Developing an Ensemble Boosting Model for Amazon Movie Review Classification

Abstract

This paper presents the development of a predictive model aimed at accurately classifying Amazon Movie Reviews based on star ratings, utilizing ensemble boosting algorithms and feature engineering techniques. The challenge was to achieve high predictive accuracy without employing deep learning methods. The final model combines three boosting algorithms—HistGradientBoostingClassifier, GradientBoostingClassifier, and XGBClassifier—using a soft voting ensemble approach. Feature engineering involved extracting sentiment scores using TextBlob and creating a helpfulness ratio from review metadata. Additionally, attempts were made to enhance the model by identifying positive and negative keywords such as "loved," "terrible," and "excellent," but this did not significantly improve results. The model achieved an accuracy of approximately 56%. This paper details the strategy, implementation, and evaluation of the model, highlighting the effectiveness of ensemble methods and feature engineering in text classification tasks.

Introduction

The exponential growth of user-generated content on e-commerce platforms like Amazon has made sentiment analysis an indispensable tool for businesses. Understanding customer feedback through reviews enables companies to improve products, tailor services, and enhance customer satisfaction. In this context, accurately classifying reviews based on their sentiment and content becomes crucial.

This project aimed to develop a predictive model to classify Amazon Movie Reviews into their respective star ratings (1 to 5 stars) using review metadata and textual content. The primary objectives were to achieve high predictive accuracy without deep learning, ensure computational efficiency suitable for large datasets, and maintain transparency in the model's decision-making process. The challenge was to leverage ensemble-based methods, feature engineering, and boosting algorithms to create a robust model capable of handling the complexities inherent in textual data.

Initial Approach

Baseline Models

The initial strategy involved experimenting with baseline machine learning models commonly used in text classification.

K-Nearest Neighbors (KNN)

KNN classifies new data points based on the majority class among its k nearest neighbors. Various distance metrics and values of k were tested. The model achieved an accuracy of approximately 40% but faced challenges, such as high computational cost during prediction and ineffectiveness in high-dimensional spaces typical of textual data.

Naive Bayes Classifier

Naive Bayes is a probabilistic classifier based on Bayes' theorem with an assumption of feature independence. Both Multinomial and Gaussian variants were evaluated, resulting in around 40% accuracy. However, the unrealistic independence assumption limited its efficacy.

Limitations of Baseline Models

The unsatisfactory performance of these models highlighted their inability to capture complex patterns and feature interactions, prompting a shift toward more sophisticated algorithms.

Transition to Boosting Algorithms

Rationale for Boosting
Boosting algorithms improve predictive performance by combining multiple weak learners into a strong learner. They capture nonlinear relationships, improve robustness, and generalize well. These qualities make them suited for handling skewed datasets, as seen in this project.

Selection of Boosting Algorithms
Three boosting algorithms—HistGradientBoostingClassifier, GradientBoostingClassifier, and XGBClassifier—were selected for their complementary strengths.

Model Development

Individual Models

- ***HistGradientBoostingClassifier***
  An efficient implementation of gradient boosting using histogram-based binning. Hyperparameters such as learning_rate, max_iter, and max_depth were tuned, with early stopping to prevent overfitting.
- ***GradientBoostingClassifier***
  A traditional gradient boosting model building additive models in a forward stage-wise manner. Key hyperparameters were tuned, including n_estimators, learning_rate, and max_depth, with subsampling to reduce overfitting.
- ***XGBClassifier (XGBoost)***
  XGBClassifier uses an optimized gradient boosting framework. Its advantages include parallel processing, L1/L2 regularization, and handling of missing data. Cross-validation determined optimal hyperparameters, including eta, gamma, max_depth, lambda, and alpha.

Ensemble Methodology

- ***Soft Voting Ensemble***
  The three models were combined using a VotingClassifier with soft voting. Soft voting averages the predicted class probabilities and considers the confidence of each model, providing robustness and reducing variance and bias.

Feature Engineering and Extraction

Data Preprocessing
Missing values in HelpfulnessNumerator and HelpfulnessDenominator were filled with zeros. Data type conversions ensured the correct formats.

Engineered Features

- ***Helpfulness Score***
  Calculated as the ratio of HelpfulnessNumerator to HelpfulnessDenominator, it was assumed that higher scores indicate more informative reviews.
- ***Sentiment Features Using TextBlob***
  Polarity and subjectivity scores were extracted from reviewText using TextBlob.

- ***Keyword Analysis***
  Positive and negative keywords were identified within the review text, but incorporating these features did not significantly improve the model's accuracy.
- ***Temporal Features***
  Additional features like Year and Month were extracted to capture temporal trends in reviews.

Sentiment Analysis Optimization
To optimize computational time, VADER was tested but ultimately TextBlob was chosen for accuracy. Parallel processing was implemented to enhance scalability.

Handling Class Imbalance
Class imbalance was managed through resampling, class weight adjustments, and using weighted F1-score as an evaluation metric.

Results

The final model achieved an accuracy of approximately 56%. High precision and recall were observed for majority classes. The ensemble model outperformed individual models, demonstrating improved stability and accuracy.

Model Comparison
The ensemble approach proved superior to individual models, highlighting the benefits of combining models for enhanced performance.

Conclusion

This project successfully developed a predictive model for classifying Amazon Movie Reviews using ensemble boosting algorithms and feature engineering. Key findings include:

1. *Boosting algorithms' effectiveness in handling complex data.*
2. *Feature engineering's critical role in model performance.*
3. *The limited impact of simple keyword-based features in enhancing accuracy.*
4. *The advantages of ensemble methods in improving accuracy and robustness.*
5. *The importance of optimization through hyperparameter tuning and regularization.*

Future Work
Future work may involve incorporating advanced text features like TF-IDF or topic modeling, using deep learning techniques, and employing model interpretability tools to understand feature contributions better.

References

1. *Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician, 46(3), 175–185.*
2. *Zhang, H. (2004). The Optimality of Naive Bayes. AAAI, 3(1), 562–567.*
3. *Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.*
4. *Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 29(5), 1189–1232.*
5. *Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.*
6. *Kuncheva, L. I. (2004). Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons.*
7. *XGBoost Documentation. (2023). Parameters for Tweedie Regression.*
8. *Loria, S. (2018). TextBlob Documentation.*
9. *Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text.*
10. *Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks.*
11. *Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions.*