5. Write a brief report or summary

5.1. A description of the dataset used

This dataset provides basic information about various products available on Amazon, along with associated reviews and details. The dataset includes various attributes such as unique identifiers (IDs), product names, brands, categories, manufacturer details, ASINs (Amazon Standard Identification Numbers, and URLs of images associated with the products. Additionally, the dataset contains information about customer reviews, including dates when the reviews were posted, the rating given by the reviewer, and review text (words or sentences giving insight on their opinions of the product). These reviews also include helpfulness, recommendations, review titles, and usernames of the reviewers. Though the dimensions of this dataset being 28,332 rows and 24 columns indicate a coprehenive view of product listing and customer feedback on amazon, this dataset is not without its faults due to numerous missing entries within columns involving whether the reviewer purchased, recommendations, review id, helpfulness, and review username. However, these missing entries did not impact the current analysis conducted on this dataset.

5.2. Details of the preprocessing steps

When defining the preprocessing function (Step 3), a text input is taken and tokenized using spaCy. Each token is then lemmatized to its base form, converted to lowercase, and joined back into a single string. The function also removes stopwords (commonly occurring words with little to no meaning such as 'the', 'is', 'and') and punctuation marks. However, upon reflection sometimes punctuation is used in reviews to further express how good or bad a product is and I am uncertain how this would translate when it comes to NLP.

I created a copy of the dataframe, containing all of the manipulations that I was performing, selecting only the 'reviews.text' and 'reviews.rating' columns (as these were the columns I was performing semantic analysis on). I then dropped all rows with missing values as instructed (Step 4). Next, I subset a sample of 5000 rows (due to the size of the dataset) and set the seed to ensure replicability. I then applied the preprocess function 'preprocess' to the 'reviews.text' column, creating a new column called 'processed_text' containing the preprocessed text (Step 5). These preprocessing steps help to clean and normalise the data before performing sentiment analysis.

5.3. Evaluation of results

I performed semantic analysis using TextBlob, which provided a polarity score ranging from -1 to 1. The polarity was then converted to sentiment labels (positive, negative, and neutral) using the 'polarity_to_sentiment' function. This was done using if, elif, else, statements and a threshold of 0. Initially a threshold of 0.33 and -0.33 was used as I thought that by ensuring that each label had the same range that this would result in higher accuracy but that was not the case. Reasons for this are unclear but it could have been due to dataset characteristics (it is possible that the majority of the sentiment in the amazon reviews dataset was clearly positive or negative). Another possible reason could be that the binary classification with a threshold of 0 simplifies the sentiment anlysis, making it easier to implement and interpret.

Additonally, the review ratings were also converted to sentiment categories based on thresholds associated with the ratings system used by amazon: ratings >= 4 as 'positive', ratings <= 2 as 'negative', and the rest as 'neutral'.

The ‘predicted_sentiment’ column contains the sentiment lables predicted by the TextBlob polarity score conversion and the ‘actual_sentiment’ column contains the sentiment lables dervied from the review ratings. The comparion between these two columns provides a better understanding on how accurate this method of natural lanuage processing (NLP) is when it comes determining whether what a reviewer states reflects their rating.

Accuracy was calculated by comparing the predicted sentiment with the actual sentiment and counting the number of correct predicitions. The accuracy of the sentiment analysis model was found to be approximatel 79.34%. This is reasonable but there is definitely room for improvement which could possibly be achieved by experimenting with other sentiment analysis techniques or even different machine learning models.

5.4. Insights into the model’s strengths and limitations

One strength of the sentiment analysis model is efficient preprocessing. The preprocessing steps, including tokenization, lemmatization, and remove of stopwords and punctuation are, are efficiently implemented using spacy. This ensures the review text is appropriately cleaned and standardised before sentiment analysis. Another strength is that TextBlob is a fairly simple and beginner friendly library for performing sentiment analysis. It provides a polarity score, indicating the sentiment of the text, which simplifies the sentiment analysis process. A third strength is the handling of missing values. Whilst it was not necessary in this case, my code handles missing values in the dataset by dropping rows with missing values for the relevant columns (reviews.text and reviews.rating). This ensures that only completed data is used for sentiment analysis.

In my case one limitation is the limited dataset size. The model is trained and evaluated on a subset of 5000 samples (out of over 28,000) from the dataset. While this improved computational efficiency, it is likely that it would have reduced the model’s accuracy and generalisability. Using the entire dataset may improve this. Another limitation is the simplicity when it comes to sentiment labelling. The model categorises sentiment into three broad categories (positive, negative, neutral) based on polarity scores or review ratings. This simplistic approach may overlook nuances in sentiment expressing, resulting in misclassifications, especially for ambiguous reviews. A third limitation is the reliance on predefined thresholds. The model relies on predefined thresholds (e.g. polarity score thresholds for positive, negative, and neutral sentiments) for sentiment labeling (something that I was struggling with as previously mentioned). These thresholds are a personal choice and may not generalise well across different datasets or domains.