Sentiment Analysis Report

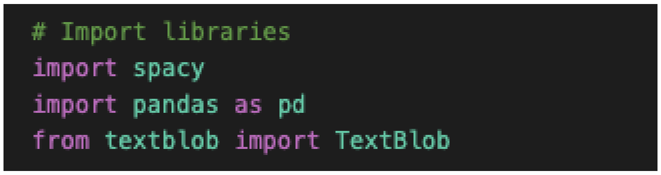
Description Of The Dataset

This dataset compiles over 34,000 consumer reviews for Amazon products. Additionally, the dataset includes basic product information such as ratings, review text, manufacturer and the sourceURLs of each review. It is to note that this dataset is a sample of a larger dataset. The variable types included within this dataset are; Categorical variables (nominal, binary), Numeric variables (discrete and continuous), Text variables, Temporal variables, Identifier variables and Image variables. This dataset is provided by Datafiniti where the full dataset can be found on their product database.

Preprocessing Steps

The preprocessing steps of this code aim to clean the data in order to prepare the Amazon Product Reviews dataset for sentiment analysis. The preprocessing can be segmented into these steps:

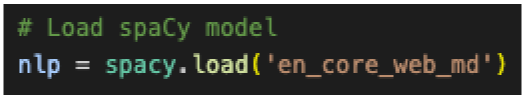
Importing Libraries



```
# Import libraries
import spacy
import pandas as pd
from textblob import TextBlob
```

First the code imports the libraries needed for the rest of the program. In this case spaCy for Natural Language Processing, pandas for data manipulation and TextBlob for sentiment analysis.

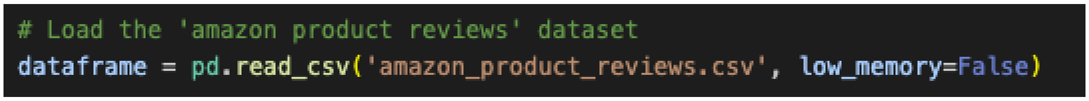
Loading the spaCy Model



```
# Load spaCy model
nlp = spacy.load('en_core_web_md')
```

The spaCy model provides the code with efficient processing of English text data with word vectors and this is through the loading of the ‘en_core_web_md’ model.

Loading the Amazon Product Reviews Dataset

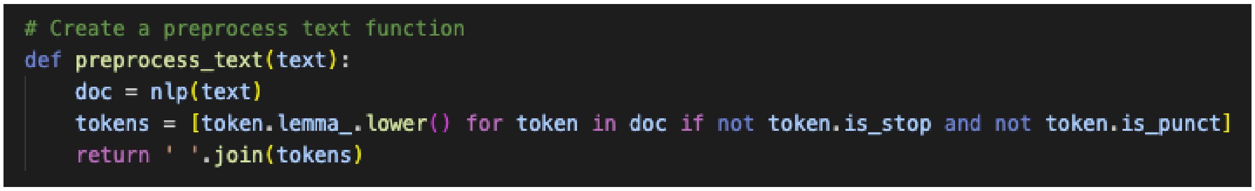


```
# Load the 'amazon product reviews' dataset
dataframe = pd.read_csv('amazon_product_reviews.csv', low_memory=False)
```

The program then loads the Amazon Product Reviews Dataset into a pandas dataframe and this is identified as ‘dataframe’.

Sentiment Analysis Report

Text Preprocessing Function

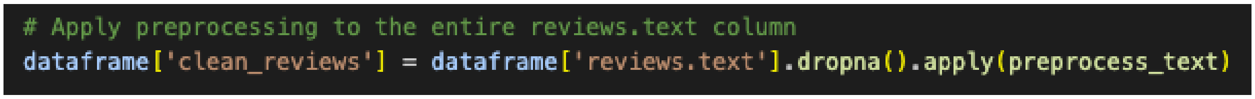


```
# Create a preprocess text function
def preprocess_text(text):
    doc = nlp(text)
    tokens = [token.lemma_.lower() for token in doc if not token.is_stop and not token.is_punct]
    return ' '.join(tokens)
```

The preprocess_text function is tailored to cleanse and prepare text data for subsequent analysis.

First the function accepts a single argument which is ‘text’ that represents the unprocessed text data. Then the code moves onto tokenization and lemmatization where the doc = nlp(text) calls on spaCy’s Natural Language Processing pipeline to process the input text. This is then followed by ‘.lemma_.lower’, ‘.is_stop’ and ‘.is_punct’ where ‘.lower’ standardises the text, ‘.is_stop’ checks if the token is a stop word and ‘if not token.is_punct’ will check whether the token is punctuation. Finally, ‘return ‘ ‘.join(tokens) takes the processed tokens and joins them into a single string and this string is now a cleaned and preprocessed input text.

Application of the Text Preprocessing Function to all reviews



```
# Apply preprocessing to the entire reviews.text column
dataframe['clean_reviews'] = dataframe['reviews.text'].dropna().apply(preprocess_text)
```

The code then applies the Text Preprocessing Function to all the Amazon Product Reviews within the reviews.text column to which ‘dataframe[‘reviews.text’]’ selects the column from the dataframe which holds the Amazon Product Text reviews. This is followed by ‘.dropna()’ which removes any rows that contain missing values and ‘.apply(preprocess_text)’ which is a function that processes each review individually.

Evaluation Of Results

The sentiment analysis through observation displays the correct alignment of sentiments with their respective reviews. Through closer inspection, we can see that positive sentiments are assigned to reviews that are generally happier in the sense that there is appreciation of the product. Moreover, we can see that with reviews that are more factual and do not display a positive or negative sentiment these were given a neutral sentiment. Finally, in terms of reviews that had negative connotations and displayed disappointing beliefs towards the product these reviews were given a negative sentiment.

Sentiment Analysis Report

The column 'clean_reviews' consists of text data that has undergone preprocessing, which involves the removal of stop words and punctuation, as well as the lemmatization of words. This preprocessing step serves to standardise the text data, making it more suitable for subsequent analysis.

The sentiment analysis results display the original reviews next to their corresponding predicted sentiments ‘Positive, Neutral and Negative’. Through observation, the majority of reviews had labeled sentiments of being positive but there is also a small proportion of reviews that were labeled as neutral and negative. This shows that the model captured a large range of sentiments within the Amazon Product Reviews dataset.

In terms of the Similarity Review of Two Amazon Product Reviews, the model successfully calculated a similarity score between the two randomly selected reviews through using spaCy which compared the word vectors of the two reviews.

The Model’s Strengths And Limitations

The strengths of this model are not limited to both the ease of use and efficiency of the program but there are also the advantages that this model uses pretrained models and has simplicity from utilising TextBlob. This model efficiently handles a moderate sized dataset which optimises spaCy’s capabilities and TextBlob’s functionality. In terms of using pretrained models, SpaCy's incorporation of pretrained word vectors across various languages facilitates precise linguistic annotations with little reliance on extensive training data or computational resources. This functionality broadens the model's applicability which therefore makes it more versatile. Continually in terms of the model’s robustness, as it uses spaCy it ensures more accurate sentiment analysis due to the fact that it enhances the robustness of text preprocessing tasks.

The limitations of this model are that of scalability and also whilst the model has strengths of using pretrained models, the dependency on these pretrained models are also a limitation. In this scenario we used a sample of a larger dataset that contains Amazon Product Reviews. However if we were to call on a larger dataset with more values then the model’s efficiency may decrease due to processing constraints. Moreover in terms of the dependency on pretrained models, the model highly depends on the quality of the models that it calls on. If the pretrained models lack coverage of particular linguistic patterns or contexts, the model's performance might be compromised.

In conclusion, the model exhibits strengths in ease of use, efficiency, and integration of pretrained models and TextBlob simplicity. It efficiently handles small to moderate-sized datasets, offering versatility in sentiment analysis tasks. However, limitations include scalability issues and dependency on pretrained models, which may impact performance. Despite this, the model presents a valuable solution for sentiment analysis with careful consideration of its constraints.