Documentation for sentiment_analysis.py

5.1. A description of the dataset used.

The dataset used: amazon_product_reviews.csv
The CSV file used contains information of online reviews that are stored as text and have the values separated by commas. Each review provides insights into the customer's experience with a specific product, including their opinions, sentiments, and ratings. The dataset can be used for various purposes, including:

- Sentiment Analysis: Analysing the sentiment of customer reviews to understand overall customer satisfaction and sentiment trends.
- Product Improvement: Identifying areas for product improvement based on customer feedback and complaints.
- Recommender Systems: Building recommender systems to suggest products to customers based on their preferences and past reviews.
- Market Research: Conducting market research to understand consumer behaviour, preferences, and trends in various product categories.

5.2. Details of the preprocessing steps.

Summary of the preprocessing steps taken:

1. Loading the Dataset: The dataset of Amazon customer reviews is loaded from a CSV file using the `pd.read_csv()` function from the Pandas library.

2. Text Preprocessing:
   - Tokenisation: The review text is tokenised using the spaCy model (`nlp`) to split the text into individual words or tokens.
   - Stopword Removal: Stopwords (commonly occurring words like 'the', 'is', 'and') are removed from the tokenised text using the spaCy stop words list (`STOP_WORDS`).
   - Punctuation Removal: Punctuation marks (e.g., '.', ',', '!') are removed from the tokenised text using the `token.is_punct` attribute.

3. Cleaning and Normalisation:
   - Lowercasing: All words in the tokenised text are converted to lowercase to ensure consistency in word matching and analysis.
   - Joining Tokens: The processed tokens are joined back into a single string, creating the cleaned and normalised review text.

4. Sentiment Analysis:
   - The cleaned review text is passed to the TextBlob library to perform sentiment analysis.

- The sentiment polarity score is calculated using the `.sentiment.polarity` attribute of the TextBlob object, which represents the sentiment of the review as a numerical value between -1 (negative) and 1 (positive).
- Based on the polarity score, the sentiment of the review is classified as 'Positive', 'Negative', or 'Neutral'.

These preprocessing steps ensure that the text data is cleaned, normalised, and ready for further analysis, such as sentiment analysis. The resulting sentiment analysis helps in understanding the overall sentiment of customer reviews and extracting actionable insights from the dataset.

5.3. Evaluation of results.

- Sample Review 1:

Review: "I thought it would be as big as small paper but turn out to be just like my palm. I think it is too small to read on it... not very comfortable as regular Kindle. Would definitely recommend a paperwhite instead."
Sentiment: Negative
The review expresses dissatisfaction with the size of the Kindle device and discomfort in using it, ultimately recommending an alternative product. The negative sentiment is appropriately identified.

- Sample Review 2:

Review: "This kindle is light and easy to use especially at the beach!!!"
Sentiment: Positive
The review expresses satisfaction with the Kindle, highlighting its lightweight and ease of use, particularly in outdoor settings like the beach. The positive sentiment is correctly identified.

- Similarity between Review 1 and Review 2:

Similarity Score: 0.7609522586736197
The similarity score indicates a moderate level of similarity between the two reviews. This score suggests that there are some commonalities in the language used in both reviews, possibly related to the topic of Kindle devices, but they are not identical. It's important to note that similarity does not necessarily indicate sentiment agreement; in this case, although the reviews may discuss Kindle devices, their sentiments differ.

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

The sentiment analysis model implemented in the provided code has both strengths and limitations:

Strengths:

1. Ease of Use: The model is relatively easy to implement and understand, making it accessible for users with varying levels of expertise in natural language processing.

2. Accuracy for Simple Texts: The model performs well for simple and straightforward texts, accurately identifying sentiments in reviews that express clear positive or negative opinions.

3. Fast Processing: The model's processing speed is generally fast, allowing for quick sentiment analysis of large volumes of text data.

4. Interpretability: The sentiment analysis results are interpretable, with clear labels (positive, negative, neutral) assigned to each review based on its sentiment polarity score.

Limitations:

1. Limited Context Understanding: The model may struggle to accurately analyse sentiment in texts with complex or nuanced language, sarcasm, or ambiguity. It lacks the ability to understand context beyond individual words and phrases.

2. Dependency on Text Preprocessing: The accuracy of the model heavily relies on the effectiveness of text preprocessing techniques such as tokenisation, stopword removal, and punctuation removal. Inaccurate preprocessing may lead to biased sentiment analysis results.

3. Over Reliance on Polarity Score: The model's classification of sentiment based solely on polarity scores may oversimplify the analysis, potentially overlooking important contextual factors or subtle nuances in language that could affect the overall sentiment.

4. Handling Negations and Contradictions: The model may misinterpret sentiments in reviews containing negations or contradictory statements, as it does not consider the context in which words are used. For example, a review containing phrases like "not bad" or "didn't like" may be misclassified.

5. Scalability: While the model is efficient for processing moderate-sized datasets, its scalability to handle very large datasets may be limited, particularly when more complex analysis techniques or additional features are required.

Overall, while the sentiment analysis model has its strengths in simplicity, ease of use, and speed, it also has limitations related to its ability to understand context, handle complex language, and provide nuanced interpretations of sentiment. Users should be aware of these limitations and consider them when interpreting the results of sentiment analysis.