Predicting Shares of Review on Yojo.com

Machine Learning Assignment

Zakra Chachar
Daniela Jaimes
Yohanes Nuwara

3 APRIL 2024

Table of Contents

Introduction

This project utilizes association mining and Gradient boosting, *interalia,* to predict the number of "shares" a review will receive based on textual features extracted from the review content. We aim to uncover key insights that can inform strategic business decisions and ultimately lead to improved product visibility and customer engagement.

We will explore various models and evaluate their performance in predicting the target variable. The Mean Absolute Error (MAE) metric will be used to assess the accuracy of our models. This report will detail the entire process, from data exploration and preprocessing to model selection and evaluation. We will also analyze the errors and feature importance of the chosen model, providing valuable insights into what drives review popularity on Yojo.com.

Overall, this project aims to bridge the gap between product features and their impact on share performance. By delving into the data and leveraging machine learning, Yoho.com can develop early intervention strategies to address potential negative feedback and ultimately enhance customer satisfaction.

Data Description

This project utilizes two separate datasets provided, both containing information on product reviews from Yojo.com. In total, there are 38,000 records across the two datasets.

model.csv: This dataset contains information for 28,000 reviews. It includes the textual content of the review title and content and other features potentially relevant to predicting review popularity. This dataset will be used to train and evaluate our machine-learning model.

predictions.csv: This dataset contains information for 10,000 reviews. It includes the same textual content from the review title and content as the model.csv dataset. However, crucially, it does not include the target variable, the number of shares. This dataset is intended to predict the number of “shares” reviews will receive using the trained model.

Target Variable:
The target variable of interest is the number of "shares" a review has received on Yojo.com. This variable is an integer and is represented in both the model.csv dataset (for training and evaluation) and the predictions.csv dataset (for prediction).

General Data Preprocessing

The data preprocessing approach employed in this project varied depending on the specific model being trained. Here's an overview of the general steps taken:

Outlier Removal:
We opted against removing outliers during the preprocessing stage. This decision was made because certain models within the project rely on outlier data for effective training.

Missing Value Imputation:
No missing values were identified within the dataset, eliminating the need for imputation techniques.

Feature Selection:
Attributes lacking informative value were removed from the dataset, **after EDA.** This process ensured our models focused on relevant features that contribute meaningfully to the analysis.

Feature Decomposition:
Dummy variable transformation, a type of one-hot encoding, was applied to decompose categorical features to allow training of the model on all numerical data.

Feature Scaling and Transformations:
Feature scaling in the form of Standard Scaler and Power Transformation were implemented varyingly because of the different strategies employed in this project. The details of these strategies will be elaborated upon in the sections dedicated to each model below.

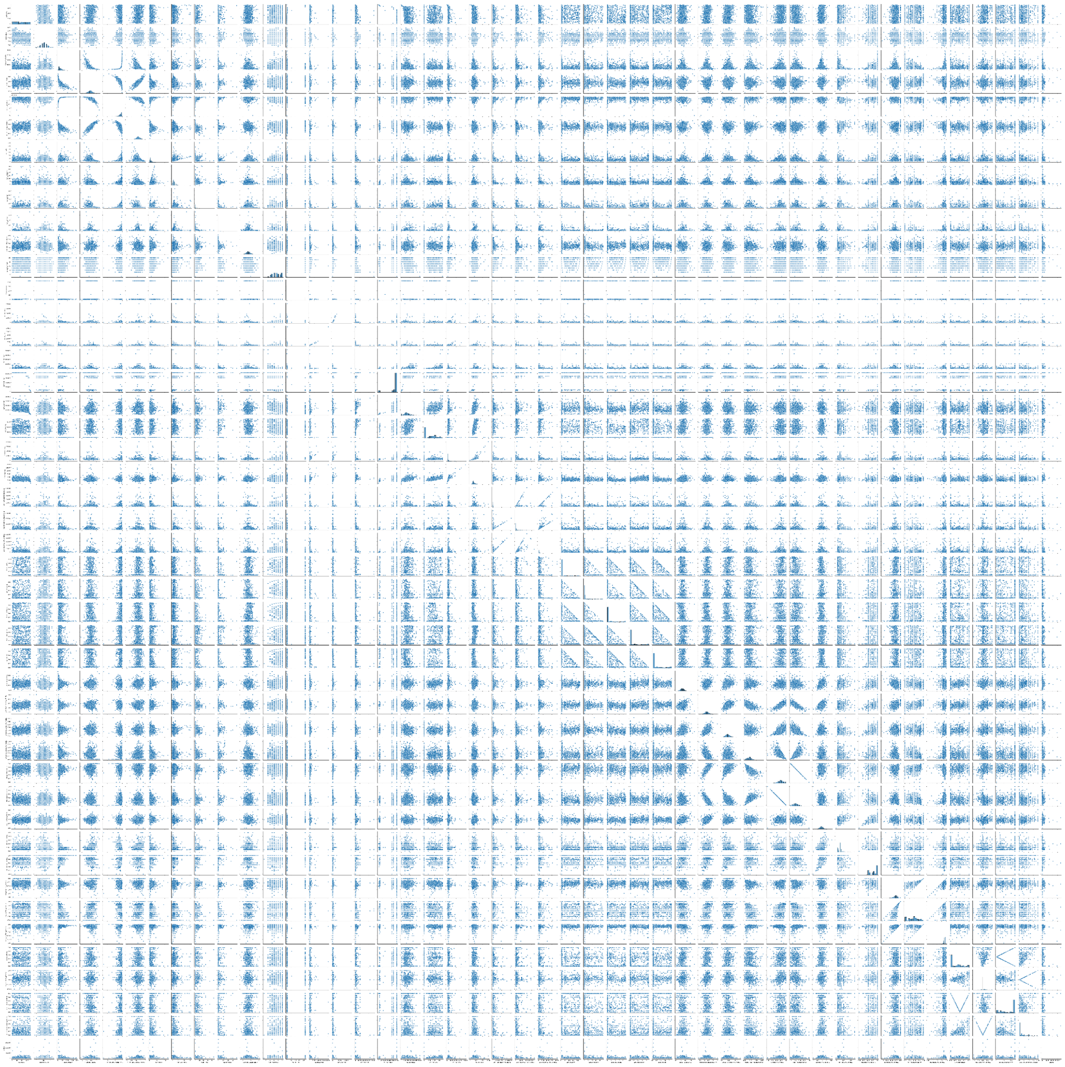
Exploratory Data Analysis (EDA)

Data Visualization & Feature Analysis

Pairplot

A pair plot was generated to visually explore the relationships between each pair of variables within the dataset. Due to the computational intensiveness of creating a pair plot for the entire dataset, a sample of 1,000 observations was used for this analysis. We recognize and acknowledge that limiting the observations might introduce slight variations compared to the full dataset.

The pair plot showed the non-linear nature of the dataset overall concerning the target variable. It also showed that linear relationships do exist between other features.



Feature Analysis

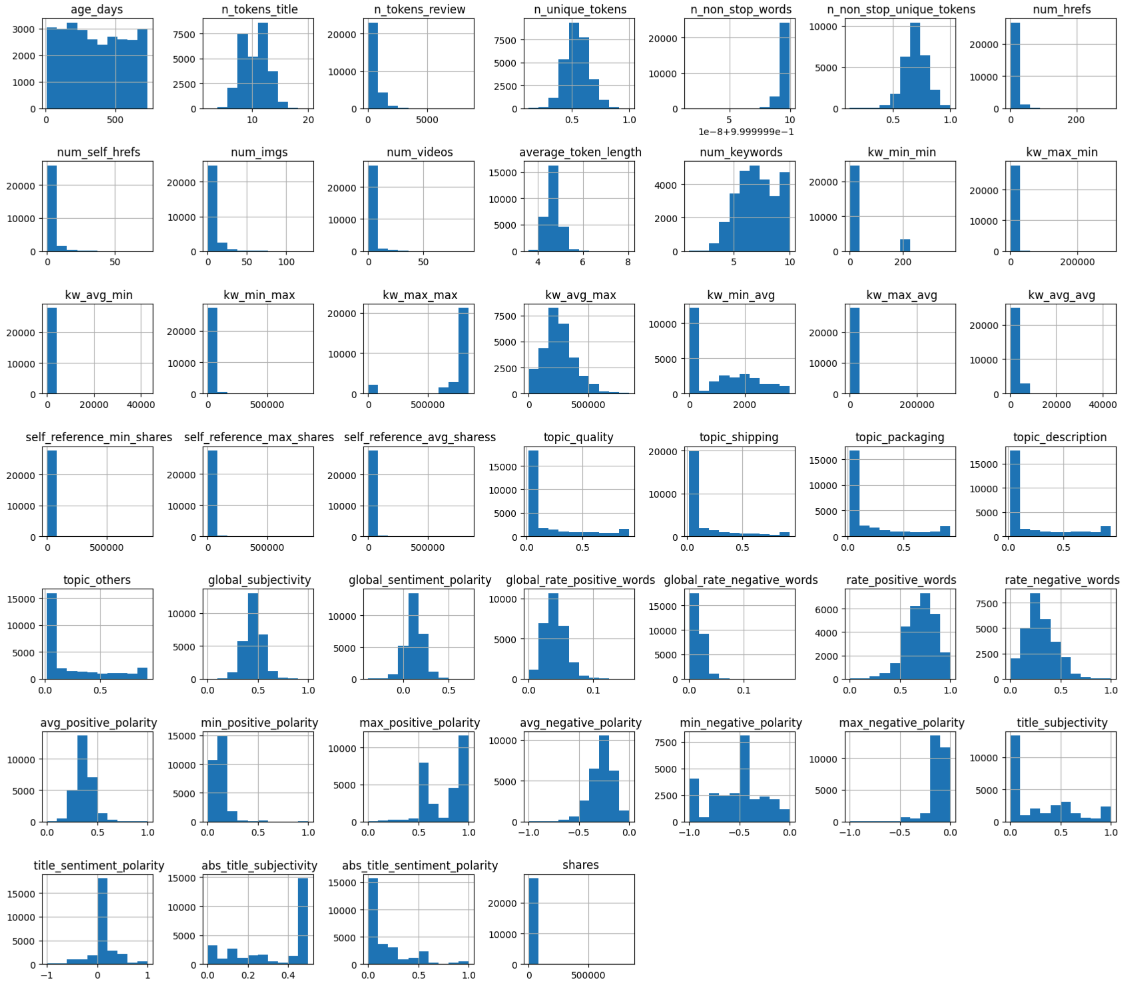
Feature Analysis provided an in-depth insight into the data and was made the basis for model improvement. There were a total of 47 features, out of which 2 were categorical features, and the remaining were numerical features.

Feature Distribution

Two plots were utilized to determine the distribution of the features, a histogram and a boxplot.

The histogram was developed to visualize each feature's distribution, which became the basis for the decision to scale features during the model selection process.

The boxplot analysis confirmed our initial suspicion that outliers were present in the features and may require treatment.



age_days, n_tokens_title, n_tokens_review, n_unique_tokens, n_non_stop_words, n_non_stop_unique_tokens, num_hrefs

num_self_hrefs, num_imgs, num_videos, average_token_length, num_keywords, kw_min_min, kw_max_min

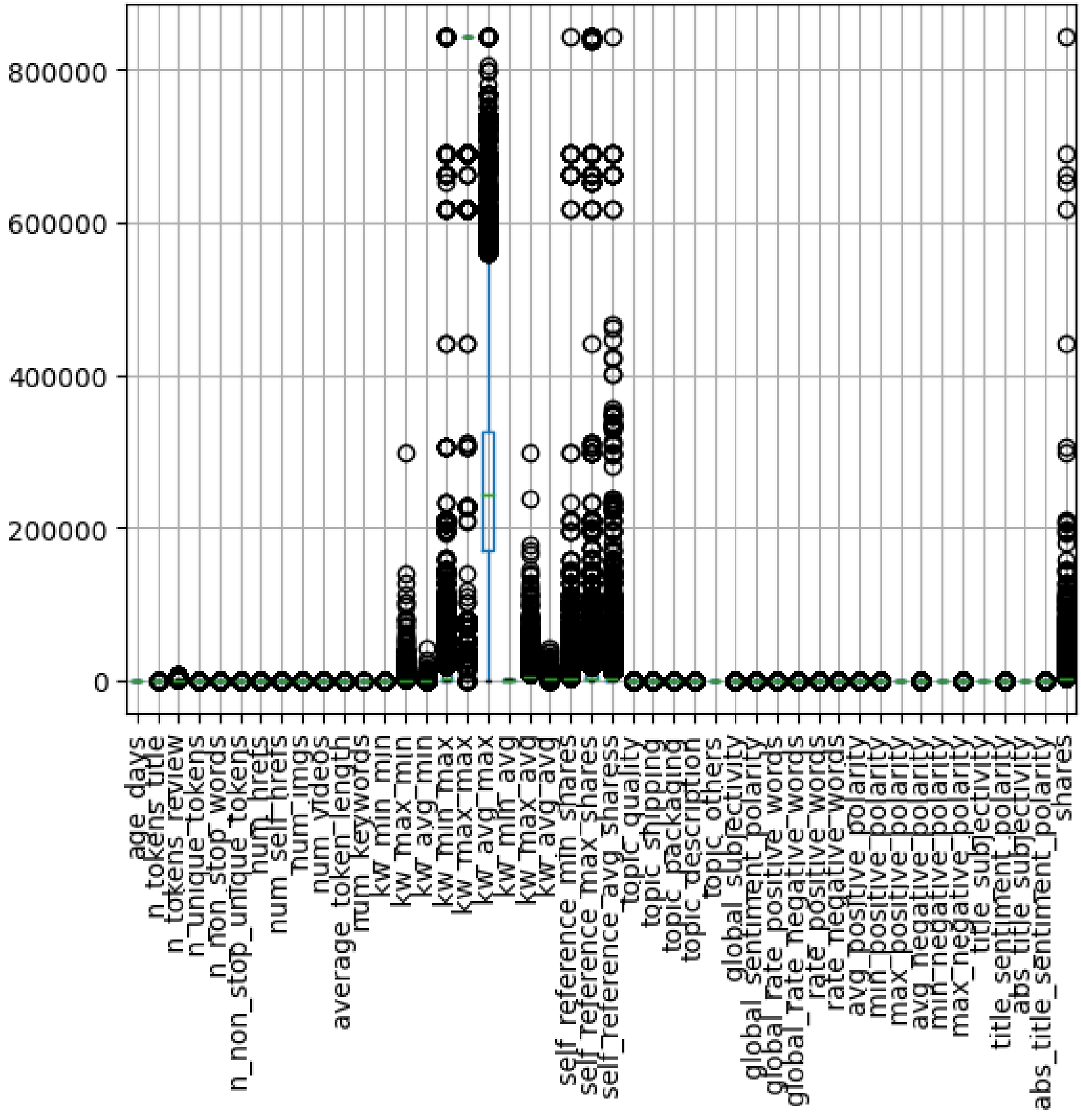
kw_avg_min, kw_min_max, kw_max_max, kw_avg_max, kw_min_avg, kw_max_avg, kw_avg_avg

self_reference_min_shares, self_reference_max_shares, self_reference_avg_sharess, topic_quality, topic_shipping, topic_packaging, topic_description

topic_others, global_subjectivity, global_sentiment_polarity, global_rate_positive_words, global_rate_negative_words, rate_positive_words, rate_negative_words

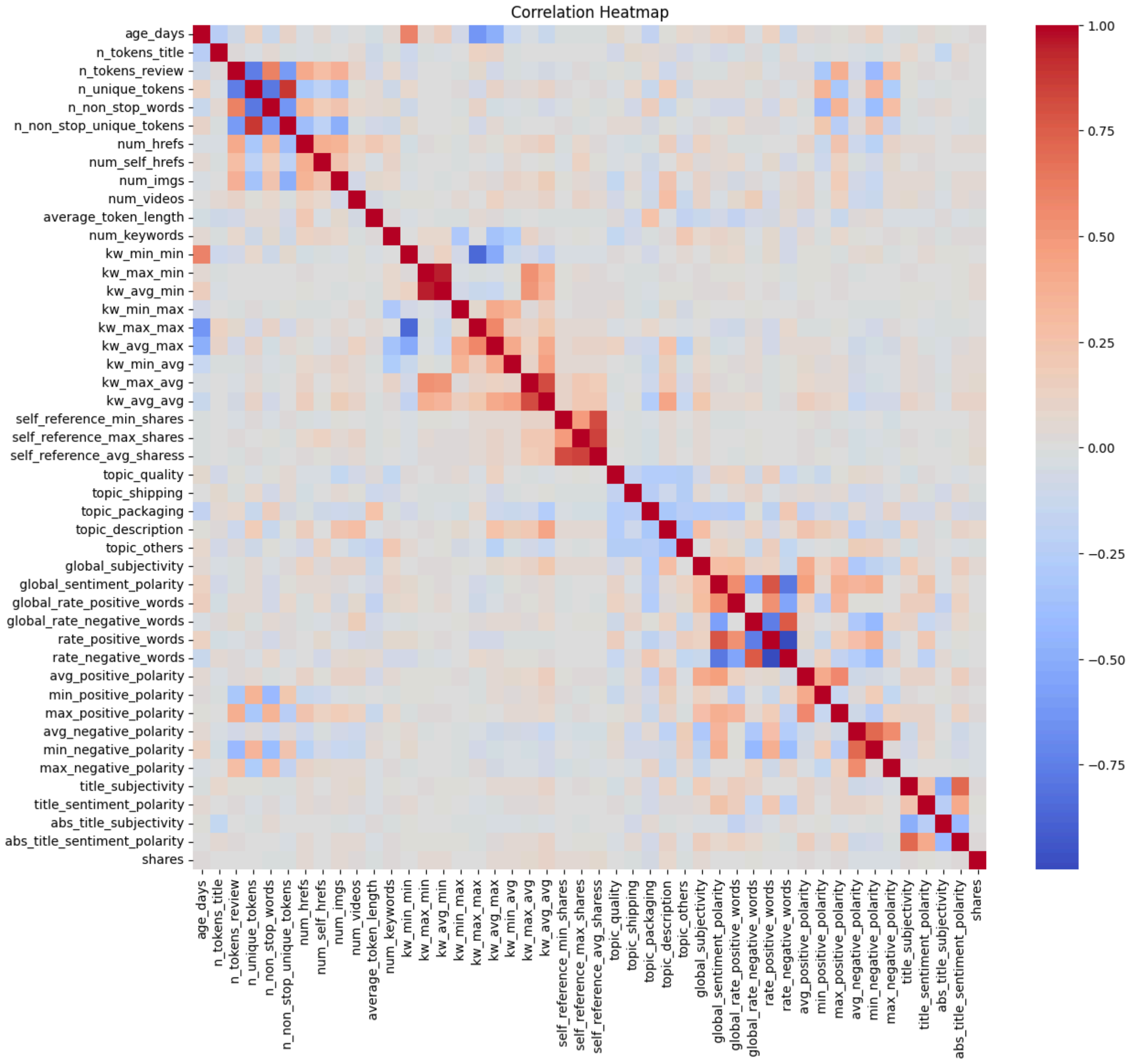
avg_positive_polarity, min_positive_polarity, max_positive_polarity, avg_negative_polarity, min_negative_polarity, max_negative_polarity, title_subjectivity

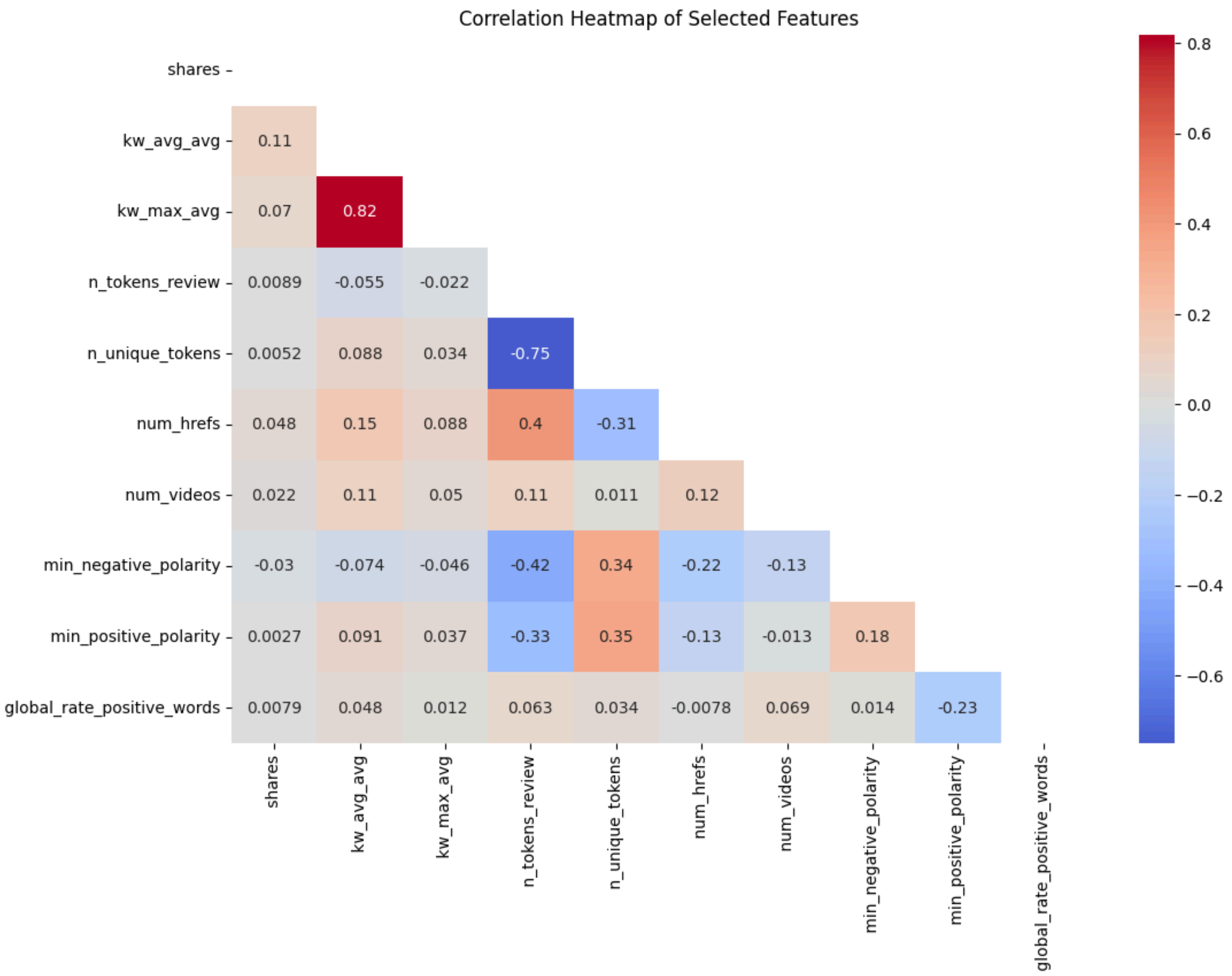
title_sentiment_polarity, abs_title_subjectivity, abs_title_sentiment_polarity, shares


800000
600000
400000
200000
0
age_days
n_tokens_title
n_tokens_review
n_unique_tokens
n_non_stop_words
n_non_stop_unique_tokens
num_hrefs
num_self_hrefs
num_imgs
num_videos
average_token_length
num_keywords
kw_min_min
kw_max_min
kw_avg_min
kw_min_max
kw_max_max
kw_avg_max
kw_min_avg
kw_max_avg
kw_avg_avg
self_reference_min_shares
self_reference_max_shares
self_reference_avg_sharess
topic_quality
topic_shipping
topic_packaging
topic_description
topic_others
global_subjectivity
global_sentiment_polarity
global_rate_positive_words
global_rate_negative_words
rate_positive_words
rate_negative_words
avg_positive_polarity
min_positive_polarity
max_positive_polarity
avg_negative_polarity
min_negative_polarity
max_negative_polarity
title_subjectivity
title_sentiment_polarity
abs_title_subjectivity
abs_title_sentiment_polarity
shares

Feature Relationships

Correlation Analysis: We first developed a heat map to identify potential correlations between features. Based on the heatmap below, we selected a few features and developed a correlation index.




Correlation Heatmap of Selected Features
shares
kw_avg_avg
kw_max_avg
n_tokens_review
n_unique_tokens
num_hrefs
num_videos
min_negative_polarity
min_positive_polarity
global_rate_positive_words
0.11
0.07
0.82
0.0089
-0.055
-0.022
0.0052
0.088
0.034
-0.75
0.048
0.15
0.088
0.4
-0.31
0.022
0.11
0.05
0.11
0.011
0.12
-0.03
-0.074
-0.046
-0.42
0.34
-0.22
-0.13
0.0027
0.091
0.037
-0.33
0.35
-0.13
-0.013
0.18
0.0079
0.048
0.012
0.063
0.034
-0.0078
0.069
0.014
-0.23
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Business Rules Based on Association Mining

As part of the exploratory analysis, we segmented the dataset by product category and employed association rule learning, this section delves deeper into the extracted insights. We'll analyze the results of the association rules for each category to identify the most influential factors impacting product share performance within those specific domains. This targeted analysis will reveal actionable business rules that can be implemented to enhance product visibility and customer engagement within each category. Finally, this analysis indicated which are the most relevant features affecting the target variable allowing us to do a more efficient feature selection.

Key Findings by Category:

Entertainment: Strong keyword usage and positive self-referencing are associated with higher shares. While a direct link is weak, encouraging positive interactions can improve brand perception.

Keyword Usage:

- The use of strong keywords, particularly the performance of minimum and maximum keywords, is consistently associated with higher shares.

Self-Referencing:

- High self-referencing behavior, where products are promoted within the content of other relevant products, is generally linked to higher shares.

Number of Keywords:

- A higher number of keywords can be beneficial for share performance, potentially increasing product discoverability.

Business: High video content usage and a strong number of external links can be beneficial for this category.

Keyword Usage:

- As in the entertainment category, strong keyword usage is positively associated with higher shares.

Video Content:

- A high number of videos is linked to higher shares.

External Links:

- A high number of external links is associated with higher shares.

Customer Reviews:

- While the direct link is weak, encouraging positive customer reviews and addressing negative feedback can contribute to a better brand image and potentially lead to increased shares.

Travel: Self-referencing within travel content and strong keyword usage are crucial.

Self-Referencing:

- Low self-referencing is linked with lower overall shares. Indicating that high self-referencing behavior leads to higher shares.

Keyword Usage:

- Just as in the other categories, strong keyword usage across various metrics (both average and maximum - kw_avg_avg_high, kw_max_avg_high) is crucial for higher shares.

Other Potential Influences:

- A weak positive association is observed between high positive sentiment analysis in reviews (global_rate_positive_words) and high shares.

Sports: Low topic quality and low self-referencing are linked to lower shares.

Topic Quality:

- Low topic quality is linked to lower shares, indicating that poor product descriptions can negatively impact performance in the sports category.

Self-Referencing:

- As in the entertainment and travel categories, in this category, low self-referencing is also a variable that is highly associated with lower shares.

Sentiment Analysis:

- While the direct link is weak, in this category we can see that having a high minimum positive polarity might be associated with lower shares.

Technology: Prioritizing strong keyword usage across various metrics is essential.

Keyword Usage:

- As in other categories, prioritizing the use of strong keywords is essential for obtaining higher shares.

Self-Referencing:

- A weak connection was found between low self-referencing and lower shares.

Review Length:

- While the direct link is weak, a high number of tokens in reviews (n_tokens_review) can be associated with higher shares, suggesting informative reviews can be beneficial.

Cleaning: Prioritizing self-referencing strategies for share improvement.

Keyword Usage:

- As in other categories, prioritizing the use of strong keywords, is essential for obtaining higher shares.

Self-Referencing:

- A weak connection was found between low self-referencing and lower shares.

Review Length:

- A high number of tokens in reviews can be associated with higher shares, suggesting informative reviews can be beneficial.

Other: Prioritizing self-referencing strategies for share improvement.

Keyword Usage:

- A connection between low keyword usage across various metrics and lower shares was found also in this category.

Self-Referencing:

- There is a weak association between low self-referencing (minimum and average) and lower shares (shares_low) with the presence of other factors (average self-referencing, minimum self-referencing).

Review Length:

- A low number of unique tokens are associated with lower shares.

Cross-Category Comparison:

Across all categories, strong keyword usage is consistently associated with higher product shares. This emphasizes the importance of optimizing product titles and descriptions with relevant keywords.

Self-referencing strategies, promoting products within relevant contexts of the same category, show potential for improvement across categories (Entertainment, Travel, Sports). However, the strength of this association varies.

Sentiment analysis in reviews can be a valuable tool for understanding customer preferences and potentially improving brand image, although the direct link with shares might be weak across most categories (Entertainment, Business, Sports, Tech). Encouraging positive customer reviews and addressing feedback is still recommended.

Conclusion:

Analyzing product share performance across categories revealed both common and distinct factors influencing shares. Strong keyword usage is crucial for all of the categories. While customer reviews play a minor role in driving shares directly, fostering positive interactions can improve brand perception. However, video content and external links seem more impactful for business products, potentially aiding education and adding credibility. Sentiment analysis in reviews can be a valuable tool for understanding customer preferences and potentially improving brand image, although the direct link with shares might be weak across most categories. Encouraging positive customer reviews and addressing feedback is still recommended.

Model Selection and Training

This section presents a detailed account of the multi-stage workflow undertaken for model selection and training. **Three distinct strategies were employed** in this phase, each utilizing different combinations of regression models to select the best model based on the lowest Mean Absolute Error (MAE)

For the selected model (strategy #3), we transformed the raw data using a power function to improve its suitability for modeling. Next, we performed a correlation analysis to uncover any relationships between the different features.

To identify category-specific patterns, we split the data by product category. Next, we used association rule learning on each category subset to find interesting relationships between features.

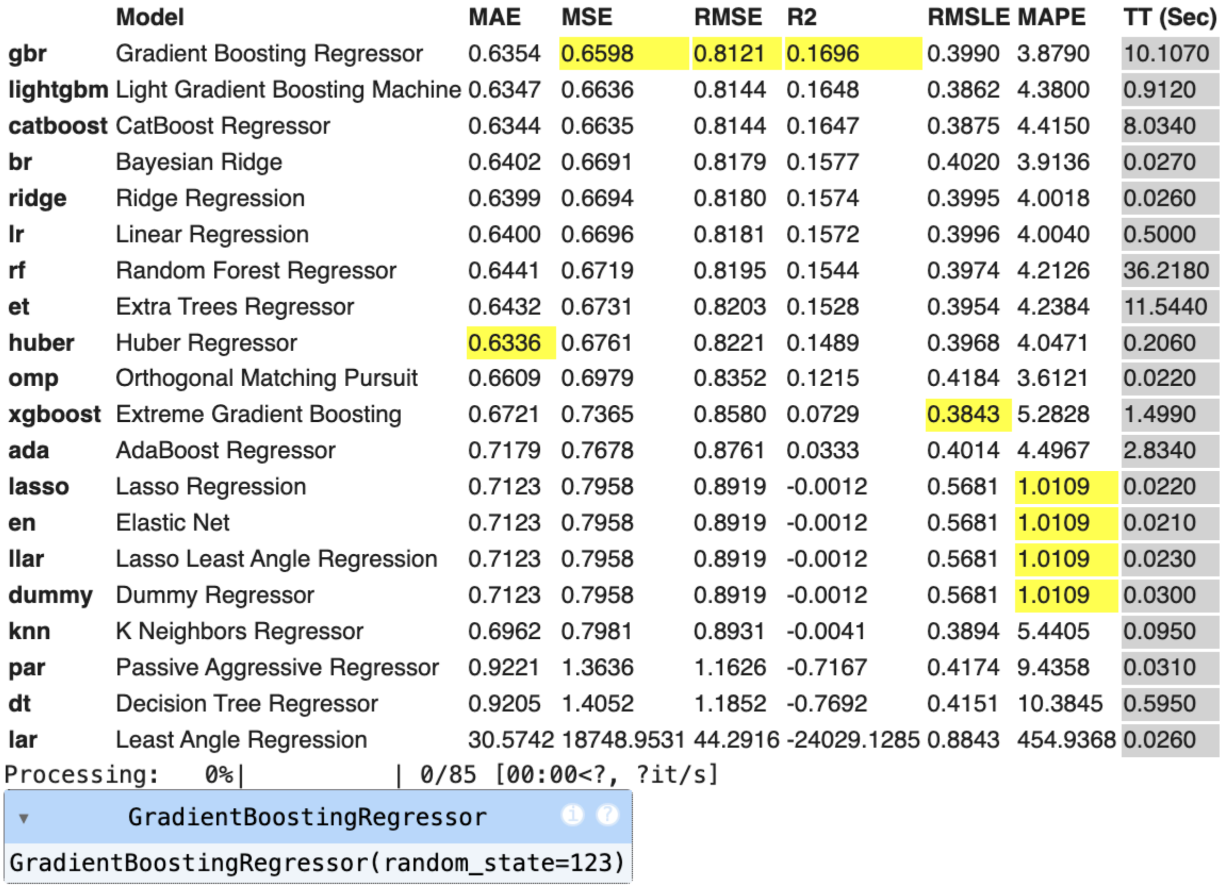
This thorough analysis provided valuable insights into feature importance. Based on these insights, we selected the five most influential features for further modeling. We trained and evaluated several regression models after removing outliers with a one-class Support Vector Machine (SVM). Gradient Boosting returned the best MAE (evidence under Hyperparameter Tuning), which was ultimately used to make predictions for the target variable.

Model Choice and Justification

The model choice is presented on a step-by-step approach, explaining each strategy before reaching the final model under strategy 3.

Strategy #1

After the completion of preliminary data processing activities, including outlier treatment, we employed the PyCaret library to facilitate a comprehensive evaluation of various machine learning models. In this initial benchmarking exercise, all numerical features were included as input variables. The PyCaret analysis yielded Gradient Boosting as the model exhibited superior performance as measured by the Mean Absolute Error (MAE) criterion.



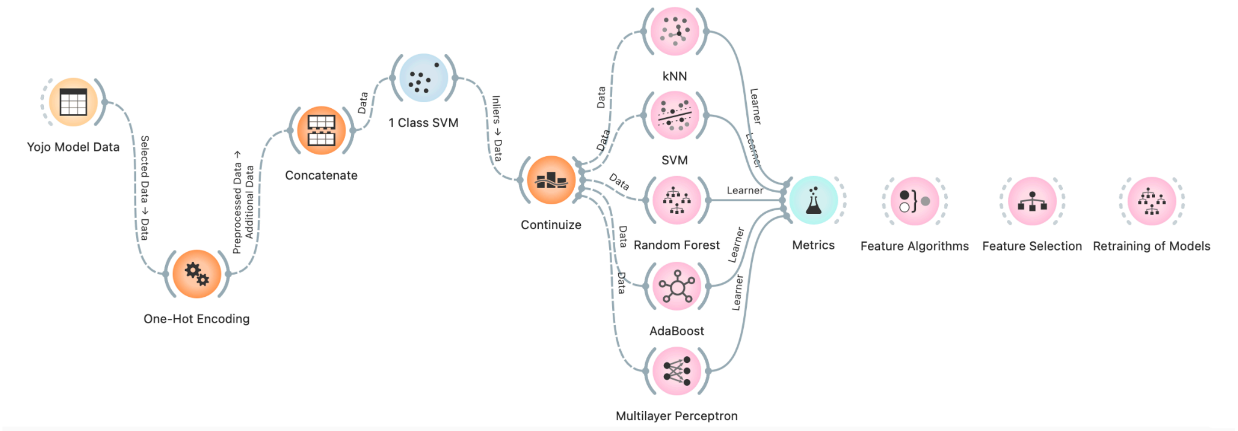
	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
gbr	Gradient Boosting Regressor	0.6354	0.6598	0.8121	0.1696	0.3990	3.8790	10.1070
lightgbm	Light Gradient Boosting Machine	0.6347	0.6636	0.8144	0.1648	0.3862	4.3800	0.9120
catboost	CatBoost Regressor	0.6344	0.6635	0.8144	0.1647	0.3875	4.4150	8.0340
br	Bayesian Ridge	0.6402	0.6691	0.8179	0.1577	0.4020	3.9136	0.0270
ridge	Ridge Regression	0.6399	0.6694	0.8180	0.1574	0.3995	4.0018	0.0260
lr	Linear Regression	0.6400	0.6696	0.8181	0.1572	0.3996	4.0040	0.5000
rf	Random Forest Regressor	0.6441	0.6719	0.8195	0.1544	0.3974	4.2126	36.2180
et	Extra Trees Regressor	0.6432	0.6731	0.8203	0.1528	0.3954	4.2384	11.5440
huber	Huber Regressor	0.6336	0.6761	0.8221	0.1489	0.3968	4.0471	0.2060
omp	Orthogonal Matching Pursuit	0.6609	0.6979	0.8352	0.1215	0.4184	3.6121	0.0220
xgboost	Extreme Gradient Boosting	0.6721	0.7365	0.8580	0.0729	0.3843	5.2828	1.4990
ada	AdaBoost Regressor	0.7179	0.7678	0.8761	0.0333	0.4014	4.4967	2.8340
lasso	Lasso Regression	0.7123	0.7958	0.8919	-0.0012	0.5681	1.0109	0.0220
en	Elastic Net	0.7123	0.7958	0.8919	-0.0012	0.5681	1.0109	0.0210
llar	Lasso Least Angle Regression	0.7123	0.7958	0.8919	-0.0012	0.5681	1.0109	0.0230
dummy	Dummy Regressor	0.7123	0.7958	0.8919	-0.0012	0.5681	1.0109	0.0300
knn	K Neighbors Regressor	0.6962	0.7981	0.8931	-0.0041	0.3894	5.4405	0.0950
par	Passive Aggressive Regressor	0.9221	1.3636	1.1626	-0.7167	0.4174	9.4358	0.0310
dt	Decision Tree Regressor	0.9205	1.4052	1.1852	-0.7692	0.4151	10.3845	0.5950
lar	Least Angle Regression	30.5742	18748.9531	44.2916	-24029.1285	0.8843	454.9368	0.0260

```
Processing:   0%|          | 0/85 [00:00<?, ?it/s]
```

```
▾     GradientBoostingRegressor
GradientBoostingRegressor(random_state=123)
```

Strategy #2

Strategy 2 at a Glance:



This approach consisted of the following steps:

1. **Feature Encoding:** We employed one-hot encoding to represent all categorical features in the data, alongside the existing numerical features. This ensures all features are compatible with the machine learning models we intend to use.
2. **Outliers and Scaling:** We implemented two outlier detection algorithms: **One-Class SVM** and **Isolation Forest**. Based on our evaluation, SVM was chosen for outlier removal. Following outlier removal, we split it into training and testing sets with a 70/30 ratio and standardized the data using **StandardScaler**.
3. **Models:** A selection of machine learning algorithms were trained and compared using the training and test data to determine the performance of different models.

Models	Parameters	Results
KNNRegressor	Neighbours: 2-20, Parameters: 1, 2, 3	MAE train: 2365.9359440267335
		MSE train: 34097387.52236151
		RMSE train: 5839.296834582184
		r2: 0.34462507738848447
		MAE test: 2512.1529524777425
		MSE test: 45036905.84830196
		RMSE test: 6710.954168246269
		r2: 0.21218211868897974

Random Forest Regressor	n-Estimators: 100	Mean Squared Error: 46172692.74060194
		R-squared: 0.020937617756944027
		Mean Absolute Error: 2875.0140021161224
Support Vector Regressor	'C': [10], 'epsilon': [0.01],'gamma':['auto'], 'kernel': ['poly','rbf', 'sigmoid'] (since Pairplot showed there is no linear relationship),'degree': [2,3,5]	MAE train: 2090.5561130974315
		MSE train: 40469476.971504815
		RMSE train: 6361.562463067136
		r2: nan
		MAE test: 2091.005078859135
		MSE test: 48514349.84415296
		RMSE test: 6965.224321165325
		r2: nan
Multi Layer Perceptron	param_grid = { 'hidden_layer_sizes': [(8,), (10, 5), (20, 10, 5)], 'solver': ['sgd'](used sigmoid as solver as it , 'alpha': 10.0 ** -np.arange(-3, 2), 'max_iter': [5000, 10000]	Best Hyperparameters: {'alpha': 100.0, 'hidden_layer_sizes': (10, 5), 'max_iter': 5000, 'solver': 'sgd'}
		MAE: 2792.2751756128946
		MSE: 47160276.94639392
		R^2: -3.472913690849566e-06
Adaboost	base1 = DecisionTreeRegressor(max_depth=3) & param_grid = { "n_estimators": [50, 100, 200], "learning_rate": [0.1, 0.5, 1.0], base2 = SVR(kernel='poly')	MAE train: 2162.693692004157
		MSE train: 40759051.481220536
		RMSE train: 6384.28159476229
		r2: nan
		MAE test: 2167.1864701479385
		MSE test: 49063336.84076905
		RMSE test: 7004.522599061913
		r2: nan

4. **Feature Selection (Attempted):** We initially explored feature selection algorithms like **RFE** and **SelectKBest** to identify the most impactful features and potentially boost model performance. This process ran for 36 hours on Colab Pro. Unfortunately, these algorithms did not yield results.

   Given the limitations of the initial approach, we opted for a different strategy. We utilized feature importance scores generated by a Random Forest model to identify the most relevant features. These selected features were then used for further analysis.

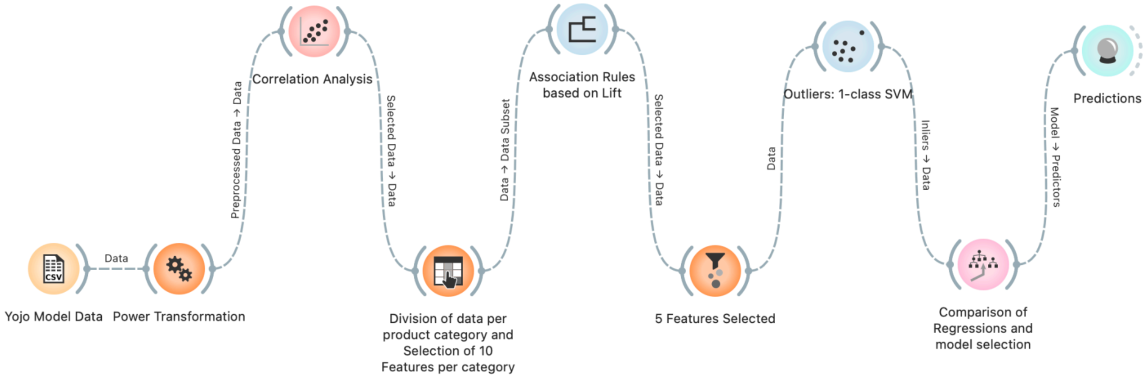
Feature Names	Scores
kw_avg_avg	0.0689
topic_shipping	0.0432
kw_max_avg	0.0431
topic_packaging	0.0371
global_sentiment_polarity	0.0351
topic_others	0.0335
global_subjectivity	0.0318
average_token_length	0.0306
kw_avg_max	0.0281
avg_positive_polarity	0.0279
age_days	0.0275
topic_description	0.0271
self_reference_avg_sharess	0.0269
topic_quality	0.0263
global_rate_positive_words	0.025
global_rate_negative_words	0.0247
n_non_stop_unique_tokens	0.0238
self_reference_min_shares	0.0232
num_hrefs	0.0225
kw_avg_min	0.0218
avg_negative_polarity	0.0216
n_unique_tokens	0.021
self_reference_max_shares	0.0206
kw_max_min	0.0204

5. **Model Retraining with Selected Features:** Despite the hurdles in feature selection, we proceeded by retraining the models using the selected features based on Random Forest and KNN
6. **Performance Comparison:** Finally, we compared the performance of these retrained models (using selected features) to the baseline models trained on all features. This comparison helps us assess the impact of feature selection on model performance.

	Models	Parameters	Results
Before Feature Selection	KNN Regressor	Neighbours: 2-20, Parameters: 1, 2, 3	MAE train: 2365.9359440267335
			MSE train: 34097387.52236151
			RMSE train: 5839.296834582184
			r2: 0.3446250773884847
			MAE test: 2512.1529524777425
			MSE test: 45036905.84830196
			RMSE test: 6710.954168246269
			r2: 0.21218211868897974
	Random Forest Regressor	n-Estimators: 100	Mean Squared Error: 46172692.74060194
			R-squared: 0.020937617756944027
			Mean Absolute Error: 2875.0140021161224
After Feature Selection	KNN Regressor	Neighbours: 2-20, Parameters: 1, 2, 3	MAE train: 2436.7425886143933
			MSE train: 34468844.54969818
			RMSE train: 5871.017335155652
			r2: 0.10916628333982714
			MAE test: 2573.130621819726
			MSE test: 45493915.98639322
			RMSE test: 6744.917789446601
			r2: 0.035330644165587195
	Random Forest Regressor	n-Estimators: 100	Mean Squared Error: 46510620.388541035
			R-squared: 0.013772078379031383
			Mean Absolute Error: 2867.656944848565

Strategy #3:

Strategy 3 at a Glance:



Building upon Strategy 1 and Strategy 2, the third strategy employs correlation analysis and association rule mining to identify the most relevant features (reduced to 9) for model training. PyCaret's model selection technique was leveraged, and hyperparameter tuning was conducted to optimize accuracy, which was further evaluated using K-fold cross-validation.

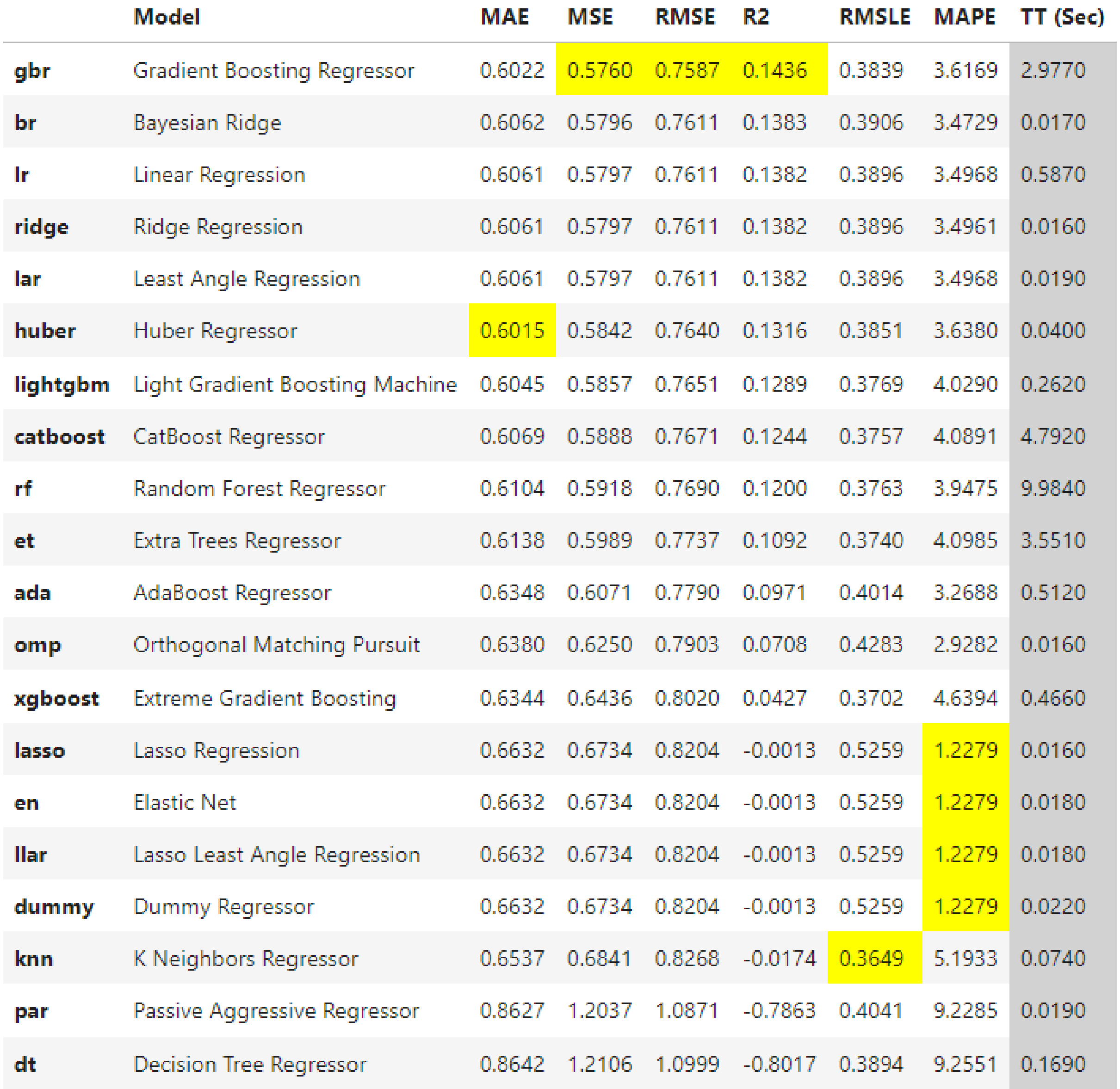
To assess the impact of data partitioning, this strategy incorporates two experiments. The first experiment utilizes the 9 features for training a model. In contrast, the second experiment segments the data by category and trains a separate model for each category.

Experiment 3.1: Correlation-Driven Feature Selection

The modeling process commenced with data import, followed by a two-step feature selection approach. Initially, Pearson's correlation coefficient identified the 9 most relevant variables from the original 37 based on their correlation with the target variable. Subsequently, association rule mining techniques were employed to explore relationships between features, with a focus on calculating Lift values to assess the predictive power of the selected features.

Next, a pre-processing pipeline was implemented. Categorical variables were encoded using one-hot encoding, while Yeo-Johnson power transformation ensured data normalization. One-class SVM with a 20% contamination factor was employed for outlier removal. To maintain the representativeness of product categories across training and testing sets, stratified sampling was utilized for data partitioning.

The model selection process involved a meticulous evaluation of various regression algorithms on the training data. Gradient Boosting emerged as the most effective model based on model evaluation metrics, as depicted below.



	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
gbr	Gradient Boosting Regressor	0.6022	0.5760	0.7587	0.1436	0.3839	3.6169	2.9770
br	Bayesian Ridge	0.6062	0.5796	0.7611	0.1383	0.3906	3.4729	0.0170
lr	Linear Regression	0.6061	0.5797	0.7611	0.1382	0.3896	3.4968	0.5870
ridge	Ridge Regression	0.6061	0.5797	0.7611	0.1382	0.3896	3.4961	0.0160
lar	Least Angle Regression	0.6061	0.5797	0.7611	0.1382	0.3896	3.4968	0.0190
huber	Huber Regressor	0.6015	0.5842	0.7640	0.1316	0.3851	3.6380	0.0400
lightgbm	Light Gradient Boosting Machine	0.6045	0.5857	0.7651	0.1289	0.3769	4.0290	0.2620
catboost	CatBoost Regressor	0.6069	0.5888	0.7671	0.1244	0.3757	4.0891	4.7920
rf	Random Forest Regressor	0.6104	0.5918	0.7690	0.1200	0.3763	3.9475	9.9840
et	Extra Trees Regressor	0.6138	0.5989	0.7737	0.1092	0.3740	4.0985	3.5510
ada	AdaBoost Regressor	0.6348	0.6071	0.7790	0.0971	0.4014	3.2688	0.5120
omp	Orthogonal Matching Pursuit	0.6380	0.6250	0.7903	0.0708	0.4283	2.9282	0.0160
xgboost	Extreme Gradient Boosting	0.6344	0.6436	0.8020	0.0427	0.3702	4.6394	0.4660
lasso	Lasso Regression	0.6632	0.6734	0.8204	-0.0013	0.5259	1.2279	0.0160
en	Elastic Net	0.6632	0.6734	0.8204	-0.0013	0.5259	1.2279	0.0180
llar	Lasso Least Angle Regression	0.6632	0.6734	0.8204	-0.0013	0.5259	1.2279	0.0180
dummy	Dummy Regressor	0.6632	0.6734	0.8204	-0.0013	0.5259	1.2279	0.0220
knn	K Neighbors Regressor	0.6537	0.6841	0.8268	-0.0174	0.3649	5.1933	0.0740
par	Passive Aggressive Regressor	0.8627	1.2037	1.0871	-0.7863	0.4041	9.2285	0.0190
dt	Decision Tree Regressor	0.8642	1.2106	1.0999	-0.8017	0.3894	9.2551	0.1690

Experiment 3.2: Product Category-Specific Model Development with Feature Refinement

In this experiment, we split the dataset into each product category and developed a different model per category.

1. **Feature Selection**: Building upon the initial feature selection process (refer to section 3.1), this round delves deeper by leveraging association rule mining on the category-partitioned data. This technique identified the most relevant features (5 out of the original 9) within each product category. Interestingly, while some features were common across categories, others emerged as unique to specific product types.
2. **Outlier Treatment:** To maintain consistency in outlier treatment across all categories, we employed the same one-class SVM with a 20% contamination factor for outlier removal within each category.
3. **Train-Test Split:** To ensure robust model evaluation for each category, we further split the data within each category into training and testing sets.
4. **Model Building and Comparison:** PyCaret was utilized to construct and compare various machine learning models within each product category. Here, our primary focus was on maximizing the accuracy metric for each category-specific model.

A table summarizing each product category, the chosen model, and the selected features for each model is presented

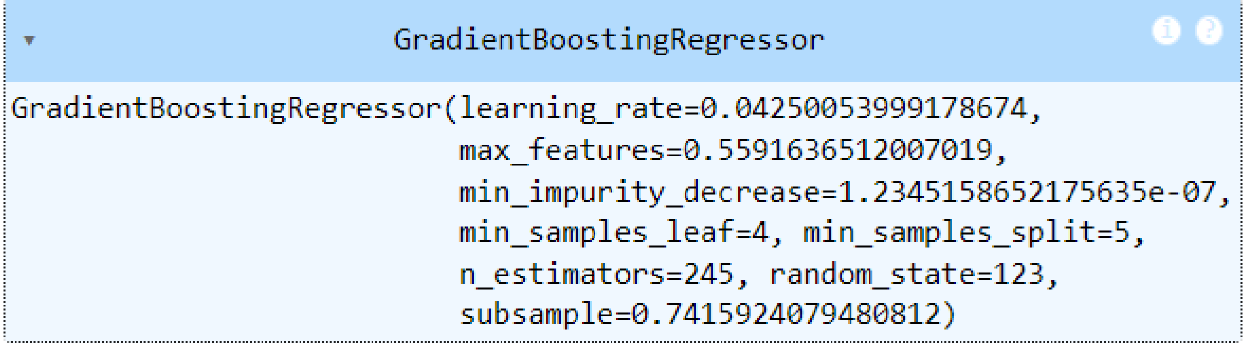
Category	Model	Selected Features
Sport	Gradient boosting	'min_positive_polarity',' self_reference_max_shares',' kw_avg_min',' self_reference_min_shares', 'topic_quality'
Travel	Gradient boosting	'self_reference_avg_sharess',' kw_avg_avg', 'topic_packaging', ' self_reference_max_shares', ' global_rate_positive_words'
Tech	Gradient boosting	' kw_avg_avg',' self_reference_avg_sharess',' min_negative_polarity',' n_tokens_review', ' n_unique_tokens'
Business	Gradient Boost	' kw_avg_avg',' n_tokens_review',' num_hrefs', ' global_rate_positive_words',' min_positive_polarity'
Entertainment	Gradient Boost	' kw_max_min',' self_reference_min_shares',' self_reference_max_shares',' kw_max_avg',' num_keywords'
Cleaning	Bayesian ridge regression	'self_reference_avg_sharess', ' self_reference_max_shares', ' num_imgs', ' num_hrefs', ' kw_avg_avg'

Other	Gradient boosting	' num_imgs',' kw_avg_avg',' self_reference_avg_sharess', ' n_unique_tokens',' kw_max_avg'

Hyperparameter Tuning

Strategy 2 employed a grid search approach to optimize hyperparameters for each model (details provided in the table above). A consistent cross-validation setting of cv=3 was used across all models in this strategy.

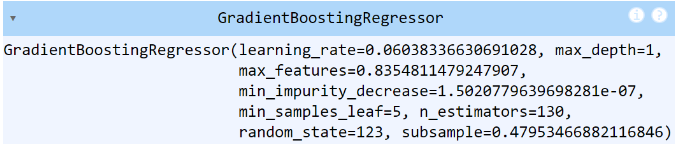
In 3.1, Hyperparameter tuning was employed using Optuna for 150 iterations to optimize the Gradient Boosting model's performance on the cross-validation set. This resulted in a reduction of Mean Absolute Error (MAE). The final, tuned model achieved promising results on the test set, with a 10-fold average of normalized MAE (0.598), MSE (0.572), RMSE (0.756), and R-squared (0.15). Specific hyperparameter values for the tuned model are displayed below.

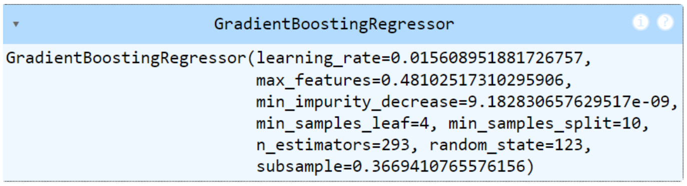


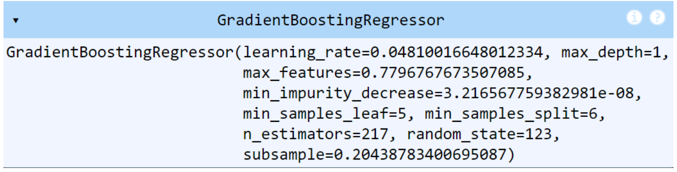
```
GradientBoostingRegressor(learning_rate=0.04250053999178674,
                          max_features=0.5591636512007019,
                          min_impurity_decrease=1.2345158652175635e-07,
                          min_samples_leaf=4, min_samples_split=5,
                          n_estimators=245, random_state=123,
                          subsample=0.7415924079480812)
```

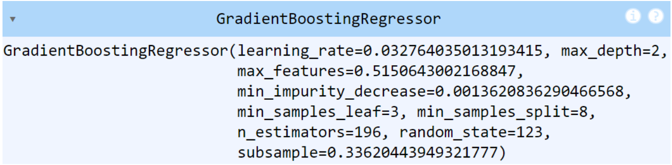
In 3.2, Hyperparameter tuning with Optuna (50 iterations) was conducted for each model developed within each product category. A table summarizes the product category, chosen model, 10-fold average normalized MAE, and the tuned hyperparameters for each category-specific model.

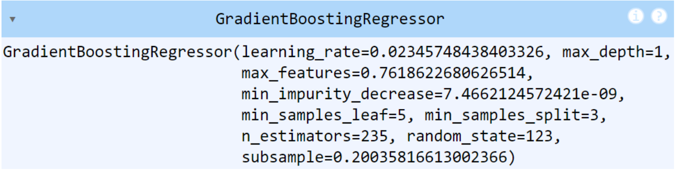
Category	Model	MAE (10-fold normalized)	Hyperparameters after tuning

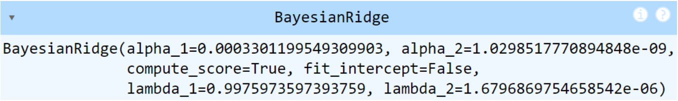
Sport	Gradient boosting	0.6188	
			GradientBoostingRegressor
			GradientBoostingRegressor(learning_rate=0.06038336630691028, max_depth=1,
			max_features=0.8354811479247907,
			min_impurity_decrease=1.5020779639698281e-07,
			min_samples_leaf=5, n_estimators=130,
			random_state=123, subsample=0.47953466882116846)

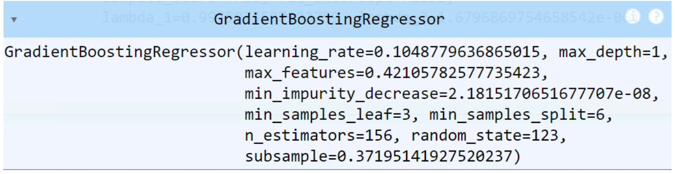
Travel	Gradient boosting	0.5461	
			GradientBoostingRegressor
			GradientBoostingRegressor(learning_rate=0.015608951881726757,
			max_features=0.48102517310295906,
			min_impurity_decrease=9.182830657629517e-09,
			min_samples_leaf=4, min_samples_split=10,
			n_estimators=293, random_state=123,
			subsample=0.3669410765576156)

Tech	Gradient boosting	0.6390	
			GradientBoostingRegressor
			GradientBoostingRegressor(learning_rate=0.04810016648012334, max_depth=1,
			max_features=0.7796767673507085,
			min_impurity_decrease=3.216567759382981e-08,
			min_samples_leaf=5, min_samples_split=6,
			n_estimators=217, random_state=123,
			subsample=0.20438783400695087)

Business	Gradient Boost	0.6141	
			GradientBoostingRegressor
			GradientBoostingRegressor(learning_rate=0.032764035013193415, max_depth=2,
			max_features=0.5150643002168847,
			min_impurity_decrease=0.0013620836290466568,
			min_samples_leaf=3, min_samples_split=8,
			n_estimators=196, random_state=123,
			subsample=0.33620443949321777)

Entertainment	Gradient Boost	0.7310	
			GradientBoostingRegressor
			GradientBoostingRegressor(learning_rate=0.02345748438403326, max_depth=1,
			max_features=0.7618622680626514,
			min_impurity_decrease=7.4662124572421e-09,
			min_samples_leaf=5, min_samples_split=3,
			n_estimators=235, random_state=123,
			subsample=0.20035816613002366)

Cleaning	Bayesian ridge regression	0.701	 BayesianRidge BayesianRidge(alpha_1=0.0003301199549309903, alpha_2=1.0298517770894848e-09, compute_score=True, fit_intercept=False, lambda_1=0.9975973597393759, lambda_2=1.6796869754658542e-06)

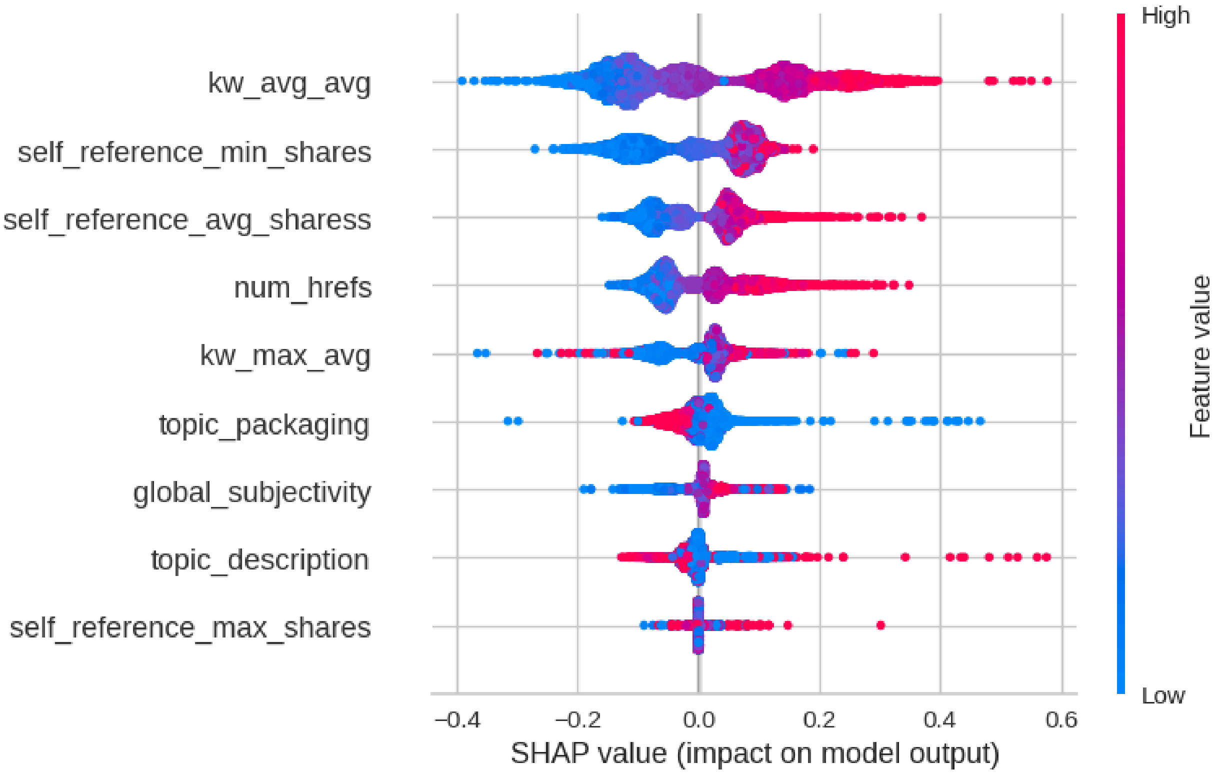
Other	Gradient boosting	0.8640	
			GradientBoostingRegressor
			GradientBoostingRegressor(learning_rate=0.1048779636865015, max_depth=1,
			max_features=0.42105782577735423,
			min_impurity_decrease=2.1815170651677707e-08,
			min_samples_leaf=3, min_samples_split=6,
			n_estimators=156, random_state=123,
			subsample=0.37195141927520237)

Selected Model

Our final model selection for prediction was the Gradient Boosting model from Strategy 3.1. This decision was driven by a comparative analysis of both accuracy and the model evaluation metric (MAE) across all strategies.

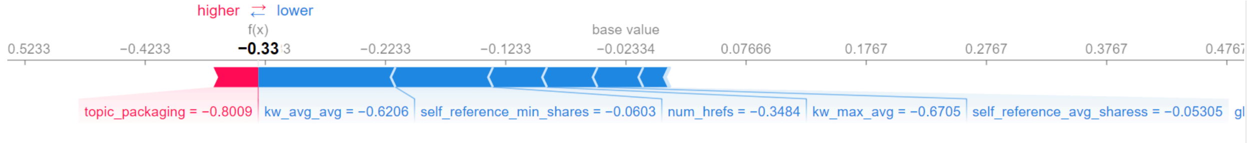
Model Interpretation

SHAP (Shapley Additive exPlanations) was employed to interpret the inner workings of our Gradient Boosting model. The resulting beeswarm plot depicts feature importance, with features ranked from highest (top) to lowest (bottom) SHAP value. This analysis reveals that "kw_avg_avg" is the most influential feature. Interestingly, "topic_description" and "topic_packaging" exhibit negative SHAP values, suggesting higher values for these features tend to decrease the predicted number of shares. Conversely, most other features contribute positively to the predicted number of shares.

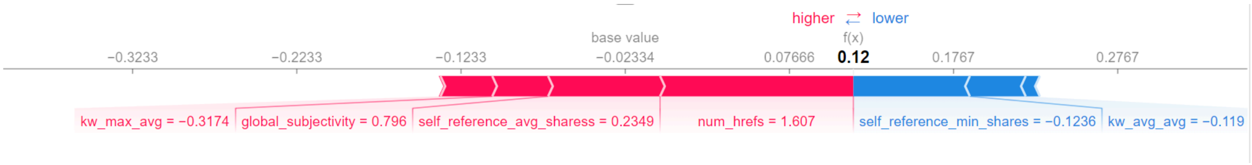


To further explore model behavior for specific cases, force plots were generated for two observations. The first observation (index #275) has a low share count (below 300). The corresponding force plot highlights that features like:
"kw_avg_avg",
"self_reference_min_shares",
"num_hrefs",
"kw_max_avg", and
"self_reference_avg_sharess"

have a negative influence, pushing the predicted share count to the right (lower values).



Conversely, the second observation (index #42) with a high share count (above 5,000) exhibits a contrasting pattern. Here, the same features
"kw_avg_avg",
"self_reference_min_shares",
"num_hrefs",
"kw_max_avg", and
"self_reference_avg_sharess"
show a positive influence, pushing the predicted share count to the left (higher values).



Conclusion

In conclusion, this report has demonstrated a comprehensive approach to predicting the number of shares a review will receive on Yojo.com using machine learning techniques. Through meticulous data preparation, exploratory data analysis, and strategic model selection and training, we have identified key factors that influence the shareability of product reviews. Our findings underscore the significance of strong keyword usage across various product categories, highlighting the importance of optimizing product titles and descriptions for increased visibility.

The application of Gradient Boosting, supplemented by association mining and correlation analysis, has proven effective in capturing the complex dynamics that contribute to a review's popularity. The model's performance, as evidenced by the lowest Mean Absolute Error (MAE), illustrates the robustness of our chosen approach. Moreover, our exploration into category-specific factors has offered tailored insights that can guide business and marketing strategies to enhance customer engagement and product reach.

By integrating SHAP values for model interpretation, we have gained deeper insights into the impact of individual features on the model's predictions. This not only enhances our understanding of the driving forces behind share performance but also provides a clear path for businesses to refine their strategies based on data-driven evidence.

Business Insights and Strategies for Yojo:

The methodologies and findings presented in this report offer a valuable roadmap for leveraging data analytics to inform strategic decisions. Based on these findings, some strategies that can be implemented to improve product share performance are:

- **Keyword Research and Optimization:** Conduct thorough keyword research for all product categories. Focus on incorporating relevant keywords with high performance into product titles, descriptions, and metadata.
- **Content Quality and Self-Referencing:** Develop high-quality content (descriptions, images, videos) for all product categories. Implement self-referencing strategies to promote products within relevant contexts where applicable (e.g., entertainment genres, travel destinations, sports equipment categories).
- **Customer Reviews and Sentiment Analysis:** Encourage customer reviews and respond to both positive and negative feedback. Utilize sentiment analysis to understand customer preferences and identify areas for improvement. While the direct impact might be weak in some categories, fostering positive customer interactions can enhance brand perception.