iHerb Hair Care Best Sellers Analysis

◆Scope:
Web Scraping, Data Visualization, Sentiment Analysis

◆Tools:
Selenium, BeautifulSoup, YData Profiling, PowerBI,
Natural Language Toolkit, Huggingface RoBERTa Transformers,
SciPy, WordCloud, TextBlob, Pandas, Numpy,
Regular Expression, Matplotlib, Seaborn

Agenda

01 Project Objective & Flow

02 Web Scraping (Product information & Customer reviews)

03 Data Preprocessing

04 Data Visualisation with a Dashboard

05 Sentiment Analysis on Customer Reviews

06 Conclusions and Future Work

Project Objective

- Get actual hair care products information and customer reviews
- Analyze hair care best sellers from iHerb

Project Flow

01
Web Scraping
(Product information)

02
Data
Preprocessing

03
Data
Visualisation

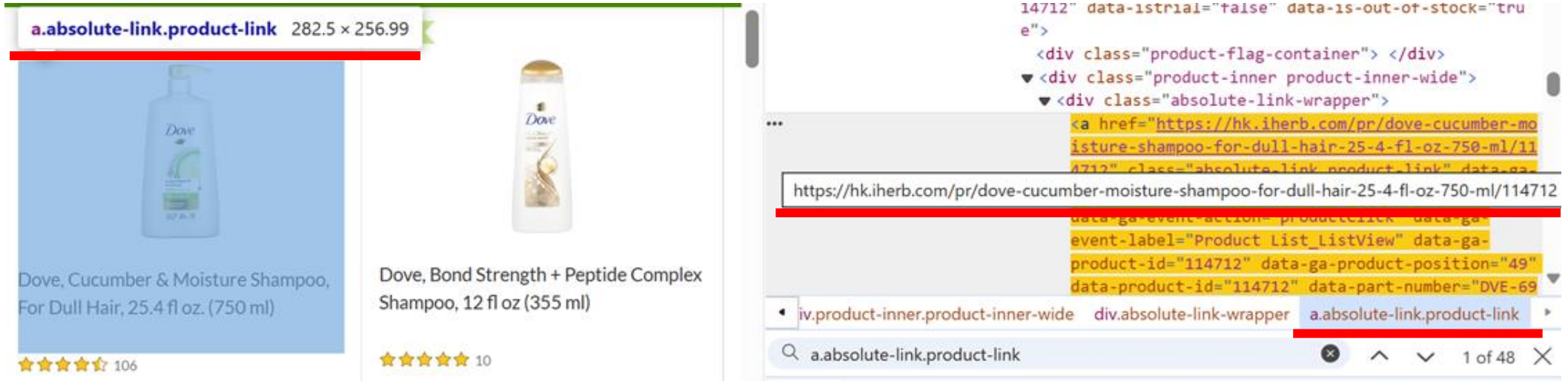
04
Web Scraping
(Customer reviews)

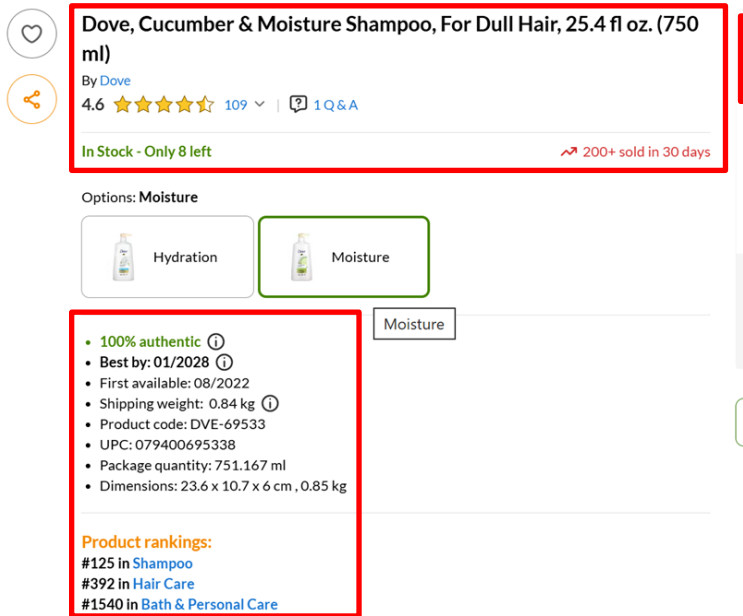
05
Text
Preprocessing

06
Sentiment
Analysis

Scraping of Product Information

Tools: **Selenium** and **BeautifulSoup**





Process:

- Defined a function to scrape data from a product page at each iteration
- Saved the scraped data into a CSV file

Scraping of Product Information

- **Bot Detection**: iHerb product pages detect bots using **XParameter**

No.	Web scrape methods we have tried		Results
1	Frameworks and Libraries	Scrapy	Fail
		Playwright	Fail
		BeautifulSoup	Success
2	Proxy and IP Management	Proxy Usage	Fail
		Switch IP Addresses in Turn	Success
3	Browser Automation	Headless Browser (Bright Data Scraping Browser)	Fail
		Selenium	Success
4	Data Extraction Techniques	Scrapfly	Fail
5	Request Management	Set User-Agent	Fail
		Set Request Headers	Fail
		Set Random Delay Time	Fail
6	No-Code Tools	Octoparse	Fail
		Instant Data Scraper	Only got infomation from 4 columns

Scraping of Product Information

- **Solution:** A **semi-manual script** was implemented as a workaround.

- **Product Links**: Scraped links for **630 targeted products** and saved them in a CSV file.
- **Removed** the corresponding product link from the **Product Link CSV** after **successful** scraping.
- If scraping **failed**, the link **remained** for future attempts.

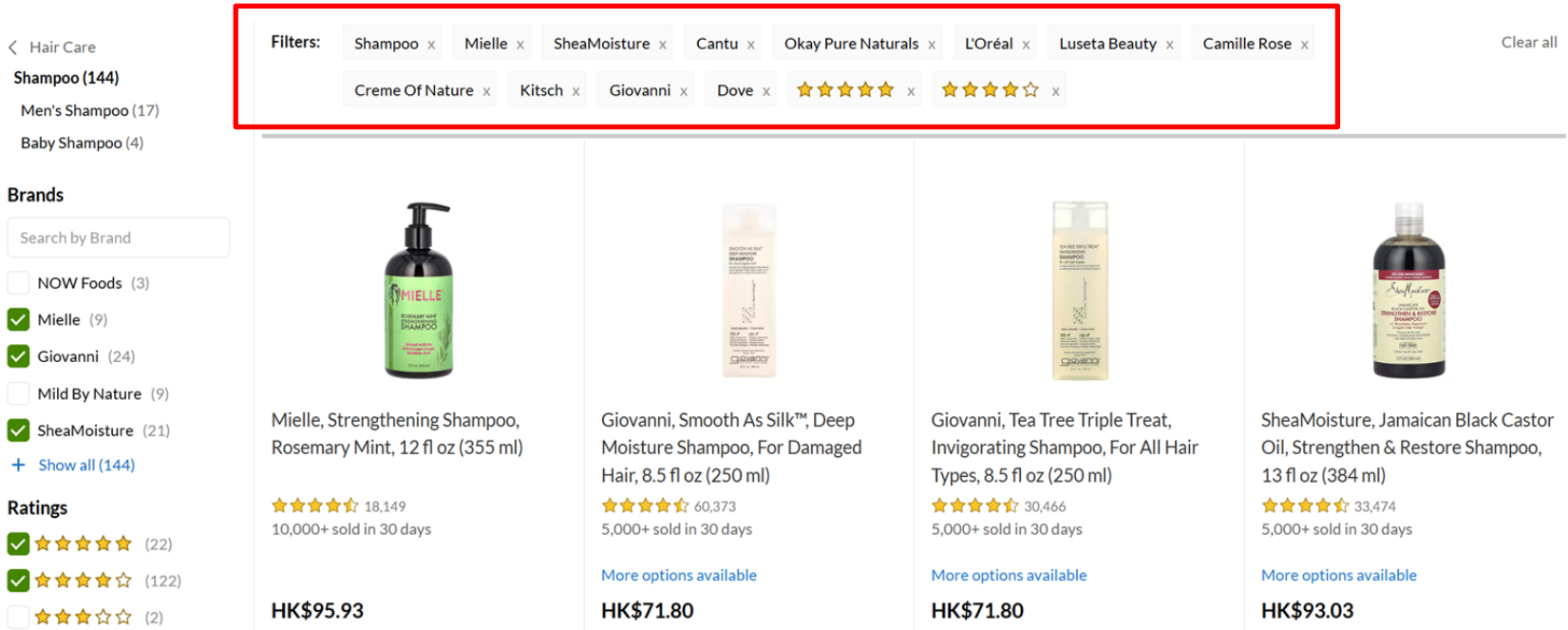
- **IP addresses Management**: Changed IP addresses using VPN and recalled the scraping function until data was successfully collected for all 630 product links.

Introduction of Dataset

- Source of dataset: [Hair Care • Natural Hair Care Products | iHerb]

- Product information:

Selection Criteria: **630 products** from **4 main categories** and **11 well-known brands**, each with over 40 hair care products, all rated **4 or 5 out of 5**.



Categories	Number of products	Percentage
Conditioner	196	31%
Shampoo	149	24%
Treatments	144	23%
Styling	141	22%

Dataset Overview (Product information)

- Total Records: 630 (12 columns)
- Unique Products: 608
- Duplicates: 22 (due to multi-category classification)

- Retention Reason: To analyze product distribution across categories.

```
# Determine the number of unique elements of the dataset
df.nunique()
```

```
product_id          608
product_name        608
brand_name           11
category              4
list_price          269
discount_price      300
sale_in_30days       14
rating               13
no_of_reviews       338
first_available      50
rankings            416
stock_alert           3
dtype: int64
```

Data Preprocessing

1. Check null value

Sale in 30 days

- Products with Sales Data: 40% (252/630)
- Interpretation: 60% had no sales in 30 days (378/630)

- Sales Range: 100 to 40,000
- Action Taken: Filled missing records with 0

```
# Determine whether there are missing values
df.isnull().sum()
```

```
product_id            0
product_name          0
brand_name            0
category              0
list_price            0
discount_price        0
sale_in_30days      378
rating                0
no_of_reviews         0
first_available       0
rankings              0
stock_alert         560
dtype: int64
```

```
# Handle sale_in_30_days

# Check the range of sales_in_30days
min_value = df['sale_in_30days'].min()
max_value = df['sale_in_30days'].max()

# Display results
print(f"Minimum sale_in_30days: {min_value}")
print(f"Maximum sale_in_30days: {max_value}")
```

```
Minimum sale_in_30days: 100.0
Maximum sale_in_30days: 40000.0
```

Data Preprocessing

Stock alert

Action Taken: Fill null values in 'stock_alert' with the string "in stock"

```
# Determine whether there are missing values
df.isnull().sum()
```

```
product_id            0
product_name          0
brand_name            0
category              0
list_price            0
discount_price        0
sale_in_30days      378
rating                0
no_of_reviews         0
first_available       0
rankings              0
stock_alert         560
dtype: int64
```

```
# Fill null values in 'sale_in_30days' with 0
df['sale_in_30days'] = df['sale_in_30days'].fillna(0)

# Change the data type of 'sale_in_30days' to integer
df['sale_in_30days'] = df['sale_in_30days'].astype(int)

# Fill null values in 'stock_alert' with the string "in stock"
df['stock_alert'] = df['stock_alert'].fillna("in stock")

# Change the data type of 'product_id to string
df['product_id'] = df['product_id'].astype(str)

# Change the data type of 'first_available' to date
df['first_available'] = pd.to_datetime(df['first_available'], format='%m/%Y')

df.head()
```

Data Preprocessing

2. Check records without volume or weight and **drop 2 products**

```
# Check records without volumne or weight
missing_all_columns = df[
    df['volume_in_floz'].isna() &
    df['volume_in_ml'].isna() &
    df['weight_in_oz'].isna() &
    df['weight_in_g'].isna()
]

# Output results
if missing_all_columns.empty:
    print("No products are missing records in all specified columns.")
else:
    print("The following products are missing records in all specified columns:")
    print(missing_all_columns[['product_name', 'volume_in_floz', 'volume_in_ml', 'weight_in_oz', 'weight_in_g']])
```


```
The following products are missing records in all specified columns:
                                    product_name  volume_in_floz  \
367  Kitsch, Scalp Exfoliator, Terracotta, 1 Exfoli...             NaN
417       Kitsch, Scalp Exfoliator, Grey, 1 Exfoliator             NaN

     volume_in_ml  weight_in_oz  weight_in_g
367           NaN           NaN          NaN
417           NaN           NaN          NaN
```

Data Preprocessing

3. Standardize capacity and

Create a new column

- Volume in ml

floz = ml

g = oz

```
# Display the updated DataFrame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 628 entries, 0 to 627
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   product_id       628 non-null    object
 1   product_name     628 non-null    object
 2   brand_name       628 non-null    object
 3   category         628 non-null    object
 4   list_price       628 non-null    float64
 5   discount_price   628 non-null    float64
 6   sale_in_30days   628 non-null    int32
 7   rating           628 non-null    float64
 8   no_of_reviews    628 non-null    int64
 9   first_available  628 non-null    datetime64[ns]
 10  rankings         628 non-null    object
 11  stock_alert      628 non-null    object
 12  volume_in_ml     628 non-null    float64
dtypes: datetime64[ns](1), float64(4), int32(1), int64(1), object(6)
memory usage: 61.5+ KB
```

Data Preprocessing

4. Create 3 more new columns

- Price per ml
- Discount %
- Sales revenue

```
# Add column price per ml
df['price_per_ml'] = (df['discount_price'] / df['volume_in_ml']).round(2)

# Add column disount %
df['discount%'] = ((df['list_price']-df['discount_price'])/df['list_price']*100).round(2)

# Add column sales revenue
df['sales_revenue'] = (df['discount_price']*df['sale_in_30days']).round(2)

# Display the updated DataFrame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 628 entries, 0 to 627
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   product_id       628 non-null    object
 1   product_name     628 non-null    object
 2   brand_name       628 non-null    object
 3   category         628 non-null    object
 4   list_price       628 non-null    float64
 5   discount_price   628 non-null    float64
 6   sale_in_30days   628 non-null    int32
 7   rating           628 non-null    float64
 8   no_of_reviews    628 non-null    int64
 9   first_available  628 non-null    datetime64[ns]
 10  rankings         628 non-null    object
 11  stock_alert      628 non-null    object
 12  volume_in_ml     628 non-null    float64
 13  price_per_ml     628 non-null    float64
 14  discount%        628 non-null    float64
 15  sales_revenue    628 non-null    float64
dtypes: datetime64[ns](1), float64(7), int32(1), int64(1), object(6)
memory usage: 76.2+ KB
```

Raw Dataset (Product information)

Categorical

1. Brand name
2. Product name
3. Category
4. First available
5. Stock alert
6. Rankings

Numeric

1. Product ID
2. List price
3. Discount price
4. Sale in 30days
5. Rating
6. No. of reviews

Clean Dataset (Product information)

Categorical

1. Brand name
2. Product name
3. Category
4. First available
5. Stock alert
6. Rankings

only take the necessary columns:
ranking_shampoo
ranking_conditioner
ranking_hair_treatments
ranking_hair_styling

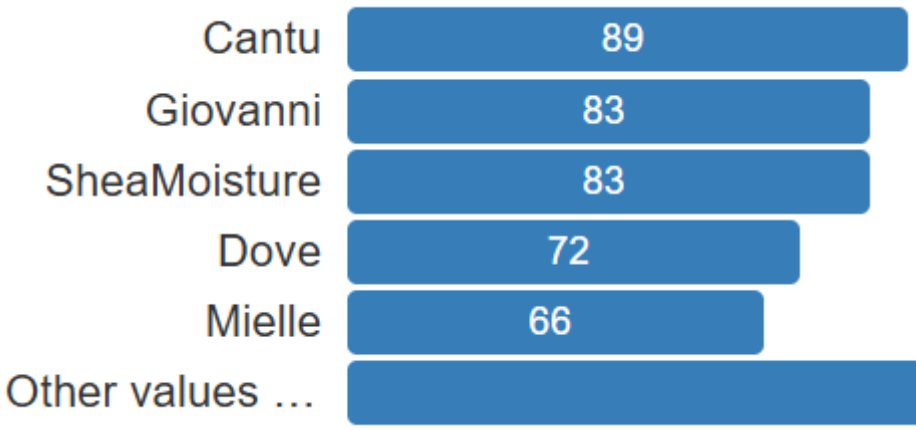
Create new columns

Numeric

1. Product ID
2. List price
3. Discount price
4. Sale in 30 days
5. Rating
6. No. of reviews
7. Volume in ml
8. Price per ml
9. Discount %
10. Sales revenue

Data Exploration

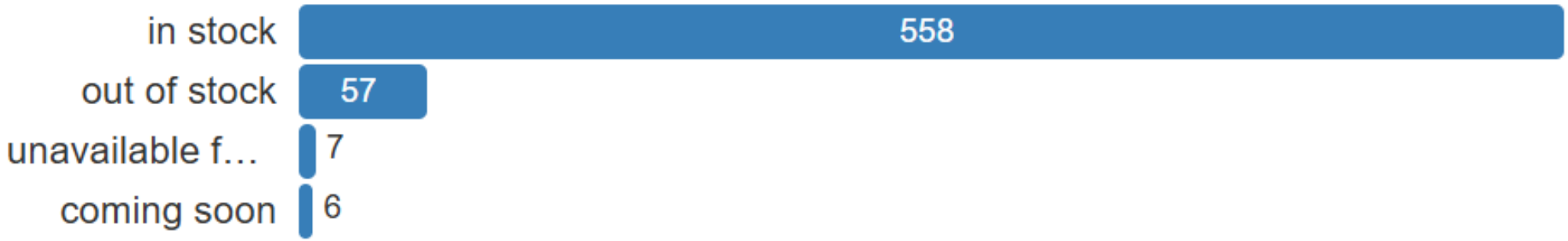
5 brands that account for a relatively large number of datasets.



Data Exploration

Stock alert situation

- Nearly 90% : in stock
- 9% : out of stock
- 1% : unavailable in Hong Kong



Data Exploration

Volume in ml

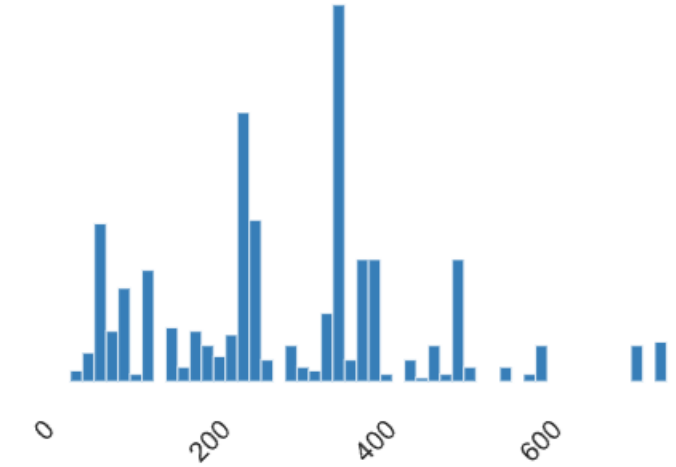
Mean: 294

Max : 750 (about **2.5 times** higher than average)

volume_in_ml
Real number (ℝ)

HIGH CORRELATION

Distinct	113	**Minimum**	30
Distinct (%)	18.0%	**Maximum**	750
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	293.89975	**Memory size**	5.0 KiB



Data Exploration

Price per ml

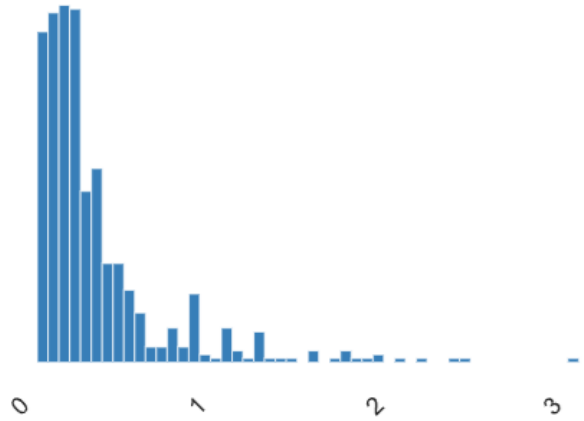
Mean: 0.42

Max : 3.14 (about **7 times** higher than average)

price_per_ml
Real number (ℝ)

HIGH CORRELATION

Distinct	98	**Minimum**	0.09
Distinct (%)	15.6%	**Maximum**	3.14
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	0.42117834	**Memory size**	5.0 KiB



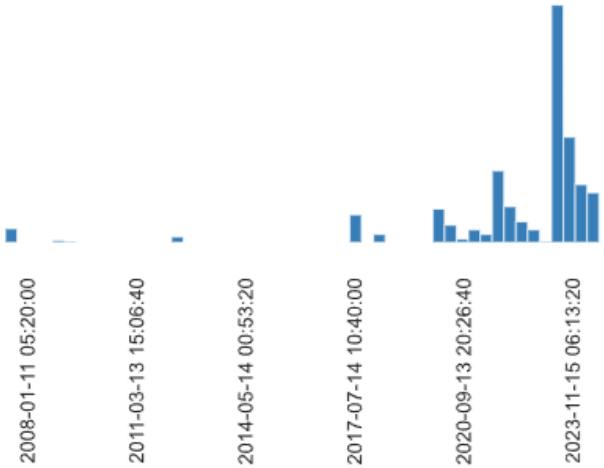
Data Exploration

A big gap in first available date.

first_available
Date

Distinct	50
Distinct (%)	8.0%
Missing	0
Missing (%)	0.0%
Memory size	5.0 KiB

Minimum	2007-05-01 00:00:00
Maximum	2024-08-01 00:00:00



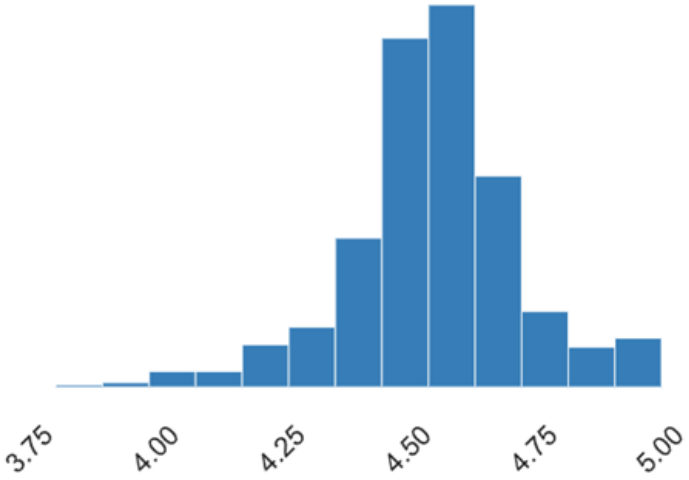
Data Exploration

Rating

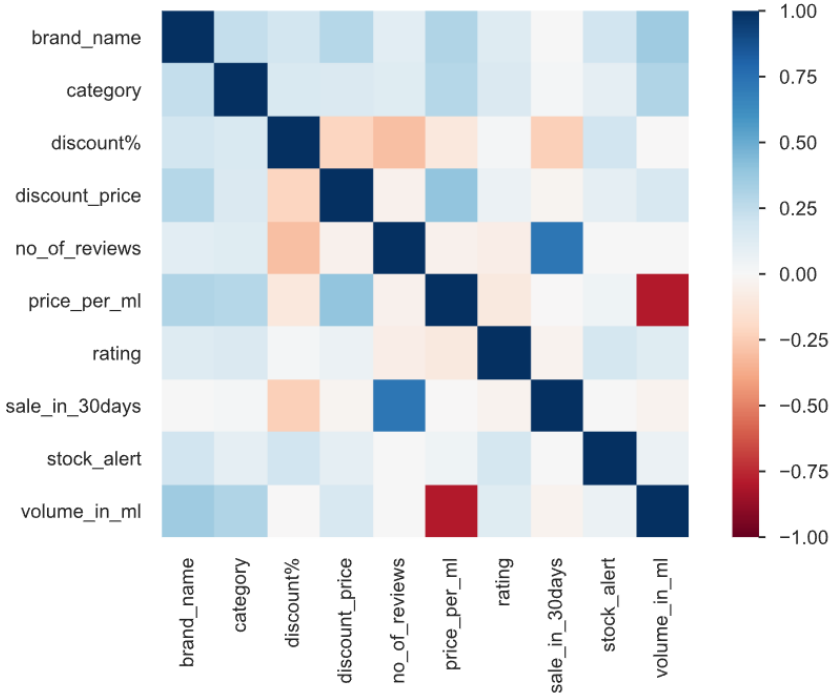
Mean: 4.56

rating
Real number (ℝ)

Distinct	13	**Minimum**	3.8
Distinct (%)	2.1%	**Maximum**	5
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	4.561465	**Memory size**	5.0 KiB



Heatmap



	brand_name	category	discount%	discount_price	no_of_reviews	price_per_ml	rating	sale_in_30days	stock_alert	volume_in_
brand_name	1.000	0.235	0.185	0.283	0.109	0.301	0.132	0.000	0.193	0.356
category	0.235	1.000	0.154	0.145	0.123	0.288	0.142	0.018	0.092	0.297
discount%	0.185	0.154	1.000	-0.213	-0.303	-0.109	0.018	-0.238	0.194	-0.004
discount_price	0.283	0.145	-0.213	1.000	-0.039	0.394	0.061	-0.024	0.086	0.159
no_of_reviews	0.109	0.123	-0.303	-0.039	1.000	-0.040	-0.063	0.721	0.000	0.001
price_per_ml	0.301	0.288	-0.109	0.394	-0.040	1.000	-0.094	-0.006	0.040	-0.796
rating	0.132	0.142	0.018	0.061	-0.063	-0.094	1.000	-0.037	0.179	0.119
sale_in_30days	0.000	0.018	-0.238	-0.024	0.721	-0.006	-0.037	1.000	0.000	-0.035
stock_alert	0.193	0.092	0.194	0.086	0.000	0.040	0.179	0.000	1.000	0.057

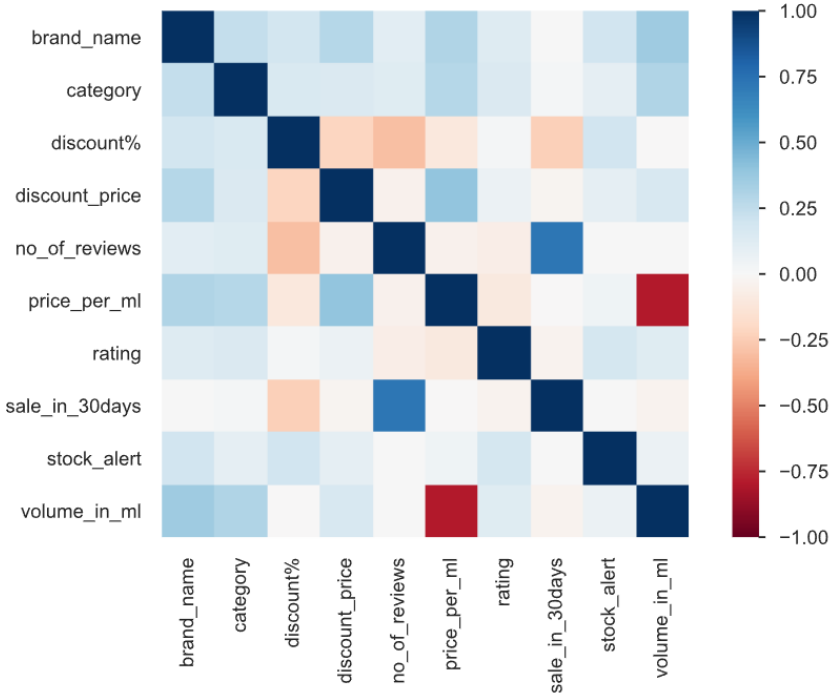
1. **No. of reviews** is inversely proportional to **discount%** (-0.303)

   Fewer reviews result in greater discounts.
2. **Price per ml** is influenced by **brand** (0.301)

   Some brands are notably more expensive.
3. **Discount price** (product prices) is proportional to **price per ml** (0.394)

   Higher product prices lead to higher unit prices.

Heatmap



	brand_name	category	discount%	discount_price	no_of_reviews	price_per_ml	rating	sale_in_30days	stock_alert	volume_in_
brand_name	1.000	0.235	0.185	0.283	0.109	0.301	0.132	0.000	0.193	0.356
category	0.235	1.000	0.154	0.145	0.123	0.288	0.142	0.018	0.092	0.297
discount%	0.185	0.154	1.000	-0.213	-0.303	-0.109	0.018	-0.238	0.194	-0.004
discount_price	0.283	0.145	-0.213	1.000	-0.039	0.394	0.061	-0.024	0.086	0.159
no_of_reviews	0.109	0.123	-0.303	-0.039	1.000	-0.040	-0.063	0.721	0.000	0.001
price_per_ml	0.301	0.288	-0.109	0.394	-0.040	1.000	-0.094	-0.006	0.040	-0.796
rating	0.132	0.142	0.018	0.061	-0.063	-0.094	1.000	-0.037	0.179	0.119
sale_in_30days	0.000	0.018	-0.238	-0.024	0.721	-0.006	-0.037	1.000	0.000	-0.035
stock_alert	0.193	0.092	0.194	0.086	0.000	0.040	0.179	0.000	1.000	0.057

4. **Volume in ml** is **highly correlated** with **price per ml (-0.796)**

   Product volume increases, the price per milliliter tends to decrease.

5. **No. of reviews** is **highly correlated** with **sale in 30 days (0.721)**

   As the number of reviews increases, sales tend to increase as well.

Data Visualization with a Dashboard

Dashboard

Sales Revenue = product price*sale in 30 days

Popularity = rating*no. of reviews / first available date of the products to 2024/9/30

The Popularity Score offers a balanced view of product appeal by combining average user ratings with the number of reviews and time since availability.

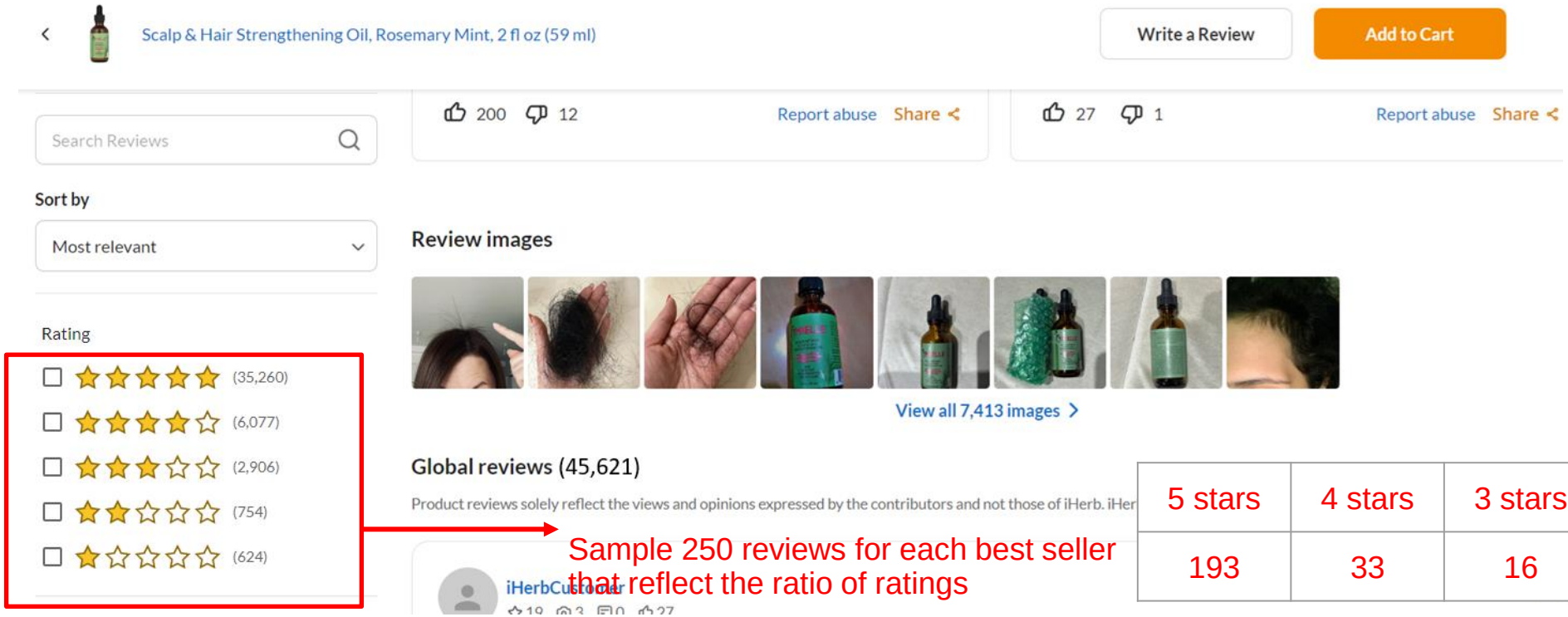
This metric helps to:

- **Mitigate Rating Bias:** High ratings alone can misrepresent a product's popularity.
- **Reflect Trendiness:** It captures genuine popularity trends beyond just raw ratings.
- **Provide Comprehensive Insights:** By factoring in average ratings, review counts, and the duration of availability, it reveals a clearer picture of a product's market standing.

Sentiment Analysis on Customer Reviews

Introduction of Reviews Dataset

- Selection Criteria: Over the past 30 days and among the 11 well-known brands, the **best-selling product** in each of the **4 categories (Shampoo, Conditioner, Treatments, Styling)**
- Data Captured: Customer reviews based on **rating ratios**
- Total Reviews Collected: 250 reviews * 4 best sellers = **1,000 reviews**
- 3 columns: product_id, rating, review_text



5 stars	4 stars	3 stars	2 stars	1 star
193	33	16	4	4

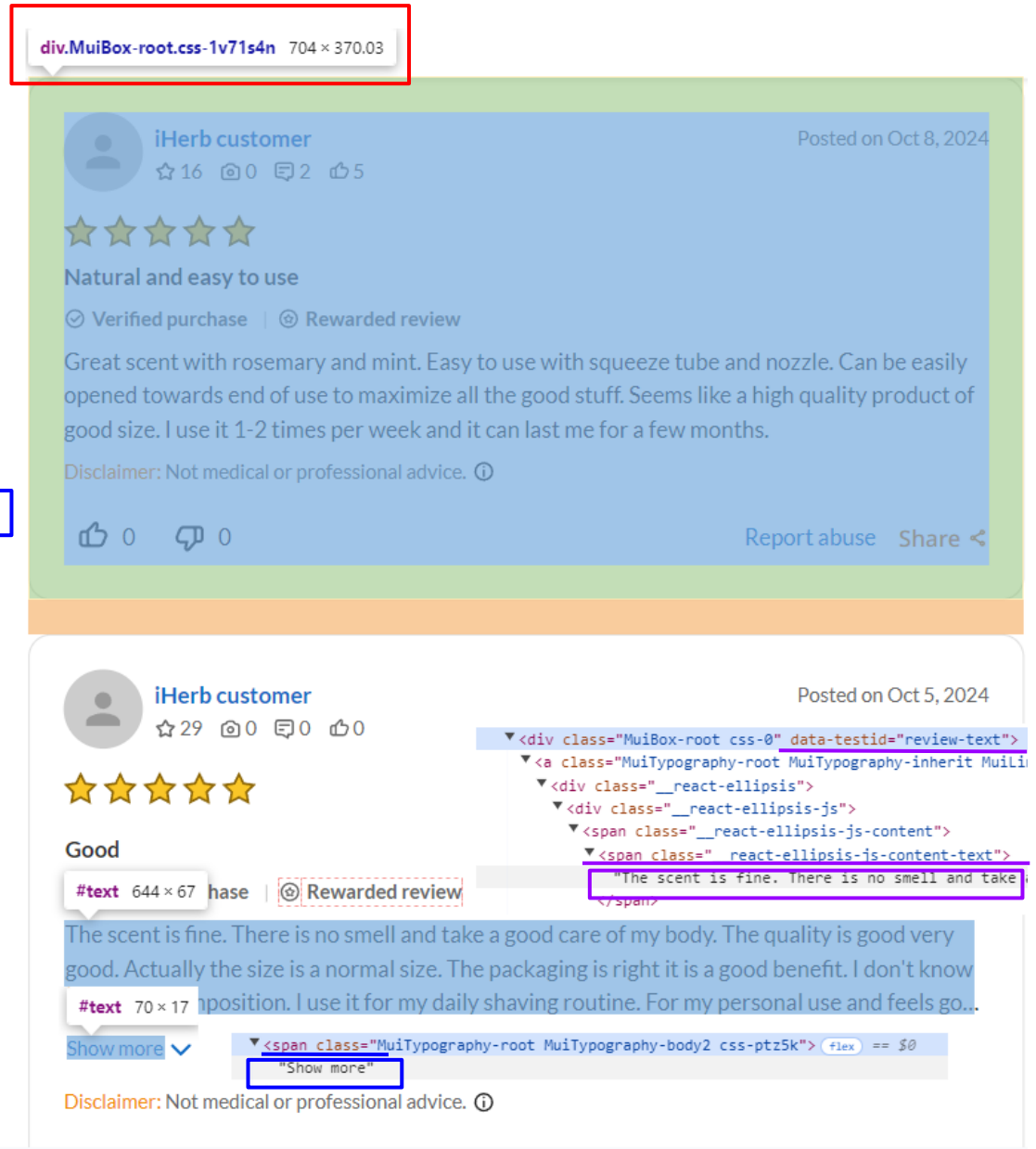
Scraping of Customer Reviews

```python
def scrape_reviews(url, max_pages):
    review_contents = []
    driver = setup_driver()
    driver.maximize_window()

    for page_num in range(1, max_pages + 1):
        review_link = f"{url}&p={page_num}"
        print(review_link)
        driver.get(review_link)

        review_containers = driver.find_elements(By.CSS_SELECTOR, "div.MuiBox-root.css-1v71s4n")

        for index, container in enumerate(review_containers):
            print(f"Processing review {index + 1} on page {page_num}...")
            try:
                driver.execute_script("arguments[0].scrollIntoView();", container)
                review_text_element = WebDriverWait(container, 10).until(
                    EC.visibility_of_element_located((By.CSS_SELECTOR, "div[data-testid='review-text'] span.__react-ellipsis-js-content"))
                )

                try:
                    show_more_button = container.find_element(By.XPATH, ".//span[contains(text(), 'Show more')]")
                    if show_more_button:
                        print(f"There is 'Show more' for review {index + 1} on page {page_num}...")
                        driver.execute_script("arguments[0].click();", show_more_button)
                        WebDriverWait(container, 10).until(
                            EC.visibility_of_element_located((By.CSS_SELECTOR, "div[data-testid='review-text'] span.__react-ellipsis-js-content"))
                        )
                except NoSuchElementException:
                    print(f"No 'Show more' button found for review {index + 1} on page {page_num}.")

                review_text_element = container.find_element(By.CSS_SELECTOR, "div[data-testid='review-text'] span.__react-ellipsis-js-content")
                review_contents.append(review_text_element.text)
                print(f"Review {index + 1} text on page {page_num}: {review_text_element.text}")
            except TimeoutException:
                print(f"The 'Show more' button is still visible after waiting for it to disappear for review {index + 1} on page {page_num}.")
            except Exception as e:
                print(f"Error processing review {index + 1} on page {page_num}: {e}")

    driver.quit()
    return review_contents
```

1. Locate all ten review containers in a page

2. Expand reviews if necessary by clicking “Show more”

3. Capture the text of each review



Text Preprocessing

- **Emoji Conversion:**
  Convert emojis to their official descriptions to standardize the text
- **Lowercasing:**
  Normalize all text to lowercase to ensure uniformity
- **Special Character Removal:**
  Remove unnecessary special characters like ,.:;@#^&_+-*/<>(){}®=…,
  while preserving emotive punctuations like ! ?
- **Stopword Removal:**
  Remove stopwords that do not carry significant meaning to reduce noise in the text
- **Spelling Correction:**
  Correct spelling errors using TextBlob to improve text quality
- **Lemmatization:**
  Reduce words to their base or root form using lemmatization to normalize variations

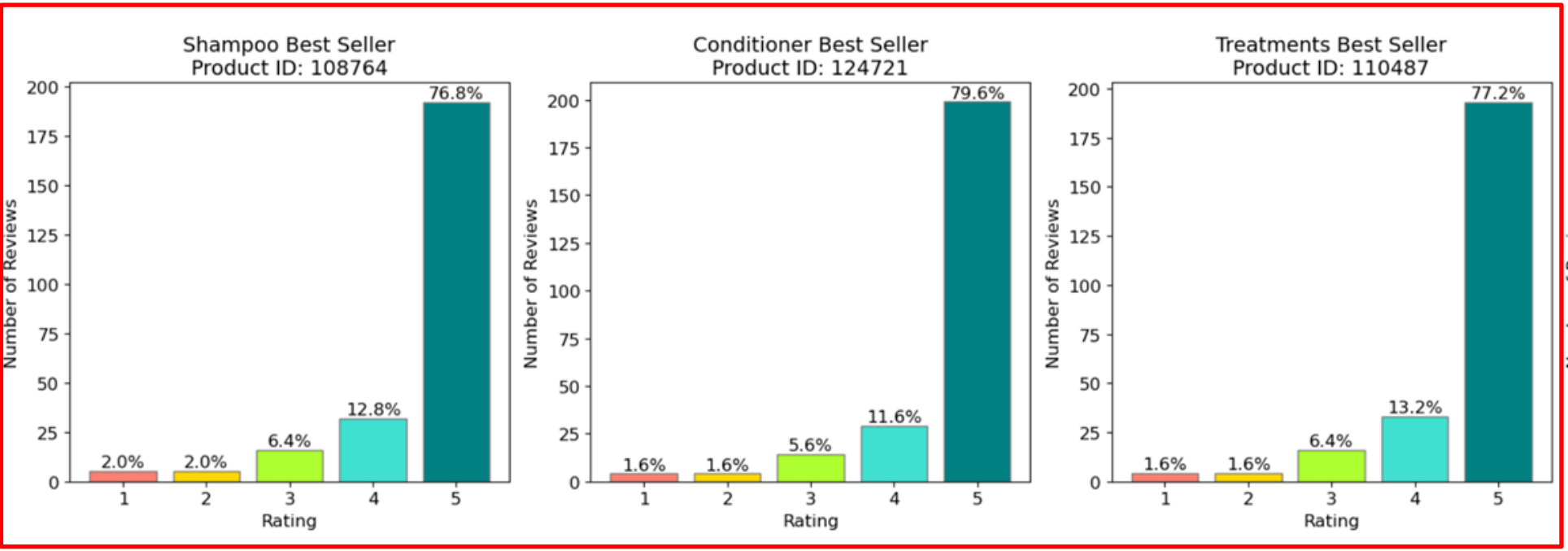
Rating Distribution

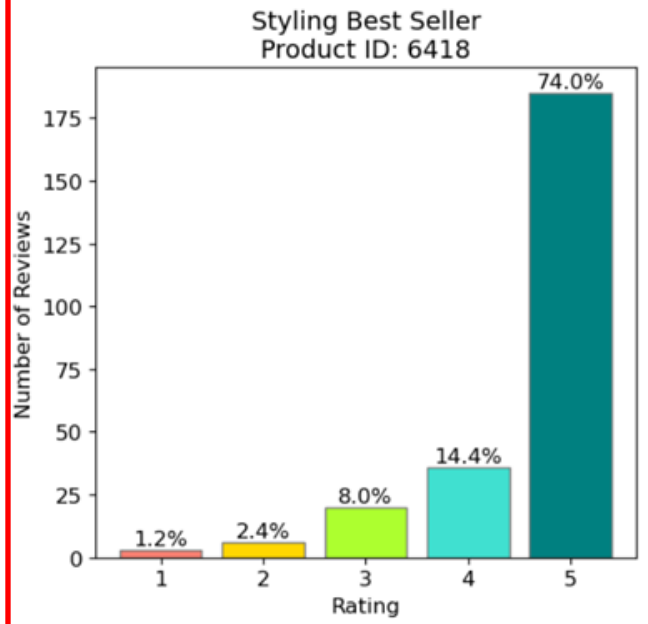
108764: **Mielle**, Strengthening **Shampoo**, Rosemary Mint

124721: **Mielle**, Strengthening **Conditioner**, Rosemary Mint Blend

110487: **Mielle**, Scalp & Hair Strengthening **Oil**, Rosemary Mint

6418: **Giovanni**, L.A. Hold **Styling Gel**, Strong Hold





Word Clouds by Rating

Analysis Plan:

Collective analysis: Shampoo, Conditioner and Treatments

Individual examination: Styling product

Visualization: Generate word clouds for each group based on rating scales

(i) Ratings 4 to 5

(ii) Rating of 3

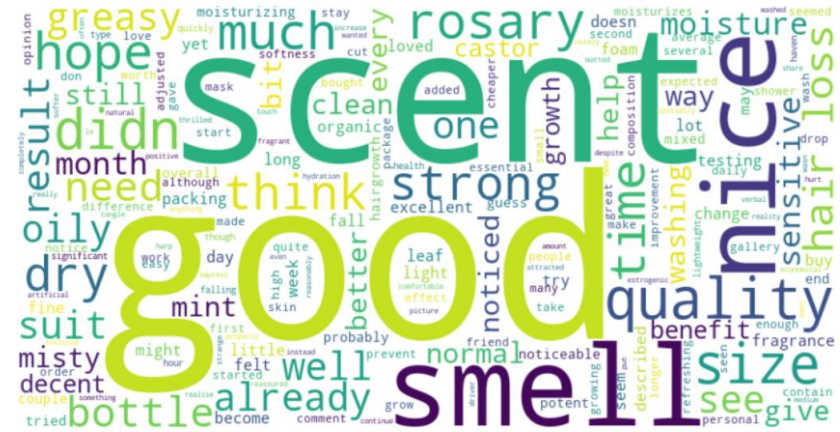
(iii) Ratings 1 to 2

Word Clouds by Rating

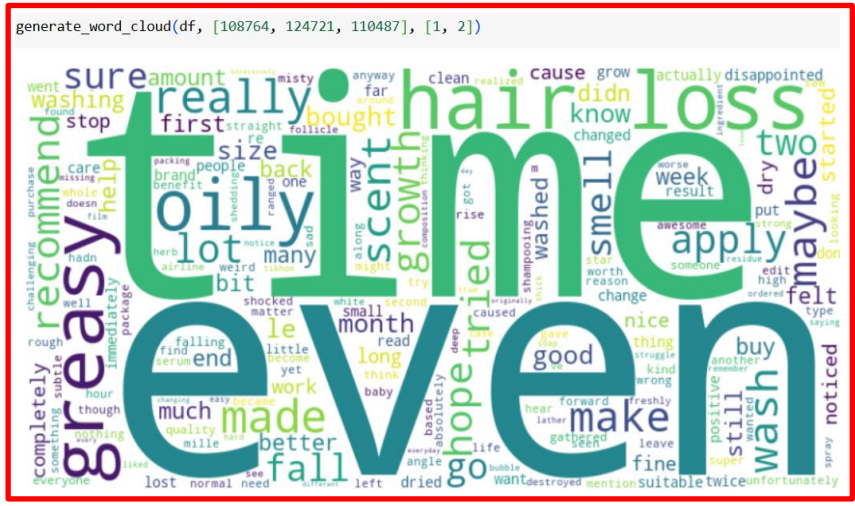
Mielle, Strengthening Hair Products, Rosemary Mint (**Shampoo, Conditioner, Treatments Best Sellers**)

```
generate_word_cloud(df, [108764, 124721, 110487], [4, 5])
```


```
generate_word_cloud(df, [108764, 124721, 110487], [3])
```



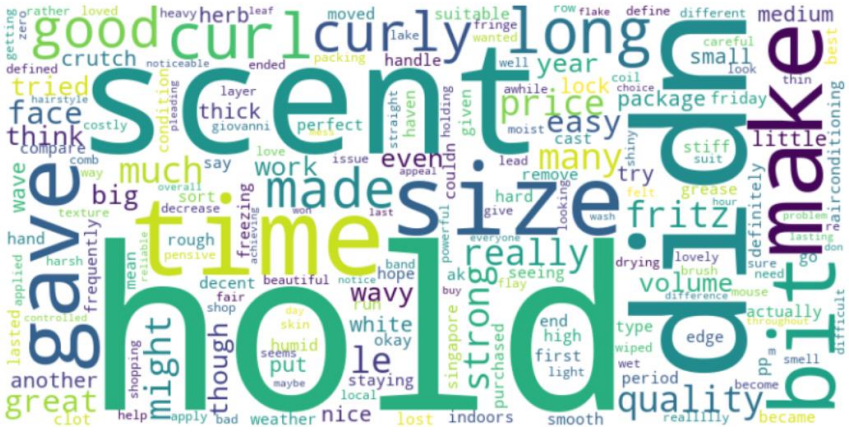

```
generate_word_cloud(df, [108764, 124721, 110487], [1, 2])
```



Giovanni, L.A. Hold Styling Gel, Strong Hold (**Styling Best Seller**)

```
generate_word_cloud(df, [6418], [4, 5])
```


```
generate_word_cloud(df, [6418], [3])
```




```
generate_word_cloud(df, [6418], [1, 2])
```

Approaches used for Sentiment Analysis:

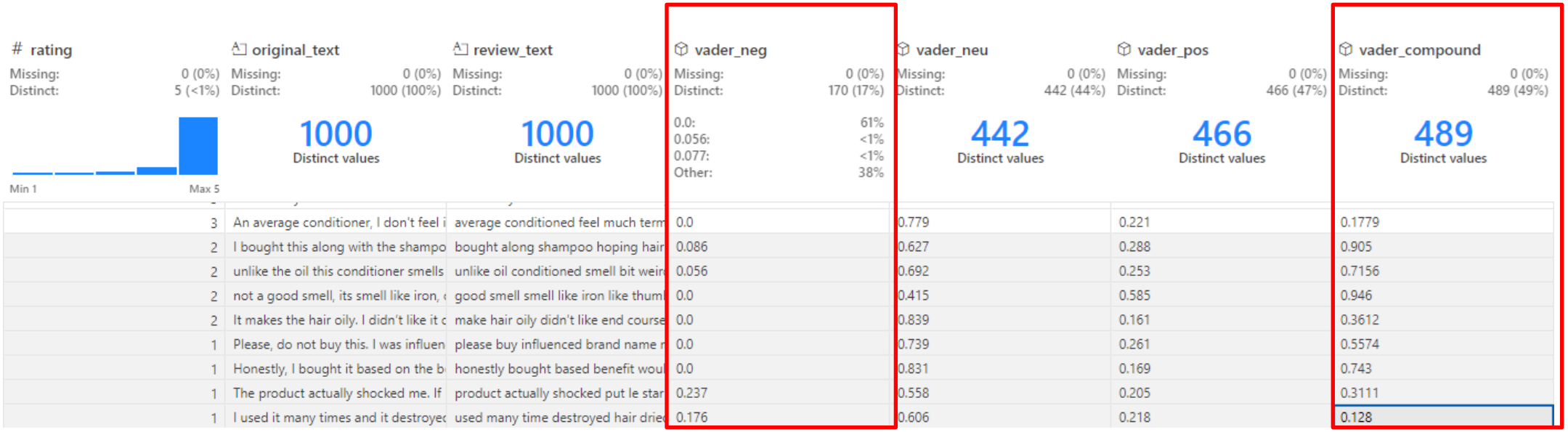
1. NLTK VADER (**V**alence **A**ware **D**ictionary **and** s**E**ntiment **R**easoner**)**

- **A lexicon and rule-based sentiment analysis tool**: a list of words with associated sentiment scores, along with a set of rules to evaluate the sentiment of text, specifically for **social media contexts**
- **'pos', 'neu', and 'neg' scores**: the **proportion** of text that falls into each category (sum up to 1)
- **'compound' score**: a normalized composite score ranging from -1 (most negative) to +1 (most positive)

2. Huggingface RoBERTa Transformers

- **Deep Learning-Based Pre-trained Model**: Uses multiple layers of Transformer architecture to learn complex features and patterns from text
- **Generation of logits**: raw scores that represent the model's preliminary assessment of the input text
- **Normalization with Scipy's Softmax**: into **probabilities** ranging from **0 to 1**
- **Positive, neutral, negative sentiment scores** (sum up to 1):
  the model's confidence level that the input text belongs to a particular sentiment class

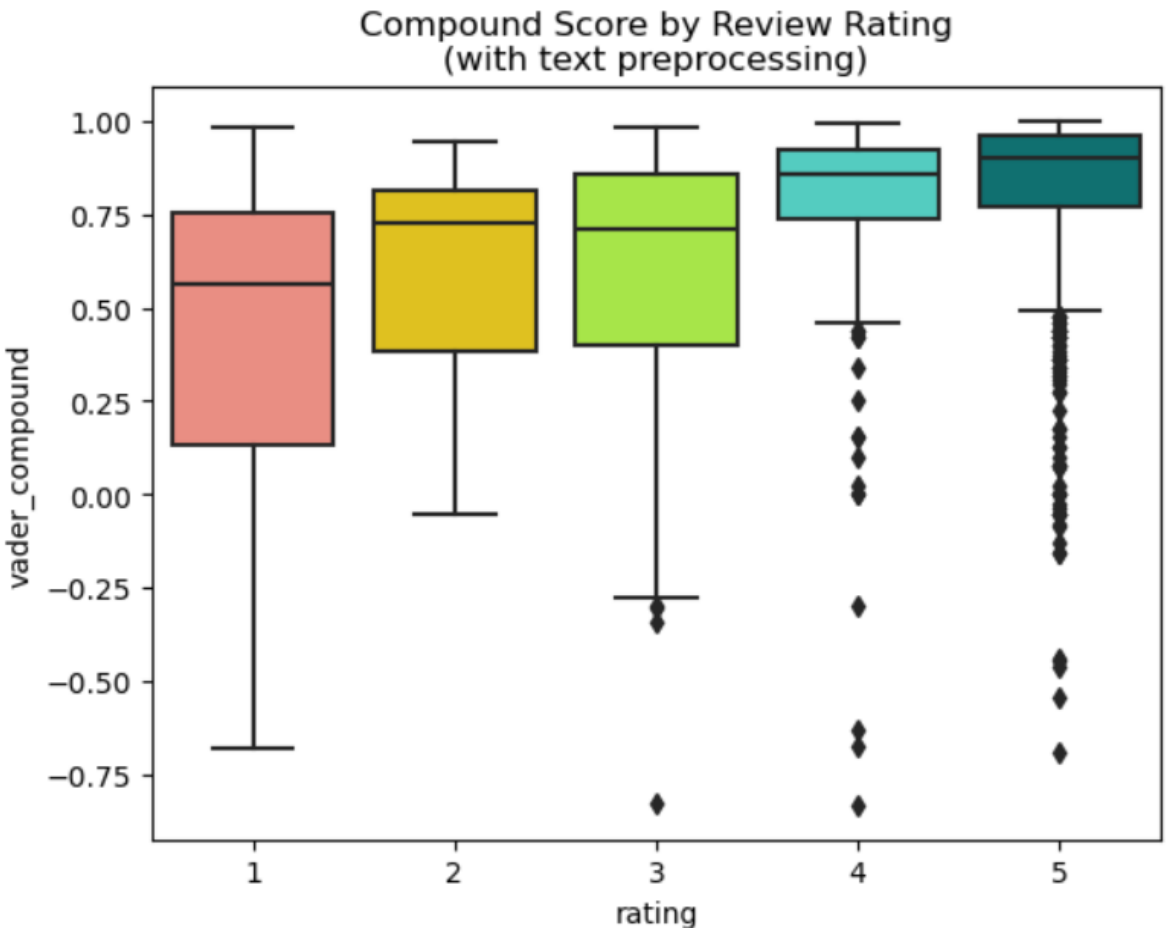
To our surprise, both approaches do not perform well on the preprocessed text, especially in scoring sentiment on reviews with low ratings.

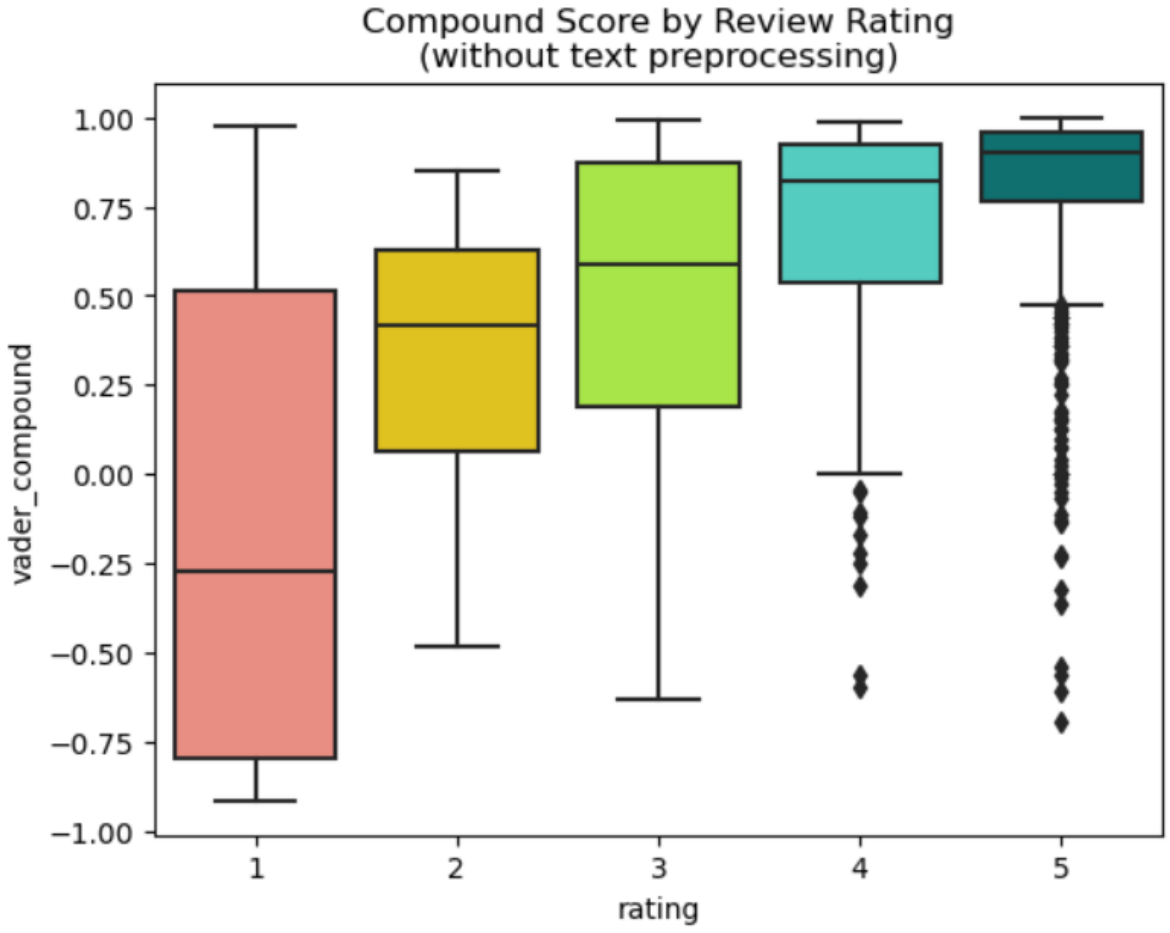


# rating	original_text	review_text	vader_neg	vader_neu	vader_pos	vader_compound
Missing: 0 (0%) Distinct: 5 (<1%)	Missing: 0 (0%) Distinct: 1000 (100%)	Missing: 0 (0%) Distinct: 1000 (100%)	Missing: 0 (0%) Distinct: 170 (17%)	Missing: 0 (0%) Distinct: 442 (44%)	Missing: 0 (0%) Distinct: 466 (47%)	Missing: 0 (0%) Distinct: 489 (49%)
Min 1 Max 5	1000 Distinct values	1000 Distinct values	0.0: 61% 0.056: <1% 0.077: <1% Other: 38%	442 Distinct values	466 Distinct values	489 Distinct values
3	An average conditioner, I don't feel i	average conditioned feel much term	0.0	0.779	0.221	0.1779
2	I bought this along with the shampo	bought along shampoo hoping hair	0.086	0.627	0.288	0.905
2	unlike the oil this conditioner smells	unlike oil conditioned smell bit weir	0.056	0.692	0.253	0.7156
2	not a good smell, its smell like iron,	good smell smell like iron like thum	0.0	0.415	0.585	0.946
2	It makes the hair oily. I didn't like it c	make hair oily didn't like end course	0.0	0.839	0.161	0.3612
1	Please, do not buy this. I was influen	please buy influenced brand name r	0.0	0.739	0.261	0.5574
1	Honestly, I bought it based on the b	honestly bought based benefit wou	0.0	0.831	0.169	0.743
1	The product actually shocked me. If	product actually shocked put le star	0.237	0.558	0.205	0.3111
1	I used it many times and it destroyec	used many time destroyed hair drie	0.176	0.606	0.218	0.128

# rating	original_text	review_text	roberta_neg	roberta_neu	roberta_pos
Missing: 0 (0%) Distinct: 5 (<1%)	Missing: 0 (0%) Distinct: 1000 (100%)	Missing: 0 (0%) Distinct: 1000 (100%)	Missing: 0 (0%) Distinct: 1000 (100%)	Missing: 0 (0%) Distinct: 1000 (100%)	Missing: 0 (0%) Distinct: 999 (>99%)
Min 1 Max 5	1000 Distinct values	1000 Distinct values	Min 0.0016079440... Max 0.87056279...	Min 0.0090356692... Max 0.903411746...	Min 0.0079993037... Max 0.98935633...
2	the scent is nice, the quality compos	scent nice quality composition lathe	0.32288322	0.5827285	0.094388306
2	First of all, I really liked the way it lat	first really liked way gathered scalp	0.0125668645	0.0686627	0.9187704
2	This product made my hair feel exce	product made hair feel excessively g	0.8705628	0.1214379	0.007999304
2	edit: i originally gave it for stars but i	edit originally gave star i'm changin	0.07461079	0.79464984	0.13073942
2	I really wanted to love this but.. may	really wanted love maybe work fine	0.021941341	0.31389585	0.6641629
1	I really didn't like the shampoo, my h	really like shampoo hair still greasy	0.6113243	0.3536271	0.03504858
1	I had so big expectation regarding th	big expectation regarding product u	0.8441516	0.14419931	0.011649103
1	i have purchased the full Mielle hair	purchased full mille hair care produc	0.2314733	0.52326244	0.24526422
1	I bought the whole range looking fo	bought whole range looking forwar	0.6541795	0.2992753	0.04654523
1	Initially, it felt like it cleaned the scalp	initially felt like cleaned scalp well b	0.5237949	0.39382058	0.082384504

Sentiment Analysis using VADER





Sentiment Analysis using VADER

1- star rating with the highest positive sentiment score

With text preprocessing:

Unwanted correction of spelling

Unexpected removal of negation words

```
vader_pos: 0.415
Original Text: i have purchased the full Mielle hair care products after the hype they got, after 2 weeks of use (I barely could use them) i wouldn't recommend them to anyone! b
Preprocessed Text: purchased full mille hair care product hope got 2 week use barely could use recommend anyone! mention help hair growth would like mention leave hair super sup
```

Without text preprocessing:

lower vader_pos: improvement

```
vader_pos: 0.248
Text: i have purchased the full Mielle hair care products after the hype they got, after 2 weeks of use (I barely could use them) i wouldn't rec
```

5- star rating with the highest negative sentiment score

With text preprocessing:

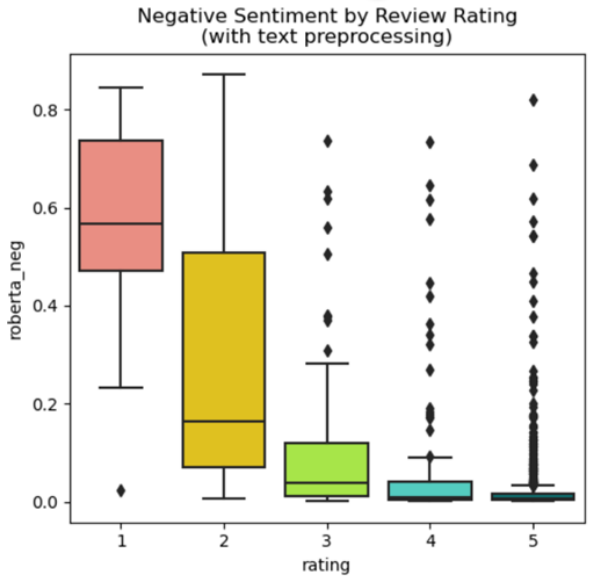
```
vader_neg: 0.435
Original Text: Crazy bittersweet that softens the hair and makes it fluffy and moisturizes it. Crazy to ask for a lot, and I will come back and order, don't miss it.
Preprocessed Text: crazy bittersweet softens hair make fluffy moisturizes crazy ask lot come back order miss
```

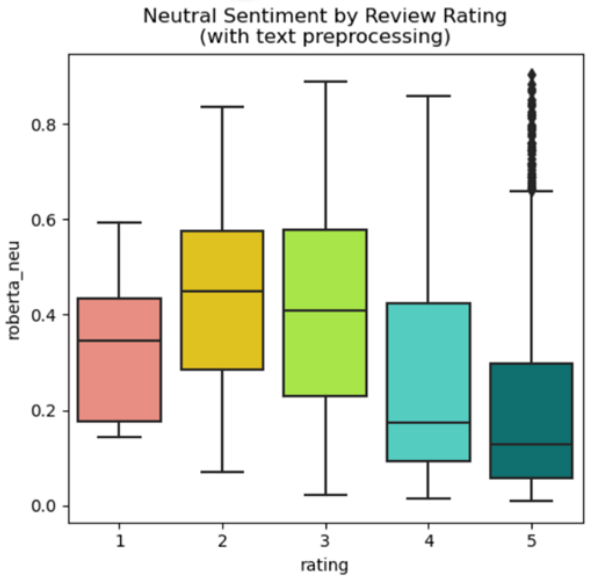
Without text preprocessing:

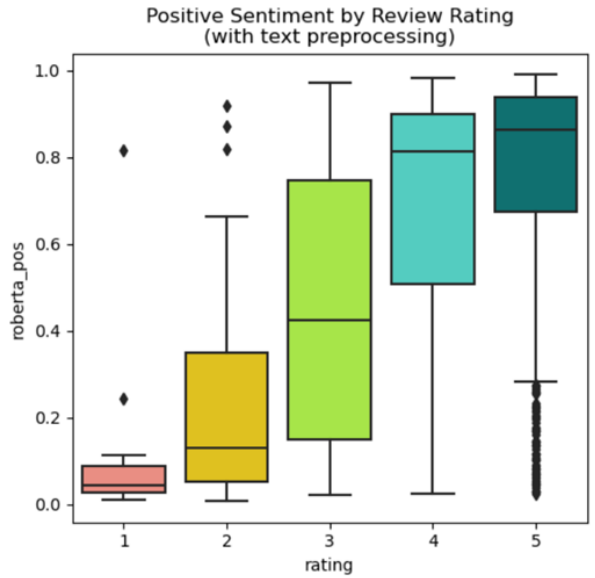
```
vader_neg: 0.26
Text: No, I haven't received it yet. No, Ingredients, Taste, Usage, Size, Package, Quantity, Like
```

Sentiment Analysis using RoBERTa

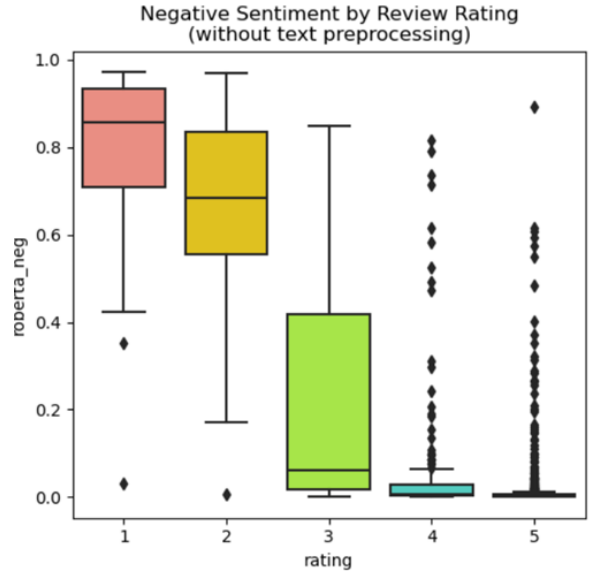
With
text preprocessing

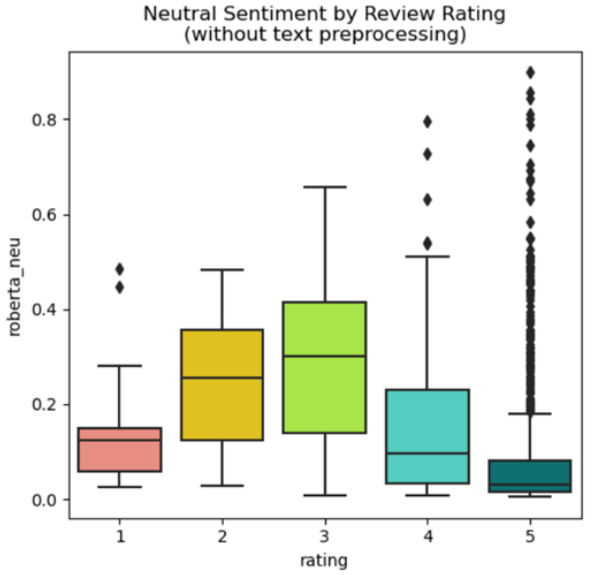


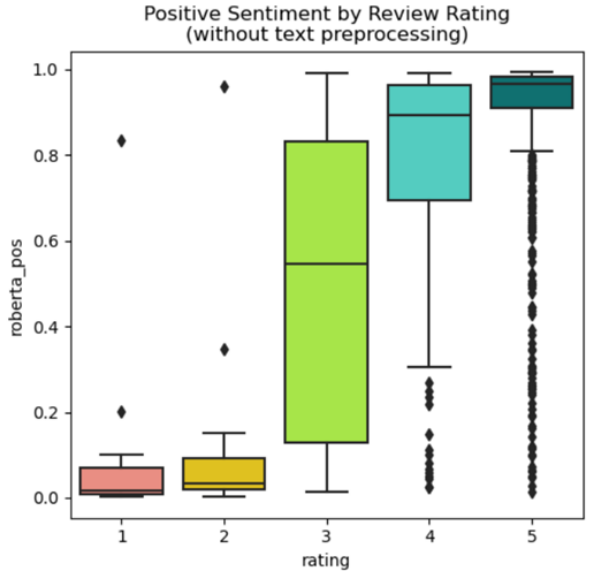




Without
text preprocessing







Sentiment Analysis using RoBERTa

1- star rating with the highest positive sentiment score

With text preprocessing:

Majority of the text carries positive sentiment;
may not be sensitvie the negative sentiment in small proportion

roberta_pos: 0.8160131573677063
Original Text: Now I finally understand the hype! After 3 months of using it I can see baby hair growing out! Noticed a little bit of difference in hair loss but I keep my hopes up after some time! The scent is subtle and doesn't bother me. My scalp is less itchy. The size is really good and goes for a while if you're oiling two times to 3 times a week. Looking forward to seeing more improvements in my hair for the next 3 months! Edit; my hair went crazy! I realized it is so heavy for my hair. Became so greasy on the scalp yet so frizzy too! Hair loss increased so I stopped using it.
Preprocessed Text: finally understand hope! 3 month using see baby hair growing out! noticed little bit difference hair loss keep hope time! scent subtle doesn't bother scalp le itchy size really good go you're boiling two time 3 time week looking forward seeing improvement hair next 3 months! edit hair went crazy! realized heavy hair became greasy scalp yet friday too! hair loss increased stopped using

Without text preprocessing: same text with similar score

5- star rating with the highest negative sentiment score

With text preprocessing:

Reasonable as seen from its small negative sentiment score

roberta_neg: 0.0236185435205698
Original Text: Very useful for girls and there is nothing better than iherb products that I recommend ❤️
Preprocessed Text: useful girl nothing better herb product recommend red heart

Without text preprocessing:

Reasonable as the review contains mostly negative wordings

roberta_neg: 0.8923622965812683
Text: Ohhh I red so many unbelievable reviews but unfortunately it didn't work for me… First of all the hair gets so greasy and you can only use it if you are home… maybe I was doing something wrong but definitely not for me … the scent is very specific the rest was ok, the package arrived well and the size is good… I guess it's just not for me

Possible reasons why both approaches perform better without text preprocessing:

145	aren't
146	couldn
147	couldn't
148	didn
149	didn't
150	doesn
151	doesn't
152	hadn
153	hadn't
154	hasn
155	hasn't
156	haven
157	haven't
158	isn
159	isn't
160	ma
161	mightn
162	mightn't
163	mustn
164	mustn't
165	needn
166	needn't
167	shan
168	shan't
169	shouldn
170	shouldn't
171	wasn
172	wasn't
173	weren
174	weren't
175	won
176	won't
177	wouldn
178	wouldn't

VADER:

1. **Relies on Lexical Cues**: Direct sentiment indicators like emojis and slang.
2. **Emoji and Emoticons Interpretation**: capable of understanding emojis and emoticons
3. **Punctuation Sensitivity**: Uses punctuation for sentiment intensity
4. **Capitalization**: May use it as a signal for emphasis

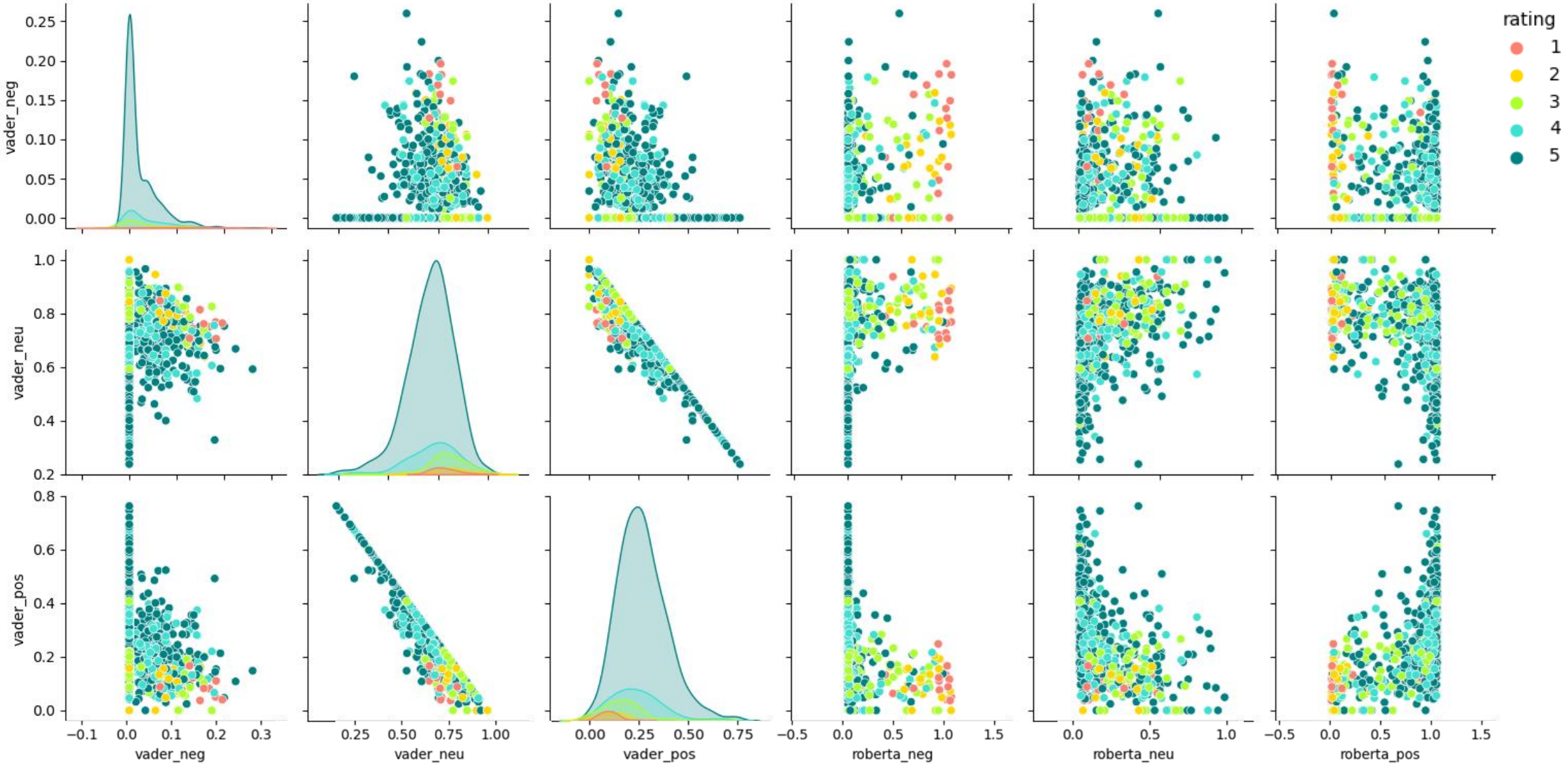
RoBERTa:

1. **Contextual Awareness**: Needs original text structure for context.
2. **Intact Syntactic Cues**: Depends on natural syntax for accurate analysis
3. **Lowercasing**: May make it harder for the model to identify contextual clues

Our findings:

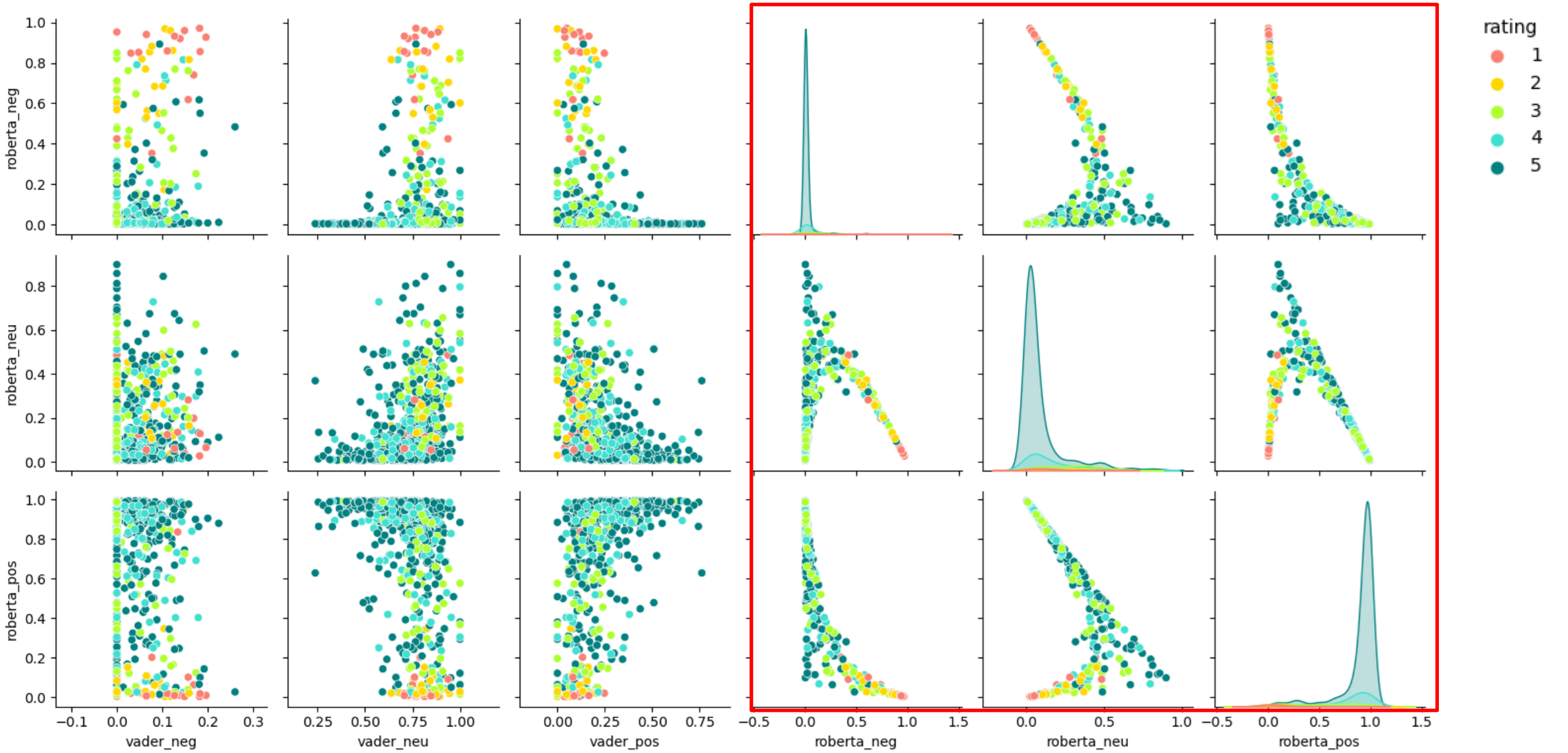
The **NLTK library's list of stopwords** includes **negation words** (as shown on the left), which play a crucial role in sentiment analysis. Stripping these words from the text during preprocessing could adversely affect the accuracy of both VADER and RoBERTa.

VADER VS RoBERTa



RoBERTa:

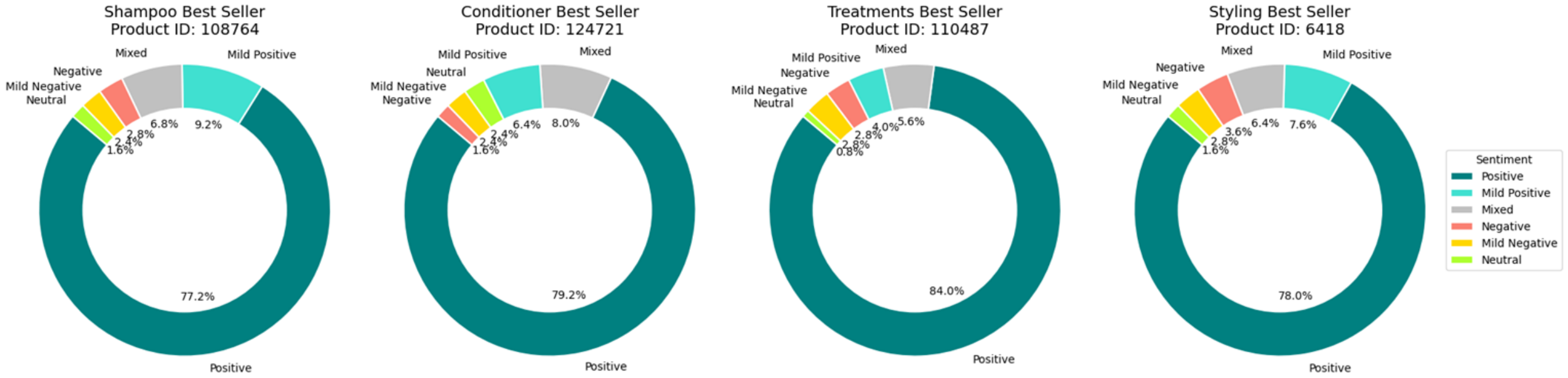
- More concentrated distribution (Points are tightly clustered)
- Consistent performance across different metrics (Less variability)
- Demonstrate more reliable performance



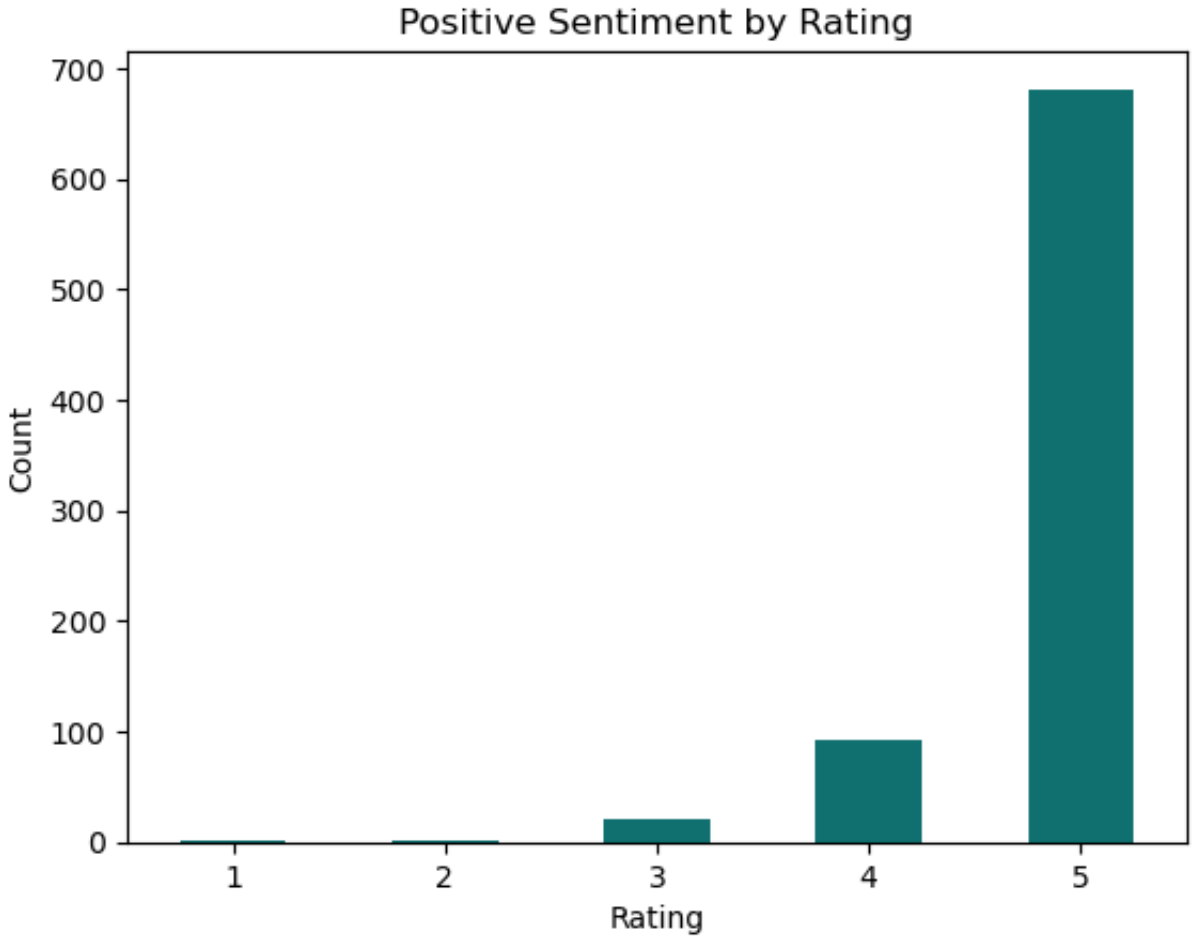
Insights from reviews

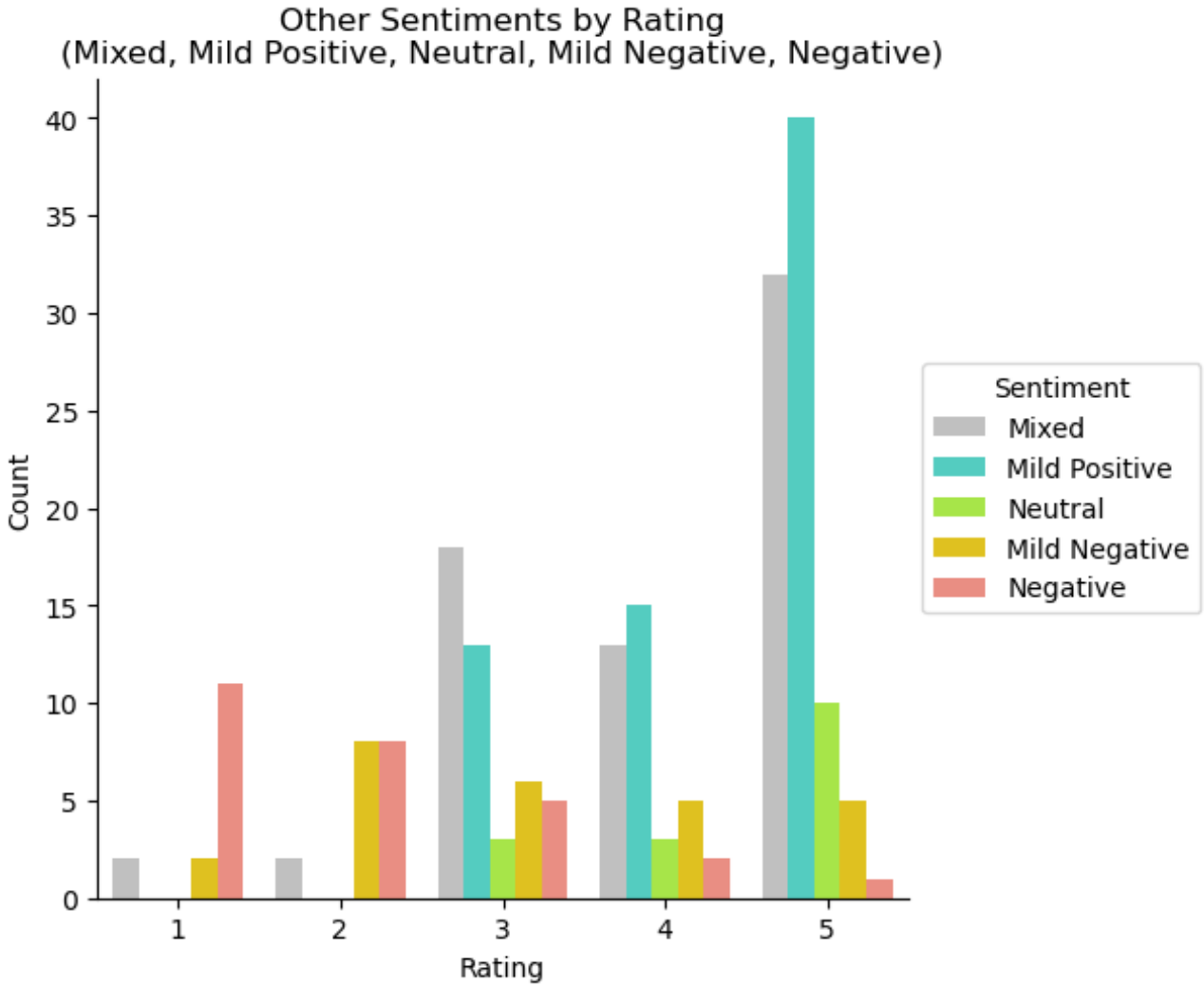
Customize sentiment classification thresholds:

- Positive: **`pos > 0.75`** and **`neg < 0.25`**
- Mild Positive: **`pos > 0.5`** and **`neg < 0.5`**
- Neutral: **`pos < 0.25`** and **`neg < 0.25`** and **`neu > 0.5`**
- Negative: **`neg > 0.75`** and **`pos < 0.25`**
- Mild Negative: **`pos < 0.5`** and **`neg > 0.5`**
- Mixed: Any instance that doesn't meet the criteria for the above categories (**pos** and **neg** are moderate; both around 0.3 to 0.5)



Sentiments by Rating





Based on our analysis, we examined the most positive, the most negative, as well as mixed reviews.

We found that mixed reviews are more reliable because

1. **Balanced View**: Capture a mix of positive and negative aspects
2. **Detailed Feedback**: Provide more detailed and specific details on product features
3. **Realistic Expectations**: Able to set accurate expectations for potential buyers
4. **Trustworthiness**: Seen as more credible and less biased, neither overly optimistic nor pessimistic
5. **Influence of Ambiguity**: Reflect varying outcomes based on individual preferences or use cases

Conclusions

Conclusions

- **Mielle** has the best sales in the past 30 days, with 3 of the 4 product categories ranking first.
- **SheaMoisture, Giovanni** and **Mielle** have four product categories on the best-selling list, which can be bundled into a series to sale to improve performance.
- **Cantu's** sales performance in hair styling is relatively good.

- Consumers are **most** willing to spend money on **treatments** and are **least** willing to spend money on **styling products.**
- Consumers like **good scent** hair care products with **great size**.
- In **shampoo, conditioner and treatments**, hair loss, greasy and oily products are not popular with consumers.
- Consumers use **styling products** to hold the hairstyle or keep their hair curly, but don't want their hair to become dry after use.

Conclusions

- **Dove** demonstrates **poor sales performance** on the iHerb platform.

  Dove did not rank as a best-seller. The dataset includes **72** Dove products, but **56 (or 78%)** had **zero sales** in the past 30 days. Additionally, Dove's highest popularity score is only 2.

- There are **7 products** rated **4 stars or above** that are not available in Hong Kong.
  It is recommended to introduce these products to **develop the Hong Kong market**.

Conclusions

- General **text preprocessing** steps may not always be required and largely **depend on the chosen sentiment analysis approach**.

- In our case, **RoBERTa** demonstrates **more consistent and reliable performance** across various metrics compared to VADER.

- In addition to Positive, Neutral, and Negative sentiments, the classification is further divided into **Mild Positive, Mixed, and Mild Negative** to provide deeper insights into customer reviews.

- The majority of **mixed sentiments** come from **ratings between 3 and 5**, making it worthwhile to investigate these reviews in more detail.

Limitations & Challenges

- Limitations

1. Less key information available
   - such as repeat purchases and customer loyalty metrics
1. Imbalanced distribution of ratings, heavily skewed towards high ratings
   - hinders the development of robust models for machine learning applications

- Challenges

1. Web scrapings
   - need to use various methods to break through anti-crawler measures on the iHerb website
2. Dataset selection
   - initially had different opinions on the theme, leading to delays due to an overload of ideas

Future Work

- A more comprehensive analysis
  - analyze lower-rated products (one-star and two-star)

- Year-over-year comparisons
  - comparison of sales of various brands in recent years
  - changes in best sellers in recent years

- Create a customized list of stopwords to enhance sentiment analysis
  - preserving negation words

- Train sentiment models using a balanced dataset
  - aiming to address overfitting issues

Reference

Dataset:
https://hk.iherb.com/c/hair-care

Use WebDriver to automate Microsoft Edge:
Use WebDriver to automate Microsoft Edge - Microsoft Edge Developer documentation | Microsoft Learn

How to create Tooltip Pages in Power BI
https://www.youtube.com/watch?v=npaQ42K1sTs

Instant Data Scraper & Octoparse
https://www.youtube.com/watch?v=0xzTzw6GQiw&t=87s

NLP & Sentiment Analysis Tutorial
https://www.kaggle.com/code/furkannakdagg/nlp-sentiment-analysis-tutorial