Sentiment Analysis Report

5.1. A description of the data set used.

The data set used is a set of Amazon product reviews from May 2017 to May 2018.

It contains 5000 rows of data and 24 columns including information such as the date of the review, the product rating, the review text and the review username.

The datatypes used are bool, float64, int64 and object, and it uses 903.4+KB.

5.2. Details of the preprocessing steps.

1. Select relevant columns from dataset
2. Remove null values
3. Define preprocessing function that:
    a. Tokenizes the input
    b. Drops stopwords and punctuation
    c. Lemmatizes the tokens
    d. Returns string of processed tokens
4. Add these processed reviews to the data frame
5. Define a function for polarity analysis that:
    a. Uses textblob to analyse sentiment
    b. Returns polarity
6. Assigned sentiment of positive, neutral or negative based on polarity using if statements and added to sentiments list.

5.3. Evaluation of results.

To test the model, I selected two sample reviews, using the length of the sentiments list and the random module to find two random indices within range.

Sample 1

Text: *“this tablet is great for reading the text is not a good as it is on my tab s2 9.7 in but this is about $200 cheaper i got it because i wanted a more natural book feel reading my books and the tab s2 was not doing it for me*

Polarity: 0.26

Sentiment: Positive

Sample 2

Text: *“Be sure to join Amazon Prime and have access to lots of free reading downloads”*

Polarity: 0.45

Sentiment: Positive

Similarity between Samples 1 and 2: 0.67

Notes:

1. Both reviews are positive
2. Sample 2 is more positive than Sample 1
3. Sample 1 comments positively on the price, the natural feel and says it is ‘great for reading’, but also says it is not as good as another product.
4. Sample 2 only advises to join Amazon Prime and mentions ‘free reading downloads’.
5. Sample 2 does not necessarily appear to be more positive than Sample 1, and doesn’t comment on the product itself.
6. In Sample 1, it is possible that the negativity about the other product has impacted the polarity result.

Both samples are mostly positive and the results given by the program are not incorrect, but they are not very useful. Sample 2 cannot really be considered a positive review of the product as it does not mention the product and to a human, Sample 1 does not seem significantly less positive than Sample 2. Using these results as they are would give a vague overview of the sentiment of the reviews but very little insight or actionable information.

The similarity between the two samples is fairly high, at 0.67. However, as all reviews of a product are likely to be broadly similar in content this doesn’t seem useful.

5.4. Insights into the model's strengths and limitations.

This model is an effective way of determining the sentiment of the reviews. It is simple and provides a clear result. However, it has various limitations.

1. Not specific

In the current model, any review with polarity between 0.1 and 1 or -0.1 and -1 is labelled positive or negative respectively, and only reviews that equal 0 are considered neutral. When all the processed reviews are analysed, the ratios are:

Positive: 88.65%
Neutral: 6.18%
Negative: 5.17%

The vast majority of the reviews are considered positive and it may be useful to divide these up. If most of the positive reviews have a polarity of less than 0.5 they would be significantly less positive than if most of them have a polarity of over 0.5 - adding further categories could help identify subsets to analyse for further information. As an example, you could use:

```
if polarity > 0.8:
        sentiment = very positive
if polarity  > 0.5:
        sentiment = positive
if polarity > 0.2:
        sentiment = somewhat positive
if polarity >= 0:
        sentiment = neutral
if polarity < 0:
        sentiment = negative
```

When the above is implemented, the results are:

Very positive: 1.72%
Positive: 22.6%
Somewhat positive: 50.34%
Neutral: 20.10%
Negative: 5.17%

Now the majority of reviews are ‘somewhat positive’ - this indicates that although many customers like the product, they have suggestions for improvements. Reviews like this would be particularly well suited for analysis to inform product development.

2. Unable to judge nuance

There are many nuances to human language and a lot of variance in how individuals express themselves, which could lead to the program misjudging the sentiment. The dataset includes the user's star ratings of the products and so to improve the accuracy of the program, these could also be taken into account. To do so, you could give weightings to the polarity and the star rating to determine an overall sentiment, like below:

```
# Change stars to a decimal value by dividing by 5.
star_rating = stars/5

# Gives equal weighting to polarity and star_rating, calculates sentiment.
overall_sentiment = (polarity * 0.5) + (star_rating * 0.5)
```

If the polarity is .9 (very positive) but the user has given ⅗ stars (somewhat positive), the overall_sentiment would be 0.75 (positive), which is likely to be a more accurate assessment.

Another limitation of the program is its inability to judge things like sarcasm or humour. A way to limit the impact of this could be to flag any reviews with a significant difference between star rating and polarity (eg. a review with a star rating of 1 and a polarity of .95) for human review, or disregard them entirely.

3. Misleading results

The most significant limitation is that if a review had multiple distinct sentiments in it, this model would not provide useful results. For example, if the review was:

"The product is amazing, I love the way it looks and it works perfectly. However, the customer service was appalling and I will never use this company again!"

The polarity would most likely balance out to give a somewhat neutral result (or, in the program as it is currently written, it could give a clear positive/negative). No part of the review is neutral so that result would be very inaccurate, but at the same time the review could not be considered either positive or negative in its entirety.

It would be more useful if the model was able to identify the different parts of the review and analyse them separately, returning the overall sentiment and other results such as:

```
product_sentiment = 'positive'
cust_service_sentiment = 'negative'
delivery_sentiment = x, subscription_sentiment = x, etc.
```

By analysing chunks of the reviews like so, it would be possible to further develop the program to return information such as specific aspects of the product that are frequently mentioned in positive or negative reviews, or the overall sentiment towards the product itself vs the subscription service. This data could inform the company on areas for improvement and potential avenues for growth.