Kelvin Kuang - U26418009 Midterm Writeup
CS506 - Data Science Tools and Applications

Data Preprocessing

The first initial preprocessing step I took was to undersample the data, given that the original dataset contained almost 800,000 five star reviews, but only about 100,000 one and two star reviews. Undersampling resulted in a balanced dataset of 89,678 samples per class. By setting each class to 89,678 samples, this approach ensures uniform representation and reduces potential biases in model predictions toward more common ratings like five-star reviews. Text null values were replaced with empty strings to avoid dropping rows, and timestamps were converted from Unix time to datetime format.

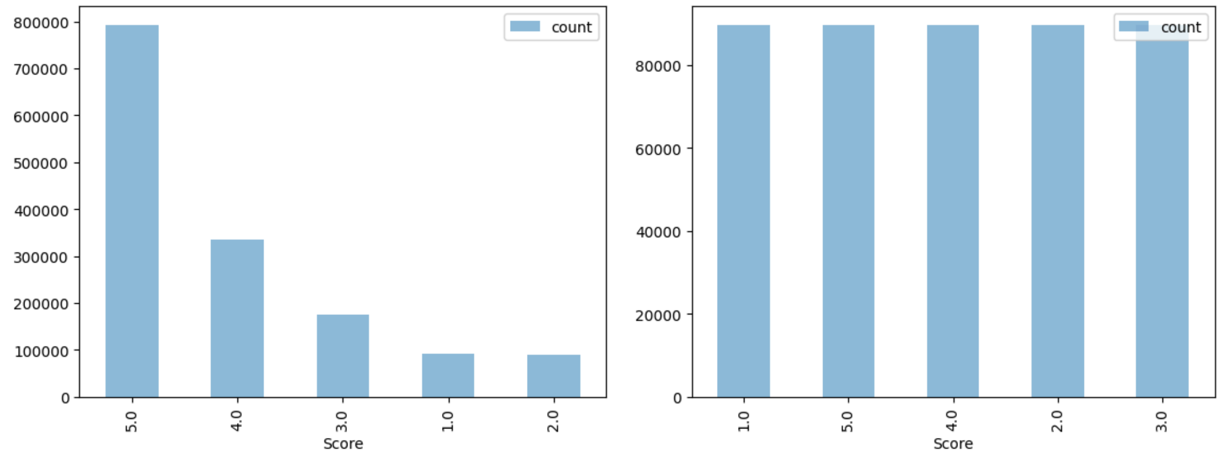

Figure 1. Original class frequency compared to undersampled class frequency. The original training dataset had a large imbalance of classes, with the majority of samples belonging to 5 star ratings. The undersampled dataset balanced the classes with each class consisting of 89,678 samples.

Feature Engineering

The features created to assess user review behavior provide a comprehensive view of each user's impact and consistency in contributing helpful reviews. Given the helpfulness denominators, which represents the total number of users who evaluated the review, whether they found it helpful or not, the helpfulness ratio was calculated by dividing the numerator by the denominator plus 1. The helpfulness ratio gives insight into the perceived quality or reliability of each review. Reviews with a high ratio are likely considered more valuable by the community, as a higher proportion of readers marked them as helpful. The average helpfulness per user feature calculates the mean helpfulness ratio across all reviews from each user, offering insight into how consistently useful their contributions are perceived by others. Rating variance per user captures the variability in a user's review scores, with a low variance suggesting the user rates products consistently, whereas a high variance indicates a broader range of opinions, potentially signifying an unbiased approach or diverse experiences. The helpfulness standard deviation per user quantifies the fluctuation in helpfulness ratings for each user's reviews, highlighting users whose contributions vary in perceived quality. To measure the proportion of helpful reviews, proportion of helpful reviews per user reflects the fraction of reviews with a helpfulness ratio exceeding a threshold (e.g., 0.5), emphasizing users who consistently write reviews regarded as valuable by others. Lastly, the high helpfulness count per user captures the number of reviews with a very high helpfulness rating (e.g., above 0.9), helping to identify users who frequently contribute top-rated reviews. Together, these features enrich the dataset with behavioral insights that can be crucial in identifying reliable reviewers and informing models aimed at predicting review helpfulness or relevance. Given the helpfulness numerators, which counts the number of users who marked the review as helpful, it reflects how many people found the review useful or insightful. For the text and summary features, the VADER sentiment analysis tool was used to extract four sentiment features from each review in the text column. These features—compound, positive, neutral, and negative scores—are useful for quantifying the emotional tone or sentiment of text. The compound score is a single, overall sentiment score for the text, ranging from -1 (most negative) to +1 (most positive). It combines all other scores into one value, giving a general indication of sentiment polarity. The positive score measures the proportion of positive words or phrases in the text. This value ranges between 0 and 1, where higher values indicate a stronger positive sentiment. The neutral score reflects the proportion of neutral language in the text, also ranging from 0 to 1. Reviews with higher neutral scores use more factual or balanced language without strong emotional cues. The negative score indicates the proportion of negative words or phrases in the text, with values ranging from 0 to 1. Higher values represent more negative language, highlighting reviews with a critical or negative tone. Term frequency-inverse document frequency (TF-IDF) vectorization was also utilized to transform the text and summary columns into numerical features. Truncated SVD was further utilized to reduce the dimensionality of these high-dimensional matrices.

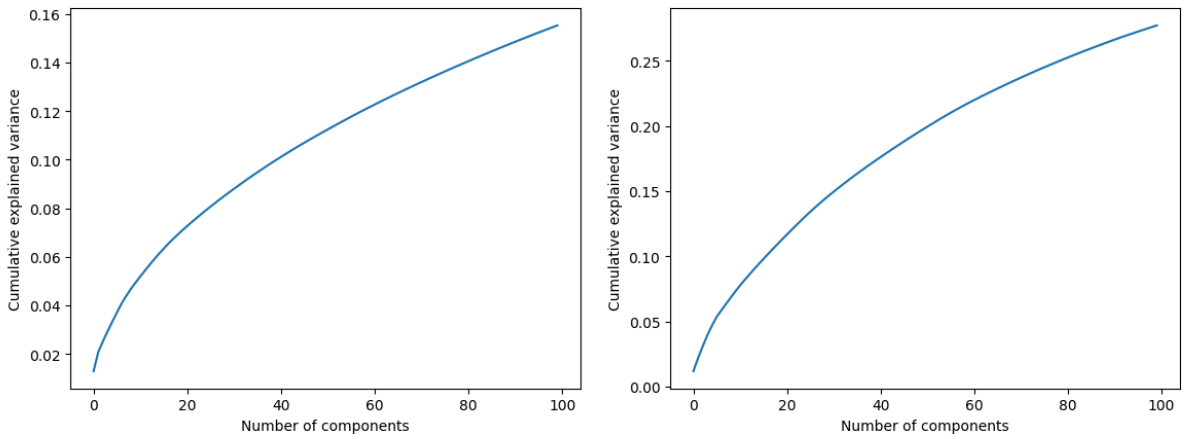

Figure 2. Scree plots after SVD for the text (left) and summary (right) columns. 100 components captures 16% of the variance for the text, and ~30% for the summary.

Model Selection

I chose to use XGBoost because of its ability to handle complex patterns in high-dimensional data, such as the TF-IDF and Truncated SVD features created. The TF-IDF vectorized features from text and summary columns can be high-dimensional and sparse, even after dimensionality reduction with Truncated SVD. XGBoost is designed to work well with such data, efficiently processing sparse matrices without memory inefficiencies, making it a great fit for the transformed text data.