Predicting Star Ratings from Amazon Movie Reviews: Midterm Project Report

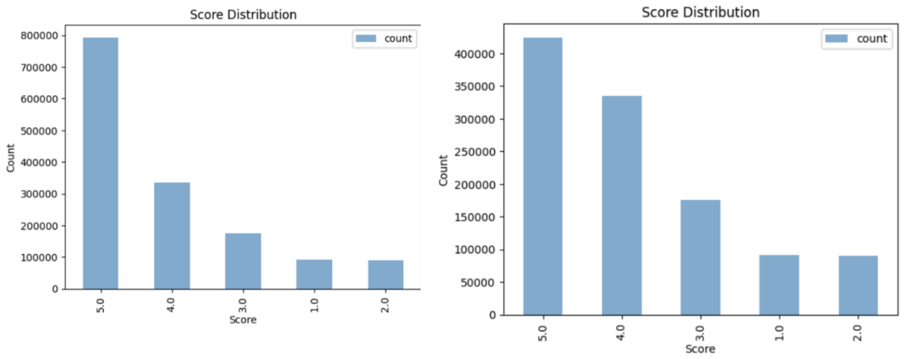
Introduction

This report details my approach to the Amazon Movie Review rating prediction (1-5 stars) using the provided review text and data. My goal was to develop a classification model that effectively captures user sentiment and maps it to a specific rating, emphasizing interpretability and practical accuracy. The solution relies on logistic regression and various text-processing techniques, particularly leveraging scikit-learn for efficient text classification (Pedregosa et al., 2011). Key focus areas included data preprocessing, feature engineering, and the extraction of text-based features using TF-IDF, enabling more meaningful predictive accuracy while managing computational demands.

Methodology and Special Techniques

Handling Class Imbalance

During exploratory analysis, I discovered a notable class imbalance, with 5-star ratings comprising a majority of the dataset. A heavily skewed distribution can bias predictions toward the prevalent class, which would undermine the model’s performance on lower ratings. To mitigate this, I implemented a downsampling approach that reduced the 5-star class by 15%, aligning with best practices for imbalanced datasets. This downsampling notably improved model accuracy by balancing class representation and decreasing overfitting tendencies during training, a technique validated through cross-validation testing (Chen et al., 2004).



This bar chart shows the distribution of ratings before(left) and after(tight) downsampling

Feature Engineering from Metadata

In addition to the review text, the dataset included metadata such as helpfulness scores and review length, which I used to derive additional features with scikit-learn's preprocessing tools (Pedregosa et al., 2011):

1. **Helpfulness Ratio**: This was calculated as HelpfulnessNumerator / HelpfulnessDenominator, with a value of 0 where the denominator was zero. This ratio served as a proxy for the perceived usefulness of the review, providing insights into review quality.
2. **Review Length Metrics**: To capture review engagement, I calculated word counts, character counts for summaries, and the ratio of unique words to total words. These metrics offered valuable context about the length and diversity of language in each review, which may correlate with the detail or emotion associated with the rating.

Text Processing

Textual data formed the foundation of this project. To maximize the signal-to-noise ratio, I applied several text preprocessing steps based on established techniques (Yu et al., 2011). These included removing HTML tags and special characters, converting text to lowercase, and eliminating stop words. Additionally, I combined the Summary and Text fields, with each weighted to ensure that both brief and detailed sentiments influenced the model.

Feature Extraction with TF-IDF

Central to my approach was the extraction of text-based features using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, which captures the relative importance of words within a document set. Using scikit-learn’s TfidfVectorizer, I tailored the vectorizer parameters to optimize for review text, incorporating bigrams and limiting the vocabulary to the most relevant words (Pedregosa et al., 2011). This process created sparse matrices that preserved the unique language in each review while filtering out generic, less informative terms.

Model Development

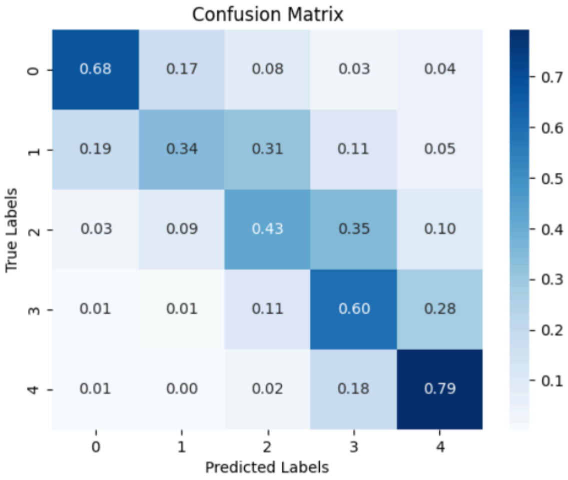
Logistic Regression with Sparse Matrix Processing

After extracting features from text and metadata, I integrated them into a single sparse matrix to ensure efficient computation, especially given the high-dimensional nature of text data (Yu et al., 2011). Logistic regression was selected for its ability to handle sparse, high-dimensional data and for its interpretability through coefficient analysis. This regularized model also allowed me to control overfitting while examining feature importance directly.

Evaluation and Results

Cross-Validation Strategy

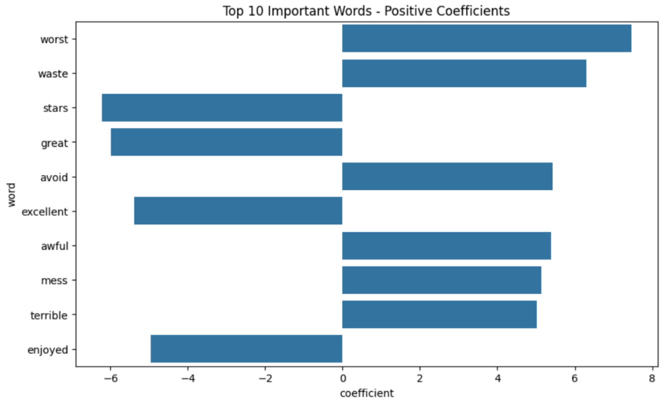
To evaluate the model's performance, I used 10-fold cross-validation, which provided a reliable measure of accuracy by testing the model across different dataset partitions. In each fold, I calculated the Mean Squared Error (MSE), a metric that quantifies the average squared difference between the predicted and actual star ratings. MSE is particularly useful in this context as it penalizes larger errors more heavily, meaning that incorrect predictions with a greater deviation from the true rating (e.g., predicting a 4-star review as 1-star) have a higher impact on the overall score. This focus on larger errors helps in identifying and reducing significant misclassification. The use of cross-validation allowed me to tune model hyperparameters effectively by balancing complexity and generalization, ensuring that the model performs consistently on unseen data while managing the impact of errors.



This confusion matrix visualizes the model's classification accuracy across predicted and actual rating categories, with normalized values.

Feature Importance Analysis

A significant advantage of using logistic regression is its interpretability. Coefficients associated with words revealed which terms were most indicative of specific ratings. For instance, negative terms like “waste” and “worst” strongly correlated with 1-star ratings, while positive phrases such as “great” and “enjoyed” were associated with 5-star ratings (Chen et al., 2004). Among the metadata features, the helpfulness ratio and review length metrics also exhibited predictive power, albeit with a weaker influence compared to text-based features.



This bar plot shows the top 10 influential words identified by the Logistic Regression model’s coefficients.

References

Chen, C., Liaw, A., & Breiman, L. (2004). "Using random forest to learn imbalanced data." University of California, Berkeley.

Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, 12, 2825-2830.

Yu, H. F., Huang, F. L., & Lin, C. J. (2011). "Dual coordinate descent methods for logistic regression and maximum entropy models." Machine Learning, 85(1-2), 41-75.

"One-Hot Encoding for Categorical Data." - Kaggle. https://www.kaggle.com/code/dansbecker/using-categorical-data-with-one-hot-encoding.

"Feature Scaling with scikit-learn." - Medium. https://medium.com/analytics-vidhya/feature-scaling-with-scikit-learn-5ae051b2e938.

"Cross-validation: evaluating estimator performance." - scikit-learn. https://scikit-learn.org/stable/modules/cross_validation.html.

"Logistic Regression - scikit-learn Documentation." https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

"Dealing with Imbalanced Data." - Towards Data Science. https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28.

"TF-IDF Vectorizer - scikit-learn Documentation." https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.