Sentiment Analysis for Amazon Reviews

Introduction

E-commerce platforms rely heavily on user data to improve their platforms and make product recommendations. Amazon, one of the world's largest distributors, handles millions of deliveries every, making it imperative to handle user data in an insightful manner to drive business decisions.

Our study explores the *Amazon Reviews Dataset (2023)*, which contains 571.54 million reviews spanning 27 years. Due to restraints on computing power and scalability, we chose to focus on the "Magazine Subscriptions" category, with 71,500 reviews, providing an ideal dataset to analyze consumer sentiment in a specific domain.

Dataset

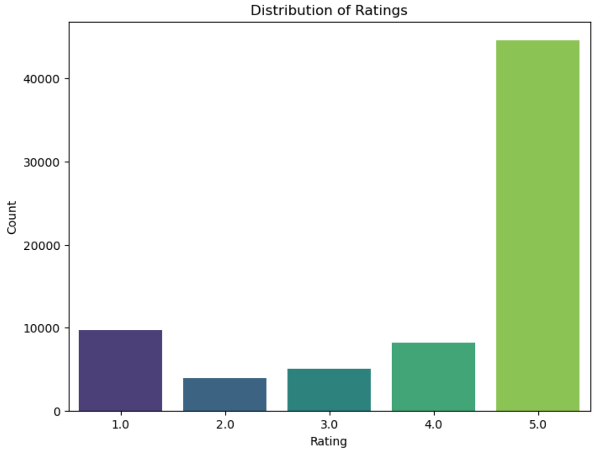
The dataset contains many columns about the product as well as the the review associated with each user and product pair, but for this analysis we focused mainly on the following columns:

- rating: number of stars given by user, ranges from 1 to 5.
- title: title of review from user
- text: string of text written for each review
- helpful_vote: number of votes given by other users to a specific review when they find it helpful
- verified_purchase: represents whether the user who wrote the review actually purchased the specific product

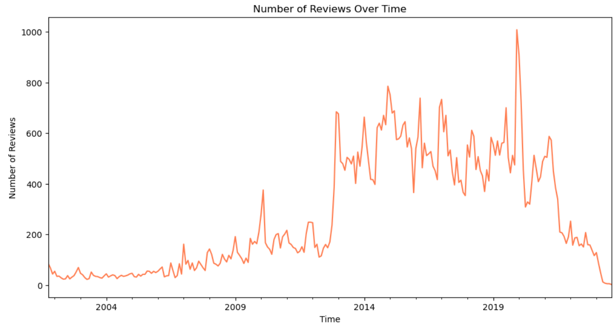
EDA and Data Cleaning

The Magazine Subscriptions dataset, a subset of the Amazon Reviews 2023 dataset, comprises reviews spanning from May 1996 to September 2023. Some basic statistics and visualizations of the dataset are as follows.

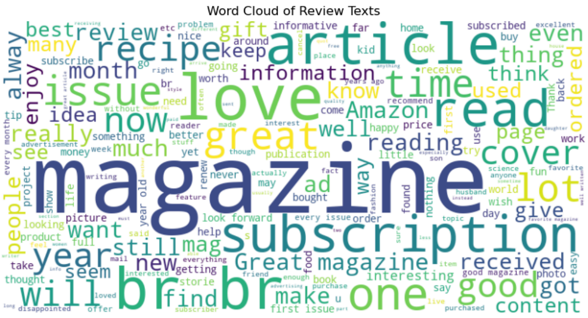
- Total reviews: ~71,500
- Unique users: ~56,000



The dataset mainly contains 5 star reviews, with 1 star reviews being second most common. This makes sense because humans are most likely to leave either a 5 star or 1 star review out of convenience if they like a product. This also implies overall customer satisfaction across Amazon's magazine subscriptions.



Upon observing temporal trends for monthly reviews, we can see a steady increase after 2013, indicating increased customer engagement with the platform. Another spike can be seen around 2020, the year when the COVID-19 pandemic began. The spikes within certain years can also be indicative of holiday periods, which usually results in increased spending.



A word cloud analysis reveals keywords present in most reviews, with common words including “magazine”, “article,” and “love.” These words seem common considering the context of the data, as it is about magazine subscriptions, and the prevalence of the word “love” aligns with the overall positive ratings across the dataset.

Task

Predictive Goal

Our main goal with this analysis is to classify customer reviews into positive and negative sentiments, excluding neutral reviews (reviews with ratings of 3 stars). While the ratings provided by each user contains information about the positivity of the review, it alone fails to explain the nuances of sentiment classification of reviews. For example, a user might rate a product 5 stars despite listing minor complaints due to the overall experience being positive. Such a rating does not accurately reflect the user’s feelings about the product, which is where more advanced sentiment analysis comes into play.

Feature Engineering

To enhance the capabilities of our models, we performed several feature engineering steps to extract more meaningful information from the data.

1. Log-transformed helpful votes: Because the helptul_vote column exhibits a heavily skewed distribution, we normalized this data by applying a log transformation, log_helpful_vote = log(1 + helpful_vote)
2. Binary verified purchase indicator: We converted the verified_purchase column to a binary indicator (0 for unverified, 1 for verified), as verified purchases typically carry more weight in influencing sentiment analysis
3. TF-IDF vectorizer: To capture the importance of words in the text column, we applied TF-IDF (Term Frequency-Inverse Document Frequency). This technique highlights words that are frequent in a specific review but less common across the whole dataset.
4. Sentiment scores: Using VADER sentiment analyzer, we computed sentiment polarity scores for both the title and text columns. These scores range from -1 (negative) to 1 (positive) and quantify the emotional tone of the review content. This was calculated using the formula, combined_sentiment = (text_sentiment + title_sentiment) / 2
5. Sentiment-Helpfulness Interaction: To explore how sentiment interacts with helpfulness, we created a new feature calculated as such, sentiment_helpful_interaction = combined_sentiment * log_helpful_vote

Baseline Concepts

We used two baseline metrics to use to compare with our final model.

1. Randomly picking reviews to be positive or negative
2. Always choosing the majority class

We will be comparing the accuracy, recall, precision, and F-1 score from this baseline metric to our final model. These metrics will assess how well our model identified positive and negative reviews. Additionally, we will use a confusion matrix to visualize the true positives, true negatives, false positives, false negatives to see where the model makes errors.

Models

We examined a variety of models, each with their own advantages and disadvantages, in order to determine the best method for sentiment analysis in the Magazine Subscriptions dataset. Logistic Regression, Random Forest, Neural Networks (MLP), K-Nearest Neighbors (KNN), LightGBM, and XGBoost were among the options. By testing several models, we were able to assess how well each performed using uniform evaluation measures, comprehend how each handled complex feature interactions, and guarantee reliable outcomes.

Baseline Models

Here are the metrics of the baseline models.

	Accuracy	Recall	F1-Score	Precision
Random Baseline	0.680	0.68	0.68	0.68
Majority Baseline	0.799	0.80	0.71	0.64

Our predicting task's performance was compared to two baseline models. The dataset's class distribution was reflected in the Random Baseline, which randomly predicts whether sentiment is positive or negative. Its accuracy, recall, F1-score, and precision were all 0.68. Due to its incapacity to accurately categorize the minority class, the Majority Baseline performed marginally better with an accuracy of 0.799 but had lower F1-score (0.71) and precision (0.64). The Majority Baseline forecasts just the majority class (positive sentiment). These baselines show that more sophisticated models are required to produce forecasts that are both meaningful and balanced.

Model Selection

To find the best model we trained and fit our features into the six different classifiers previously mentioned.

	Accuracy	Recall	F1-Score	Precision
Random Forest	0.915	0.92	0.92	0.92
Logistic regression	0.905	0.91	0.91	0.92
Neural Network	0.913	0.91	0.92	0.92
XGBoost	0.914	0.91	0.92	0.92
KNN	0.754	0.75	0.78	0.87
LightGBM	0.909	0.91	0.91	0.92

Because of its ease of use and interpretability, logistic regression was selected as the initial model. It was a great place to start because of its capacity to offer unambiguous insights regarding feature relevance. However, because it was a linear model, it had trouble capturing nonlinear interactions between characteristics like helpful votes and sentiment ratings. With an accuracy of 90.5%, it did rather well, but its ability to manage the complexity of this dataset was constrained by its reliance on linear assumptions.

With an accuracy of 91.5%, Random Forest showed a remarkable capacity to handle both text-derived and numerical information. We were able to confirm the contribution of engineering features like combined_sentiment thanks to its ensemble technique, which decreased overfitting and produced interpretable feature significance scores. In contrast to gradient boosting techniques like XGBoost, Random Forest was somewhat less effective at capturing complex feature interactions, despite its resilience.

The ability of neural networks (MLP) to represent intricate, non-linear interactions showed promise. They were comparable to Random Forest in terms of accuracy (91.3%), but they needed a lot of processing power and hyperparameter adjustment. It was really difficult to figure out the right number of layers and the correct number of nodes for optimization. Furthermore, they were less useful for drawing conclusions from the data, which was a major objective of this study, due to their lack of interpretability.

A more straightforward model, K-Nearest Neighbors (KNN), was evaluated. However, its accuracy was only 75.4% because of its difficulties with the high-dimensional TF-IDF features. Finding neighbors became ineffective for this activity due to the considerable increase in processing cost that occurred with the quantity of the dataset. Its efficacy was further constrained by its sensitivity to feature scaling. KNN's inefficiency with high-dimensional data demonstrated the importance of more sophisticated models.

LightGBM delivered competitive results with a 90.9% accuracy rate. It was attractive for scalability because it was quicker and used less memory than XGBoost. In contrast to XGBoost, it scored marginally worse on recall for the minority class, which affected its robustness. This discrepancy demonstrated how crucial customized gradient boosting models are for managing unbalanced datasets.

As shown in the table, every classifier performed better than our two baselines except KNN. Random Forest Classifier ended up having the highest accuracy, but XGBoost and Neural Networks were comparable. To determine which had the best overall performance, we performed GridSearchCV on all three classifiers. GridSearchCV optimized all three models and determined which model was best for our predictive goal.

Final Model

Due to XGBoost's overall performance, adaptability, robustness, and its ability to manage the intricacies of the Magazine Subscriptions dataset, we determined that it would be the best for our predictive goal and task. XGBoost's gradient boosted framework, which continually enhances weak learners to handle complex feature interactions and subtle patterns in the data, displayed its prowess generating a 92% accuracy and consistent scores for precision, recall, and F1-score. XGBoost was especially good at tackling class imbalance, a major issue in this data, because of its capacity to prioritize incorrectly categorized samples during training.

To make sure we got the full potential out of XGBoost we performed hyperparameter tuning to optimize the parameters within the framework.

- n_estimators: set to 200 to balance model complexity and boost efficiency
- max_depth: set to 10 to capture complex patterns without overfitting
- learning_rate: set to 0.2 to balance training speed and performance
- subsample and colsample_bytree: both set to 0.8 to use randomness and prevent overfitting

Adjusting these parameters allowed for the model to generalize effectively while maintaining a high accuracy, recall, precision, and F1-score for the positive and negative classes.

Even though XGBoost performed well, there were several difficulties when it was implemented and optimized. Scalability was a major problem since it took a lot of processing power to train XGBoost using high-dimensional TF-IDF features and a sizable dataset. To solve this, we used parallel processing (n_jobs=-1) to shorten training time and established a max_features constraint to limit the number of TF-IDF features. Another issue was overfitting, which caused the model to disproportionately favor the majority class, especially when longer trees (max_depth > 10) were used. Using regularization through colsample_bytree allowed for tree depth control and it reduced the chances of overfitting. The dataset's intrinsic class imbalance also presented challenges because the model originally favored the majority class due to the large number of positive ratings. In order to get around this, we used scale_pos_weight to modify the loss function and SMOTE to balance the classes during training. We made sure that XGBoost's accuracy and generalization skills were unhindered by these constraints by tackling these issues with meticulous tuning and preprocessing.

To sum up, XGBoost was selected due to its exceptional performance, robustness, and capacity to manage the dataset's intricacies, such as class imbalance and high-dimensional features. We achieved the maximum accuracy and generalization across all tested models by carefully adjusting the hyperparameters and tackling issues like scalability and overfitting. It is the best option for sentiment classification in the Magazine Subscriptions dataset due to its interpretability and well-balanced trade-offs.

Literature

The Amazon Reviews Dataset, provided by McAuley Lab, is one of the most comprehensive public datasets for analyzing user-generated content on e-commerce platforms. The 2023 dataset, the focus of this study, contains over 571 million reviews spanning 27 years and has been widely used in research on recommender systems, sentiment analysis, and user behavior modeling. Previous studies using this dataset include:

- Interaction between natural language and item metadata, revealing nuanced correlations between user sentiments and product descriptions
- Pipelines to extract "good" justifications from user reviews, capturing only the most relevant portions from reviews
- One-Class Collaborative Filtering that combines high level visual features with temporal modeling
- Modeling for feature vectors for product images using both visual similarity and latent stylistic coherence.

Similar datasets include the Yelp dataset, which emphasizes location-based user behaviors and preferences, the MovieLens dataset, which is used for studying collaborative filtering and hybrid recommender systems in the entertainment domain, as well as the Goodreads dataset, which focuses on textual reviews, ratings, and user interactions to understand preferences in book recommendations.

Recent methods for analyzing these datasets include deep learning, neural networks, and multimodal models that integrate image and text data. The conclusions from existing work often align with the general goal of leveraging user reviews for better recommendation systems and sentiment analysis. However, they differ in focus and implementation. Our analysis mainly focuses on extracting meaningful information from text data as well as product metadata, while previous methods may involve other forms of analysis such as image processing.

Results

To maximize performance, three models—Random Forest, XGBoost, and Neural Networks—were adjusted. Although all three models produced competitive results, XGBoost stood out as the most efficient and well-balanced:

Performance

From the Models Section, we hyperparameter tune these three main models.

	Accuracy	Recall	F1-Score	Precision
Random Forest	0.916	0.92	0.92	0.92
XGboost	0.92	0.92	0.92	0.92
Neural Network	0.914	0.91	0.92	0.92

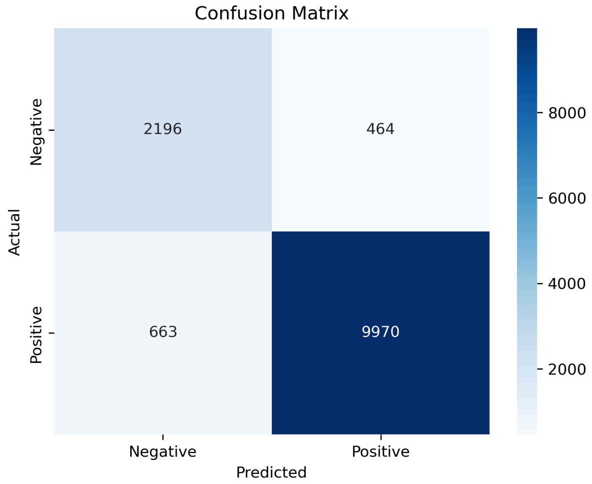
With a precision, recall, and F1-score that were all in balance, XGBoost achieved the greatest accuracy of 0.92. XGBoost's resilience in identifying both positive and negative reviews was demonstrated by the confusion matrix, which showed the lowest false positive (362) and false negative (726) rates.

With an accuracy of 0.916, Random Forest closely followed. It performed consistently across measures and handled feature interactions well, despite being marginally less effective than XGBoost. However, compared to XGBoost, it exhibited a marginally larger rate of false positives (464).

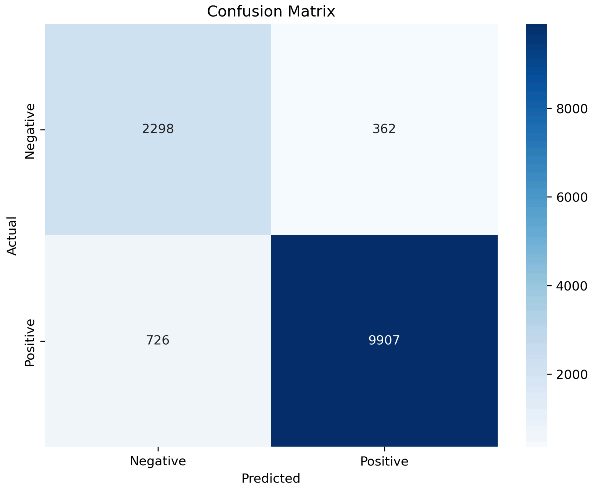
Neural Networks: Neural networks needed a lot of fine-tuning and processing power, but they were competitive, with an accuracy of 0.914. They were less useful for drawing conclusions from feature importance since they were not interpretable.

The findings highlight the value of strong models like as XGBoost, which are able to successfully manage the difficulties of feature complexity, class imbalance, and subtle feature interactions. For real-world uses like product feedback analysis and e-commerce platform optimization, the models' excellent precision and recall validate their capacity to accurately categorize both positive and negative evaluations.

The confusion matrix plots for Random Forest and XGBoost further illustrate their performance:



Strong performance on both classes are shown from the Random Forest confusion matrix above, which has 2,196 true negatives and 9,970 true positives. However, there are more false positives compared to the XGboost confusion matrix below.



With 2,298 true negatives and 9,907 real positives, the XGBoost confusion matrix performs somewhat better, lowering the quantity of false positives (362), at the risk of a slight increase in false negatives (726). The stability of XGBoost in managing unbalanced data is demonstrated by this balanced classification across positive and negative classifications. XGBoost is better suited for unbalanced data as it is able to predict the right label for the minority class.

Feature Representation

To determine the best features for classification, we tried modeling with 3 different sets of features

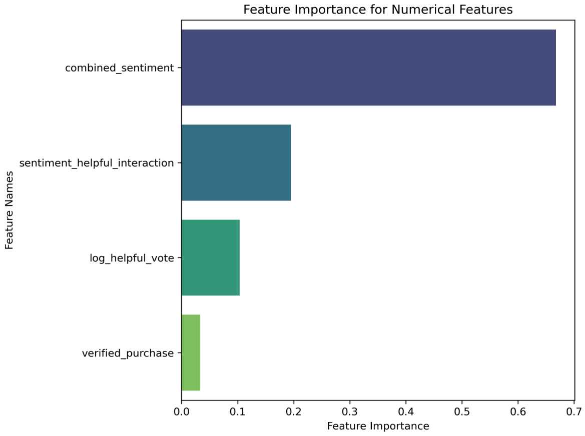
1. helpful_vote, verified_purchase
2. combined_sentiment, helpful_vote, verified_purchase
3. TF-IDF, combined_sentiment, helpful_vote, verified_purchase

The results of each model is shown on the table below.

	Accuracy	Recall	F1-Score	Precision
Helpful, Verified	0.706	0.71	0.73	0.79
Sentiment, Helpful, Verified	0.849	0.85	0.86	0.87
TF-IDF, Sentiment, Helpful, Verified	0.915	0.92	0.92	0.92

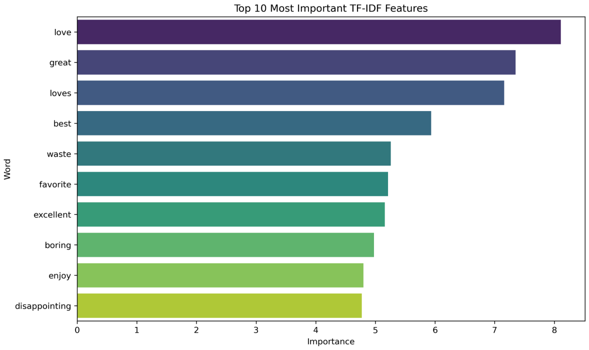
With just numerical features like verified_purchase and log_helpful_vote, the model's accuracy was 0.706, its recall was 0.71, and its F1-score was 0.73. Although these characteristics helped to comprehend the helpfulness and authenticity of reviews, they were not enough to appropriately identify sentiment since they lacked the emotional complexity and contextual depth needed to successfully differentiate between positive and negative reviews. This model also did not beat out our baseline model.

The model's performance was much improved by including sentiment features, particularly combined_sentiment. With a recall of 0.85 and an F1-score of 0.86, the accuracy rose to 0.849, underscoring the importance of sentiment in expressing the emotional tone of reviews. By bridging the gap between strictly numerical measurements and the subjective meaning ingrained in literary material, these ratings successfully quantified the positive and negative statements in the review text and title.



The significance of emotional tone and review helpfulness in sentiment classification was highlighted by the above visualization. This graph revealed that combined_sentiment was the most important feature, followed by sentiment_helpful_interaction and log_helpful_vote. Verified_purchase, on the other hand, was comparatively less significant because of its binary form and low variability.

The best performance was obtained by combining TF-IDF features with numerical and sentiment characteristics; XGBoost was able to attain an accuracy of 0.92. This shows that by detecting particular word usage and patterns that are associated with sentiment, granular textual features offer significant predictive value.



According to the TF-IDF feature importance plot, words like "love" and "great" were highly suggestive of positive sentiment, whilst words like "boring" and "disappointing" were essential for identifying negative assessments. These textual elements improved the model's comprehension of reviews by encapsulating both the emotional tone and the semantic structure.

The combination of numerical features and sentiment with TF-IDF portrays the importance of having a diverse set of features that encapsulate all of the data. The numerical features provided the foundation for our insights, but sentiment analysis provided the depth and context for model enhancement and nuanced classification.

Interpretation

The model’s parameters show why XGBoost is the best choice for this task.

- Max_depth = 10: This limits each tree's maximum depth, preventing overfitting and enabling the model to recognize intricate patterns. A depth of 10 achieves a balance between generalization and complexity.
- N_estimators = 200: The model will have enough iterations to learn complex patterns without becoming overly computationally intensive.
- Learning_rate = 0.2: This regulates the training weight update step size. Consistent convergence without overfitting is guaranteed by a reasonable learning rate.
- Subsample = 0.8 : This adds randomness to each tree's training data to avoid overfitting and enhance generalization
- Colsample_bytree = 0.8: The percentage of features taken into account for each split is managed, lowering the possibility of overfitting and increasing tree diversity.
- Regularization: To avoid overfitting, XGBoost penalizes excessively complex trees using internal regularization approaches (L1/L2 penalties).
- Class imbalance Handling: To improve the categorization of minority classes, the model uses a combination of feature sampling and weight modifications to handle class imbalance.

Together, these settings maximize XGBoost's capacity to identify intricate patterns in the data while preserving robustness and preventing overfitting.

Conclusion

Impressive sentiment classification results were obtained by combining textual and numerical information with sophisticated modeling techniques. The confusion matrix plots' insights show how XGBoost outperforms alternatives by skillfully balancing recall and precision. The relevance of sentiment features (combined_sentiment and sentiment_helpful_interaction) is shown by the numerical feature importance plot, whereas the TF-IDF feature important plot highlights the rich semantic value of particular words like "love" and "disappointing."

These tables and graphs highlight the importance of feature engineering and robust modeling by offering comprehensible insights into how features affect model performance. They confirm that attaining good performance in sentiment classification tasks requires a blend of textual features, sentiment analysis, and numerical indicators.

Our analysis concludes by emphasizing how important it is to combine strong modeling techniques with sophisticated feature engineering in order to achieve high-performance sentiment categorization. We were able to capture the complex interaction between user activity and emotional tone by utilizing textual characteristics, sentiment scores, and numerical indicators through TF-IDF. The top-performing model was XGBoost, which successfully managed feature complexity and class imbalance while striking a balance between accuracy, recall, precision, and F1-scores. The value of sentiment-driven features and granular textual representations in influencing model performance is confirmed by the insights obtained from feature importance and confusion matrix analysis. These results highlight the importance of well considered feature design as well as machine learning's capacity to provide insightful data that can be easily interpreted by e-commerce platforms looking to improve user experiences and decision-making processes.