Amazon Movie Reviews Rating Prediction

Xiling Wang
CS506

Midterm Write-up
October 28, 2024

1 Introduction

In this project, we aim to predict the star ratings for Amazon Movie Reviews based on review text, metadata, and user engagement features. This task presents challenges in handling a large dataset with missing values, high dimensionality, and class imbalance. Our solution involves targeted feature engineering, class balancing techniques, and careful model selection.

2 Data Preprocessing and Feature Engineering

In this section, I focus primarily on processing the review text, with emphasis on detecting positive and negative sentiment. In the third block in the notebook, we can see that ratings of five are significantly higher than all the others. Thus, if we predict all the five right and extract helpful features for low ratings, it will be a profound model.

2.1 Initial Observations and Assumptions

It is evident that temporal factors are less influential than text-based features; however, fully capturing a review's meaning is complex. Judging attitudes based on keywords like "great" or "garbage" provides useful insights.

2.2 Feature Engineering Steps

Text Sentiment Features:

- **Negation and Positive Word Flags:** We created `negation_flag` and `positive_flag` features to indicate the presence of negative or positive language in each review. Each flag was scaled to enhance its impact, as these markers correlate strongly with user satisfaction. Also, I add a n amplifier to these two flag, to make it more important than all the other attribute, this will be reflect in the part below
- **Net Positive Flag:** We combined the sentiment flags to form a `net_positive_flag`, representing the overall tone of each review.

Engagement Metrics:

- **Helpfulness Ratio:** Calculated as the ratio of users finding a review helpful to the total number of votes (`HelpfulnessNumerator / HelpfulnessDenominator`). We also implemented an adjusted ratio to handle cases with low vote counts.
- **Vote Counts:** Included the raw counts of helpful and total votes, as high engagement often correlates with more informative reviews.

Text Length Feature:

- **Review Length:** Counted the words in each review as `review_length`, assuming that longer reviews might indicate stronger engagement or opinion.

3 Class Balancing

Due to the significant imbalance in star ratings, with certain classes like 1, 2, and 3 underrepresented, we applied a mix of undersampling and oversampling to create a balanced training set. This approach ensured that each class was adequately represented during training, improving generalization.

- **Undersampling:** Higher-count classes (e.g., Score of 5) were undersampled to reduce dominance. We set the count for class 4 as a standard, ensuring that class 5 did not exceed twice the count of class 4.
- **Oversampling:** Classes with fewer instances (e.g., 1, 2, and 3) were resampled to match a target count. In this case, we grew these classes to 0.8 times the count of class 4.

4 Model Selection and Training with XGBoost

We chose XGBoost for its robustness in handling large datasets, interpretability, and effectiveness in multiclass classification. The model's hyperparameters were optimized through cross-validation, with early stopping applied to prevent overfitting.

4.1 Feature Scaling

The input features were scaled using `StandardScaler` to enhance model stability, which is particularly beneficial in tree-based models like XGBoost. This scaling was consistently applied to both training and validation sets.

4.2 Early Stopping and Evaluation

Early stopping was set with `early_stopping_rounds=5` to halt training when validation performance plateaued, ensuring an optimal model without unnecessary training rounds.

5 Results and Patterns Observed

5.1 Model Performance

The Random Forest model achieved an average cross-validated accuracy of 55%, indicating robust performance across classes. High accuracy in classes with extreme sentiments (ratings like 1 or 5) highlighted the effectiveness of sentiment and engagement features.

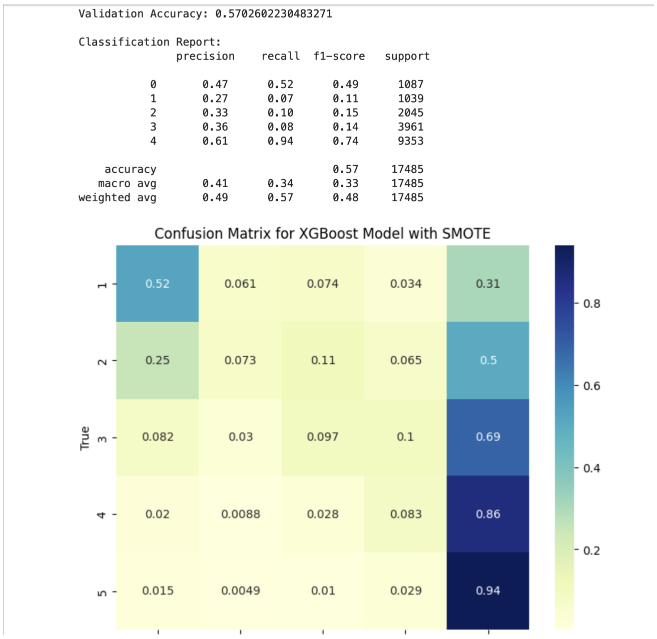
5.2 Aggregated Features for Users and Products

To enrich our feature set and capture behavioral patterns, we created **user-level** and **product-level aggregates** using helpfulness metrics and engagement attributes. These aggregates, based solely on non-target columns, help the model leverage broader contextual information without introducing data leakage. Below is a breakdown of these aggregated features:

5.2.1 User-Level Aggregates

For each user, we calculated the following:

- **User Helpfulness Average**: The mean helpfulness ratio (`HelpfulnessRatio`) across all reviews by the user, giving insight into the overall helpfulness of their reviews.
- **User Helpful Votes Average** and **User Total Votes Average**: Average counts of helpful and total votes per user, which indicate engagement with the user's reviews.
- **User Text Characteristics**: Calculated the average length of each review (`text_length`), the average number of words (`word_count`), and the average number of unique words (`unique_word_count`). These characteristics capture a user's typical review style, providing the model with a sense of review verbosity and diversity.



```
Validation Accuracy: 0.5702602230483271

Classification Report:
              precision    recall  f1-score   support

           0       0.47      0.52      0.49      1087
           1       0.27      0.07      0.11      1039
           2       0.33      0.10      0.15      2045
           3       0.36      0.08      0.14      3961
           4       0.61      0.94      0.74      9353

    accuracy                           0.57     17485
   macro avg       0.41      0.34      0.33     17485
weighted avg       0.49      0.57      0.48     17485
```

Confusion Matrix for XGBoost Model with SMOTE

1	0.52	0.061	0.074	0.034	0.31
2	0.25	0.073	0.11	0.065	0.5
3	0.082	0.03	0.097	0.1	0.69
4	0.02	0.0088	0.028	0.083	0.86
5	0.015	0.0049	0.01	0.029	0.94

Figure 1:

5.2.2 Product-Level Aggregates

For each product, we computed similar metrics to capture collective engagement patterns:

- **Product Helpfulness Average**: Average helpfulness ratio for each product, reflecting how helpful reviews of that product generally are.
- **Product Helpful Votes Average** and **Product Total Votes Average**: The mean counts of helpful and total votes for reviews on each product, providing insights into overall product engagement.
- **Product Text Characteristics**: Average length of reviews, average word count, and average unique word count for each product, capturing a summary of typical review structure and content diversity.

5.3 Observed Patterns

- **Sentiment's Predictive Power:** Reviews with distinct positive or negative tones aligned well with ratings, confirming that sentiment features significantly contributed to model accuracy.
- **User and Product Biases:** User and product historical ratings provided stability and helped prevent extreme fluctuations, particularly useful when textual data was limited.
- **Engagement Correlation with Critical Reviews:** High `HelpfulnessRatio` was commonly observed in lower-rated reviews, as critical reviews tend to attract more helpful votes.

6 Conclusion

From the Figure 1, we can see that the model isn't directly give every prediction 5.0 instead, given most of them 5.0 by predicting based on engineered data, and it successfully predict the 1.0 ratings for most cases. The model successfully combined sentiment, engagement, and behavioral patterns to predict Amazon Movie Review ratings with a significant degree of accuracy.