Report: Sentiment Analysis and Similarity of Amazon Reviews

Prepared by: *Veli Nhlapo*

Course: *Data Science*

Date: *10 September 2025*

Introduction

This report explores customer product reviews from the Amazon dataset. The main goal was to apply **Natural Language Processing (NLP)** techniques to analyze the sentiment of reviews and to measure the similarity between them. We used the **spaCy** library with the **en_core_web_md** model and integrated the **spacytextblob** pipeline for sentiment scoring.

Dataset Overview

The dataset is titled *Datafiniti Amazon Consumer Reviews of Amazon Products (May 2019)*. It contains 24 columns with product details, brand information, and customer reviews.

For this analysis, we focused only on the **reviews.text** column, which stores the main body of customer feedback. Before processing, we dropped missing values (**NaN**).

A sample of the dataset is shown below:

Example review: "I order 3 of them and one of the item is bad quality."

Another review: "Bulk is always the less expensive way to go for products."

Approach

1. Analysis

We used spacytextblob, which provides a polarity score for each review:

- Polarity > 0 → Positive
- Polarity < 0 → Negative
- Polarity = 0 → Neutral

This rule was applied to all reviews to categorize their sentiment.

2. Similarity Analysis

We also used the vector embeddings from **en_core_web_md** to measure **cosine similarity** between reviews. This allowed us to see how closely related two reviews are in meaning.

Results

Sentiment Analysis

We tested the system on five randomly selected reviews. The results were:

“Awesome tablet. I was amazed how fast it is. And the software is very user friendly” → **Positive**

“They don't last. Used in electronics (like computer mice, computer keyboards). Energizer or Duracell last easily 3x longer. Not worth the savings.” → **Positive** (misclassified, should be Negative)

“Thx.” → **Neutral**

“Kids love it, easy to use, great quality. Bought this for the grandkids and it has a 2-year warranty.” → **Positive**

“The kids feature is great. My 18-month-old takes it with her everywhere. Very kid friendly.” → **Positive**

Observation:

Most reviews were identified as **Positive**, which reflects Amazon’s general review trend. However, one clearly negative review was misclassified as Positive, showing a limitation of the sentiment model.

Similarity Analysis

We compared two reviews using spaCy embeddings:

Review 1: “I order 3 of them and one of the item is bad quality.”

Review 2: “Bulk is always the less expensive way to go for products.”

The similarity score was 0.95, indicating the reviews are semantically very close, even though the wording differs.

Observation:

High similarity scores can group reviews that discuss the same product features (e.g., quality and price).

Challenges and Limitations

- **Misclassifications**: Some negative reviews were labeled Positive due to the simplistic polarity thresholding.

- **Model differences**: If **en_core_web_sm** was used instead of **en_core_web_md**, similarity results would be weaker since the small model does not include word vectors.
- Processing speed: The dataset is large, and analyzing all reviews may take time.

Conclusion

This project showed that:

- Sentiment Analysis revealed a strong bias toward positive reviews, with occasional misclassifications.

- Similarity Analysis helped identify closely related reviews, even when worded differently.

Future Improvements

- Use transformer-based models such as BERT or RoBERTa for more accurate sentiment classification.

- Perform a balanced analysis by including more negative and neutral reviews.

- Extend similarity analysis to cluster reviews by product or feature.