Predicting Amazon Movie Review Ratings Using Data Science Principles

by: Wasay Rizwan

Introduction

The task of predicting star ratings from Amazon movie reviews presents an interesting challenge in NLP (natural language processing) and machine learning. Our goal was to predict user ratings (1-5 stars) based on review text and associated metadata. This report details the methodology and approach used to tackle this prediction task.

Data Overview

Initial data exploration revealed some interesting patterns:

- Review scores are heavily skewed toward 5 stars
- There is significant variation in review length and helpfulness votes
- Many users have multiple reviews in the dataset

Feature Engineering

The core of my approach focused on comprehensive feature engineering across multiple dimensions:

1. **Text-Based Features**:
    - Review and summary length
    - Word count and average word length
    - Punctuation analysis (exclamation marks, question marks)
    - Capitalization patterns
    - Enhanced sentiment analysis using custom word lists
2. **User Behavior Features**:
    - User review count
    - Average user helpfulness ratio
    - User helpfulness standard deviation
    - User rating consistency metrics
3. **Product Features**:
    - Product review count
    - Product popularity metrics
4. **Temporal Features**:
    - Review year
    - Review month

    - Day of week patterns

Enhanced Sentiment Analysis

A particularly effective feature was my enhanced sentiment analysis system that incorporated:

- Expanded positive/negative word lists
- Negation handling
- Contextual scoring
- Sentiment pattern recognition
- Combined text and summary sentiment

The sentiment analysis was implemented with curated word lists and negation handling. I noticed simple positive/negative word counting wasn't enough. The way people express sentiments in reviews is complex - multiple positive words together usually indicate stronger positive sentiment, negations can negate one word or a whole phrase, etc.

Weighted Scoring in my enhanced sentiment analysis explained:

1. **Base Scoring**
    - Positive word: +1
    - Negative word: -1
    - Negated positive word: -1
    - Negated negative word: +1
2. **Multiplier Effects**
    - Clustered sentiments get 1.5x multiplier
    - Example: "amazing fantastic movie"
        - "amazing": +1
        - "fantastic": +1.5 (boosted due to previous positive)
        - Total score: +2.5 instead of just +2

Model Selection and Optimization

I chose the k-Nearest Neighbors (kNN) algorithm for several reasons:

- It is the algorithm I am most familiar with and that we have reviewed and used extensively in-class (lecture, assignment, lab)
- Ability to capture complex rating patterns
- No assumptions about feature relationships
- Effective with our engineered feature space
- Interpretable predictions based on similar reviews

Key model optimizations included:

- Sample size optimization for computational efficiency
- Ball-tree algorithm for faster neighbor searches
- Distance-weighted voting
- Parallel processing with n_jobs=-1

Key Patterns Discovered

What I Noticed:

- Reviews with multiple exclamation points (!!!) strongly correlated with 5-star ratings
- ALL CAPS words were strong indicators of extreme ratings (1 or 5 stars)
- Short summaries with strong words ("Amazing!" or "Terrible!") were highly predictive

This pattern directly informed my sentiment scoring system, giving extra weight to these intensity indicators.

- Some users consistently rate higher or lower than average
- Users with many reviews tend to give more moderate ratings
- Highly active reviewers often use more nuanced ratings (2-4 stars)

This led me to create user-specific features that significantly improved prediction accuracy.

- Very short reviews (<50 characters) strongly correlate with extreme ratings
- Longest reviews often correspond to 3-star ratings
- 4-star reviews tend to have more balanced positive/negative points

I used these patterns to create weighted features where review length influenced the final prediction differently based on length ranges.

Conclusion

The key to this project was identifying and leveraging actual patterns in how users write and rate reviews. Rather than just throwing features at the problem, I focused on understanding and implementing features that captured real behavioral patterns. Valuable patterns included user rating consistency, sentiment intensity markers, review structure indicators, and temporal effects. These patterns formed the foundation of my feature engineering decisions. My model aims to capture many of the subtle patterns that influence how people rate movies.

The most valuable lesson was learning how to combine different types of information - text analysis, user behavior, and timing - to better understand and predict human behavior.