Project Overview and Analysis Write-Up

Data

The dataset we are working with comprises Amazon reviews from various products and product categories. It includes feature variables such as if the reviewer is verified, the time that the review was posted, the text in the review, unique identifying IDs of the reviewer and the product, summary of the review, how many people found the review helpful, the image added into the review, product metadata, and product category. These features serve as independent variables in our analysis. Our main objective in this project is to construct a classification model for predicting the sentiment of a product review on Amazon based on text and other data provided, in this case our target variable comes from the overall product rating. Through this analysis, we aim to identify the key factors impacting customer satisfaction for Amazon products across different categories, helping enhance the overall user experience on the platform.

Methods

To conduct data preprocessing by performing one hot encoding on the 'category' feature in both the training and testing datasets. Since we felt that this variable was useful in modeling the overall rating, we continued with converting the categorical 'category' feature into a set of binary columns. Next, we converted the ‘overall’ column into a binary ‘rating’ variable. Ratings >3 were labeled as 1 and ratings <=3 were labeled as 0. Following that, we extracted the date-time features including 'hour,' 'weekday,' 'month,' and 'year' from the 'unixReviewTime' column in the training dataset. The ‘category’ column was mapped into numerical values using ‘category_mapping.’ We converted the ‘verified’ column to binary and removed stopwords from the 'reviewText' and 'summary' columns. We Lastly, we created 'combined_text' in both the training and testing datasets by concatenating the 'reviewText' and 'summary' columns to be used for text analysis in the following steps.

We allocated 80% of the data to the training set (X_train and y_train) and the remaining 20% to the test set (X_test and y_test). Next, we selected a group of features that we felt were most important in modeling the data. Our features chosen kept changing as we logically thought through whether or not a column will affect the price. After conducting multiple tests, we concluded using the most influential feature ‘combined_text’ as our selected parameter.

We selected a logistic regression model to perform binary classification on the Amazon review dataset. This model was trained on our training dataset. Our primary evaluation metric for this classification task was the F1-Score, which is important for assessing the model's ability to correctly classify positive and negative sentiments in reviews. A higher F1-Score indicates a more accurate model. While we did not explicitly perform hyperparameter tuning in the provided code, we experimented and found the feature of ‘combined_text’ to be the best in the training of our model. We used a CountVectorizer to transform the 'combined_text' feature to find the most used words in the summary and trained the model on the transformed text data and evaluated its performance. We transformed the text data with TF-IDF vectorization and then applied logistic regression for classification. We used Grid Search for hyperparameter tuning to find the best hyperparameters.

For our second model, we used a Random Forest classifier for binary classification. This model was trained on the same training dataset and used the same features as the previous model. The Random Forest model combines multiple decision trees to make predictions. While hyperparameter tuning was again not performed, we played with different features to identify the most informative ones for sentiment prediction.

Similar to the logistic regression, we used a CountVectorizer for text feature transformation and evaluated its performance.

Our third model was an XGBoost classifier trained on the training dataset with the same features used for the previous models. We again used a CountVectorizer for text feature transformation.

Results

Part 1:

Model	ROC_AUC	Macro F1	Accuracy	Confusion Matrix
Logistic Regression	0.94295	0.86716		[[3264 292] [477 1805]]
Random Forest	0.92952	0.82983		[[3385 171] [786 1496]]
XGBoost	0.926375	0.84889		[[3242 314] [558 1724]]
Decision Trees	0.7611	0.77093		[[2859 697] [643 1639]]

Part 2 (Sentiment only):

Model	ROC_AUC	Macro F1	Accuracy	Confusion Matrix
Logistic Regression	0.77625	0.71090	0.7159	[[2901 655] [1003 1279]]
Random Forest	0.74435	0.6961	0.6992	[[2786 770] [986 1296]]
XGBoost	0.7788	0.72058	0.7228	[[2837 719] [899 1383]]
Decision Trees	0.6504	0.6654	0.66700	[[2639 917] [1027 1255]]

Part 2 (Text + Sentiment)

Model	ROC_AUC	Macro F1	Accuracy	Confusion Matrix
Logistic Regression	0.9254	0.84322	0.8446	[[3201 355] [552 1730]]
Random Forest	0.9066	0.80778	0.81466	[[3317 239] [843 1439]]
XGBoost	0.9116	0.8265	0.8280	[[3150 406] [598 1684]]
Decision Trees	0.7369	0.748097	0.74768	[[2798 758]

				[715 1567]]

These metrics are appropriate for this task because they help understand the model's accuracy and effectiveness. A high ROC AUC tells us that the model can tell the difference between positive and negative sentiments. The F1 score lets us evaluate the model's performance for positive and negative sentiments. The accuracy is important for binary classification tasks because it allows us to measure the model's likelihood of classifying the correct sentiment. Lastly, the confusion matrix helps us evaluate the performance of a classification model.

Error Analysis - Sentiment Analysis Model

By solely relying on the VADER model, we obtain three basic interpretations: neutral, negative, and positive sentiment. However, the VADER model lacks the ability to provide the intensity or frequency of words used within the sentences. This becomes a significant limitation when solely using the VADER model. For instance, consider an instance where a word like "unexpected" or "expectations" appears repeatedly, such as in ID 117. While the VADER model calculates the sentiment of these words, incorporating TF-IDF (Term Frequency-Inverse Document Frequency) provides further insight into the ultimate classification of the review. TF-IDF allows us to understand the significance and relevance of words within the context, which can enhance the accuracy of the final sentiment classification.

Combining both the TF-IDF and sentiment analysis approaches allows the model to gain insights from various perspectives within the text and data. TF-IDF assists in understanding the importance and relevance of words by considering their frequency and significance within the context. On the other hand, sentiment analysis, such as VADER, provides an initial understanding of the general sentiment. When used together, these approaches provide the model with multiple angles and a richer understanding of the text, which generally enhances the performance by capturing both the semantic meaning and the context in which words are used.

Future Work

We would likely try to use a different sentiment analysis model that would be able to understand the relevance of words and their significance within certain contexts.