.78    102801
    macro avg       0.64      0.38      0.37    102801
 weighted avg       0.74      0.78      0.70    102801
```

Classification Report for SVM:

	precision	recall	f1-score	support
Average	0.49	0.03	0.06	2430
Bad	0.70	0.45	0.55	3516
Good	0.83	0.98	0.90	19054
accuracy			0.81	25000
macro avg	0.67	0.49	0.50	25000
weighted avg	0.78	0.81	0.77	25000

One tool for assessing a classification model's performance is a classification report. It demonstrates how successfully the model recognised various data categories.
Six classifiers are used in the report: SVM, Random Forest, Decision Trees, Multinomial Naive Bayes, and Logistic Regression. The classifiers are assessed using a dataset that has been classified as "good" or "bad". Each classifier's F1-score, recall, and precision for both classes are displayed in the report.

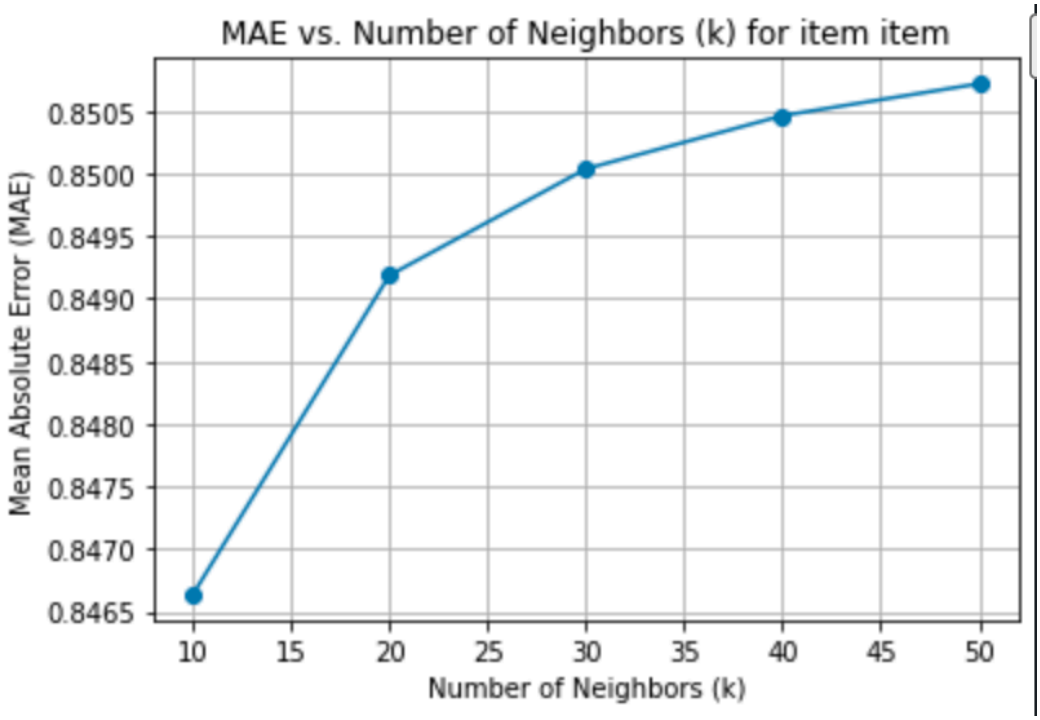
- Precision is the proportion of positive identifications that were actually correct. For example, if a classifier identifies 100 items as "Good" and 90 of them are actually good, then the precision is 0.9.
- Recall is the proportion of actual positive cases that were identified by the classifier. For example, if there are 100 items that are actually good, and the classifier identifies 80 of them, then the recall is 0.8.
- F1-Score is a harmonic mean of precision and recall. It is a way to balance the two metrics into a single score.

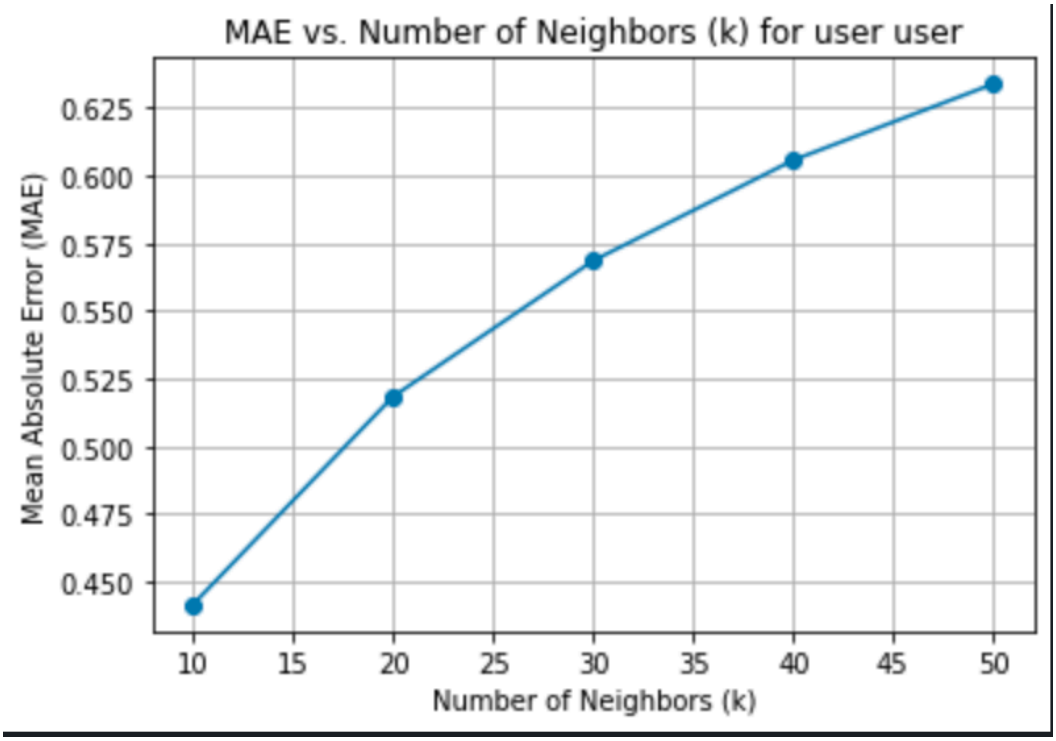
The report also shows the accuracy of each classifier. Accuracy is the proportion of all predictions that were correct.

One statistical technique that's frequently applied to categorization issues is logistic regression. Based on a collection of input data, this linear model forecasts the likelihood of a particular result.
A machine learning model called Decision Tree bases its choices on a structure like a tree. The nodes and branches comprise the tree. Every branch indicates a potential value for each characteristic, and every node represents a particular data feature. After being trained on a dataset, the decision tree follows the branches of the tree to identify new data points.

An ensemble of decision trees makes up the machine learning model Random Forest. Every tree in the forest receives training on a distinct subset of the data, and the average of the trees' predictions serves as the final forecast. Because random forests are less prone to overfit the data, they are frequently more accurate than single decision trees.Several-nominal One kind of naive Bayes classifier used for multi-class classification issues is called Naive Bayes. The foundation of naive Bayes classifiers is the idea that an item's features are unrelated to one another. Although naïve Bayes classifiers can still be useful for certain classification applications, this assumption is frequently untrue.
Machine learning models for regression and classification are called SVMs (Support Vector Machines). In order for SVMs to function, a hyperplane dividing the data points of one class from the data points of another class must be found. In order to maximize the margin between the two classes, the hyperplane is selected.

the logistic regression classifier, along with precision, recall, f1-score, and support metrics, evaluates its performance in predicting instances correctly for each class. precision measures accuracy, recall captures the proportion of correctly identified positive instances, and f1-score balances precision and recall. support indicates the number of occurrences for each class. in addition, accuracy represents overall correct classifications divided by total instances. macro and weighted averages provide an overall assessment of precision, recall, and f1-score.
In comparison, the logistic regression model demonstrates reasonable overall performance, achieving the highest accuracy among the models. However, it struggles notably with the "bad" class, exhibiting low precision and recall for that category.
Similarly, the decision tree classifier displays acceptable accuracy but performs inadequately in distinguishing between the "bad" and "average" classes, reflected in lower precision, recall, and f1-score for these categories.
the random forest model exhibits performance akin to the decision tree model, albeit with slightly improved results, particularly in accuracy.
On the other hand, the multinomial naive bayes classifier shows high accuracy but notably struggles with the "bad" and "average" classes, evident from considerably low precision and recall scores for these categories.
overall, logistic regression and random forest models outperform others in accuracy. Nevertheless, all models face challenges in accurately classifying instances in the "bad" and "average" classes, with multinomial naive bayes showing the poorest performance in this regard. logistic regression shines due to its highest overall accuracy and relatively consistent performance across all classes, albeit with room for improvement in classifying "bad" instances.


MAE vs. Number of Neighbors (k) for item item
Mean Absolute Error (MAE)
0.8505
0.8500
0.8495
0.8490
0.8485
0.8480
0.8475
0.8470
0.8465
10
15
20
25
30
35
40
45
50
Number of Neighbors (k)


MAE vs. Number of Neighbors (k) for user user
Mean Absolute Error (MAE)
0.625
0.600
0.575
0.550
0.525
0.500
0.475
0.450
10
15
20
25
30
35
40
45
50
Number of Neighbors (k)

The two diagrams displayed depict mean absolute error (mae) for item-based collaborative filtering (left) and user-based collaborative filtering (right), where mae assesses the variance between predicted and actual ratings, with lower values indicating better performance.
item-based collaborative filtering emphasizes finding similar items. the approach involves identifying the k most comparable items based on rating history and determining the average rating given by the user among these neighbors, which then serves as the predicted rating for the unrated item.
Conversely, user-based collaborative filtering centers on discovering similar users. it entails identifying the k most analogous users based on rating history and determining the average rating these neighbors gave to the unrated item, which becomes the predicted rating for the target user.
The parameter k, representing the number of neighbors in knn, significantly influences the recommender system's performance. The graphs illustrate how mae varies with different k values.
In the item-based collaborative filtering graph, mae initially starts high at small k values (around 0.85) and gradually decreases with increasing k. This trend reflects the system's transition from potentially insufficiently considering similar items to incorporating more, leading to improved predictions and reduced mae. However, excessive k values may introduce dissimilar items, elevating mae again.
In contrast, the user-based collaborative filtering graph depicts a sharper decline in mae. it begins high (around 0.625) and rapidly decreases as k increases. mae reaches its minimum around k=20 before slightly increasing, indicating that this method may be less sensitive to k values compared to item-based collaborative filtering. it achieves satisfactory prediction accuracy even with lower k values.
In summary, determining the optimal k value depends on the dataset and recommender system. experimentation with various k values is crucial to identify the one that minimizes mae and maximizes prediction accuracy.
Item-Item Collaborative Filtering: The MAE reduces as the number of neighbors (K) increases, and the lowest MAE is reached at K=50 with a value of approximately 0.846.
User-User Collaborative Filtering: The MAE also reduces here with an increase in the number of neighbors (K), but the decrease is steeper compared to the item-item case. The lowest MAE is reached at K=50 with a value of approximately 0.45.
In conclusion, the item-item collaborative filtering approach seems to yield better results (lower MAE) than the user-user based approach based on the data shown in the image.

Conclusion:

A product recommendation system based on Amazon review data is successfully developed by the project. Through the application of collaborative filtering and machine learning algorithms, the system provides users with personalised recommendations, increasing user satisfaction and engagement. The project

emphasises how crucial feature engineering, data preprocessing, and model evaluation are to creating efficient recommendation systems. To continuously improve the recommendation experience, more adjustments and optimisations can be done in response to user feedback and interactions.