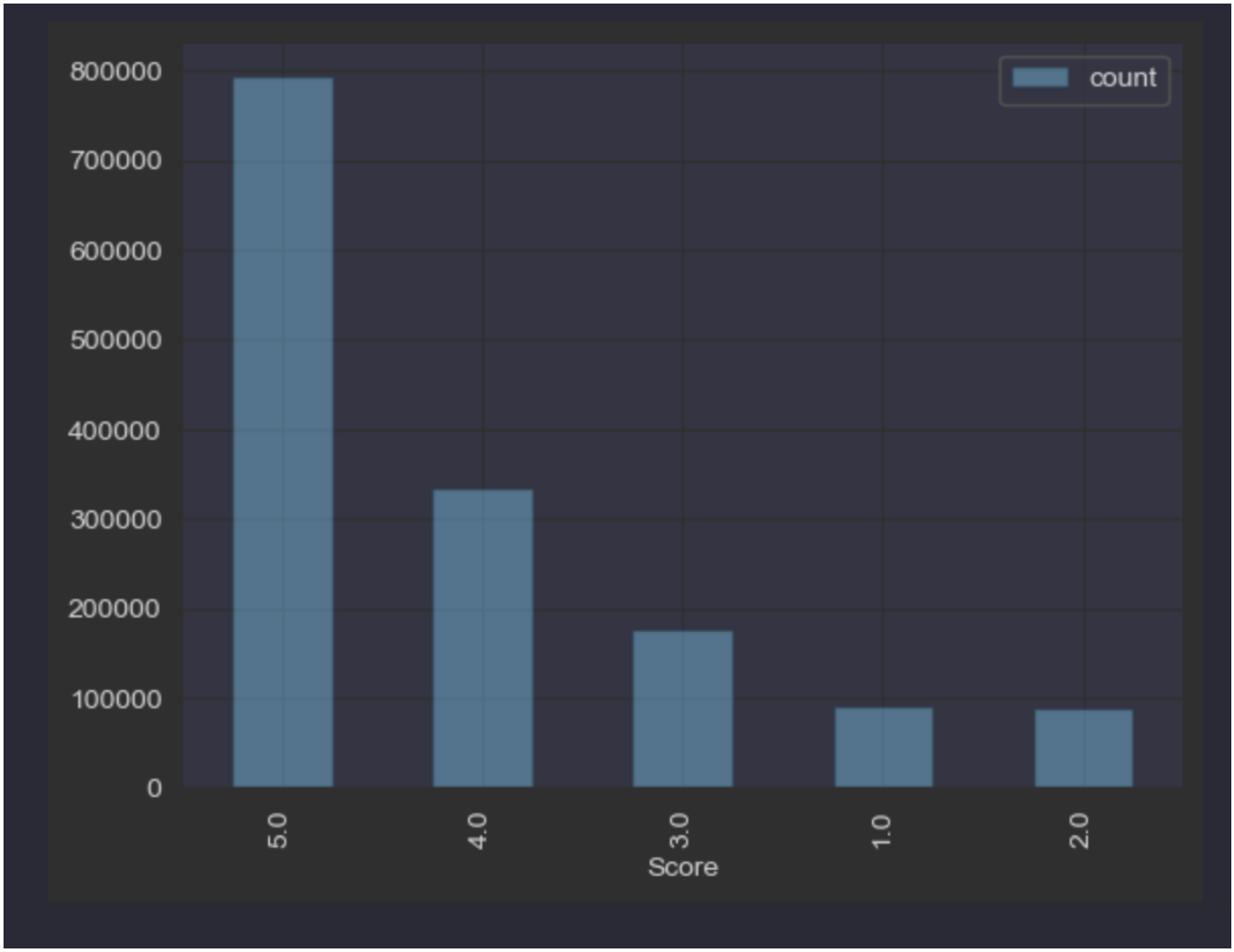
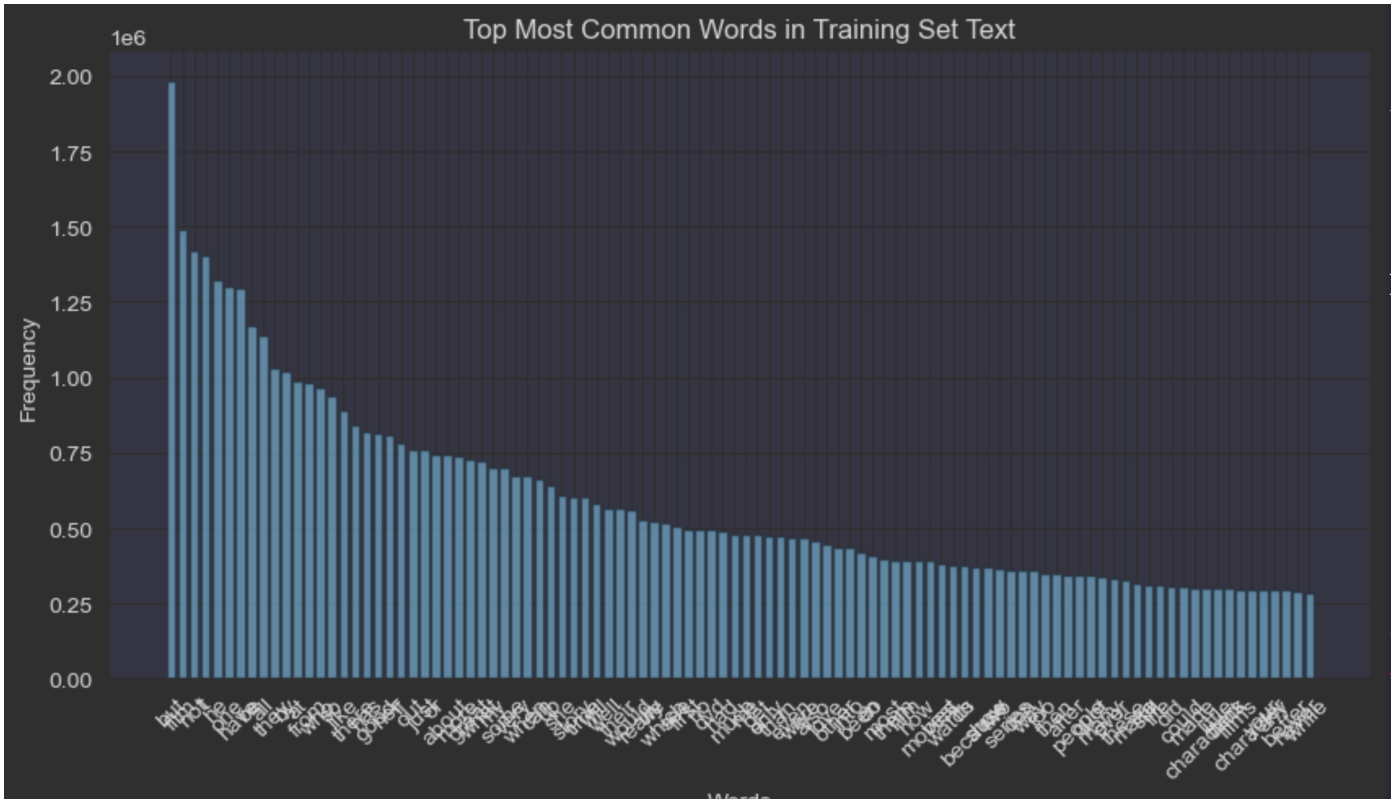
CS-506 Midterm - Fall 2024

Model

The K-Nearest Neighbors (KNN) algorithm is used for this project, with `n_neighbors` set to 500.

Data Distribution





Feature Engineering

Feature engineering is a crucial part of this project, as it allows us to derive meaningful insights from the data and improve model performance.

Basic Features

Helpfulness

- **Formula**: `Helpfulness = HelpfulnessNumerator / HelpfulnessDenominator`
- **Description**: This feature represents the helpfulness of a given review, where higher values indicate more helpful reviews.

Helpfulness Difference

- **Formula**: `HelpfulnessDiff = HelpfulnessDenominator - HelpfulnessNumerator`
- **Description**: `HelpfulnessDiff` captures how unhelpful a review is perceived to be by indicating the difference between the total helpfulness denominator and the actual numerator.

Text-Related Features

These features capture the sentiment and characteristics of the review text:

- **TextLen**: The length of the review text.
- **TestCos**: Cosine similarity between the text vector and a random vector.
- **SummaryLen**: The length of the review summary.

- **SummaryCos**: Cosine similarity between the summary vector and a random vector.

Aggregated Features

The following features capture the aggregated helpfulness values for individual users and products:

- **User Helpfulness Numerator Median**: The median value of the helpfulness numerator for a particular user.
- **User Helpfulness Numerator Difference**: The difference between the maximum and minimum helpfulness numerators for a user.
- **Product Helpfulness Numerator Median**: The median value of the helpfulness numerator for a specific product.
- **Product Helpfulness Numerator Difference**: The difference between the maximum and minimum helpfulness numerators for a product.

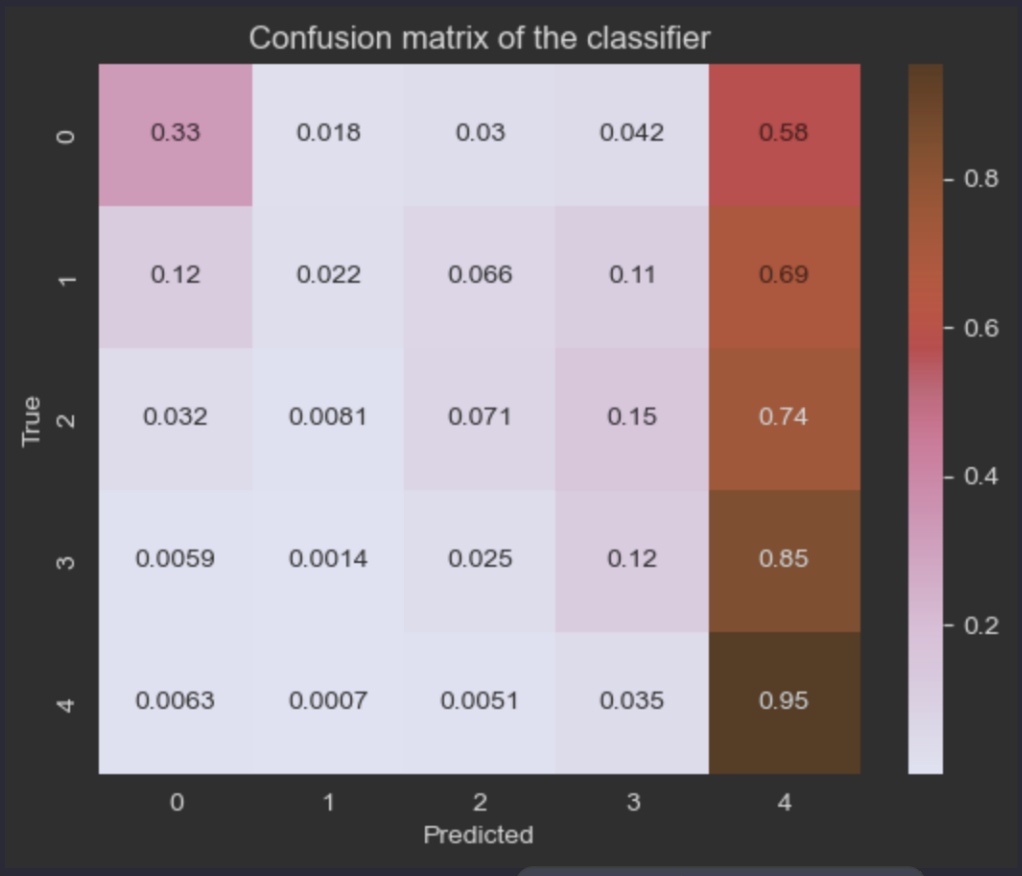
These aggregated features provide insights into the consistency of helpfulness scores across users and products, helping to reveal trends in user and product review patterns.

Confusion matrix

```
plt.show()
```

✓ [9] 408ms

```
Accuracy on testing set =  0.5660453066764332
```



Code | M↓ Markdown