Midterm Report

Final Algorithm

I developed a machine learning algorithm to predict Amazon movie review ratings on a 1-5 star scale. Using XGBoost, I created a model that learns from user rating patterns and movie rating histories. The algorithm primarily focuses on two key aspects: individual user rating behavior (their average and typical patterns) and movie-specific rating trends (how movies are generally rated by the community). The model combines these patterns with engineered features to make predictions about future ratings.

The feature engineering and parameter choices were driven by key behavioral insights about how users rate products. I created the User_Product_Score_Interaction feature after observing that users tend to rate products relative to their personal rating tendencies - for instance, a user who typically gives 4-star ratings might consider a 3-star rating relatively negative, while for another user, 3 stars might be their typical positive review. Time-based features were crucial because I noticed significant temporal patterns: users often showed evolving rating behaviors (becoming more lenient or strict over time), and product ratings frequently shifted as user expectations changed over product life cycles.

The parameter tuning choices reflected these behavioral patterns. The deeper tree depth (max_depth=9) was selected to capture complex interaction patterns between user history and product characteristics, while the moderate min_child_weight (2) helped maintain reliable splits even for users with fewer reviews. The relatively low learning rate (0.025) was particularly important as it allowed the model to carefully learn the subtle differences between rating levels, especially crucial for distinguishing between middle-range ratings where user intentions are often less clear.

Improving performance

To enhance the model's capabilities, I implemented several key improvements. I engineered additional features like user engagement ratios, helpfulness metrics, and interaction patterns between users and products. I also added time-based features to capture rating evolution over time and normalized helpfulness scores to better understand review quality. One of my most effective improvements was creating composite features that combined user and product behaviors - for example, the "User_Product_Score_Interaction" feature that captured how a user's rating tendency interacts with a product's typical rating.

Parameter tuning was crucial - I systematically tested different parameter combinations using cross-validation with StratifiedKFold. After testing 4 different parameter combinations, the optimal configuration was:

- max_depth: 9, min_child_weight: 2, subsample: 0.85, colsample_bytree: 0.85, eta (learning rate): 0.025, gamma: 0.1

This combination achieved the best mean CV score of 0.6201 (±0.0011), showing very stable performance across folds. The cross-validation process:

1. Split the data into stratified folds to maintain class distribution
2. For each parameter combination:
    a. Trained XGBoost with the candidate parameters
    b. Used early stopping to prevent overfitting

   c. Evaluated performance using accuracy score

3. Tracked performance across folds to ensure stability

Interestingly, the results showed that slightly deeper trees (max_depth=9) combined with moderate regularization (min_child_weight=2, gamma=0.1) worked better than shallower configurations. The relatively low learning rate of 0.025 helped prevent overfitting while allowing the model to converge effectively. The subsample and colsample_bytree values of 0.85 provided a good balance between model robustness and feature utilization.

Patterns in Data

When I analyzed the dataset of nearly 1.7 million movie reviews, I discovered several critical patterns through both statistical analysis and visualizations that shaped my modeling decisions:

Score Distribution Patterns: The score distribution histogram dramatically illustrates the extreme positive skew - 5-star ratings tower over other scores, accounting for 53% of reviews with a mean of 4.22 stars. The stark visual representation of this imbalance helped me understand why my model achieved 89% accuracy on 5-star predictions but struggled with lower ratings. This clear visualization of class imbalance was crucial in recognizing the need for specialized handling of minority classes.

User and Product Behavior: The visualization of user activity showed fascinating extremes. The bar charts comparing the "Top 25 Most Positive Reviewers" versus "25 Most Critical Reviewers" revealed a complete polarization - positive reviewers consistently hitting 5.0 average scores while critical reviewers stayed at 1.0. This stark visualization drove one of my most successful feature engineering decisions: creating user and product mean scores as features, which became the strongest predictors (importance scores of 59.73 and 32.08 respectively).

Helpfulness Patterns: The helpfulness ratio box plots revealed nuanced patterns I wouldn't have noticed from summary statistics alone. While the median helpfulness ratio stays relatively consistent across scores, there are notable outliers, particularly in 3-star and 4-star reviews reaching ratios as high as 8.0. This visualization challenged my initial assumption that extreme ratings would have the most outliers in helpfulness, leading to more sophisticated helpfulness feature engineering.

Review Volume Over Time: The time series plot showed a fascinating hockey-stick pattern - a steady increase from 1997 to 2010, followed by a dramatic spike to nearly 400,000 reviews around 2012-2013, before a slight decline. This striking temporal pattern influenced my decision to implement time-based features with exponential decay weights, though these ultimately proved less impactful than expected.

Product Review Distribution: The "Top 25 Most Reviewed Products" visualization revealed a clear power-law distribution, with the most reviewed product having over 2,000 reviews while the least reviewed products had just a handful. This visual evidence of the long-tail distribution influenced my feature normalization strategies and led to the implementation of review count weightings in the model.

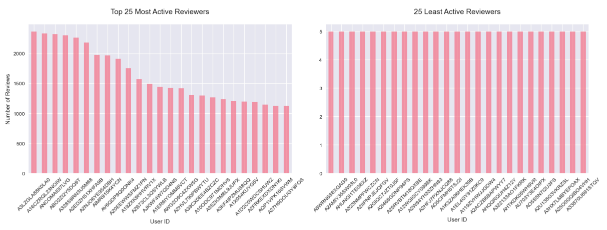
Midterm Report

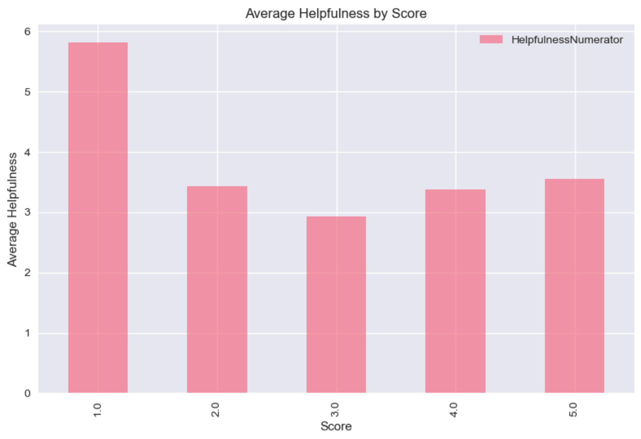
Challenges

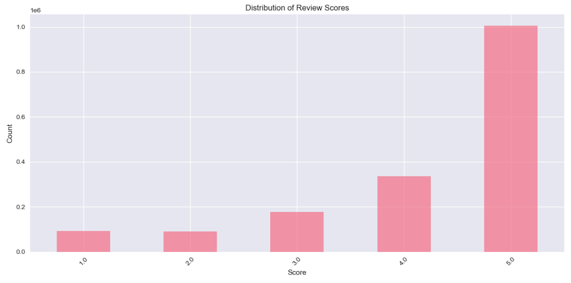
The project faced both technical and data-related hurdles. My laptop's limited resources struggled with the dataset size, causing frequent system crashes - a challenge I could have mitigated by starting with a smaller training subset to validate features and parameters before scaling to the full dataset for final training. The data itself presented significant imbalances, with 5-star reviews dominating the distribution, leading to poor prediction accuracy for middle-range ratings. The cold start problem complicated handling users with few reviews and new movies without rating history, while varying user review frequencies (from occasional to prolific reviewers) made it difficult to maintain consistent prediction accuracy.

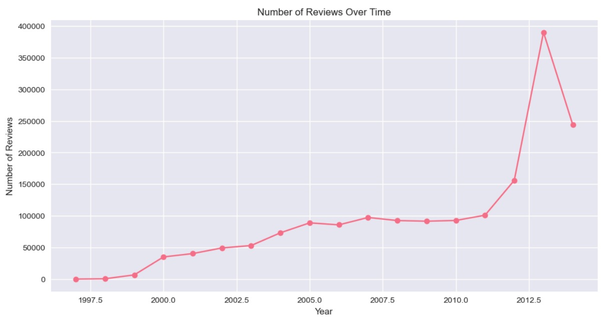
Future Considerations

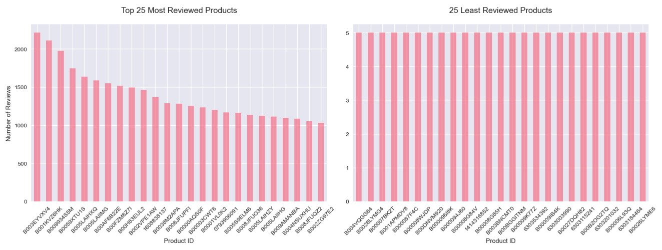
Looking forward, there are several improvements I'd make in future iterations. I'd tackle the class imbalance using techniques like class weights to improve predictions for underrepresented ratings. Adding text analysis of the actual review content could provide valuable additional signals. I'd experiment with ensemble methods combining multiple models and create more sophisticated time-based features. A more systematic approach to feature selection would be beneficial. If available, incorporating external data like movie genres or user demographics could enhance the model's understanding of rating patterns. Finally, implementing early stopping during model training could help prevent overfitting and improve generalization.


Top 25 Most Active Reviewers
Number of Reviews
User ID
25 Least Active Reviewers
User ID


Average Helpfulness by Score
HelpfulnessNumerator
Average Helpfulness
Score


Distribution of Review Scores
1e6
Count
Score


Number of Reviews Over Time
Number of Reviews
Year


Top 25 Most Reviewed Products
Number of Reviews
Product ID
25 Least Reviewed Products
Product ID