Sentiment Analysis Report

The data

The dataset used in this python programme is named ‘amazon_product_reviews.csv’. This dataset provides a rich source of information that allows us to conduct a detailed sentiment analysis. By examining various aspects such as the textual content of the reviews, the ratings, and the reviewer’s recommendation status, we can gain insights into customer opinions towards different products.
In this report, we will analyse the sentiment of customer reviews for Amazon products that were produced using several methods. These include preprocessing to ensure the data is clean and ready for analysis, and employing logistic regression as our machine learning model for sentiment analysis.

The preprocessing stage

In the preprocessing stage of our analysis, we undertake several steps to clean and prepare the data for sentiment analysis. We begin by loading the data, the CSV file containing the Amazon product reviews is loaded using pd.read_csv with low_memory=False to prevent DtypeWarning during the loading of large files. We then remove any leading or trailing whitespaces in the column names to ensure consistent and error-free columns. To ensure that the dataset is free of missing data we drop NaN values specifically from 'reviews.text' and ‘reviews.rating’. To initialise the Tokenizers and Stemmer we use the English Tokenizers from SpaCy and the NLTK Porter Stemmer. A custom tokenisation function is defined to handle punctuation removal (text is tokenised and punctuation removed) and lemmatization (tokens are lemmatised to reduce words to their base form).
To preprocess the reviews, each review is then processed by the custom tokenization function, followed by the stemming of the lemmatized tokens. The processed tokens are then joined back into cleaned review text.
Lastly, a few examples of the preprocessed reviews are printed to verify the cleaning process. This detailed preprocessing ensures that the text data is properly cleaned and transformed, making it suitable for further sentiment analysis and model training.

The evaluation of the model

Acurracy represents the proportion of correctly classified instances among all instances in the test set. In this case the accuracy is approximately 72.2%. This indicates that the model correctly predicted the sentiment rating for about 72.2% of the reviews in the test set. While accuracy provides a general sense of the model’s performance, it can be misleading in cases where class distribution is imbalanced. Therefore, its important to also consider other metrics like precision, recall, and F1-score to get a complete picture of the model’s performance.

The classification report

The classification report breaks down the model's performance for each sentiment rating class (1.0, 2.0, 3.0, 4.0, and 5.0) using four key metrics: precision, recall, F1-score, and support.

Precision is the ratio of correctly predicted positive observations to the total predicted positives. High precision indicates that the model has a low false positive rate.

- Class 1.0: 0.53
- Class 2.0: 0.50
- Class 3.0: 0.33
- Class 4.0: 0.54
- Class 5.0: 0.76

Recall, also known as sensitivity, is the ratio of correctly predicted positive observations to all observations in the actual class. It answers the question: "Of all instances that truly belong to a given class, how many were correctly predicted?" High recall indicates that the model has a low false negative rate.

- Class 1.0: 0.11
- Class 2.0: 0.01
- Class 3.0: 0.10
- Class 4.0: 0.29
- Class 5.0: 0.94

F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is useful when you need a single metric to evaluate the model's performance, especially when you have an uneven class distribution.

- Class 1.0: 0.19
- Class 2.0: 0.03
- Class 3.0: 0.16
- Class 4.0: 0.38
- Class 5.0: 0.84

Support is the number of actual occurrences of each class in the test set. It provides context for interpreting the precision, recall, and F1-score. For instance, a high F1-score in a class with low support might be less impressive than a similar F1-score in a class with high support.

- Class 1.0: 80
- Class 2.0: 77
- Class 3.0: 278
- Class 4.0: 1756
- Class 5.0: 4735

In conclusion, from the classification report, the model performs best on Class 5.0 (rating 5 stars). With high precision (0.76) and F1-score (0.84), these scores indicate that the model is very good at correctly identifying positive reviews. On the other hand, for classes 1.0, 2.0, and 3.0 (ratings 1, 2, and 3 stars), the model performs poorly, with very low recall (0.11, 0.01, and 0.10 respectively) and correspondingly low F1-scores (0.19, 0.03, and

0.16). This suggests that the model struggles to correctly identify negative and neutral reviews. For class 4.0 (rating 4 stars), the model's performance is moderate, with a precision of 0.54, recall of 0.29, and an F1-score of 0.38. This indicates some difficulty in distinguishing this class from others, but it performs better than for the lower ratings.

Overall Evaluation

The classification report provides a detailed breakdown of the model's performance for each sentiment rating class, allowing for a comprehensive evaluation of its predictive capabilities. The macro average (averaged across all classes) and weighted average (weighted by the number of true instances for each class) provide additional insights:

- Macro avg: Precision (0.53), recall (0.29), and F1-score (0.32) indicate that, on average, the model performs moderately but is skewed by the poor performance on the less frequent classes.
- Weighted avg: Precision (0.68), recall (0.72), and F1-score (0.68) show a more optimistic view, heavily influenced by the dominant 5-star class.

This detailed analysis highlights the strengths and weaknesses of the model, showing its effectiveness in identifying positive reviews but also indicating areas for improvement, especially in correctly predicting negative and neutral sentiments.