Rating Prediction for Amazon Products Based on Customer Reviews using Machine Learning

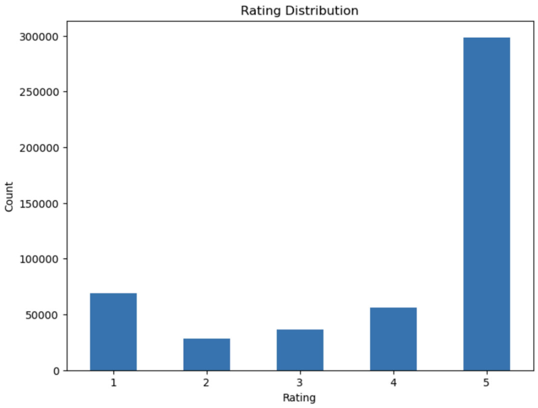
1 INTRODUCTION

In the age of digital commerce, customer reviews have become a cornerstone for evaluating products and services. In the Health and Personal Care sector, reviews are particularly significant as they influence purchasing decisions for products that directly impact well-being and the success of business. Given the personal and sensitive nature of these products, understanding customer feedback is essential not only for helping companies better understand customer satisfaction and preferences to improve product quality but also for fostering consumer trust. Therefore in this project paper, we intend to predict a product's star rating based on the textual content of its user review. The ability to predict ratings from review text has significant business value. For instance, it can enhance product search engines by prioritizing items with high predicted satisfaction or refine recommendation systems by gaining deeper insights into customer sentiment. By analyzing customer reviews and ratings, we hope our findings can empower businesses to better meet consumer needs and improve their products.

1.1 Dataset Description

To construct a robust rating prediction model, we will need a sufficiently large and diverse model to capture the complexities of customer feedback. The dataset we used in this study can be found on the website Recommender Systems and Personalization Datasets [1] provided by professor Julian McAuley from UCSD. It originally comprised 494,121 records across 10 variables, including attributes such as ratings, review content, helpful vote counts, and verified purchase indicators. The data has no missing value but 5,130 duplicate records, we have removed those. The dataset reveals an average rating of 3.9965, suggesting that customers generally leave positive feedback. However, the rating distribution (Figure 1) indicates a significant skew toward high rating of 5, which is disproportionately represented compared to other ratings:

Figure 1: Review Rating Distribution



This trend suggests a potential bias in the dataset, perhaps customers are more inclined to review products they are satisfied with. This makes us curious about the underlying patterns in customer feedback, especially why so many reviews result in a high rating of 5, so we examined the review texts using a Word Cloud (Figure 2): the bigger the word, the more frequently the word is used in the reviews. We can see larger words such as *use*, *product*, *love*, *easy*, *make*, *good*, and *time* indicate their prevalence in customer feedback. This observation aligns with the skewed distribution of ratings, as many customers who leave high ratings (5 stars) also use consistently positive terms in their reviews. This rating distribution and WordClouds reflect the challenges in building balanced predictive models, as it may lead to over-prediction of higher ratings and underperformance for lower ratings.

Figure 2: WordCloud for Text Reviews

2 PREDICTIVE TASK

The task we are addressing is predicting the star rating (rating) given by users based on their review text (text). This involves leveraging natural language processing (NLP) techniques and embedding methods to capture the semantic meaning of reviews, as well as utilizing other relevant features from the dataset to build a predictive model. Our primary approach involves using Word2Vec embeddings to process the text and title columns. These embeddings are numerical representations that encode the semantic relationships between words and phrases, allowing us to extract meaningful patterns from the review content that are likely to correlate with the given rating.

2.1 Feature Engineering and Data Processing

To prepare the data for modeling, several preprocessing steps will be applied. The text and title columns will undergo cleaning to remove noise, such as punctuation, special characters, and stopwords, and converted to lowercase for consistency.

- **TF-IDF:** As a baseline, we used TF-IDF to convert the review text into numerical representations to capture term importance across the dataset
- **Word2Vec**: We trained Word2Vec embeddings on the dataset to capture semantic relationships between words. The embeddings were aggregated using mean pooling to represent each review as a fixed-length vector.
- **Doc2Vec**: As an exploratory approach, We used Doc2Vec to directly generate dense embeddings for entire reviews, capturing contextual information.

While textual features were the primary focus, other features in the dataset provide valuable information as well. The helpful_vote column, which indicates the number of users who found the review helpful, can be normalized to prevent scale issues and capture its relative importance. The verified_purchase column can be encoded as a binary feature, as it may provide more reliable indicators of genuine reviews. The asin and parent_asin columns, representing product IDs, and user_id will be mapped to numerical indices for compatibility with machine learning models. These features can enable collaborative filtering techniques in advanced modeling stages.

2.2 Model Evaluation

To evaluate the performance of our model, we will use metrics tailored for ordinal data like ratings, specifically Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - x_i\right|$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2}$$

These metrics effectively quantify the alignment between predicted and actual ratings, with lower values indicating better performance. The dataset was split into training and test sets using an 80/20 ratio to ensure a fair evaluation of generalization. Additionally, cross-validation could be applied in future iterations to validate performance on different subsets of the data. To establish a baseline for comparison, simple models like predicting the mean rating for all users or applying linear regression with bag-of-words representations of the text and title will be implemented. These models provide a reference point for assessing the performance gains achieved by more sophisticated approaches.

2.3 Relevant Baselines

To evaluate our approach, we implemented several baseline models for comparison. The simplest baseline predicts the mean rating for all users, providing a foundational benchmark. Building on this, we will also incorporate the three regression methods as described in Section 2.1. By concatenating the embeddings and structured features, the logistic regression can capture both textual semantics and non-textual signals to predict the rating. To validate the model's predictions, we will conduct a detailed error analysis, examining instances where the predicted ratings deviate significantly from the actual ratings. This analysis will help identify potential biases, such as an over-reliance on specific features or misinterpretations of unusual review content.

3 MODEL

We chose logistic regression as the foundation for predicting user ratings because it offers a practical balance between efficiency, interpretability, and predictive capability. By framing the rating prediction as a classification problem, we can use features extracted from the dataset and the simplicity of logistic regression to make predictions. This model is particularly appealing because it provides transparency in understanding the relationship between input features and predicted outcomes, making it a strong baseline for comparison with more complex models.

3.1 Model Optimization

To optimize the logistic regression model, we employed several strategies. For TF-IDF + Logistic Regression, we perform hyperparameter tuning to identify the best regularization strength (λ) using grid search. This involves testing a range of regularization parameters for both L1 (lasso) and L2 (ridge) penalties:

- L1 regularization helps in feature selection by driving less important coefficients to zero
- L2 regularization prevents overfitting by penalizing large coefficients.
- L1 and L2, or a combination (elastic net), depends on cross-validation results, ensuring the model balances simplicity and generalization.

For Word2Vec and Doc2Vec, we explored different parameters:

- vector_size (50, 100, 200)
- window (3, 5, 10)
- min_count (1, 2, 5)

Both Word2Vec and Doc2Vec required significantly more computational resources than TF-IDF. The training embeddings for word2vec was time-consuming, and the quality was sensitive to the dataset size while the inferencing document embeddings for test data was much slower than TF-IDF transformation. There are some challenges encountered, including high-dimensional embeddings and overfitting. Regularization proved effective in mitigating overfitting, while careful feature selection reduced computational complexity. Scalability was addressed by optimizing Word2Vec embedding generation, limiting embedding size and training iterations to balance resource usage and model quality.

3.2 Model Strengths and Weakness

The TF-IDF + Logistic Regression model demonstrated strong performance, with the lowest RMSE and MAE among the three approaches. Its simplicity, efficiency, and interpretability make it a reliable baseline, but it cannot capture semantic relationships or word order. In contrast, the Word2Vec + Logistic Regression model leverages word embeddings to capture semantic relationships between words, providing a richer representation of text. However, it underperformed due to the limitations of mean pooling, which discards word order and context, and the dataset size was insufficient for high-quality embeddings. Finally, the Doc2Vec + Linear Regression model provides context-aware representations, but it struggles with prediction accuracy that is likely due to the simplicity of linear regression and has higher computational cost compared to TF-IDF.

3.3 Alternative Models Considered

In comparison to logistic regression, we considered alternative models like neural networks and gradient-boosted trees. While these models often deliver higher accuracy, they require more extensive tuning and computational resources and lack the interpretability of logistic regression. Other simpler models, such as linear regression, were unsuitable due to their inability to handle the ordinal nature of ratings. Logistic regression strikes a balance, providing a strong baseline with room for interpretability and efficient optimization.

4 LITERATURE

The problem of predicting user ratings based on review text has been widely studied, particularly in the context of e-commerce platforms. The dataset used in this study originates from the Amazon Reviews dataset, specifically focusing on the Health and Personal Care category.

Prior work on Amazon review datasets have explored various predictive and analytical tasks. He and McAuley [1] utilized the dataset to investigate personalized recommendation systems using deep learning techniques. Their work developed collaborative filtering approaches augmented with text-based features, demonstrating the importance of integrating review content with user-product interactions. Similarly, Chen et al. [2] focused on sentiment analysis and employed transformer-based models like BERT to predict user sentiments and ratings, achieving state-of-the-art performance on several subsets of the Amazon dataset. Merve Esra Taşcı et al. [3] expanded on this by testing multiple machine learning algorithms, including Passive Aggressive, Random Forest, and AdaBoost, to classify sentiments from Amazon reviews, showing the versatility of combining textual analysis with various classifiers. These studies highlight the effectiveness of combining textual and structured data to enhance predictive accuracy.

4.1 Related Datasets and Comparisons

The Amazon Reviews dataset has also been compared to similar datasets in the domain of e-commerce and online reviews. For example, the Yelp Dataset and IMDB Movie Reviews Dataset share similarities in terms of user-generated content and metadata, but they differ in scale and domain specificity. Studies using the Yelp dataset have often focused on business recommendations and service ratings, while IMDB reviews have been primarily used for sentiment classification in the context of entertainment media. The Amazon dataset, by contrast, encompasses a broader range of products and domains, making it particularly versatile for studying e-commerce behavior. In our project, we specifically focus on how user feedback, as captured in reviews, can inform the improvement of recommendation systems and influence commercial strategies. By analyzing review text and associated features, we aim to uncover patterns in user preferences and satisfaction levels that can directly guide product recommendations, marketing efforts, and inventory management.

4.2 State-of-the-Art Methods

State-of-the-art methods for analyzing such datasets typically involve advanced machine learning and natural language processing models. Transformer architectures, such as BERT and GPT, have been widely adopted for sentiment analysis and rating prediction tasks due to their ability to capture contextual nuances in text. For instance, Dang et al. [4] explored the integration of sentiment analysis with collaborative filtering recommenders, applying advanced NLP models like BERT to address data sparsity and improve recommendation accuracy. Neural networks, including recurrent and convolutional models, have also been used to process textual data and combine it with metadata features for prediction. Additionally, regression-based methods and logistic models have been applied as interpretable baselines, particularly in studies emphasizing feature contributions and the transparency of predictions.

Despite the success of these advanced methods, simpler models still play a critical role. For instance, logistic regression offers interpretability and efficiency, making it a valuable baseline. Moreover, work such as the sentiment analysis study by Taşçı et al. [3] has shown that even traditional machine learning algorithms can achieve competitive performance when coupled with robust feature engineering. Another example is a study by Dang et al. [4], which demonstrated that prompt-based sentiment analysis using NLP models could significantly enhance collaborative filtering methods. These studies collectively underscore the breadth of approaches available for leveraging review data.

5 RESULT

The results demonstrate that the TF-IDF + Logistic Regression model outperformed the others in predicting user ratings. With an RMSE of 0.8773 and an MAE of 0.3535, this model proved to be the most accurate and reliable approach. The success of this model can be attributed to the simplicity and effectiveness of TF-IDF in capturing term importance and its seamless integration with logistic regression.

Table 1: Performance Comparison

Model Type	RMSE	MAE
TF-IDF + Logistic Regression	0.8773	0.3535
Word2Vec + Logistic Regression	1.0897	0.4837
Doc2Vec + Linear Regression	1.393	1.1346

In contrast, the Word2Vec + Logistic Regression model achieved an RMSE of 1.0897 and an MAE of 0.4837, underperforming due to limitations in the embedding aggregation method (mean pooling) and the smaller dataset size, which impacted embedding quality. However, the use of logistic regression as the underlying model allowed for efficient handling of the extracted embeddings. The Doc2Vec + Linear Regression model performed the worst, with an RMSE of 1.393 and an MAE of 1.1346. This approach was intended to capture contextual information by generating dense representations of entire reviews, but its combination with linear regression was likely suboptimal.

5.1 Model Parameters Interpretation

The logistic regression models provided interpretable coefficients that highlighted the contribution of individual features to the prediction. For example, higher coefficients for certain TF-IDF terms or Word2Vec dimensions are likely aligned with positively or negatively skewed ratings. The use of regularization (e.g., L1 or L2 penalties) helped ensure that the model prioritized the most impactful features while reducing the influence of irrelevant or noisy dimensions. These regularization techniques also addressed potential overfitting issues, particularly with the high-dimensional embeddings produced by Word2Vec and Doc2Vec.

For embedding-based approaches, parameters such as vector_size and min_count were instrumental in shaping the quality of Word2Vec and Doc2Vec embeddings. Larger vector sizes captured more nuanced semantic relationships but required more computational resources and risked introducing noise. The min_count parameter controlled the minimum frequency threshold for words included in the vocabulary. Lower values (e.g., 1 or 2) ensured rare but meaningful terms were captured, while higher values excluded noise from very infrequent words.

Table 2: Feature Performance

Model Type	Effectiveness	Remarks
TF-IDF + Logistic Regression	HIGH	Simple & computation efficient
Word2Vec + Logistic Regression	MED	Limit to data size & pooling
Doc2Vec + Linear Regression	LOW	Computation intensive

5.3 Success and Failures

The success of the TF-IDF + Logistic Regression model can be attributed to its balance of simplicity, interpretability, and effectiveness in capturing key textual patterns. the importance of keywords such as "good," "fantastic," and "terrible" in accurately representing user preferences and predicting ratings. TF-IDF excels at extracting the most important and relevant words from documents, making it more effective for this task.

It is computationally efficient and does not rely on extensive training or hyperparameter tuning compared to Word2Vec and Doc2Vec. Conversely, the Word2Vec + Logistic Regression model's underperformance highlights the challenges of embedding-based approaches when using relatively small or skewed datasets. Similarly, the Doc2Vec + Linear Regression model failed due to lacked sufficient data to construct a robust model capable of analyzing semantic relationships effectively and the mismatch between the embedding complexity and the simplicity of linear regression.

REFERENCES

[1] He, R., & McAuley, J. (2016). *VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback*. Retrieved from https://arxiv.org/abs/1510.01784

[2] Chen, T., Zhang, Z., & Liu, J. (2020). *Sentiment Analysis on Customer Feedback Data: Amazon Product Reviews*. IEEE. Retrieved from https://ieeexplore.ieee.org/document/8862258

[3] Taşcı, M. E., Rasheed, J., & Özkul, T. (2024). *Sentiment Analysis on Reviews of Amazon Products Using Different Machine Learning Algorithms*. In *Lecture Notes in Networks and Systems* (pp. 328–338). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-031-62881-8_26

[4] Dang, E., Hu, Z., & Li, T. (2022). *Enhancing Collaborative Filtering Recommender with Prompt-Based Sentiment Analysis*. Retrieved from https://arxiv.org/abs/2207.12883