Data Mining Final Project CSCI 6366

Team Name: The Roasters

Team Members:
Lesli Perez
Jennifer Pedraza
Rebecca Fernandez
Ashley Gomez

May 09, 2024
CSCI 6366

Final Report

In preparing our dataset for analysis, several preprocessing steps were employed. Firstly, null values were eliminated to ensure data integrity. Following this, an 90/10 split was applied for cross-validation purposes. Subsequently, punctuation was removed and all text was converted to lowercase to standardize the review data. Punctuation and uppercase letters caused the most frequent words to be duplicated, so they needed to be removed. Stop words, including custom ones, were then removed to focus on relevant content. The custom words were manually filtered to remove words that we noticed didn't offer any value or interpretation, such as "cup" and "notes". Additionally, the unique identifier attribute "coffee ID" was dropped from consideration because each ID is individual and was not valuable information for our model. Additionally, positive and negative words were extracted from the reviews for the sentiment analysis, which will be discussed in subsequent paragraphs. Finally, the review column was dropped post-sentiment analysis from the training, validation, and testing datasets to streamline the analysis process.

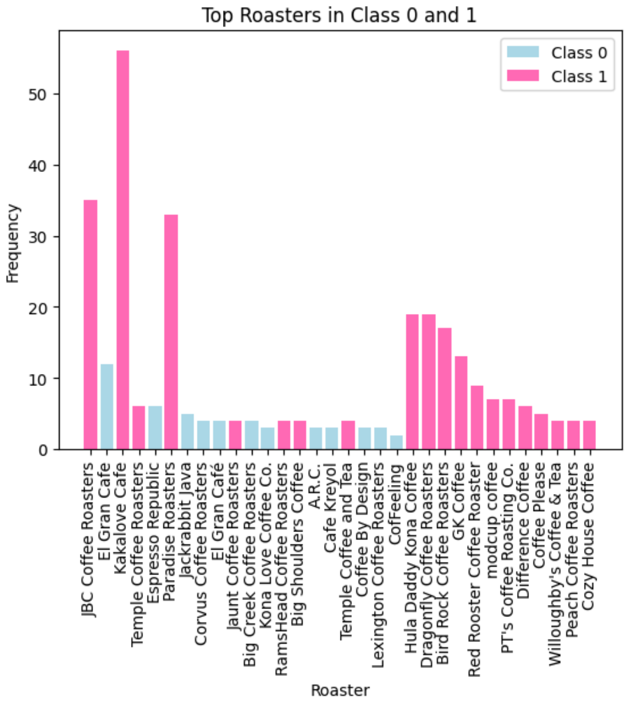

Figure 1.) *Frequency of Top Roasters in Class 1 and Class 0*

Methodology and Implementation

We began our analysis by preprocessing the coffee dataset. This involved several steps:

- Dropping Coffee ID: Initially, we removed the Coffee ID attribute from our dataset. Despite this removal, the model's accuracy remained consistent, indicating that the Coffee ID attribute did not contribute significantly to the predictive performance.
- Converting Categorical Variables: Next, we addressed any remaining categorical variables in the dataset by converting them into dummy variables. This transformation ensured that the categorical attributes were suitable for consumption by machine learning algorithms.
- Sentiment Analysis: We conducted sentiment analysis on the descriptive data associated with coffee reviews. This allowed us to extract sentiment-related features that significantly distinguished between the two classes, which increased the predictive power of the model.
- Frequency Analysis: Furthermore, we performed frequency analysis on certain attributes to gain insights on their distributions and identify any patterns or anomalies that could impact our modeling approach.
- Evaluation and Cross-Validation.

To evaluate and cross-validate the performance of our models, we employed the following strategies:

- Splitting Data: We split the dataset into training and validation sets using three different ratios: an 80-20 split, a 70-30 split, and a 90-10 split.
- F1 Score Calculation: We calculated the F1 score for both the training and validation sets. The F1 score (i.e., “harmonic mean of precision and recall” [Arize AI]) provided a valid measure of model performance.
- Confusion Matrix Analysis: Additionally, we analyzed confusion matrices to gain insights into the model's performance in terms of true positives, false positives, true negatives, and false negatives (See ***Figure 4***).

Our experimental results are summarized as follows:

Attribute	**Split**	**Accuracy**	**F1 Score**
Removed Coffee ID	80-20	0.7419	0.7479
Removed Coffee ID	90-10	0.7419	0.7536
Removed Coffee ID	70-30	0.7204	0.7267

Table 1.) *Accuracy and F1 Scores Across Different Cross-Validation Split Ratios*

Despite the relatively consistent accuracy results across different split ratios, we observed a slight improvement in the F1 score when transitioning from an 80-20 split to a 90-10 split compared to the transition to 70-30 split . This suggests that the latter split ratio may provide a better balance between the training and validation datasets, resulting in improved model performance.

Insightful Findings

Our team generated word cloud diagrams to showcase the most frequently occurring words within the "Review" attribute of Class 1 (i.e., positive) and Class 0 (i.e., negative) (See ***Figures 2 and 3***). The diagrams present a clear visual summary of flavors, aromas, and aesthetics that are preferred by consumers in their coffee. As a Coffee Roaster, the outcome of specific descriptive words can inform marketing strategies, such as initiating adjustments to menu offerings to better align to consumer needs, while also considering the removal of products correlated with average descriptives. For example, a product development opportunity could be the introduction of star, jasmine-like flavors in future coffee options.

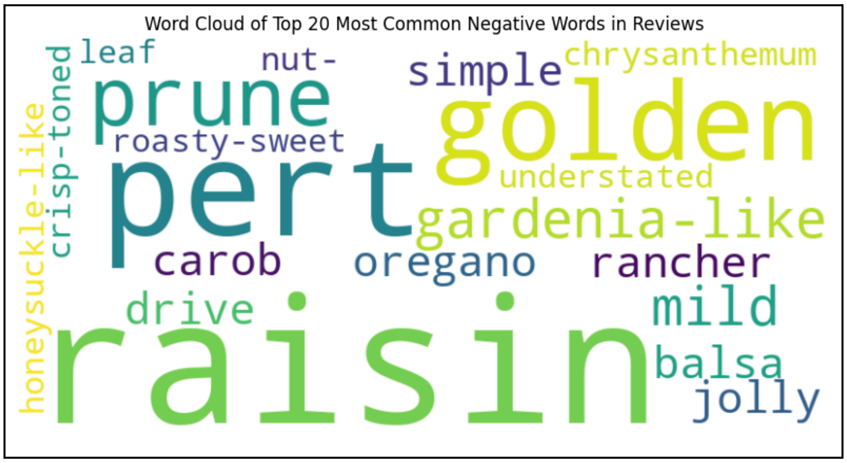

Figure 2.) *Word Cloud Representing the Negative Words Associated with Class 0*

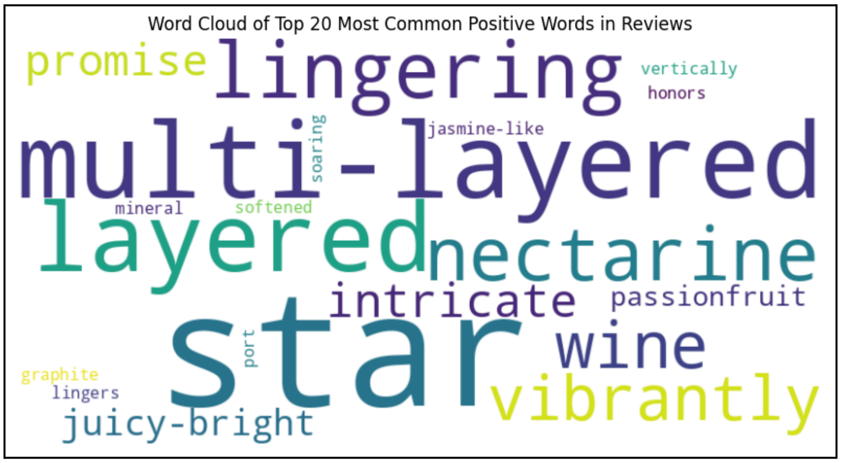

Figure 3.) *Word Cloud Representing the Positive Words Associated with Class 1*

Lastly, our team conducted a prediction accuracy analysis for each class based on the resulting confusion matrix (See ***Figure 4***). The model performed better predicting an outstanding rating, achieving an accuracy of 82.79%. However, the model only achieved an accuracy of 51.56% when predicting an average rating (i.e the model will mislabel an "average" coffee as "outstanding" 48.44% of the time).

From a consumer perspective, placing more reliance on outstanding ratings over average ones can help mitigate disappointment. With the significant mislabeling of Class 0, consumers are also more likely to be satisfied in the event an average coffee turns out to be outstanding. From a Coffee Roaster's perspective, the accuracy of outstanding predictions is equally important. Consistently delivering outstanding coffee products but receiving an average rating has the potential to distort consumer perception and negatively impact their revenue stream. A similar outcome may result if the business fails to meet customer expectations for "outstanding" products while serving "average" coffee products.

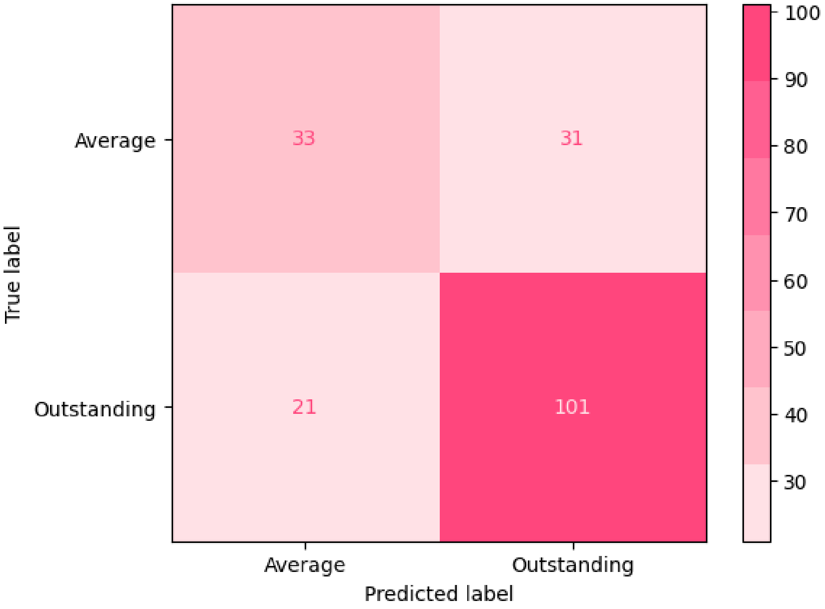

Figure 4.) *Confusion Matrix Displaying True Average(TA), False Average(FA) , True Outstanding(TO) and False Outstanding(FO)*

Challenges in Training

One of the key challenges we encountered was effectively preprocessing the data to extract meaningful insights. Initially, we experimented with various methods, some successful and others less so. Our first attempt involved breaking down the review column by separating sentences, which required removing punctuation and standardizing the text to lowercase. However, this approach revealed many irrelevant words, including filler words and common terms present in both positive and negative reviews. To address this, we refined our strategy by implementing sentiment analysis and converting the results into Boolean variables, enabling us to filter out words shared across different sentiment classes. Additionally, we faced difficulties with the model's constraints on string categories, such as roaster names. We implemented dummy variables to avoid any more issues with this problem. Initially, we applied these variables after the 90/10 split, inadvertently resulting in an imbalance within the training data. Therefore, we revised our approach, implementing the dummy variables before the split to ensure a more balanced distribution of the data.

Group Contributions

All tasks were evenly distributed among all group members. Each member contributed to both the coding and report writing phases at the same time. Throughout the project, we collectively indulged in coffee to immerse ourselves in the tasks and to enhance our focus on the project. <3

References

Awan, Abid Ali, and Avinash Navlani. “Naive Bayes Classifier Tutorial: With Python Scikit-Learn.” *DataCamp*, DataCamp, 3 Mar. 2023, www.datacamp.com/tutorial/naive-bayes-scikit-learn?irclickid=WXVSfGSShxyPUSG21BzY-wwRUkHTMs23xy15VU0&irgwc=1&utm_medium=affiliate&utm_source=impact&utm_campaign=000000_1-2003851_2-mix_3-all_4-na_5-na_6-na_7-mp_8-affl-ip_9-na_10-bau_11-Bing+Rebates+by+Microsoft&utm_content=BANNER&utm_term=EdgeBingFlow.

prashant111. “Naive Bayes Classifier in Python.” *Kaggle*, Kaggle, 28 Aug. 2020, www.kaggle.com/code/prashant111/naive-bayes-classifier-in-python.

“NLTK Documentation.” *NLTK*, www.nltk.org/howto/sentiment.html. Accessed 8 May 2024.

Savio, Arulius. “Mastering Pandas Get_dummies(): A Guide for Python Users.” *AskPython*, 18 Feb. 2023, www.askpython.com/python-modules/pandas/pandas-get-dummies.

Ali, Moez. “NLTK Sentiment Analysis Tutorial: Text Mining & Analysis in Python.” *DataCamp*, DataCamp, 23 Mar. 2023, www.datacamp.com/tutorial/text-analytics-beginners-nltk.

“How to Count Distinct Values of a Pandas Dataframe Column?” *GeeksforGeeks*, GeeksforGeeks, 1 Dec. 2023, www.geeksforgeeks.org/how-to-count-distinct-values-of-a-pandas-dataframe-column/.

“Sklearn.Metrics.F1_score.” *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html. Accessed 8 May 2024.

“Understanding and Applying F1 Score: Ai Evaluation Essentials with Hands-on Coding Example.” *Arize AI*, 8 Apr. 2024, arize.com/blog-course/f1-score/#:~:text=F1%20score%20is%20a%20measure,can%20be%20modified%20into%20F0.