Amazon Product Review Helpfulness Prediction Report

Executive Summary

This report details my analysis of Amazon product reviews to predict likelihood of review being helpful. My champion model was the XGBoost that achieved 76.4% ROC-AUC and 83.3% PR-AUC in identifying helpful reviews, with an optimal decision threshold at 0.4. My analysis revealed that star rating, review text features, and verified purchase status are key predictors of review helpfulness.

1. Data Exploration and Analysis

Dataset Overview

- **Total reviews**: 1.2 million (2003-2005)
- **Product category**: Books (370,978 unique products)
- **Data quality**: No missing values in target or features

Key Findings

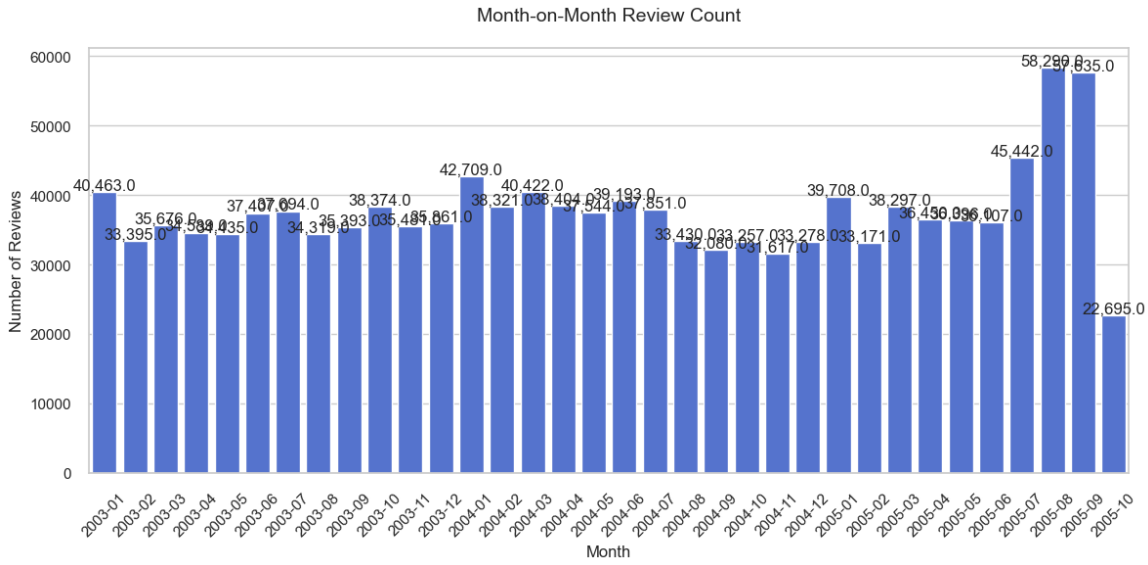
1. **Helpfulness distribution**:
    a. Helpful reviews (≥0.7 cut off ratio): 63.9%
    b. Unhelpful reviews: 36.1%
2. **Duplicate handling**:
    a. Removed reviews appearing >3 times (10,403 duplicates)
    b. Kept first review when ≤3 duplicates
3. **Vote analysis**:
    a. 27% of reviews had only 1-3 votes
    b. Filtered to only include reviews with ≥3 votes for reliability
4. **Star rating impact**:
    a. Strong positive correlation with helpfulness
    b. Most ratings clustered at 3+ stars
5. **Verified purchases**:

a. 10% of reviews from verified purchasers

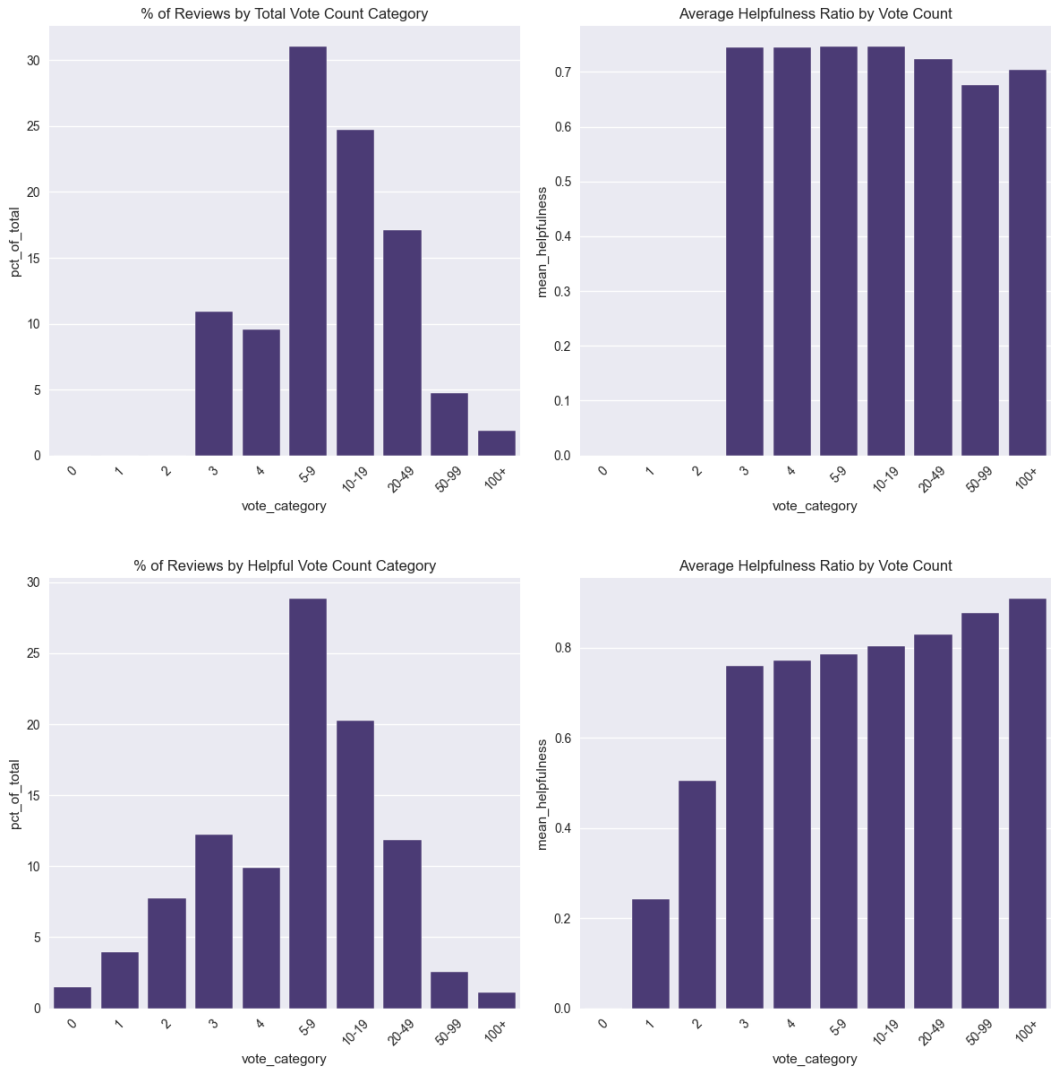
b. Positive correlation with helpfulness

6. Some interesting findings:

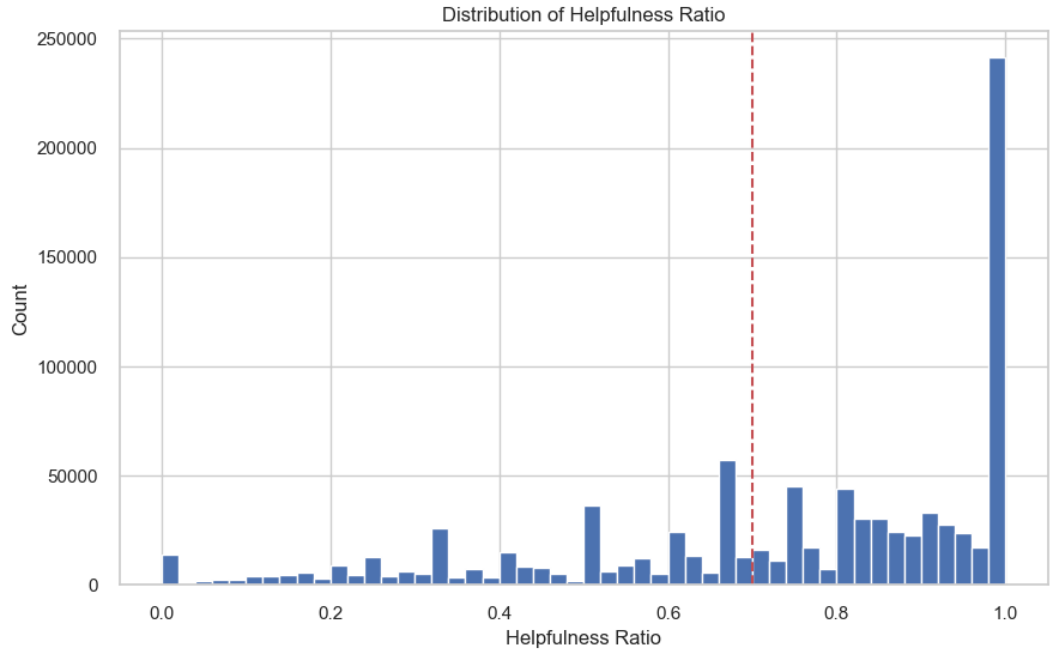
a. Month on month review count looks consistent



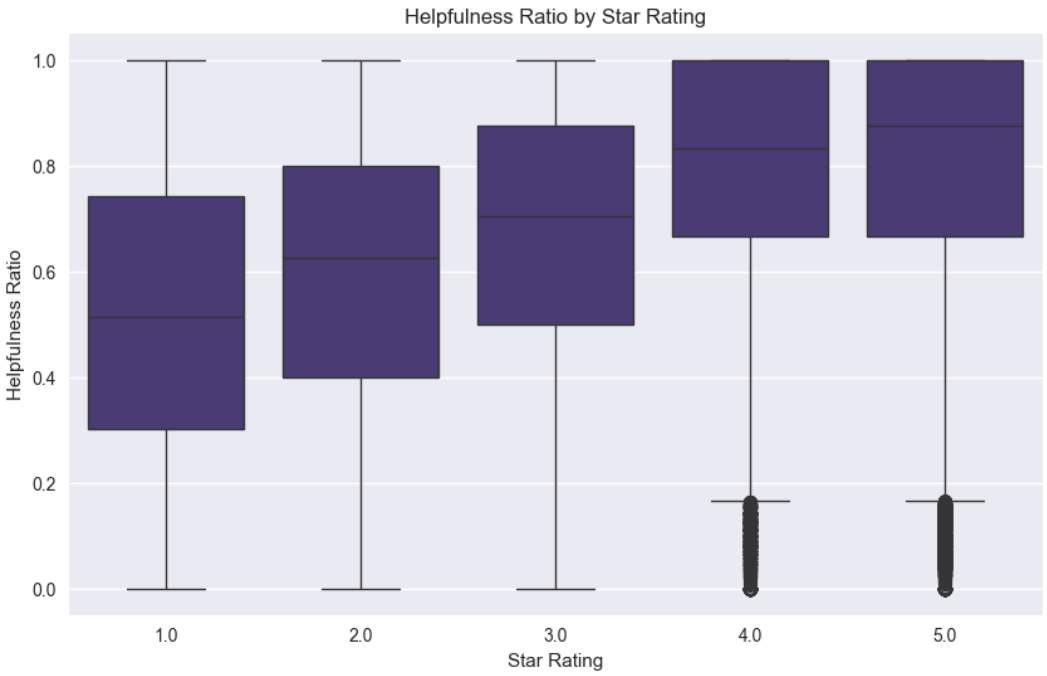
b. 27% of the reviews have just 1-3 total votes. And the number of helpful votes seems to be high for 1-3 votes. Which means although the the ratio might seem high, the magnitude is low. Hence, we could either consider weighted ratio as target or drop the 1-3 votes data.



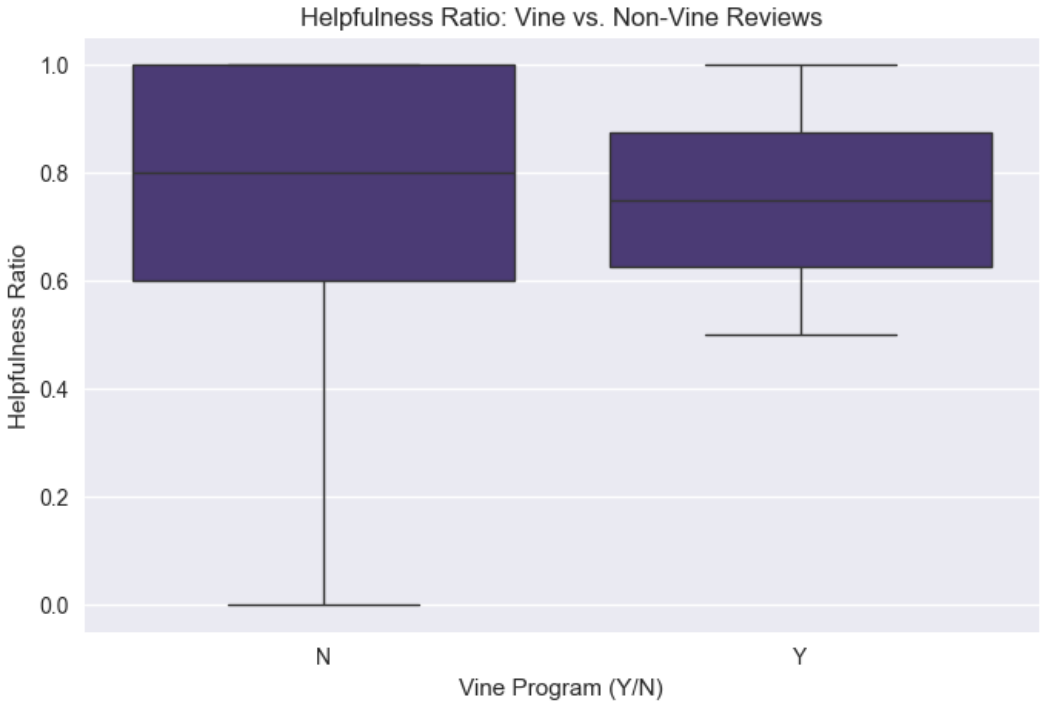
c. Chose 0.7 as the classification cut off for modeling target

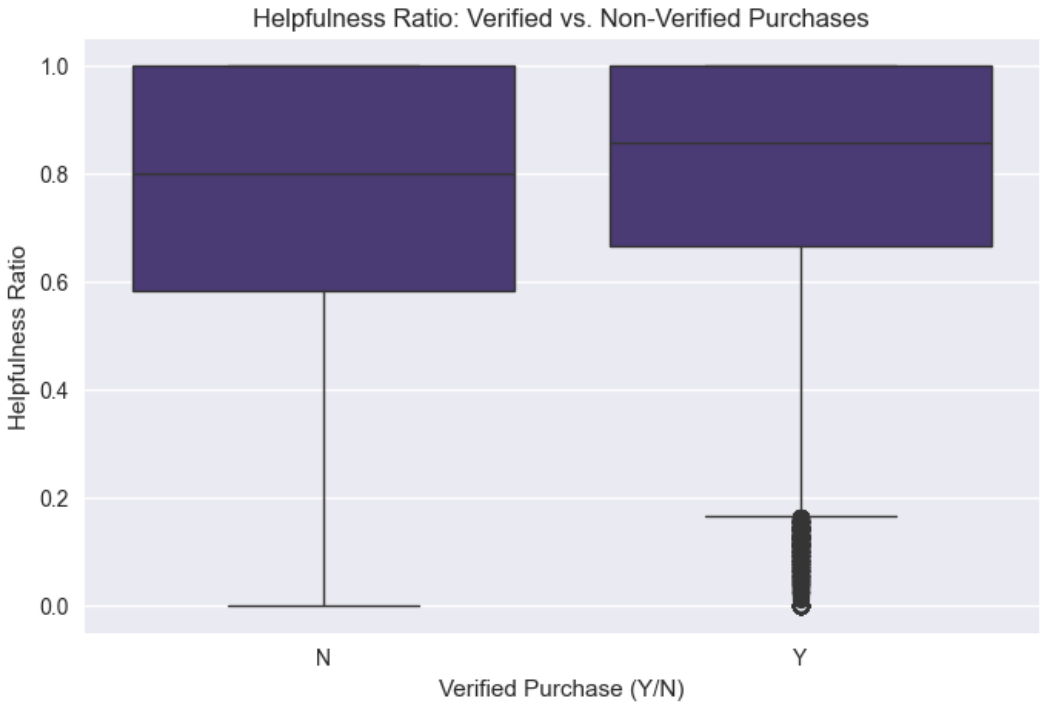


d. Star rating has positive correlation



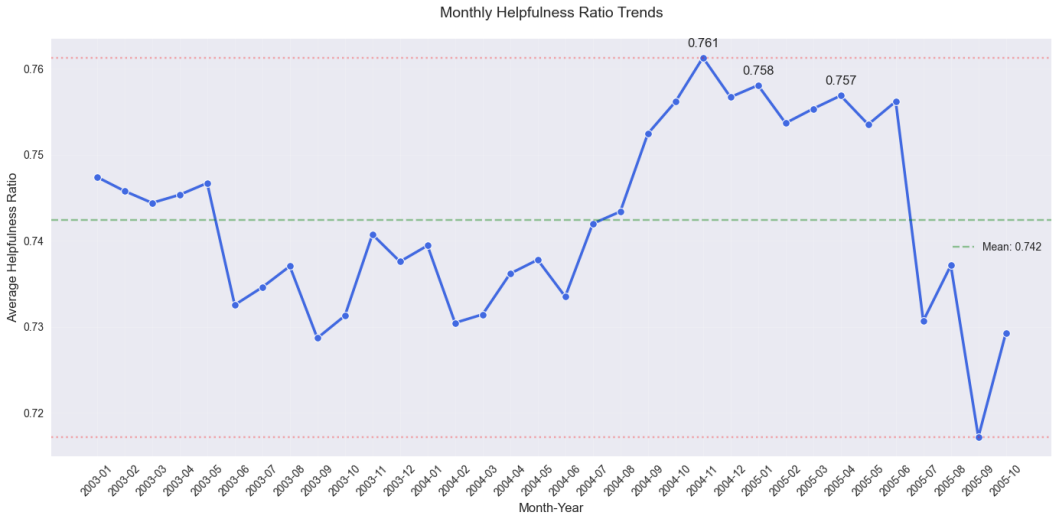
e. Vine program was a loss but verified customers seem to be helful





f. Features on raw length didn't contribute much for helpfulness ratio. TF-IDF generated feature signal feature had the highest correlation

g. Very low temporal trends



2. Data Preprocessing

Cleaning Steps

1. Removed HTML tags and special characters from review text
2. Handled duplicate reviews:
    a. Removed exact duplicates (>3 occurrences)
    b. Kept earliest review for ≤3 duplicates
3. Filtered time period (2003-2005)
4. Excluded reviews with <3 total votes
5. Created binary target (helpful ≥0.7 ratio)

Text Processing

- Removed stopwords and punctuation
- Standardized capitalization
- Handled HTML entities (br, quot, etc.)

3. Feature Engineering

Created Features

1. **Basic metrics**:

    a. Title/headline/body length
    b. Word counts
    c. Exclamation/question counts
    d. ALLCAPS count
2. **Sentiment analysis**:
    a. Polarity (positive correlation)
    b. Subjectivity (no significant correlation)
3. **Text analysis**:
    a. Helpful/unhelpful keyword flags
    b. Keyword ratio
    c. Helpfulness signal (TF-IDF weighted)
4. **Enhanced features**:
    a. Combined helpfulness signals with text metrics

4. Feature Selection

a. Used Cliques based Pearson feature selection for numerical columns
b. Used Bivariate analysis for categorical feature selection

5. Model Building and Evaluation

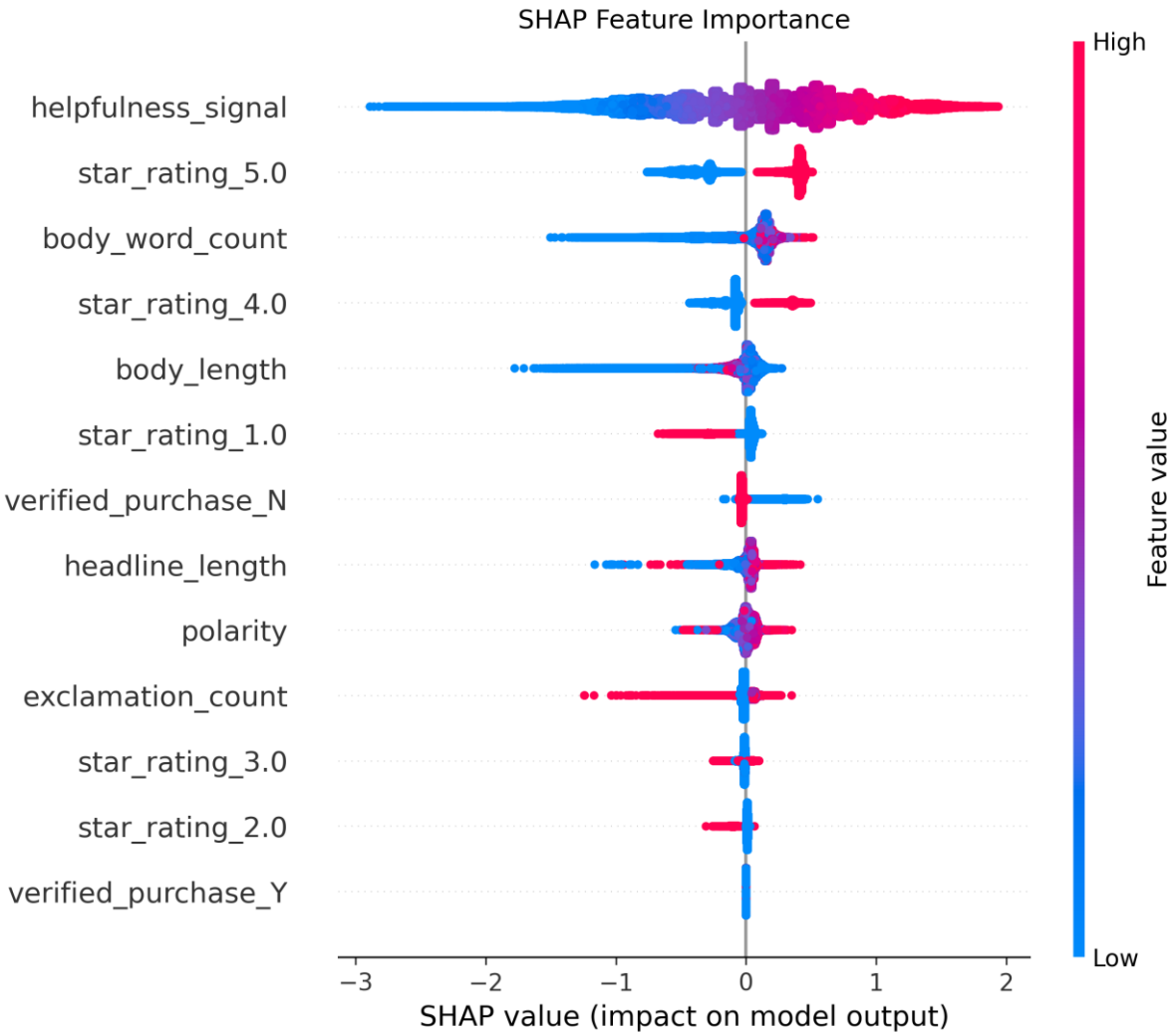
Models Tested

1. **Gradient Boosting (GBM)**:
    a. Ensemble of decision trees with sequential error correction
    b. Robust to outliers and mixed data types
2. **XGBoost**:
    a. Optimized gradient boosting with regularization
    b. Handles missing values and prevents overfitting
3. **LightGBM**:
    a. Leaf-wise tree growth for efficiency
    b. Excellent for large datasets

Final Model Selection

- **XGBoost** performed best with:
    - 5-fold cross-validation
    - 20% test set
    - Hyperparameter tuning

SHAP Analysis



1. **helpfulness_signal** (Top feature):
    a. Strongest predictor of helpfulness
    b. Higher values (red dots on right) increase helpfulness prediction
    c. Lower values (blue dots on left) decrease helpfulness prediction
2. **star_rating_5.0**:
    a. 5-star ratings strongly predict helpful reviews (red clustered on right)

    b. This makes sense as users often find positive reviews helpful
3. **body_word_count**:
    a. Moderate length reviews tend to be most helpful
    b. Very short (blue) or very long reviews may be less helpful
4. **verified_purchase_N**:
    a. Non-verified purchases (red) slightly decrease helpfulness
    b. Verified purchases (bottom feature) slightly increase helpfulness

5-Fold CV Evaluation Metrics

Metric	Value
CV Mean AUC	0.765 ± 0.001
Test AUC	0.764
Optimal Threshold	0.397
Precision (Helpful)	0.72
Recall (Helpful)	0.92
F1 (Helpful)	0.81

6. Evaluation

1. Core Metrics Evaluation

The model demonstrates solid performance across key evaluation metrics:

- **ROC-AUC**: 0.7644 (moderate discrimination ability)
- **PR-AUC**: 0.8328 (strong performance for imbalanced classification)
- **Accuracy**: 0.72 (72% overall correct predictions)

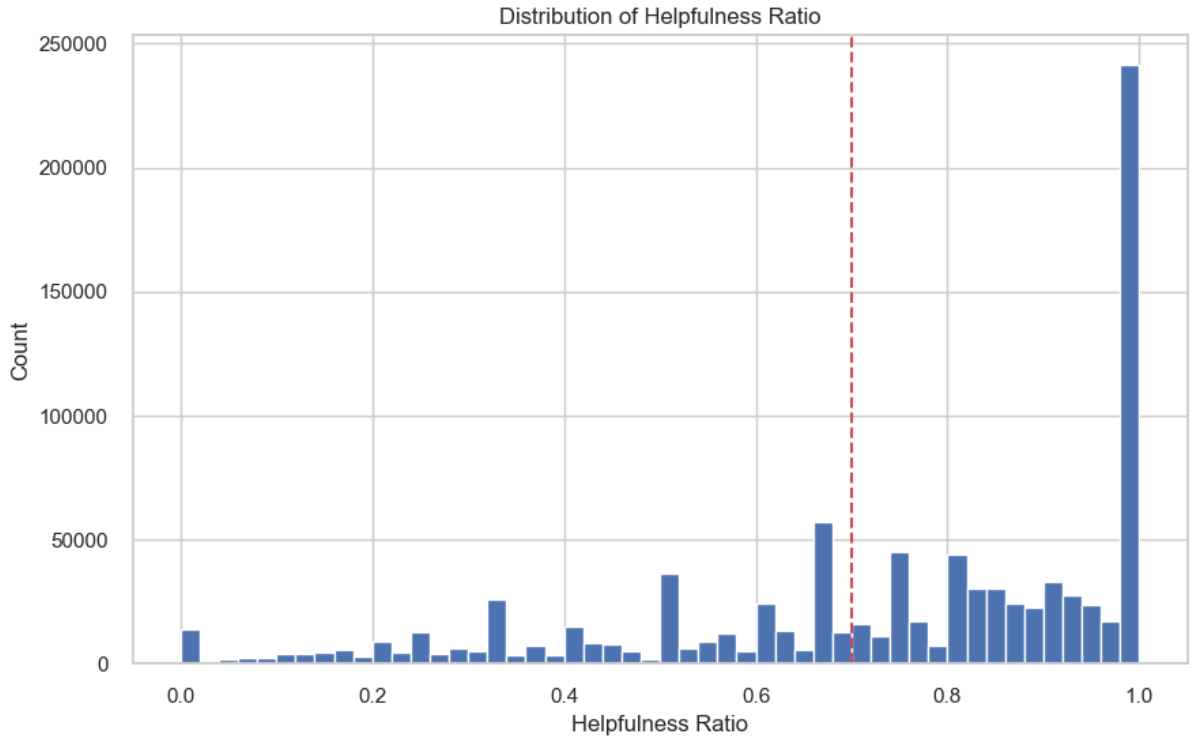
Classification Report:

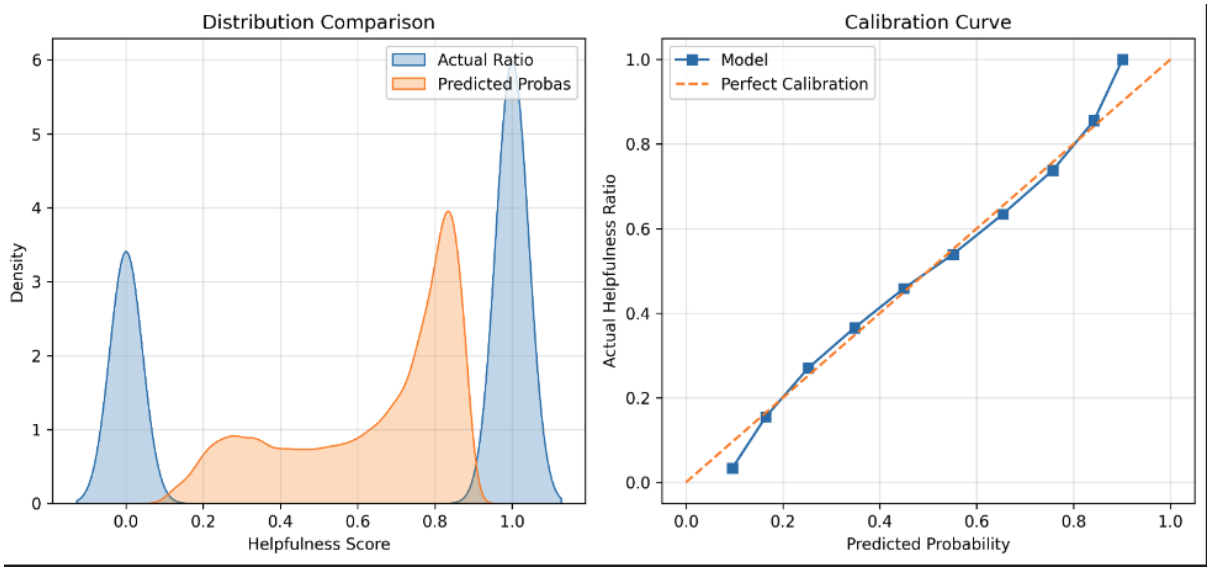
Class	Precision	Recall	F1-Score	Support
0	0.71	0.40	0.51	68,022
1	0.73	0.91	0.81	119,438

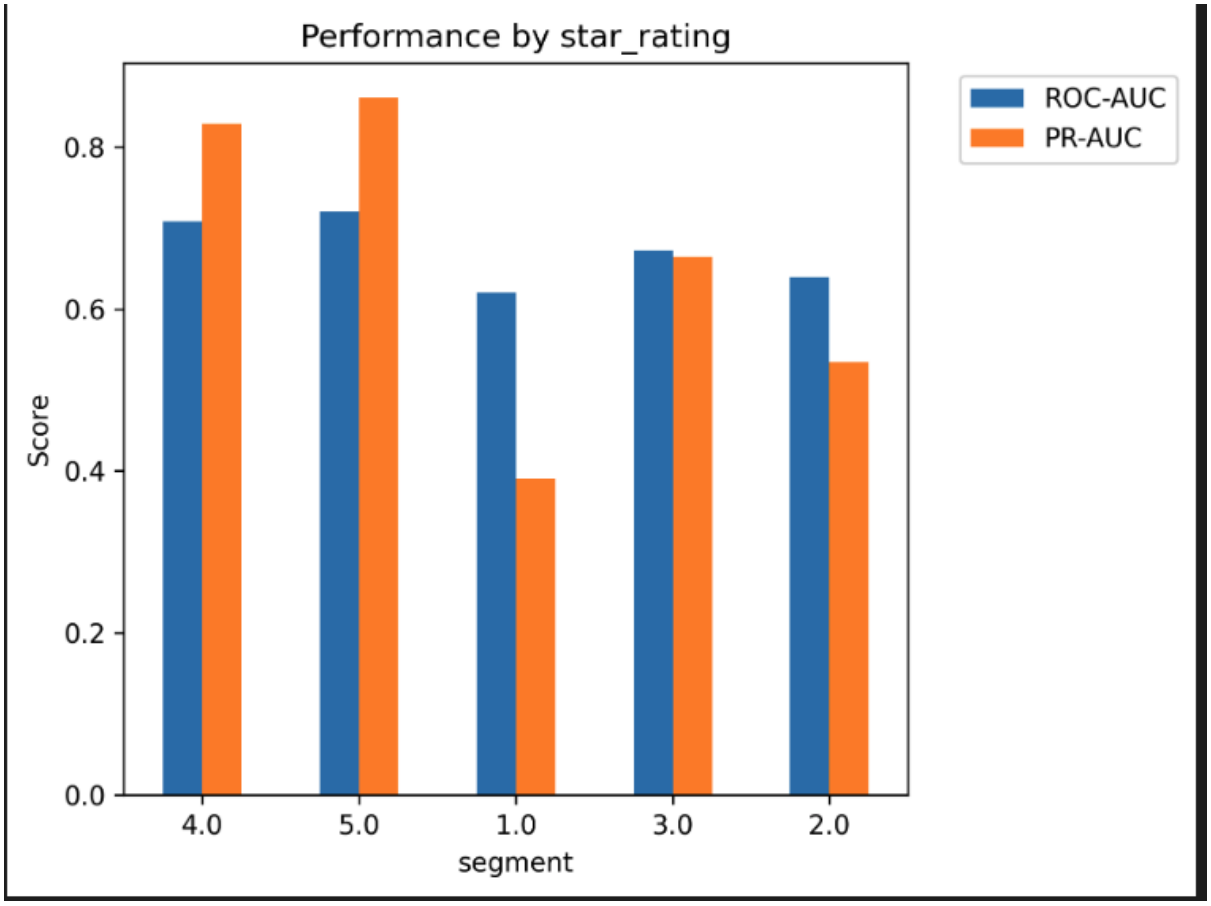
Key observations:

- The model shows better performance for class 1 (higher recall of 0.91)
- Class 0 has lower recall (0.40), indicating room for improvement in identifying negative cases
- Good precision for both classes (0.71 and 0.73 respectively)

2. Drift Analysis







The data distibution in fig 1 and 2 shows the distribution to be bipolar that resonates across the predicted probability scores too.

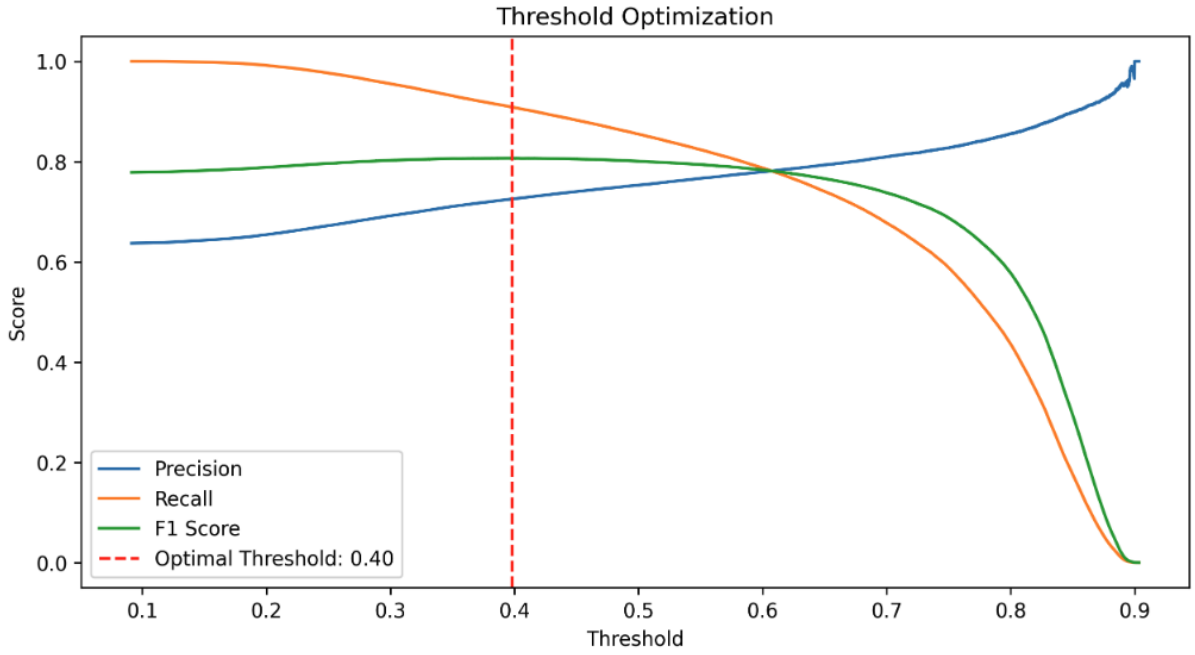
Performance by star rating segments shows variation:

- Highest performing segments: 4.0 and 5.0 star ratings

- Lowest performing segments: 1.0 and 2.0 star ratings
- Moderate performance for 3.0 star rating

This suggests potential data distribution differences across rating segments that may require investigation.

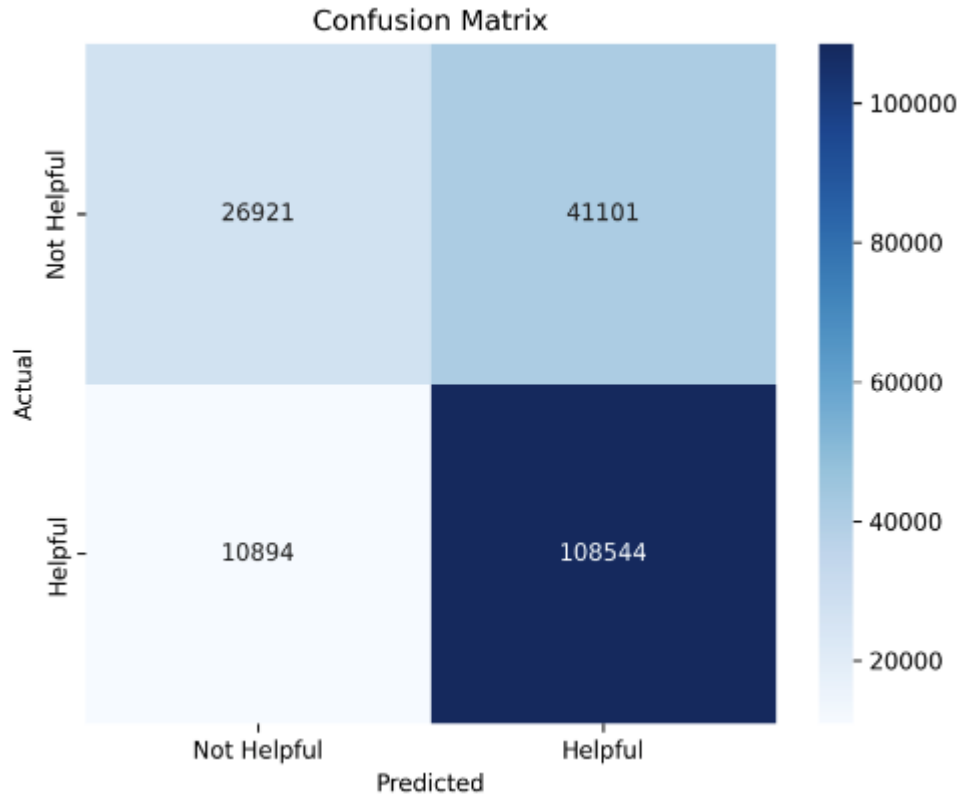
3. Threshold Analysis



- **Optimal Threshold**: 0.3980 (automatically determined for balanced performance)
- The ROC curve shows reasonable separation (AUC = 0.76)
- Precision-Recall curve shows strong performance (AP = 0.83)

Recommendation: Consider threshold adjustment if business requirements prioritize either precision or recall more heavily.

4. Segment Analysis



The confusion matrix reveals:

- True Positives: 108544 (correctly predicted helpful reviews)
- False Positives: 41104 (not helpful reviews predicted as helpful)
- False Negatives: 10894 (helpful reviews predicted as not helpful)

Key insights:

- The model tends to err on the side of predicting reviews as helpful (higher false positives than false negatives)
- This may be appropriate if the business prioritizes capturing helpful content over filtering unhelpful content

7. Potential improvements and Discussions:

- Investigate feature engineering for better discrimination of not helpful reviews

- Consider class weighting or downsampling techniques to address the recall imbalance
- Explore segment-specific models for star ratings with significantly different performance characteristics
- Explore more sophisticated nltk libraries to capture symantics and sentiment
- Threshold optimisation of target variable and logic for creating target (Weighted)
- Improved feature selection
- Identify patterns such as
    - High votes + high ratio: Consensus helpful (gold standard)
    - Low votes + high ratio: Potentially hidden gems
    - High votes + low ratio: Controversial but attention-grabbing