Predicting Product Ratings for Amazon Electronics With User Reviews

Joyce Shiah, Sneha Shah, Sean Deering, Mengkong Aun
University of California, San Diego, Halıcıoğlu Data Science Institute

Background and Literature

The data utilized in this study were sourced from the website: https://cseweb.ucsd.edu/~jmcauley/datasets.html and contains various datasets assembled for research purposes in Dr. Julian McAuley's lab. Some of the ways in which this dataset was previously used include: Establishing connections between products and their respective reviews, analysis of user behavior patterns, and building recommender systems. For example, the recommender systems were constructed using the significance of features in the underlying structured and textual data herein [1]. Further, user preferences were connected to the contents of reviews, which, generally speaking, can be used to examine and create recommendations tailored to individual users. This framework provides a useful technique to influence reviews for explainable AI which is connected to suggestions for products in this article [2]. User preferences can be analyzed through utilization of visual information and accompanying metadata. Dynamic user profiling can be observed from non-textual data in relation to reviews through the following article [3]. By combining user review data with metadata, one can look into recommendations based on images and substitutes and this showcases a forerunner for systems that are multimodal with data that is visual and textual, enabling better recommendations to be made [4] . Other related datasets that have been utilized in the past include the *Amazon Question and Answer Dataset,* which consists of questions and answers related to the *Amazon Product Reviews Dataset* and the *BeerAdvocate Dataset* (which is similar to the *Amazon Product Reviews Dataset*, except that the reviews are specific to different kinds of beer). The *Amazon Question and Answer Dataset* has been explored for the utilization of question and answering systems and customer reviews and focuses on personality and uncertainty in questions and answers related to products [5]. It has also been used to find solutions to questions connected to products that are intricate and personal [6]. The *BeerAdvocate Dataset* was used to gain an understanding of user behavior and preferences, as evidenced by a research study that looked into comprehending product features and user perspectives [7]. Another related study investigated time-related patterns connected to user competence [8]. State-of-the-art methods that are currently being employed to explore these sorts of data are: Neural Collaborative Filtering, Aspect-Based Sentiment Analysis (ABSA), transformer models like Bidirectional Encoder Representations from Transformers (BERT), Multimodal Learning, and Explainable AI.

Getting to Know the Dataset

The dataset we chose was the *Amazon Product Reviews Dataset*. This dataset comprises reviews and metadata that are grouped by category. We chose to explore the **Electronics** reviews and accompanying metadata. First, we loaded the data, limited the length of reviews and metadata to 20,000 observations, and investigated the underlying structure. Then, we computed summary statistics. The rating distribution showed that many of the reviews had a rating between 4.25 and 5 stars (out of 5 possible stars). The distribution of helpful votes suggested that the helpfulness feature was not commonly utilized by many users. Out of the 20,000 reviews examined, 14,405 were verified purchases, indicating that this dataset contained many real user interactions with a product (as reviews with verified purchases are generally perceived as more trustworthy than ones that are not verified). The dataset contained reviews spanning from May 19, 2003 to March 17, 2023**.** The average review length was 417.25 characters, and as review length increased, the frequency of reviews tended to decrease (suggesting that customers did not typically write long reviews).

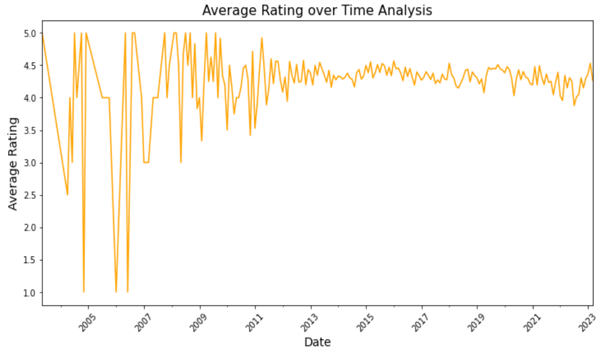

Figure 1a. Average Rating over Time Analysis

Figures 1 a&b provide insight into customer satisfaction trends over time. The initial fluctuations in ratings highlight areas for improvement in product development as well as addressing early issues with products. As ratings stabilize, it is suggestive of the presence of a more refined and consistent product line. Businesses can use this as a benchmark for maintaining quality. Occasional dips and spikes in ratings can help businesses understand and respond to changing customer needs and seasonal preferences, guiding better product design and marketing strategies. These insights can ultimately enhance customer satisfaction and support more informed customer decision-making.

The average item rating observed in the metadata was 4.08 stars (out of 5) . The top 10 categories were **Computer** (which included about 6,000 items); **All Electronics** (fewer than 5,000 items); **Camera & Photo** (~2,500 items); **Cell Phones & Accessories** (~2,000 items); and **Home Audio & Theater** (~1,000 items). **Industrial & Scientific** (~500 items, and **Amazon Fashion**, **Tools & Home Improvement**, **Car Electronics**, and **Amazon Home** were all around the same range (between 0 and 500 items). In addition, the earliest product availability date observed was 9/4/1973, and the latest product availability date observed was 5/26/2023.

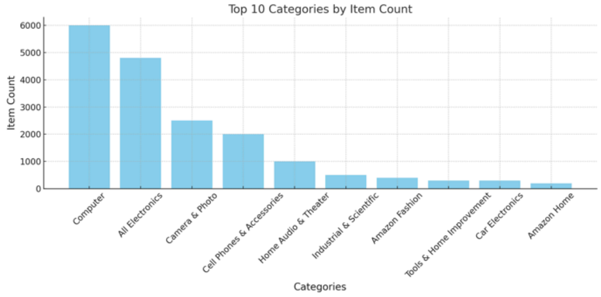

Figure 1b. Top 10 Categories and Number of Items

Finally, we explored the relationship between user reviews and metadata by inspecting the rating distribution for certain categories, creating a word cloud to analyze the presence of repeated text, and identifying insights specific to certain categories. We found that the range of the average user rating for each category was between 1 and 5, with categories like **GPS & Navigation** and **Industrial & Scientific** having an average rating of 5, while categories like **Health & Personal Care** and **Sports & Outdoors** having an average rating of 1.

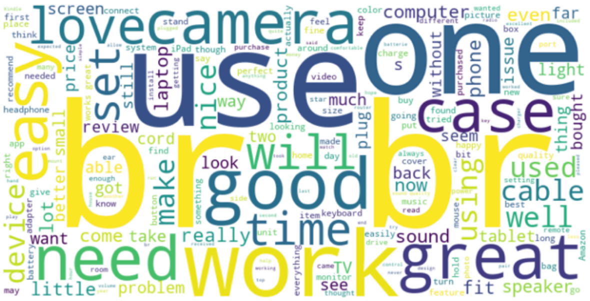

Figure 2. Frequently Used Words in Amazon Electronics Reviews

A word cloud was also created from the user review text data (Figure 2), revealing that the most frequently used words included "one," "use," and "camera", while the words that appeared least frequently included "phone," "device," and "tablet". The code used for our data exploration can be found here.

Predictive Tasks

The predictive task chosen for this dataset involved predicting product ratings based on user review data and associated features.

This task is crucial for understanding customer satisfaction and improving recommendation systems by identifying key factors influencing product ratings. Accurate predictions provide actionable insights for businesses to optimize product offerings, enhance user experience, and make informed decisions regarding product improvements.

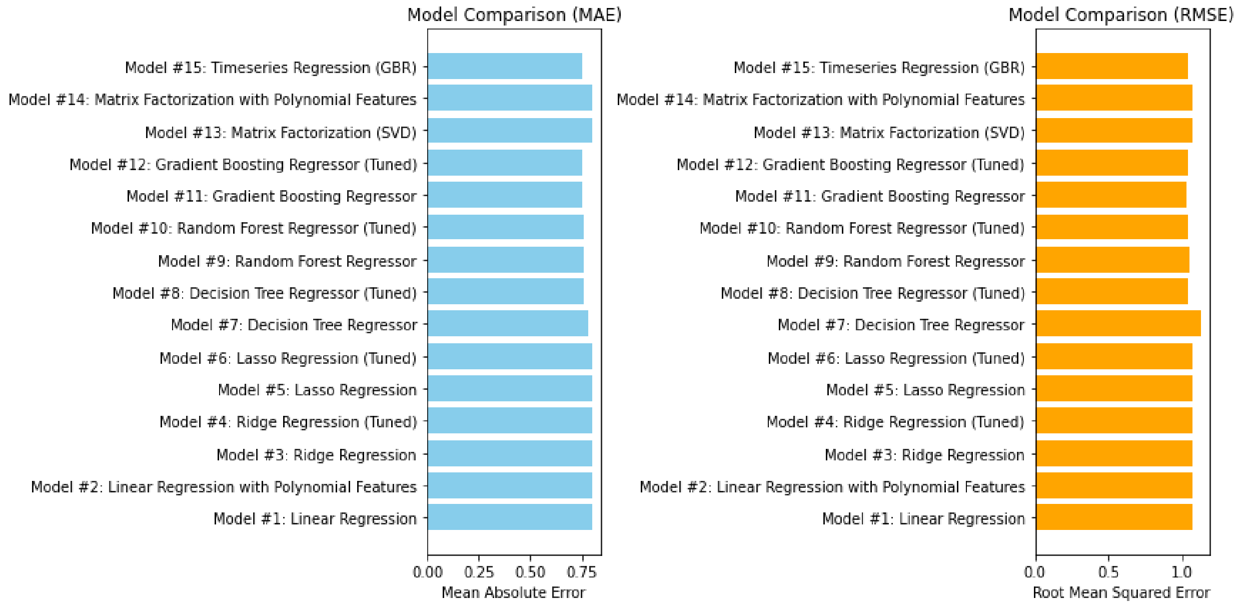

Figure 3. Model Performance Comparison

The performance of the models was evaluated using **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**.

- **MAE** provides an intuitive measure of the average magnitude of prediction errors, making it straightforward to interpret how close predictions are to the actual ratings.
- **RMSE**, which penalizes larger errors more heavily, is useful for identifying models prone to significant deviations.

The dataset was split into train (70%) and test (30%) sets while maintaining correct temporal chronological order to prevent look-ahead biases. Additionally, k-fold cross-validation or time-series cross-validation were applied to ensure robust generalization to unseen data. Baseline models for comparison included simpler approaches like **Linear Regression, Ridge Regression, and Lasso Regression** for their interpretability and computational efficiency, as well as more advanced models like **Decision Trees, Random Forests, Gradient Boosting, and Matrix Factorization (SVD)**. The inclusion of **Time Series Regression** ensured that the temporal structure of the data was explicitly considered in the evaluation.

Model Description

The **Time Series Regression Model (Model #15, Figure 3)** was selected as the best-performing model, achieving the lowest **Mean Absolute Error (MAE)** of **0.752109** and a competitive **Root Mean Squared Error (RMSE)** of **1.031324**. This model leveraged **Gradient Boosting Regression** while maintaining the correct temporal structure of the dataset, ensuring that predictions were made using past data only (preventing look-ahead bias). Gradient Boosting is a powerful ensemble learning algorithm that builds models iteratively, with each subsequent model improving upon errors made by their predecessor. This model's ability to capture complex, non-linear relationships and interactions between features made it an excellent choice for this dataset. The time-series structure provided an added advantage of preserving temporal dynamics, making the model generalizable to real-world applications.

Hyperparameter Optimization

The tuning of key hyperparameters contributed significantly to the model's performance:

1. **n_estimators**:
   - This parameter defines the number of boosting iterations (trees). A value of **100** was chosen to balance accuracy and training time. Increasing n_estimators improves complexity handling, but can risk overfitting unless paired with a lower learning rate.

2. **learning_rate**:
   - A value of **0.1** was used to ensure that each boosting step made small, incremental improvements. A lower learning rate reduces the risk of overfitting but requires more boosting iterations to achieve convergence.

3. **max_depth**:
   - The maximum depth of individual trees was limited to **3**, ensuring that the trees did not overfit the training data. Shallower trees promote better generalization.

4. **subsample**:
   - This was set to the default value of **1.0**, meaning all samples were used for training. Introducing subsampling (e.g., 0.7–0.9) future iterations could add randomness to reduce overfitting further.

5. **min_samples_split, min_samples_leaf**:
   - These were left at their defaults to balance tree complexity and leaf size. Fine-tuning these parameters may have helped to stabilize the model further.

The tuning process focused on minimizing MAE and RMSE while preventing overfitting. Time Series regression has additional constraints due to its chronological data handling, but the chosen parameters allowed the model to generalize effectively. Future experiments could include early stopping based on validation error or exploring subsampling strategies for further robustness.

Comparison with Other Models

Several other models were considered for this task, but none could match the Time Series Regression model's performance:

1. **Gradient Boosting Regressor (Tuned) - Model #12**:
   - Performed nearly as well, with an **MAE** of **0.753699** and **RMSE** of **1.032081**, making it a strong contender. However, it did not explicitly account for temporal relationships, which are critical for tasks involving sequential data.

2. **Random Forest Regressor (Tuned) - Model #10**:
   - Achieved a slightly higher **MAE** of **0.758120** and **RMSE** of **1.033031**. While it handled feature complexity well, its lack of iterative error correction limited its performance compared to Gradient Boosting.

3. **Gradient Boosting Regressor (Tuned) - Model #12**:
   - Performed nearly as well, with an **MAE** of **0.753699** and **RMSE** of **1.032081**, making it a strong contender. However, it did not explicitly account for temporal relationships, which are critical for tasks involving sequential data.

4. **Random Forest Regressor (Tuned) - Model #10**:
   - Achieved a slightly higher **MAE** of **0.758120** and RMSE of **1.033031**. While it

handled feature complexity well, its lack of iterative error correction limited its performance compared to Gradient Boosting.

5. **Decision Tree Regressor (Tuned) - Model #8**:
   - **MAE** of **0.759601** and **RMSE** of **1.040152**. Despite improvements after hyperparameter tuning, it remained less robust than ensemble methods like Gradient Boosting and Random Forest.

6. **Linear Regression, Ridge, and Lasso**:
   - These models were computationally efficient but could not capture non-linear relationships, resulting in higher **MAE** and **RMSE** values (e.g., **MAE** ~ 0.8018 for Linear Regression).
7. **Matrix Factorization (SVD)**:
   - While computationally efficient, it failed to leverage the dataset's complexity fully, resulting in higher **MAE** and **RMSE** (e.g., **MAE** ~ 0.8018)

Model	MAE	RMSE
Model #1: Linear Regression	0.801821	1.063462
Model #2: Linear Regression with Polynomial Features	0.801821	1.063462
Model #3: Ridge Regression	0.801771	1.063454
Model #4: Ridge Regression (Tuned)	0.797971	1.06333
Model #5: Lasso Regression	0.797865	1.06332
Model #6: Lasso Regression (Tuned)	0.797865	1.06332
Model #7: Decision Tree Regressor	0.779989	1.120806
Model #8: Decision Tree Regressor (Tuned)	0.759601	1.040152
Model #9: Random Forest Regressor	0.761625	1.044109
Model #10: Random Forest Regressor (Tuned)	0.75812	1.033031
Model #11: Gradient Boosting Regressor	0.753651	1.0294
Model #12: Gradient Boosting Regressor (Tuned)	0.753699	1.032081
Model #13: Matrix Factorization (SVD)	0.801821	1.063462
Model #14: Matrix Factorization with Polynomial Features	0.801821	1.063462
Model #15: Timeseries Regression (GBR)	0.752109	1.031324

Table 1. Model Performance Comparison: **MAE** and **RMSE** Across Tested Models

Results and Conclusions

The Time Series Regression using Gradient Boosting Regressor (**Model #15**) outperformed all evaluated models, achieving the lowest **MAE** (0.752109) and a competitive **RMSE** (1.031324). This indicated that the model provided the most accurate predictions with minimal average errors and reduced large deviations compared to alternatives. Its success stemmed from leveraging temporal patterns while capturing non-linear relationships, feature interactions, and subtle data patterns. The Gradient Boosting Regressor's iterative learning approach, combined with respect for temporal dependencies, made it the most robust and reliable choice.

Comparison to Alternatives

While **Model #15** excelled in overall performance, other models offered varying degrees of success. Gradient Boosting Regressor (Tuned) - **Model #12** came close, with an **MAE** of 0.753699, but its lack of temporal awareness limited its real-world applicability. Simpler models like Linear

Regression, Ridge, and Lasso failed to capture the complexity of the dataset, with higher **MAE** (~0.80) and **RMSE** (~1.06). Although the Decision Tree Regressor (Tuned) achieved an **MAE** of 0.759601, it was more prone to overfitting compared to ensemble methods. The Random Forest Regressor (Tuned) performed well, with an **MAE** of 0.758120, but its slightly higher **RMSE** (1.033031) indicated occasional large prediction errors. Finally, Matrix Factorization Models (**Models #13–#14**) performed identically to Linear Regression, as they did not fully leverage feature interactions or temporal patterns. A detailed model performance comparison can be found in **Table 1**.

Feature Representations

Key features like sentiment, review length, and temporal variables (month, year, day of the week) significantly contributed to the model's performance. These features provided meaningful patterns that the Gradient Boosting algorithm could exploit. On the other hand, binary features like verified purchase had limited impact due to their simplicity but still contributed when combined with richer features. Future experiments could explore engineered interaction terms, such as sentiment multiplied by review length, to further enhance performance.

Interpretation of Model Parameters

- **n_estimators (100)**: This parameter determined the number of boosting iterations. A moderate value allowed sufficient learning while preventing overfitting.

- **learning_rate (0.1)**: This controlled the contribution of each boosting step, balancing model complexity and improving generalization.

- **max_depth (3)**: This restricted the depth of each tree, ensuring that only the most relevant patterns were captured while avoiding overfitting.

- **subsample (1.0)**: Using the entire dataset for each tree maintained consistency, though future experiments with subsampling (e.g., 0.7) could potentially add randomness and reduce overfitting further.

Why the Proposed Model Succeeded

The Time Series Regression model succeeded due to its ability to respect the temporal structure of the data while iteratively refining predictions. By learning from past errors and modeling complex, non-linear interactions between features, it balanced bias and variance effectively. Hyperparameter tuning further enhanced its ability to generalize, ensuring it focused on meaningful patterns while avoiding overfitting. Its flexibility and capacity to integrate both temporal and non-temporal features made it uniquely suited for this dataset.

Why Other Models Failed

Other models struggled with inherent limitations. Linear models like Ridge and Lasso could not capture non-linear relationships, while Decision Trees lacked the robustness of ensemble methods and overfit the data. Although Random Forest performed well, it lacked the precision of Gradient Boosting in minimizing both average and large errors. Matrix Factorization models, while computationally efficient, failed to utilize temporal or interaction-based features effectively.

Significance of Results

The results underscore the importance of combining temporal modeling with advanced machine learning techniques for datasets with sequential and complex relationships. The success of the Time Series Regression model demonstrates the value of respecting data chronology and highlights the critical role of feature engineering and hyperparameter tuning. These findings validate the use of Gradient Boosting for predictive tasks requiring nuanced handling of non-linear and interaction patterns, providing a robust foundation for future applications.

References

[1] Hou, Y., Li, J., He, X., Yan, A., Chen, X., McAuley, J. 2024. Bridging Language and Items for Retrieval and Recommendation. arXiv:2403.03952. [Online]. Available: https://arxiv.org/pdf/2403.03952.

[2] Ni, J., Li, J., McAuley, J. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. https://cseweb.ucsd.edu/~jmcauley/pdfs/emnlp19a.pdf

[3] He, R., McAuley, J. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. https://cseweb.ucsd.edu/~jmcauley/pdfs/www16a.pdf

[4] McAuley, J., Targett, C., Shi, J., Hengel, A. 2015. Image-Based Recommendations on Styles and Substitutes. https://cseweb.ucsd.edu/~jmcauley/pdfs/sigir15.pdf

[5] Wan, M., McAuley, J. 2016. Modeling Ambiguity, Subjectivity, and Diverging Viewpoints in Opinion Question Answering Systems. https://cseweb.ucsd.edu/~jmcauley/pdfs/icdm16c.pdf

[6] McAuley, J. Yang, A. 2016. Addressing Complex and Subjective Product-Related Queries with Customer Reviews. https://cseweb.ucsd.edu/~jmcauley/pdfs/www16b.pdf

[7] McAuley, J., Leskovec, J., Jurafsky, D. 2012. Learning Attitudes and Attributes from Multi-Aspect Reviews. https://cseweb.ucsd.edu/~jmcauley/pdfs/icdm12.pdf

[8] McAuley, J., Leskovec, J. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. https://cseweb.ucsd.edu/~jmcauley/pdfs/www13.pdf