Stars Aligned

Data Driven Solutions for Amazon Review Predictions

October 28, 2024

Anamu Uenishi

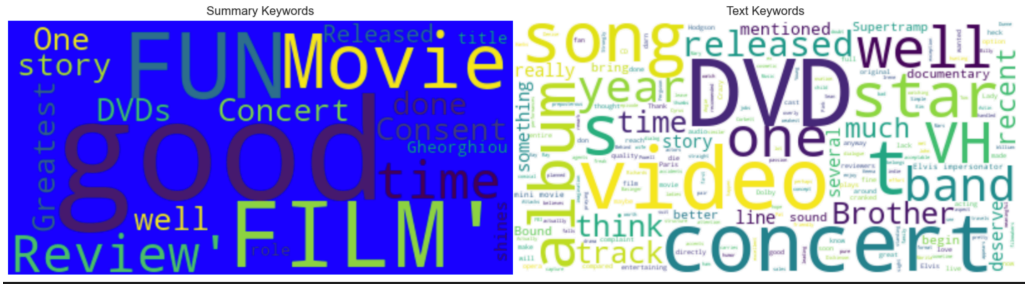
CS 506

Part 1: Feature Engineering and Understanding the Data

The success of any predictive model is strongly tied to identifying which features are most relevant. Thus, the initial focus of this project was on thoroughly analyzing the data, understanding patterns, and engineering features that could provide meaningful insights for the model.

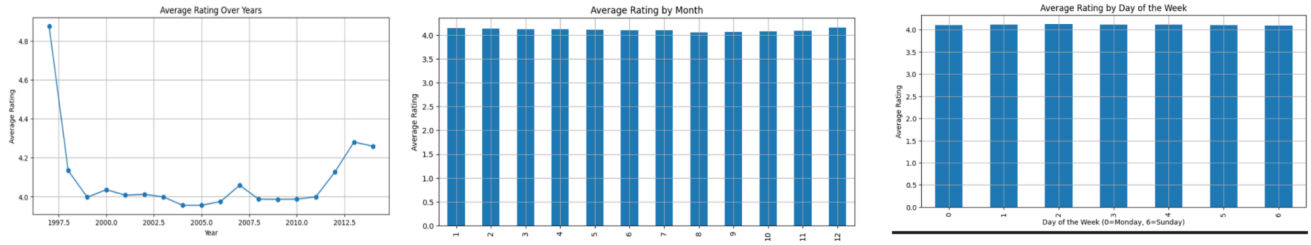
Sentiment Analysis and Text Features

To identify keywords that might predict review scores, I created word clouds for both 'Summary' and 'Text.' Words like *good* and *fun* in positive reviews suggested that sentiment could be predictive of scores. I applied VADER sentiment analysis, initially including compound, positive, neutral, and negative scores as features. However, further analysis showed that only the compound score had significant predictive value (feature importance: 0.0965), while the other scores added unnecessary complexity (importance around 0.004). Using only the compound score retained key sentiment information without inflating the model's dimensionality.



Temporal Patterns: Time Feature Analysis

Time data often holds hidden insights, and I explored the Unix time feature in search of temporal trends in review scores. To achieve this, I converted the Unix timestamp into days, months, and years, aiming to identify any correlation between when a review was written and its corresponding rating. While day and month exhibited almost negligible correlations with the review scores, the year feature demonstrated a correlation coefficient of 0.0863 with ratings. This indicated that review trends could change over time - perhaps reflecting shifts in product quality or changing user preferences. By including 'year' as a feature, I allowed the model to account for these subtle time-related patterns.

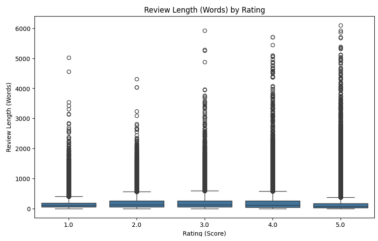


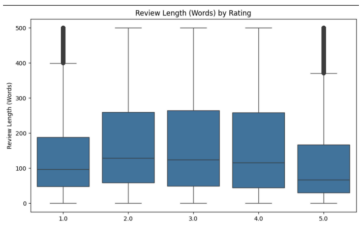
Review Length and Rating Patterns

In examining the relationship between review length and rating, I hypothesized that longer reviews might indicate a stronger emotional investment, which could correspond to more extreme ratings (either very positive or very negative). The initial boxplot (left) includes all review lengths and shows a vast number of outliers, especially at higher word counts, which obscures any discernible patterns.

To clarify this relationship, I clipped review lengths to a maximum of 500 words, based on the mean word account, producing the plot on the right. This approach reveals a more structured trend: reviews with ratings of 1.0 and 5.0 tend to be longer on average than those with mid-range scores, indicating that people with strong opinions are more

likely to write detailed reviews. This insight supports the inclusion of review length as a feature, as it may help the model distinguish between neutral and extreme sentiments.

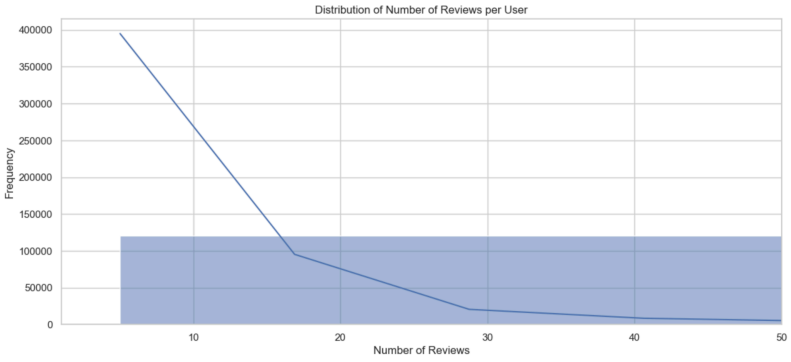




User and Product Rating Patterns

In many e-commerce datasets, users often have reviewing habits, and products can have a natural tendency towards certain ratings. This prompted a closer look at how users and products behaved within the dataset:

- **User Rating Patterns:** Analysis revealed that 123,960 users submitted multiple reviews, as shown by the high frequency of users with multiple reviews in the graph. This observation suggested potential biases, with certain users more likely to rate positively or negatively. To leverage this insight, I created a ‘user_mean_rating’ feature to capture each user's average rating tendency, helping the model account for individual rating behaviors.



- **Product Rating Patterns:** Analysis revealed that 50,052 products had multiple reviews, as shown in the graph with the frequency distribution of reviews per product. This observation suggested that products with higher review counts might have more established reputations, either positively or negatively. To capture this insight, I added a ‘product_mean_rating’ feature, allowing the model to account for the overall perception of each product based on its average rating across users.

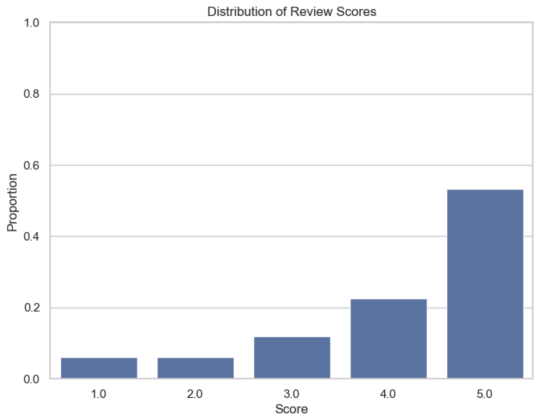
Variability in Ratings: Capturing Uncertainty

To further refine these insights, I introduced measures of variability for both users and products. By calculating the standard deviation of ratings given by a user 'user_rating_variability' and the standard deviation of ratings received

by a product 'product_rating_variability', I captured the consistency or inconsistency in their ratings. High variability could indicate a user's tendency to give very different ratings based on their mood or experience, or a product's tendency to elicit a wide range of reactions from different users. This variability feature added another layer of nuance to the model, enabling it to understand not just the average behavior but also the uncertainty associated with user and product ratings.

Part 2: Tackling Data Imbalance

This chart reveals a significant imbalance, with over 50% of reviews rated as 5 stars, while only around 6% are 1-star ratings. This distribution poses challenges, as models may favor the dominant 5-star class, ignoring lower ratings. To mitigate this, I considered several techniques: using class weighting, synthetic data generation, and stratified splits to preserve class distribution. However, generating synthetic data on a large dataset (1.6 million entries) proved computationally intense, and class weights often overcompensated, leading to reduced accuracy on the majority class. Instead, I focused on models robust to imbalances and employed cross-validation methods that preserved the original class distribution for better evaluation.

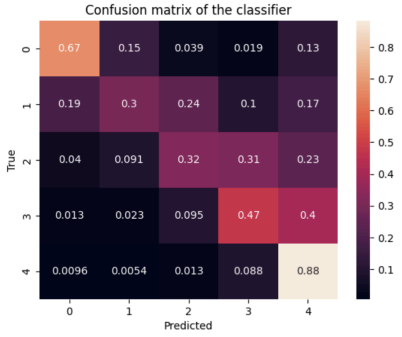


Part 3: Model Optimization and Training

To select the optimal model, I initially tested Support Vector Machine (SVM), Random Forest, and Gradient Boosting Classifier, with Gradient Boosting emerging as the best performer, achieving a baseline accuracy of 0.6626. I focused on fine-tuning this model using Bayesian Optimization combined with Stratified K-Fold cross-validation (SKF) to maintain class proportions. Bayesian Optimization was chosen over exhaustive search methods for its efficiency, honing in on promising parameter ranges and reducing computational costs. Running hyperparameter tuning on an NVIDIA RTX 3090 overnight increased accuracy to 0.6750. Further feature importance analysis showed that the ‘review_length_words’ feature had minimal impact (0.0079), so I removed it to reduce dimensionality, resulting in a final accuracy of 0.6854.

Part 5: Model Evaluation

The model evaluation reveals a clear challenge in predicting middle ratings (2, 3, and 4), as seen in both the confusion matrix and classification report. This difficulty likely stems from the underrepresentation of these classes in the training data. The model shows a tendency to misclassify reviews into the majority class, particularly rating 5, due to class imbalance, which skews the model towards predicting higher scores. Interestingly, while class 1 (the most underrepresented class) has a relatively high precision, the recall remains low, indicating the model's hesitation to classify reviews as strongly negative unless highly certain. Overall, the model achieves reasonable performance with a weighted average F1-score of 0.69, but it struggles with underrepresented classes, highlighting a potential area for further improvement.



	precision	recall	f1-score	support
1.0	0.67	0.63	0.65	24240
2.0	0.30	0.39	0.34	17142
3.0	0.32	0.45	0.38	31043
4.0	0.47	0.54	0.50	73799
5.0	0.88	0.78	0.83	225112
accuracy			0.68	371336
macro avg	0.53	0.56	0.54	371336
weighted avg	0.71	0.68	0.69	371336