Predicting Amazon Review Star Ratings

CS 506 Data Science Tools and Applications

28 October 2024

1 Introduction

The goal of this project is to predict the star ratings of Amazon movie reviews based on review text and other associated metadata. We approached this problem using various machine learning models, focusing on both ensemble methods and linear models to determine which approach yields the highest accuracy while managing computational efficiency. The dataset contains Amazon movie reviews with features such as text, helpfulness scores, and time information.

The target variable is the star rating, ranging from 1 to 5 stars. This report outlines the methods used, the models applied, and the evaluation results. We specifically used models like Gradient Boosting, Linear Support Vector Machines (SVM), and Naive Bayes to analyze their effectiveness in handling text data and predicting review ratings.

2 Data Preprocessing and Feature Engineering

To effectively utilize the review data, several preprocessing and feature engineering steps were taken.

2.1 TF-IDF Vectorization

The review text was converted into numerical features using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. TF-IDF helps identify the most important words in each review, giving higher weight to words that are frequent in a particular review but less common across all reviews.

To reduce the computational load, the number of featuers was limited to 100 by setting the `max_features` parameter in the `TfidfVectorizer`. This step helped to manage the high dimensionality inherent in text data.

2.2 Additional Features

- **Sentiment Proxy Features**: Used a predefined list of positive and negative words to weigh reviews based on the sentiment of the language. Some exploratory analysis was extremely helpful in finding the usefulness or lack thereof of certain words. For example, ”good” was frequently used in negative, as well as positive reviews and was removed from the set of common positive words.

3 Modeling Approaches

Several models were tested, with Gradient Boosting ultimately being the most accurate.

3.1 Gradient Boosting

Gradient Boosting is an ensemble learning technique that combines multiple weak learners (decision trees) to create a strong predictive model. We used `GradientBoostingClassifier` from `scikit-learn` with 100 estimators and a learning rate of 0.1.

The model performed well due to its ability to capture complex relationships between the features. However, training time was relatively long, especially with a large dataset like ours.

3.2 Linear SVM

The intent was initially to use a method from lecture to create the final submission file. SVMs are effective for high-dimensional data like text, so it was theoretically ideal to be used in this problem. However, even with putting limits on max_iterations for the model, the LinearSVC program failed to finish in a reasonable amount of time. A significantly smaller dataset would need to be used, and I moved on to other methods.

3.3 Naive Bayes

`MultinomialNB` was applied, a Naive Bayes variant suitable for count-based features like TF-IDF. Naive Bayes models are computationally efficient and ideal text classification. However, Naive Bayes assumes feature independence, which may not hold for our dataset, and limited certain features such as my Sentiment variable that features negative numbers. It was not as accurate as the final Gradient Boosting method, and is not used for the final results.

4 Evaluation

4.1 Accuracy and Confusion Matrix

The accuracy of each model was evaluated on a held-out test set. The best performing model, `GradientBoostingClassifier`, achieved an accuracy of approximately 56.0%.

4.2 Feature Importance

Feature importance analysis showed that sentiment features and helpfulness scores were among the most influential factors in predicting the star rating. This insight aligns with expectations, as positive sentiment and helpful reviews are likely indicators of higher ratings.

5 Conclusion

This project demonstrated the effectiveness of feature engineering and different machine learning models in predicting Amazon review star ratings. The models tested included

Gradient Boosting, Linear SVM, and Naive Bayes.

Gradient Boosting provided the most accurate model, as Naive Bayes was slightly less accurate.

Future work to improve the model would likely involve more intensive data processing involving the written reviews. This model used a manually inputted list that was influenced by some top words generated from the data set. Based on the feature importance output, the manually generated positive/negative word sets were impactful to the model, but a more robust method could improve our accuracy.