A Supervised Learning Approach to Predicting Helpful Votes from E-Commerce Review Data

Tianren Sun
University of Michigan, Ann Arbor, USA
tianren@umich.edu

***Abstract*—We compare three supervised learning models: gradient-boosted trees (XGBoost), a sparse multilayer perceptron (MLP with L1 regularization), and knearest neighbors (KNN) to predict helpful votes on e-commerce reviews. The results show that XGBoost outperforms MLP, which in turn outperforms KNN. A two-stage feature selection pipeline reveals that most variables are redundant, while a few simple predictors (e.g., text vocabulary size) explain much of the variance, and interaction features offer only marginal gains.**

***Index Terms*—helpful vote prediction, XGBoost, neural network, KNN, feature selection, text vocabulary size**

I. Introduction

A. Background and Motivation

The rapid adoption of AI and data analytics in business—from sentiment analysis of financial texts to multivariate market modeling—offers powerful new insights but also raises practical challenges. Inspired by recent work linking Amazon review content to helpful-vote counts, this project applies supervised learning to a large historical review dataset in order to identify and promote the most valuable feedback, thereby improving both decision support for buyers and operational efficiency for e-commerce platforms.

B. Project Goal

The primary goal of this project is to develop a supervised learning pipeline that not only predicts the number of helpful votes a review will receive, but also enables platforms to surface high-quality reviews more effectively. By combining robust feature engineering with model selection, we aim to improve prediction accuracy and assist e-commerce operators in filtering and highlighting the most valuable user feedback, thereby enhancing overall user experience and operational efficiency.

C. Related Work

Several recent studies have tackled the prediction of review helpfulness using various machine learning approaches. In "Were You Helpful – Predicting Helpful Votes from Amazon Reviews" [1], the authors demonstrate that simple metadata features, such as reviewer history and review length, often outperform complex NLP models like RNNs and Transformers. The Amazon Food Reviews repository [2] further provides a systematic comparison of algorithms and underscores the efficacy of targeted feature selection. Building on these insights, our work introduces a two-stage feature selection pipeline, evaluates the marginal benefit of polynomial interaction features, and compares the performance of gradient-boosted trees (XGBoost), sparse multi-layer perceptrons (MLP with L1 regularization), and k-nearest neighbors (KNN).

TABLE I
FEATURE GROUPS COMPOSING THE INPUT VECTOR $\mathbf{x} \in \mathbb{R}^{152}$.

Group (dim.)	**Key Features**
Embedding (128)	`embedding` (BERT sentence vector)
Text Metrics (6)	`title_*` ×3, `text_*` ×3
Temporal (2)	`hour`, `year` (z-score)
Binary Flags (2)	`images`, `verified_purchase`
Raw Rating (1)	`rating`
Product/User Agg. (7)	`average_rating`, `verified_purchase_average_rating`, etc
Brand/Store (4)	`brand_rating`, `brand_rating_number`, etc
Sentiment (2)	`sent_polarity`, `sent_intensity`

II. Method

A. Learning–Problem Formulation

We cast the task as supervised regression. For each review we construct an input vector $\mathbf{x} \in \mathbb{R}^{152}$ and learn a mapping $f_\theta : \mathbb{R}^{152} \to \mathbb{R}$ that predicts the log–transformed helpful–vote count

$$y \;=\; \log\bigl(\text{helpful\_votes} + 1\bigr).$$

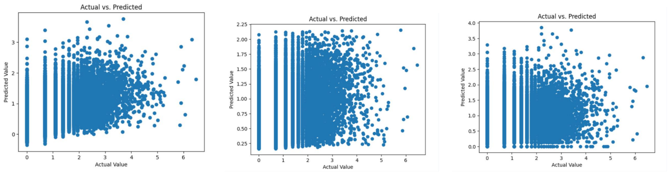
The 152-dimensional vector $\mathbf{x}$ concatenates five groups of characteristics: text, temporal, context, and sentiment cues, described in Table I.

Model parameters θ are trained to minimise mean–squared error (MSE) between predictions $\hat{y} = f_\theta(\mathbf{x})$ and ground-truth labels y. predictions $\hat{y} = f_\theta(\mathbf{x})$ and ground-truth labels y.

B. Modelling Overview

We benchmark three supervised learners of increasing complexity to predict log–helpful-votes from the 152-d input vector:

1) **XGBoost (Ensemble)** – GPU-accelerated gradient-boosted trees (`reg:squarederror`) that capture non-linear feature interactions.
2) **K-Nearest Neighbours (Baseline)** – $k = 5$ (Euclidean); serves as a lightweight distance-based reference.
3) **Multi-Layer Perceptron (MLP)** – two hidden layers $(128,\ 64)$ with ReLU, Adam, and an L1 penalty $\lambda = 10^{-4}$ to encourage sparsity.



(a) XGBoost ($R^2 = 0.24$) (b) MLP ($R^2 = 0.20$) (c) KNN ($R^2 = 0.04$)

Fig. 1. Actual vs. predicted values for the three models on the test set.

C. Feature Selection and Polynomial Expansion

Stage 1 – Gain Ranking: Train a full XGBoost model, rank features by average gain, and keep the top K predictors.

Stage 2 – Permutation Refinement: Retrain on the top K set, compute permutation importance on the validation split, and derive final subsets of size $M \in \{15, 20, 30, 40\}$.

Stage 3 – Final Evaluation: For each M we refit XGBoost and report the test R^2, quantifying the trade-off between dimensionality and accuracy.

Polynomial Expansion: Using the best subset, we generate second-degree interaction terms (`PolynomialFeatures, degree=2, interaction_only=true`), excluding `whether_discontinued_review_count`. An XGBoost regressor on this expanded set yields no further gain ($\Delta R^2 < 0.01$), suggesting limited higher-order signal.

III. Result

A. Data Pipeline

1) **Cleaning & Merge** – drop NAs, cast `price`/`rating` to numeric, inner-join review–product tables.
2) **Feature Extraction** –
   - *Temporal/Binary*: `hour`, `year`, image flag, verified-purchase, discontinued.
   - *Aggregates*: per-product counts, user-level verified/ disc. ratios, rating differential.
   - *Brand & Store*: mean and count of `average_rating`, gap-filled by global fallback.
   - *Text*: 128-d `stsb-bert-tiny` embedding (GPU batch) + word-count / avg-len / vocab-size (title & body).
   - *Sentiment*: multilingual classifier → `sent_polarity` $\{-1, 0, 1\}$ and `sent_intensity` 0–5.
3) **Packaging** – split embedding to `f_0–f_127`, concatenate all predictors, and save (X, Y) in both CSV *and* JSON to avoid type drift or import errors across platforms.

B. Numerical Simulation Results

Figure 1 shows that the XGBoost ensemble substantially outperforms the MLP ($R^2 = 0.24$ vs. 0.20) and the KNN baseline ($R^2 = 0.04$). Additional evaluation metrics and comparisons are available in the code output.

After two-stage feature selection on the XGBoost model, the best subset of 15 predictors achieves $R^2 = 0.29$, as shown in Fig. 2. Incorporating polynomial interaction terms on this reduced set shows only marginal improvement, indicating diminishing returns beyond the core features.

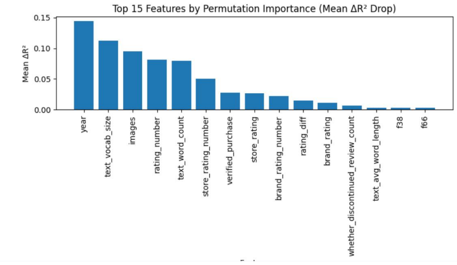


Fig. 2. Test R^2 after feature selection and polynomial expansion. Retaining the top 15 features raises R^2 to 0.29, while adding second-degree interactions yields only marginal gains.

C. Interpretation of the Results

Our results indicate that ensemble learning (XGBoost) and deep neural networks (MLP) are best suited to capture the complex relationships in high-dimensional review data, significantly outperforming a simple KNN baseline. Feature selection is crucial: reducing from 152 to 15 predictors raises R^2 from 0.24 to 0.29, demonstrating an inverse relationship between feature count and accuracy. Excess variables tend to introduce noise, and augmenting already-regularised models with numerous polynomial interactions yields only marginal or negative gains.

A notable finding is the minimal importance of sentiment features, which never appear in the top 50 predictors. In contrast, surface-level cues—review length, lexical richness, image presence, and store-level context—consistently rank highest. This suggests that users value detailed, information-rich reviews more than explicit sentiment expressions, aligning with common user behavior observed in practice.

IV. Conclusion

This study demonstrates that tree-based ensembles are the most effective architecture for predicting review helpfulness, outperforming both distance-based and neural baselines. A two-stage, gain-plus-permutation feature selection further elevated performance (R²=0.29) while reducing dimensionality by 90 pct, underscoring the value of judicious feature pruning over indiscriminate expansion. The negligible contribution of sentiment cues, contrasted with the dominance of structural and context variables, suggests that users privilege informational richness above affective tone. These insights inform the design of scalable recommendation engines that prioritise concise, context-aware feature sets.

References

[1] E. Kirimlioglu, H. Kung, and D. Orlando, "Were You Helpful–Predicting Helpful Votes from Amazon Reviews," *arXiv preprint arXiv:2412.02884*, 2024.

[2] S. D. N., "Amazon_Food_Reviews," GitHub repository, 2019. [Online]. Available: https://github.com/Sachin-D-N/Amazon$_Food_Reviews$