# Capstone Project - NLP Applications

The goal of this project is to categorise a dataset of Amazon reviews into three groups, identifying the sentiment of each review: Positive, Neutral, and Negative. To achieve this, I will use various preprocessing techniques to clean and prepare the data before using Natural Language Processing (NLP) to identify the sentiment expressed within each review and classify them.

## Dataset

The dataset used is a list of 28,332 consumer reviews for Amazon products such as the Kindle, Fire TV Stick, and more from Datafiniti's Product Database updated between February 2019 and April 2019. The dataset can be downloaded from Kaggle.com.

There are 24 data columns that use a mixture of float, integer and object data types. The majority of these columns are not relevant to the objective of this project and can be dropped in order to enable faster data analysis and processing.

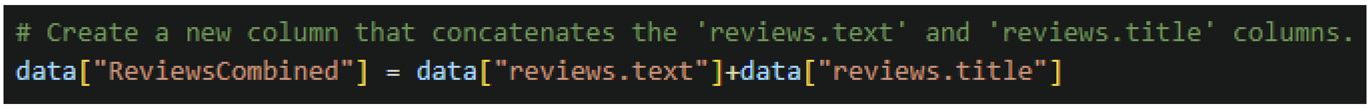
The dataset contains 65 unique entries in the ‘id’ and ‘name’ columns that indicates that there are 65 different products that the reviews relate to. There are also 3 unique entries in the ‘brand’ column, but one of these is only unique due to the case of a letter, so in fact there are only 2 unique brands in the dataset: ‘AmazonBasics’ and ‘ Amazon’.

The columns that have been identified as being relative to the objective are: ‘reviews.rating’, ‘reviews.text’, and ‘reviews.title’. The ‘reviews.text’ and ‘reviews.title’ columns will contain the text that the sentiment analysis model will analyse and the ‘reviews.rating’ column will be useful to compare against the results of the sentiment analysis in order to assess its accuracy.

## Preprocessing

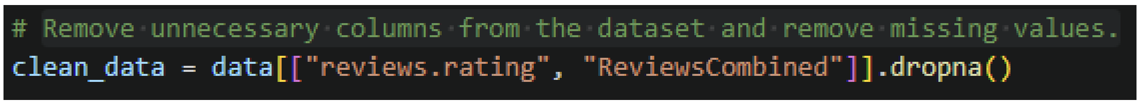
The goal of data preprocessing in sentiment analysis is to convert the dataset into a structured and clean format that can be readily fed into a sentiment classification model. Various techniques are employed during this preprocessing phase to extract meaningful features from the dataset while eliminating noise and irrelevant information. The ultimate objective is to enhance the performance and accuracy of the sentiment analysis model. Preprocessing techniques that have been used in this dataset include: combining columns, dropping columns, tokenisation, lemmatization, lowercasing, removing stop words, and removing punctuation.

**Create a new column by combining the ‘reviews.text’ and ‘reviews.title’ columns**.
This new column, named ‘ReviewsCombined’ brings together all the data needed to be used for sentiment analysis. Whilst the sentiment analysis could have been performed only on the ‘reviews.text’ column, the title of reviews often give an emotive summary of the review and I felt that it would produce more accurate results if they were included.



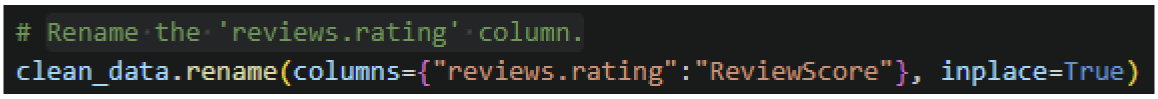
```
# Create a new column that concatenates the 'reviews.text' and 'reviews.title' columns.
data["ReviewsCombined"] = data["reviews.text"]+data["reviews.title"]
```

**Remove unnecessary columns from the dataset and remove missing values.**
Dropping all of the columns apart from ‘reviews.rating’ and the newly created ‘ReviewsCombined’ removes any noise from the dataset and will allow for faster data processing. The .dropna() method also removes any null values from the data, if any.



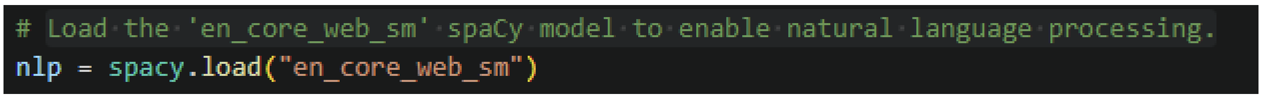
```python
# Remove unnecessary columns from the dataset and remove missing values.
clean_data = data[["reviews.rating", "ReviewsCombined"]].dropna()
```

**Rename the 'reviews.rating' column.** Renaming this to ‘ReviewScore’ is mostly presentational.



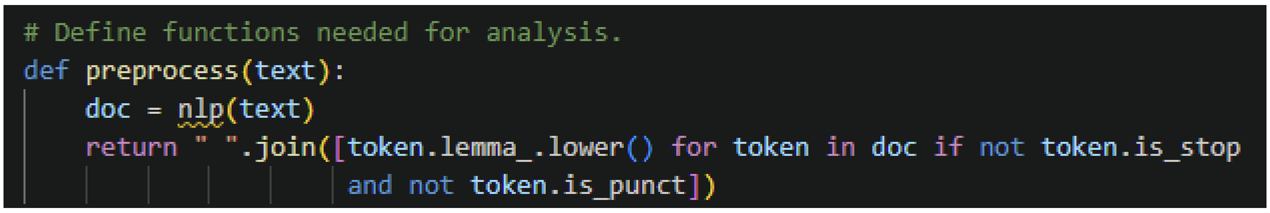
```python
# Rename the 'reviews.rating' column.
clean_data.rename(columns={"reviews.rating":"ReviewScore"}, inplace=True)
```

**Load the 'en_core_web_sm' spaCy model to enable natural language processing.**
This is the model that will analyse and classify the sentiment of the product reviews and needs to be loaded before any sentiment analysis can take place. For this project I am using the ‘small’ model as I am not using vectors and so the processing time will be reduced.



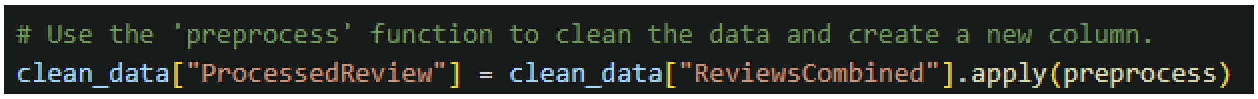
```python
# Load the 'en_core_web_sm' spaCy model to enable natural language processing.
nlp = spacy.load("en_core_web_sm")
```

**Create a function that deals with tokenisation, lemmantisation, lowercasing and removal of stop words and punctuation.**
The function ‘preprocess(text)’ removes spaces between words before tokenising them. These tokens are then lemmatised to change the inflections of a word to their base word. This will help the model produce more accurate results. All tokens are also transformed to lowercase, to remove duplication of words. Stop words (e.g. ‘the’, ‘and’, ‘is’) are removed as these are not relevant to the sentiment and can negatively affect accuracy. Lastly, all punctuation is removed to reduce noise as these are also not relevant to the sentiment.



```python
# Define functions needed for analysis.
def preprocess(text):
    doc = nlp(text)
    return " ".join([token.lemma_.lower() for token in doc if not token.is_stop
                     and not token.is_punct])
```

**Use the ‘preprocess(text)’ function to clean the data and create a new column.**
The tokens are stored in the newly created column, ‘ProcessedReview’. This is the data that will be used for sentiment analysis. The original, unprocessed column is not dropped so that it is possible to compare the two and determine how accurate the tokens are.



```python
# Use the 'preprocess' function to clean the data and create a new column.
clean_data["ProcessedReview"] = clean_data["ReviewsCombined"].apply(preprocess)
```

## <u>Processing</u>

**Create a function to calculate the subjectivity of a review.**
Using the sentiment.subjectivity method from the TextBlog library, the function ‘getSubjectivity(text)’ returns a score for each review between 0 and 1. The higher the score, the more subjective it is and the lower the score, the more objective it is. From these results we can determine if subjectivity is related to how a product is rated. The results are stored in a new column named ‘Subjectivity’.

```python
def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity
```

```python
# Use the 'getSubjectivity' function to apply scores to the 'ProcessedReview' data.
clean_data["Subjectivity"] = clean_data["ProcessedReview"].apply(getSubjectivity)
```

**Create a function to calculate the polarity of a review.**
Also using TextBlob, the function ‘getPolarity(text)’ uses sentiment.polarity to return a score for each review between -1 and 1:  -1 defines a negative sentiment and 1 defines a positive sentiment.  These are the scores that will be used to determine the sentiment of each review.  The results are stored in a new column named ‘Polarity’.

```python
def getPolarity(text):
    return TextBlob(text).sentiment.polarity
```

```python
# Use the 'getPolarity' function to apply scores to the 'ProcessedReview' data.
clean_data["Polarity"] = clean_data["ProcessedReview"].apply(getPolarity)
```

**Create a function to predict sentiment.**
The function ‘getSentiment(score)’ takes the polarity score for a review and classifies it.  A score below 0 is predicted to be negative and a score greater than 0 is predicted to be positive.  A score of 0 is classified as a neutral sentiment.  The results are stored in a new column named ‘Sentiment’.

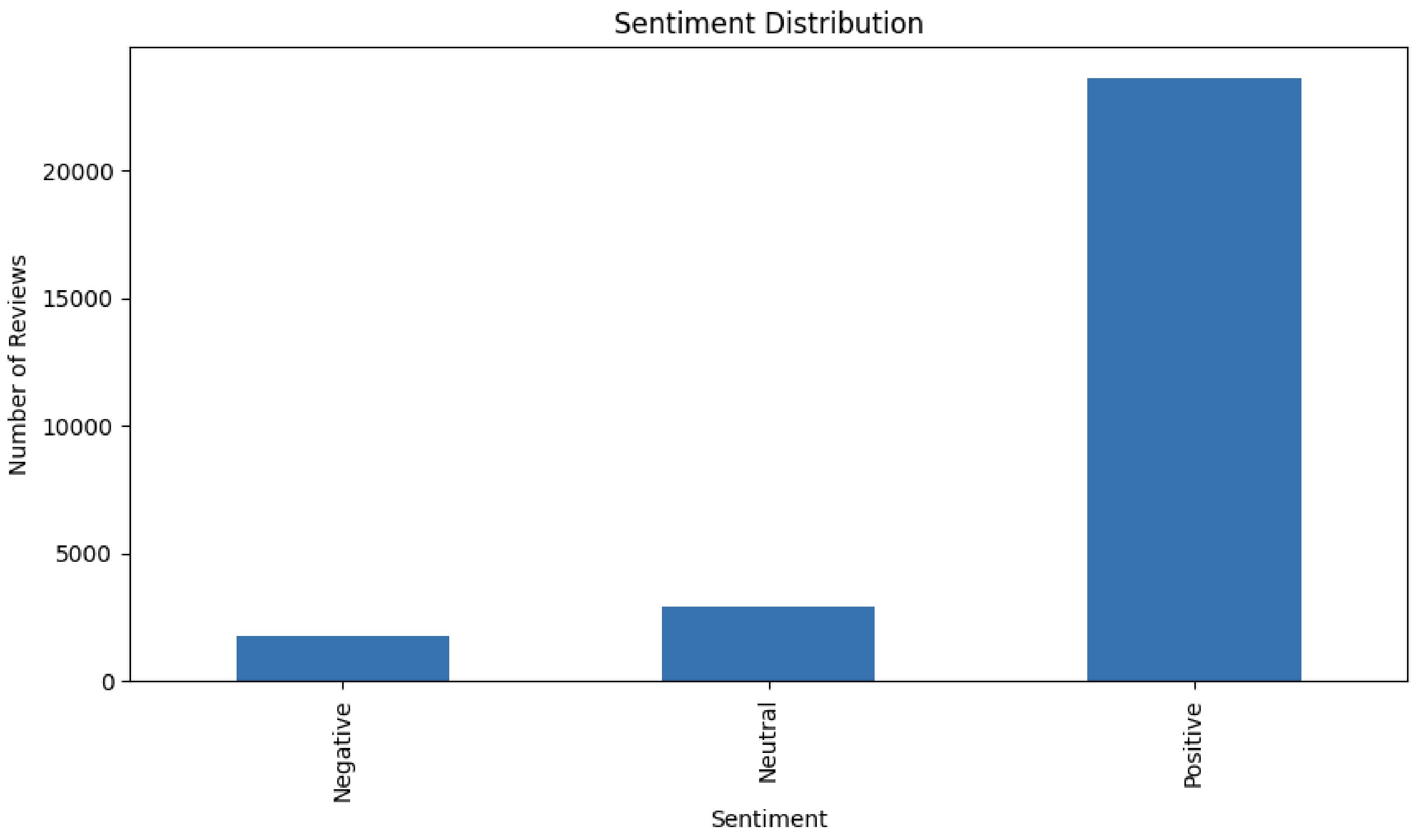
```python
def getSentiment(score):
    if score < 0:
        return "Negative"
    elif score == 0:
        return "Neutral"
    else:
        return "Positive"
```

```python
# Use the 'getSentiment' function to apply a sentiment to the 'Polarity' scores.
clean_data["Sentiment"] = clean_data["Polarity"].apply(getSentiment)
```

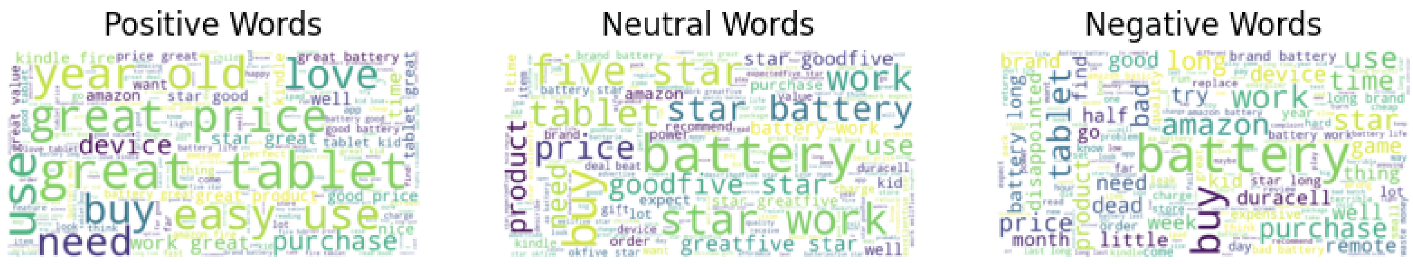
## <u>Results</u>

The sentiment analysis model has predicted that the vast majority (83%) of reviews are positive.  11% of reviews were predicted to be neutral and only 6% were negative.  This gives a good indication that customers like the products offered and the reasons for this can be explored further.

```
Sentiment
Positive    23654
Neutral      2913
Negative     1765
```



By creating a Word Cloud for each of the classifications, it is possible to see the most frequently used words.

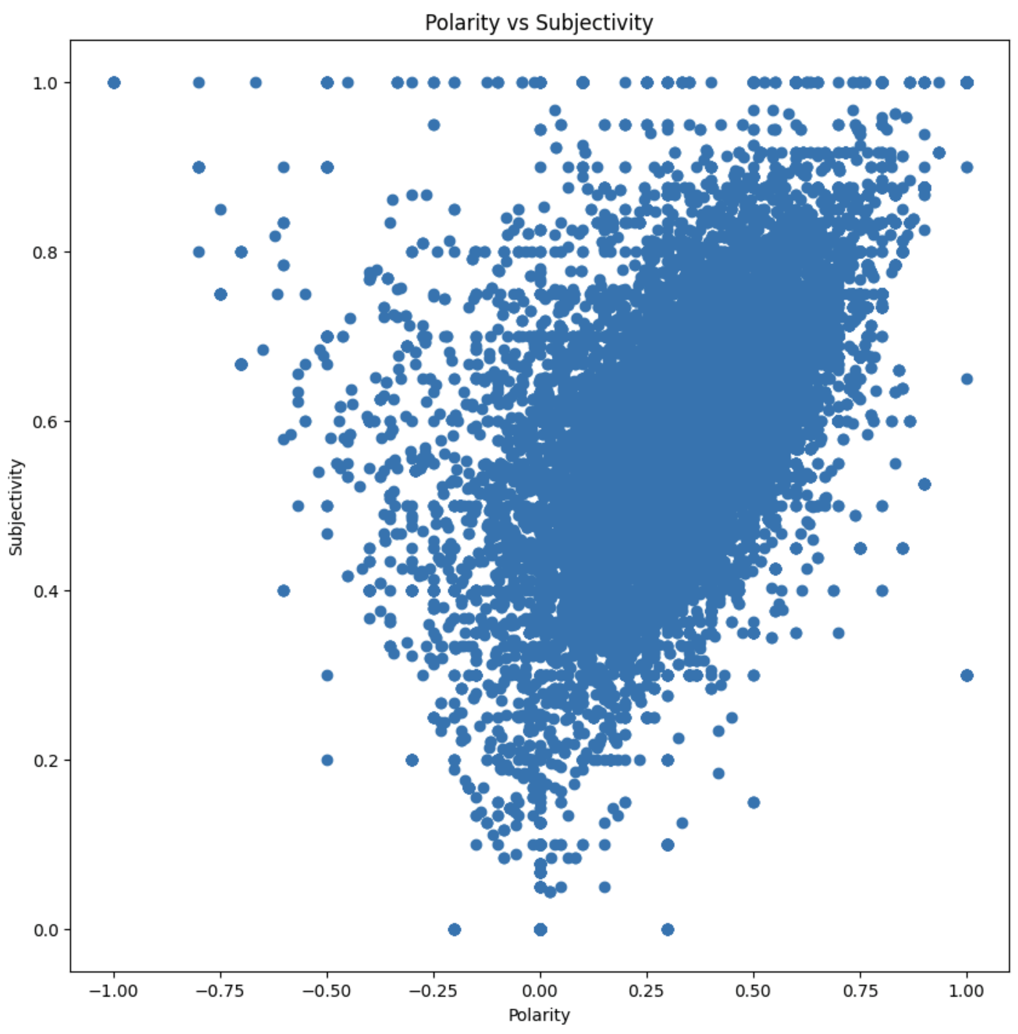


‘Great’, ‘Love’ and ‘Easy’ were some of the most frequently used words in reviews that were classified as positive.  This suggests that the model has correctly analysed these words as they clearly reflect positive emotions.  ‘Price’ was also a common word found in positive reviews which would indicate that this is another product attribute that customers like.

‘Battery’ was commonly found in both neutral and negative classifications which suggests that customers would like to see the battery’s lifespan and/or capacity improved in these products.

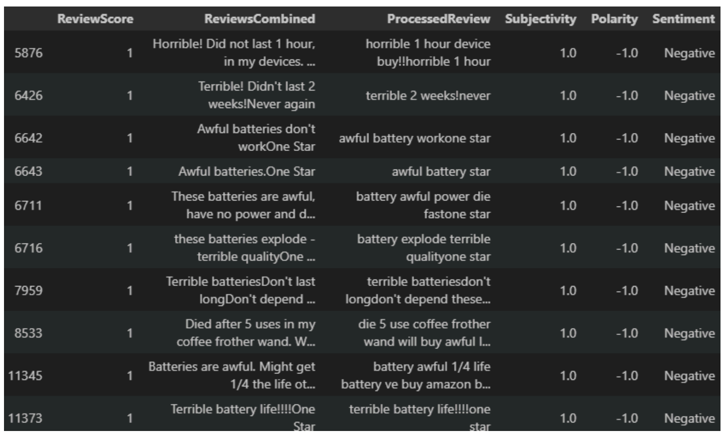
It is apparent that there are some words that may have been incorrectly classified.  For example, ‘five star’ and ‘great’ appear in the neutral classification but are obviously positive words.  However, it may be that the balance of the review was ultimately neutral, even though positive words were used.

The following graph compares the subjectivity scores to the polarity scores:

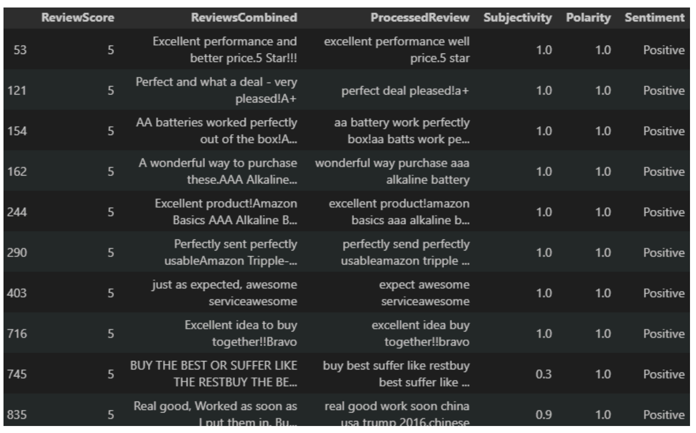


There appears to be a correlation between the subjectivity of a review and its sentiment. Reviews with a high subjectivity are more positive in attitude. Reviews that are more objective score lower on polarity, trending towards 0. This suggests that objective reviews have less personal feelings and so the strength of these decreases.

In order to test the accuracy of the model, I took a sample of the 10 highest polarity reviews and the 10 lowest polarity scores. From this sample, all reviews have been correctly classified.

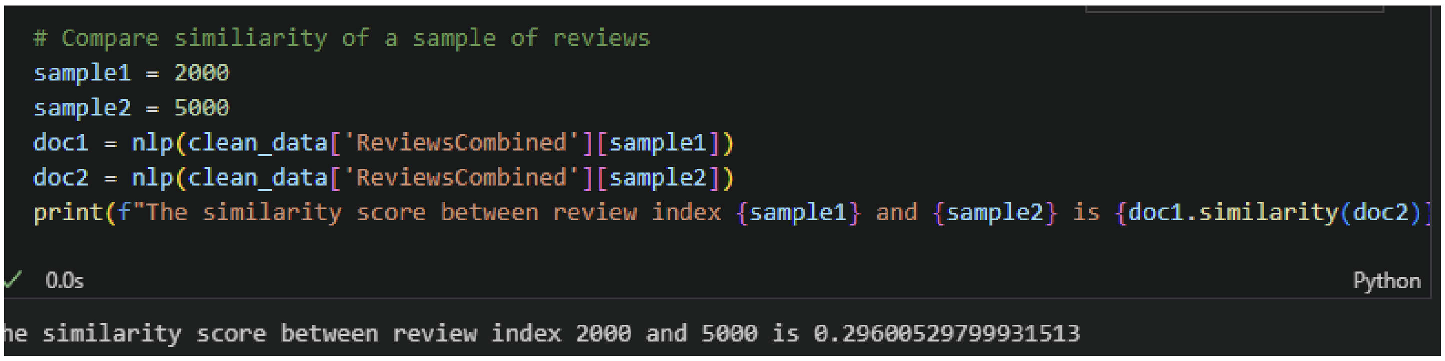


| | ReviewScore | ReviewsCombined | ProcessedReview | Subjectivity | Polarity | Sentiment |
|---|---|---|---|---|---|---|
| 5876 | 1 | Horrible! Did not last 1 hour, in my devices. ... | horrible 1 hour device buy!!horrible 1 hour | 1.0 | -1.0 | Negative |
| 6426 | 1 | Terrible! Didn't last 2 weeks!Never again | terrible 2 weeks!never | 1.0 | -1.0 | Negative |
| 6642 | 1 | Awful batteries don't workOne Star | awful battery workone star | 1.0 | -1.0 | Negative |
| 6643 | 1 | Awful batteries.One Star | awful battery star | 1.0 | -1.0 | Negative |
| 6711 | 1 | These batteries are awful, have no power and d... | battery awful power die fastone star | 1.0 | -1.0 | Negative |
| 6716 | 1 | these batteries explode - terrible qualityOne ... | battery explode terrible qualityone star | 1.0 | -1.0 | Negative |
| 7959 | 1 | Terrible batteriesDon't last longDon't depend ... | terrible batteriesdon't longdon't depend these... | 1.0 | -1.0 | Negative |
| 8533 | 1 | Died after 5 uses in my coffee frother wand. W... | die 5 use coffee frother wand will buy awful l... | 1.0 | -1.0 | Negative |
| 11345 | 1 | Batteries are awful. Might get 1/4 the life ot... | battery awful 1/4 life battery ve buy amazon b... | 1.0 | -1.0 | Negative |
| 11373 | 1 | Terrible battery life!!!!One Star | terrible battery life!!!!one star | 1.0 | -1.0 | Negative |



| | ReviewScore | ReviewsCombined | ProcessedReview | Subjectivity | Polarity | Sentiment |
|---|---|---|---|---|---|---|
| 53 | 5 | Excellent performance and better price.5 Star!!! | excellent performance well price.5 star | 1.0 | 1.0 | Positive |
| 121 | 5 | Perfect and what a deal - very pleased!A+ | perfect deal pleased!a+ | 1.0 | 1.0 | Positive |
| 154 | 5 | AA batteries worked perfectly out of the box!A... | aa battery work perfectly box!aa batts work pe... | 1.0 | 1.0 | Positive |
| 162 | 5 | A wonderful way to purchase these.AAA Alkaline... | wonderful way purchase aaa alkaline battery | 1.0 | 1.0 | Positive |
| 244 | 5 | Excellent product!Amazon Basics AAA Alkaline B... | excellent product!amazon basics aaa alkaline b... | 1.0 | 1.0 | Positive |
| 290 | 5 | Perfectly sent perfectly usableAmazon Tripple-... | perfectly send perfectly usableamazon tripple ... | 1.0 | 1.0 | Positive |
| 403 | 5 | just as expected, awesome serviceawesome | expect awesome serviceawesome | 1.0 | 1.0 | Positive |
| 716 | 5 | Excellent idea to buy together!!Bravo | excellent idea buy together!!bravo | 1.0 | 1.0 | Positive |
| 745 | 5 | BUY THE BEST OR SUFFER LIKE THE RESTBUY THE BE... | buy best suffer like restbuy best suffer like ... | 0.3 | 1.0 | Positive |
| 835 | 5 | Real good, Worked as soon as I put them in. Bu... | real good work soon china usa trump 2016.chinese | 0.9 | 1.0 | Positive |

As a further exercise, I used the SpaCy .similarity() function to compare two reviews, chosen at random. This yielded a similarity score of 0.296, which would indicate a relatively low similarity between the two reviews. However, as I am using the ‘en_core_web_sm’ model which doesn’t contain word vectors, the results would be improved by using a larger model.



```
# Compare similiarity of a sample of reviews
sample1 = 2000
sample2 = 5000
doc1 = nlp(clean_data['ReviewsCombined'][sample1])
doc2 = nlp(clean_data['ReviewsCombined'][sample2])
print(f"The similarity score between review index {sample1} and {sample2} is {doc1.similarity(doc2)}
```

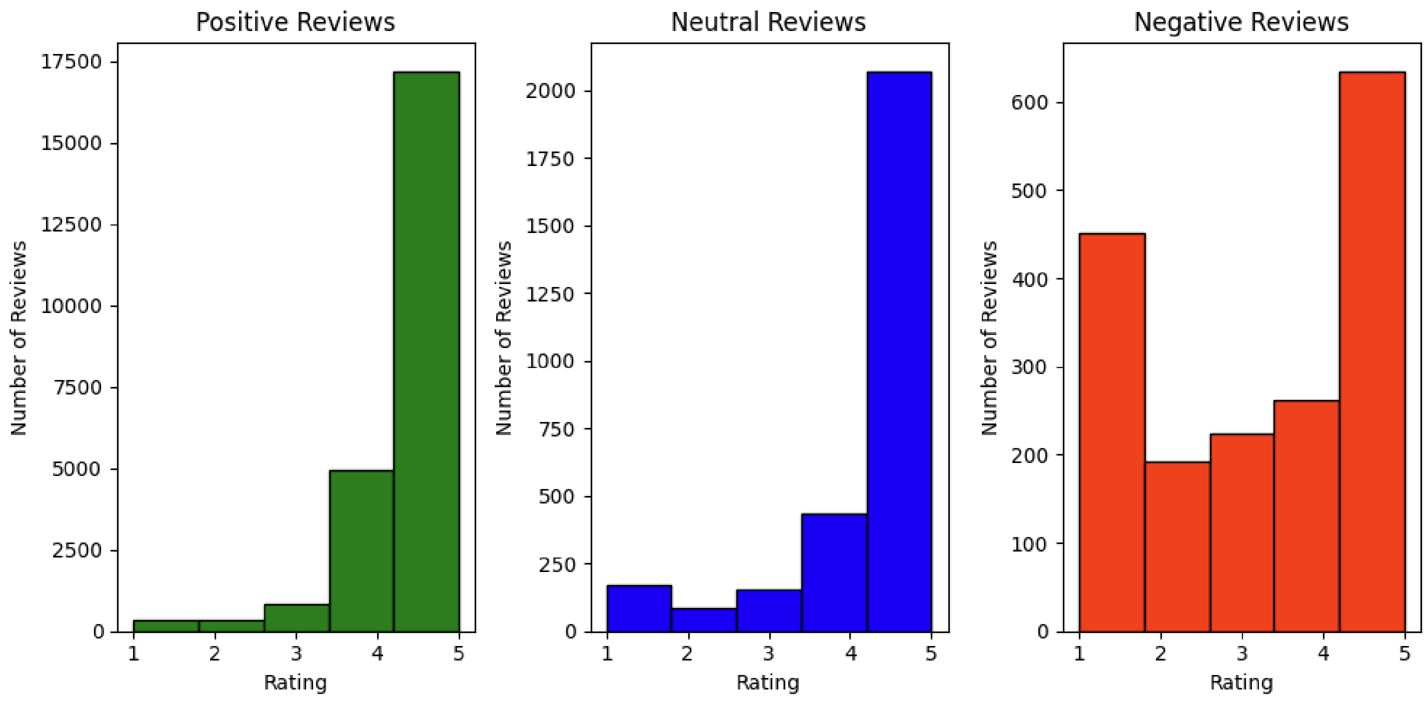
```
he similarity score between review index 2000 and 5000 is 0.29600529799931513
```

## <u>Strengths and Limitations</u>

SpaCy is a relatively new library, but provides out-of-the-box support for many NLP tasks. It is easy to use and has optimised algorithms and data structures for fast performance. SpaCy also provides high accuracy in linguistic annotations, powered by machine learning.

There may be some inaccuracy with how the model is classifying negative reviews. By creating graphs showing the customer review score distribution by sentiment classification, we can see that a number of 5-star reviews, which would be expected to contain a positive statement, have been classified as Neutral or Negative. However,the majority of reviews appear to be correctly analysed, as 17,000 5-star reviews have been classified as positive, compared to circa 600 that have been classified as negative.

For this project I have used the 'small' model but the accuracy of the analysis may be improved by using the 'medium' or 'large' models as these have more data to draw from. Further analysis is required to understand why there is a discrepancy between sentiment and review score for some reviews.



Finally, the dataset used may not be complete and so may not provide the full-picture. As the data is for Amazon branded products sold on the Amazon website, it is possible that some negative reviews may have been removed in order to make the products seem more attractive to customers. Conversely, positive reviews that are not from real customers may have been added to produce the same outcome. The completeness and integrity of the data should always be considered when using sentiment analysis to inform decision making.