

NLP Sentiment Analysis for Amazon Products

By Juan Moctezuma-Flores

Intro - Sentiment Analysis for Amazon Product Reviews

- This project's goal is to analyze comments provided by customers by determining if the given review is associated with the following sentiment: 'positive', 'negative' or 'neutral'.
- In this data science project, we are exclusively analyzing reviews of every *whey protein* product (as of February 2023) for sell at Amazon. Whey protein powder is a sporting supplement (based on dairy milk) consumed commonly by athletes, body-builders and anyone who wants or needs additional protein on their diet.
- How do we obtain our data? By scraping (automatically extracting) pages one through ten from the comments' section for every single product available when searching for the terms 'Whey Protein' at Amazon's (E-Commerce website) searchbar.
- What is the main tool for this project? We used Python 3 – programming language for both web scraping and applying both multiple Natural Language Processing (NLP) techniques and statistical methods / data science prediction models.



Intro - Sentiment Analysis for Amazon Product Reviews (Part 2)



Image of 2 whey protein powder scoops by Harvard.Health.edu

Intro – Table of Contents

- Title – Page 1
- Introduction – Pages 2-3
- Web Scraping (Data or text extraction phase) – Page 5
- Data Preprocessing (Data or column manipulation) – Page 6
- Exploratory Data Analysis (Result summary) – Pages 7-8
- NLP Sentiment Prediction Analysis
 - *Vader* Model vs. *Roberta* – Pages 9-12
 - *Bert* and Linear Regression Model for Bert results' analysis – Pages 14-15
 - *TextBlob* approach – Pages 16-19
- Naïve Bayes Classifying Algorithm (Statistical Method) – Pages 20-23
- Business Analysis – Understanding the root cause of negative sentiment for quality improvement purposes – Pages 24-27
- Conclusion – Page 28
- Reference Links – Page 29

Web Scraping for Amazon Product Reviews

- BeautifulSoup is Python's library that is used for extracting data from Amazon's front-end (public website). A sample size of 100 comments (for each whey protein product during Q1 - 2023) from *See All Reviews* section gets scraped (and saved into a CSV file).

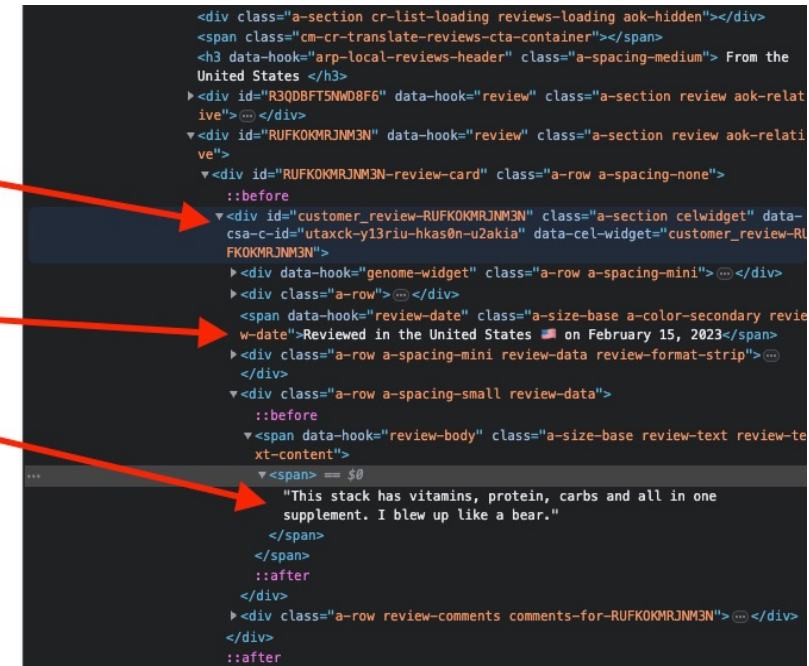
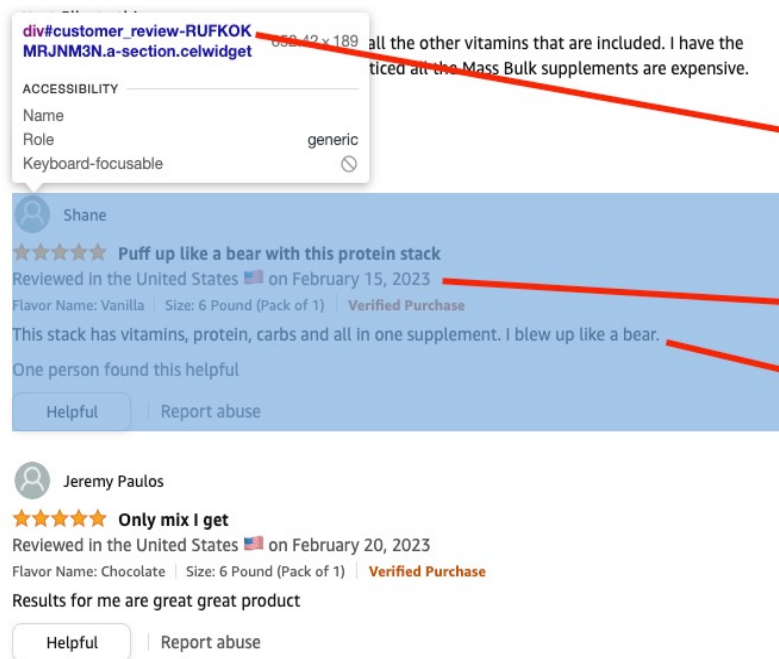


Image of Google Chrome's Developer tool showing the pages' sections and text from HMTL tags where the data comes from

Data Preprocessing for Reviews' Results

- Pandas is Python's library used for tabular data manipulation. Pandas reads the CSV file containing the web scraped text results. Unnecessary columns get removed, some data gets formatted correctly, and new IDs get created based off current webpage links.

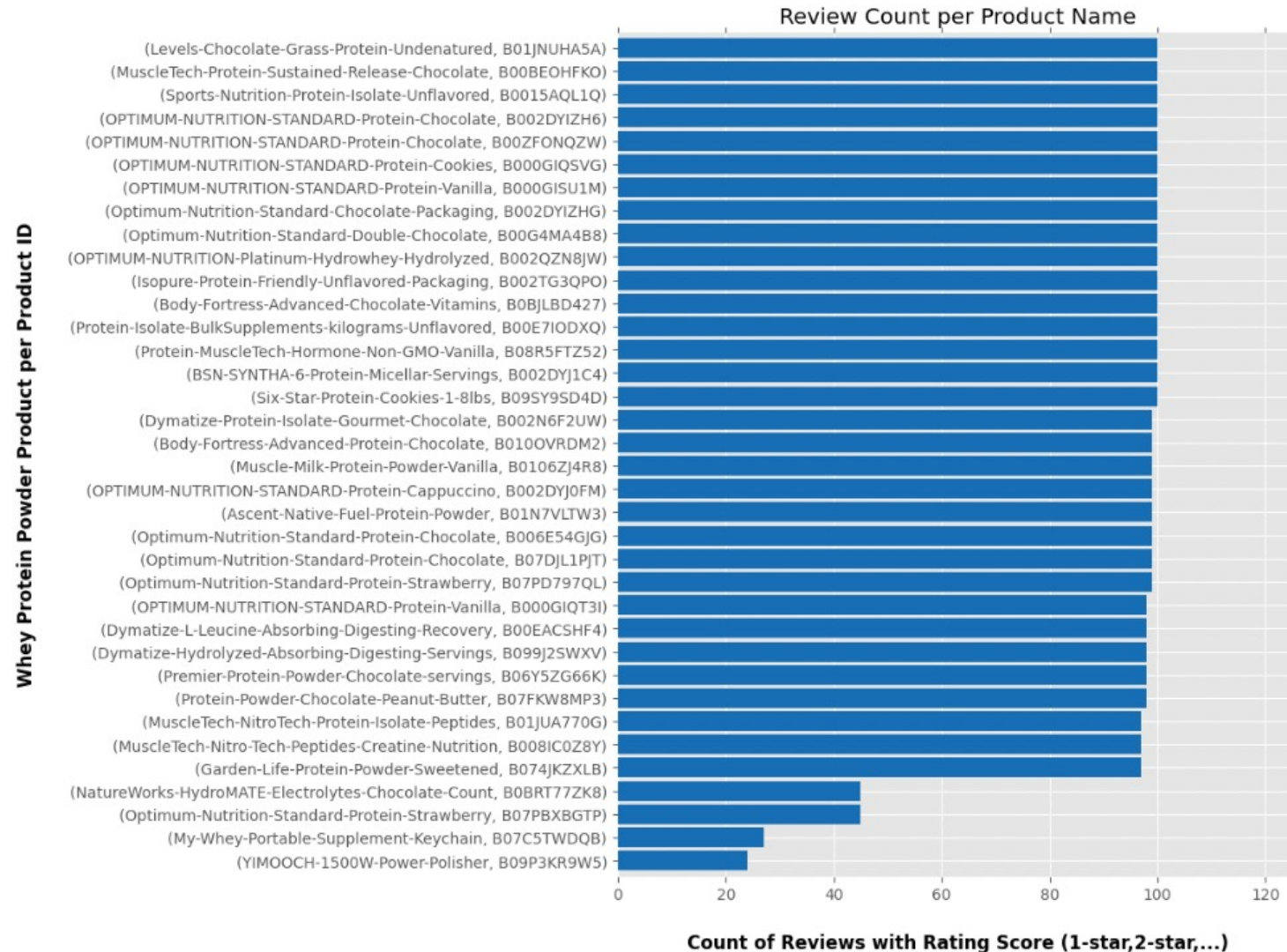
In [3]: `df.head(5)`

Out[3]:

	ID	Product_Name	Date	Rating_Score	Reviews	Link	Product_ID
0	0	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-25	5.0	I love this. I make it for myself and my kids...	https://www.amazon.com/NatureWorks-HydroMATE-E...	B0BRT77ZK8
1	1	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-02-06	5.0	Takes away lightheadedness and makes my husba...	https://www.amazon.com/NatureWorks-HydroMATE-E...	B0BRT77ZK8
2	2	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-27	5.0	The chocolate tastes delicious! I drink it ev...	https://www.amazon.com/NatureWorks-HydroMATE-E...	B0BRT77ZK8
3	3	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-27	5.0	I absolutely love this! My buddy gave me a fe...	https://www.amazon.com/NatureWorks-HydroMATE-E...	B0BRT77ZK8
4	4	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-02-18	4.0	I like to work out regularly. This includes w...	https://www.amazon.com/NatureWorks-HydroMATE-E...	B0BRT77ZK8

Image of Jupyter Notebook showing a few results from the author's preprocessing Python code

Exploratory Data Analysis for Preprocessed Results – Part 1



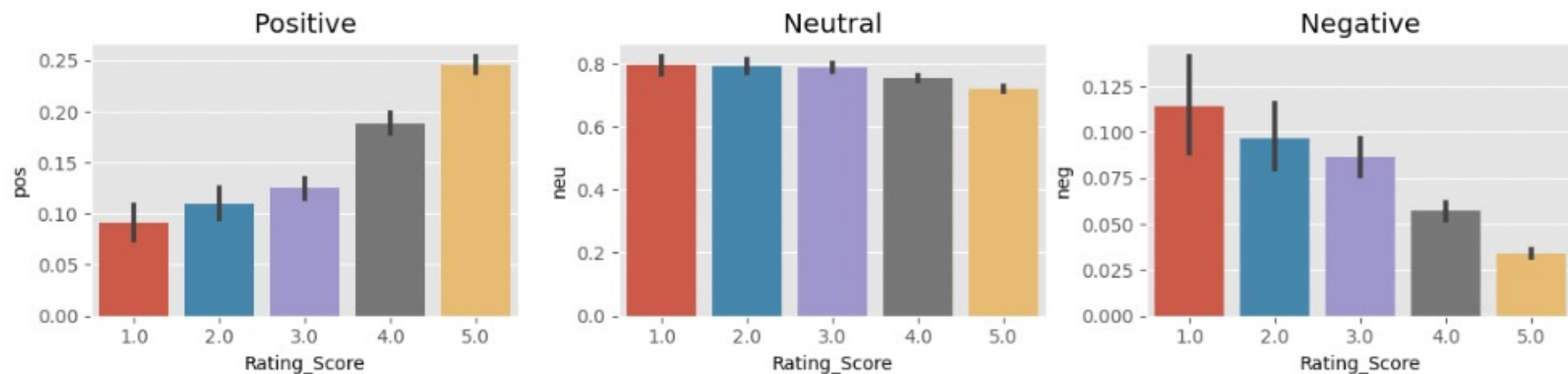
Exploratory Data Analysis for Preprocessed Results – Part 2

- Most customer rated-comments are biased towards the 'positive' sentiment (4 or 5-star reviews), which means that our quick exploratory analysis suggests that the overall quality of all the whey protein products available for sell at Amazon is high. Therefore, our sentiment analysis models should predict positive 'feelings' as well.
- Out of the 3314 rows of extracted data, 2100 reviews were given 5 stars & 738 reviews were given 4 stars.



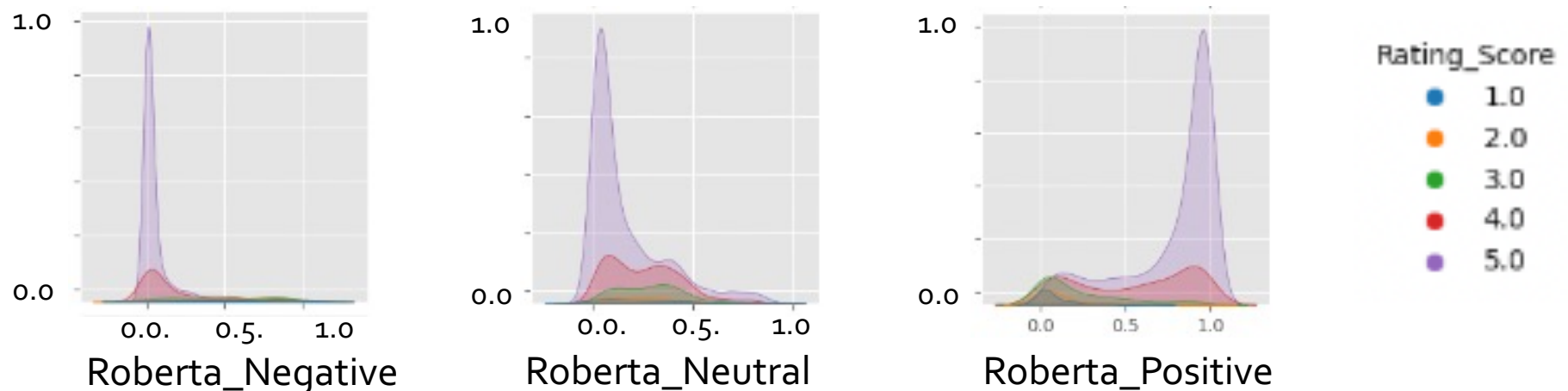
NLP Sentiment Analysis – Vader Model

- Vader (Valence Aware Dictionary & Sentiment Reasoner) pre-trained model is based on a 'bag of words' approach, which means that it goes over every word in a text or sentence and assigns a negative, neutral, positive decimal score (which adds up to a sentiment compound value).
- The disadvantage of this model is that it doesn't take into consideration human tone, sarcasm, relationship among words, context, etc.
- Observe the 'positive' chart below - Vader predicts a 'positive' sentiment towards customers who provided a 5-star comment. Interestingly, the 'neutral' chart is slightly biased towards a 'negative' prediction.

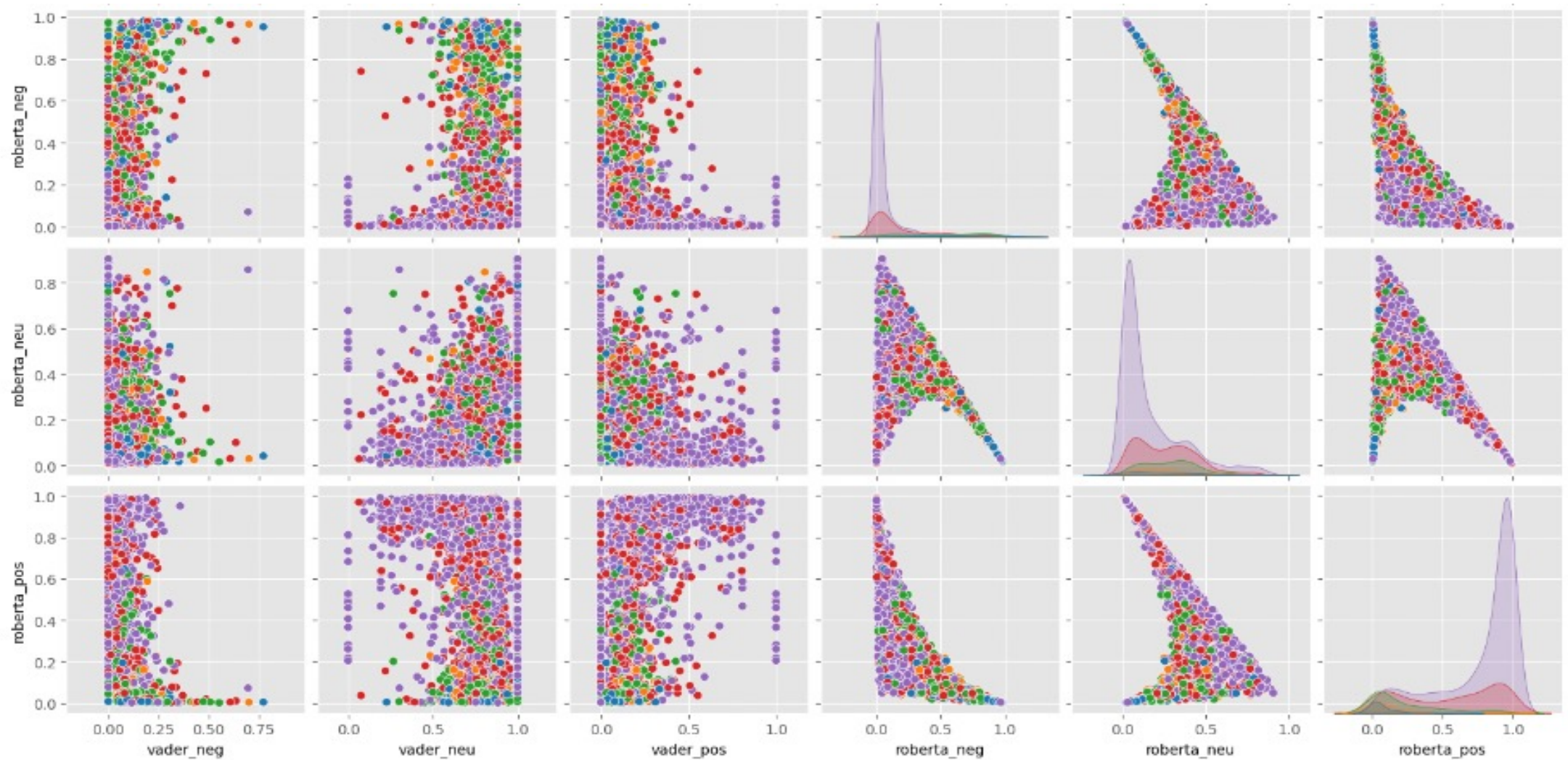


NLP Sentiment Analysis – Roberta Model

- A Robustly Optimized BERT Pretraining Approach (Roberta) is a pre-trained model that goes over every word and its respective combinations in a text or sentence; and assigns a negative, neutral, positive decimal score.
- The advantage of this model is that it does take into consideration human tone, sarcasm, relationship among words, context, etc.
- Observe the 'Roberta_Positive' chart below - Roberta predicts a 'positive' sentiment towards customers who provided 4 and 5-star comments (as expected). Interestingly, the 'neutral' and 'negative' charts are slightly biased towards a 'negative' prediction for some 4 and 5-star comments .



NLP Sentiment Analysis – Vader vs. Roberta (Part I - Comparison)



NLP Sentiment Analysis – Vader vs. Roberta (Part II - Conclusion)

- Most comments for every whey protein product offered at Amazon were given 5-stars, hence most data points are purple (assigned by Python to comments with rating score of 5) – *See image below as a reference for slide 11.*
- Notice that there are correlations between Roberta & Vader - it's hard to tell but notice that every Roberta vs Roberta graph reflect 'skewed' or 'V-shaped'-looking pattern. However, every Roberta vs Vader graph doesn't reflect a 'well-defined' pattern – as the data points are mostly scattered.
- In conclusion, Vader model produces less confident / precise results – the reason being that Vader predicts sentiment scores based on words only, while Roberta is highly trained as that model considers not only words, but combinations and context or relationship between words as well.



NLP Sentiment Analysis – Bert Model (Part I)

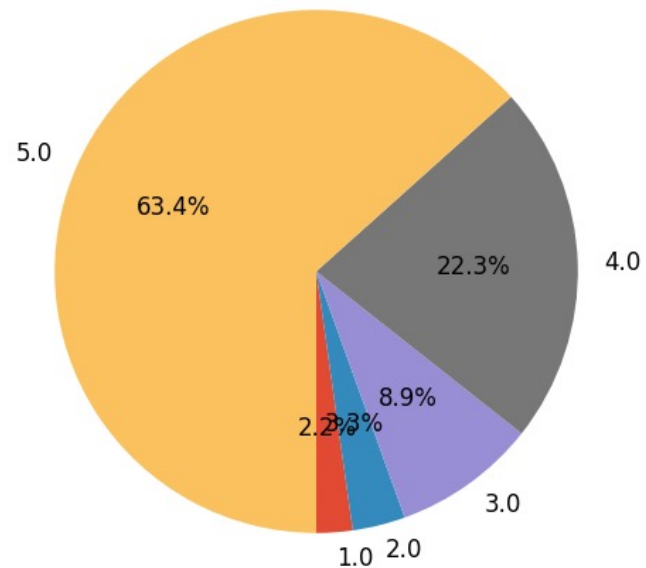
- Bert (Bidirectional Encoder Representations from Transformers) is a pre-trained model that goes from left to right, and right to left over every word in a text or sentence and assigns a negative, neutral, positive decimal score (which get rounded to an integer to simulate a predicted 'star'-rating). Note that Roberta is the extended version of Bert, that is why we are not comparing it to Roberta Model, nor using decimal-sentiment values.
- The disadvantage of this model is that it is currently limited to 514 sequences per text, meaning that if a comment is too long, it won't provide an accurate sentiment prediction compared to shorter comments (hence less precise when comparing Bert's results with the actual number of stars).

	ID	Product_Name	Date	Rating_Score	bert_sentiment	Reviews	Product_ID
0	0	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-25	5.0	5	I love this. I make it for myself and my kids...	B0BRT77ZK8
1	1	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-02-06	5.0	5	Takes away lightheadedness and makes my husba...	B0BRT77ZK8
2	2	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-27	5.0	5	The chocolate tastes delicious! I drink it ev...	B0BRT77ZK8
3	3	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-27	5.0	5	I absolutely love this! My buddy gave me a fe...	B0BRT77ZK8
4	4	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-02-18	4.0	4	I like to work out regularly. This includes w...	B0BRT77ZK8

Image of Jupyter Notebook showing a few Bert sentiment (integers) results from the author's script

NLP Sentiment Analysis – Bert Model (Part II)

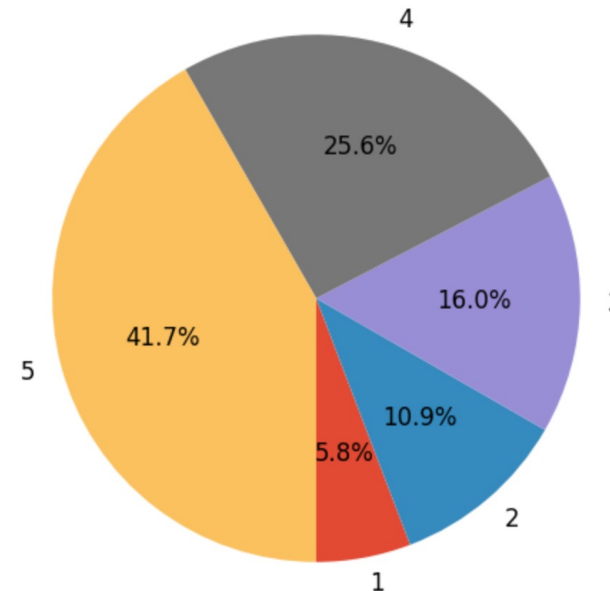
Actual Results



Rating_Score

```
1.0      74
2.0     108
3.0     294
4.0     738
5.0    2100
dtype: int64
```

Bert Prediction Results



bert_sentiment

```
1      192
2     361
3     529
4     849
5    1383
dtype: int64
```

NLP Sentiment Analysis – Bert Model (Part III - Conclusion)

- If we compare the pie chart results, noticed that Bert's sentiments are less biased towards a score of a '5-star' review (unlike the actual customers' rating scores), hence we can expect this model to be less precise and less efficient when it comes to prediction.
- However, Bert's model was still able to predict that most comments were given 5 and 4 star-ratings. Let's keep in mind that the margin of error increases as Bert can't predict long text (no more than 514 sequences) unlike Roberta and Vader which do not have this technical limitation or condition.
- The percentage errors are the following using the error formula $((Estimated\ Number - Actual\ Number) / Actual\ number) \times 100$: (See previous charts as a reference)

$$E_5 = 5\text{-Star Comment Error} = ((1383 - 2100) / (2100)) \times 100\% = 34.14\%$$

$$E_4 = 4\text{-Star Comment Error} = ((849 - 738) / (738)) \times 100\% = 15.04\%$$

$$E_3 = 3\text{-Star Comment Error} = ((529 - 294) / (294)) \times 100\% = 79.93\%$$

$$E_2 = 2\text{-Star Comment Error} = ((361 - 108) / (108)) \times 100\% = 234.26\%$$

$$E_1 = 1\text{-Star Comment Error} = ((192 - 74) / (74)) \times 100\% = 159.46\%$$

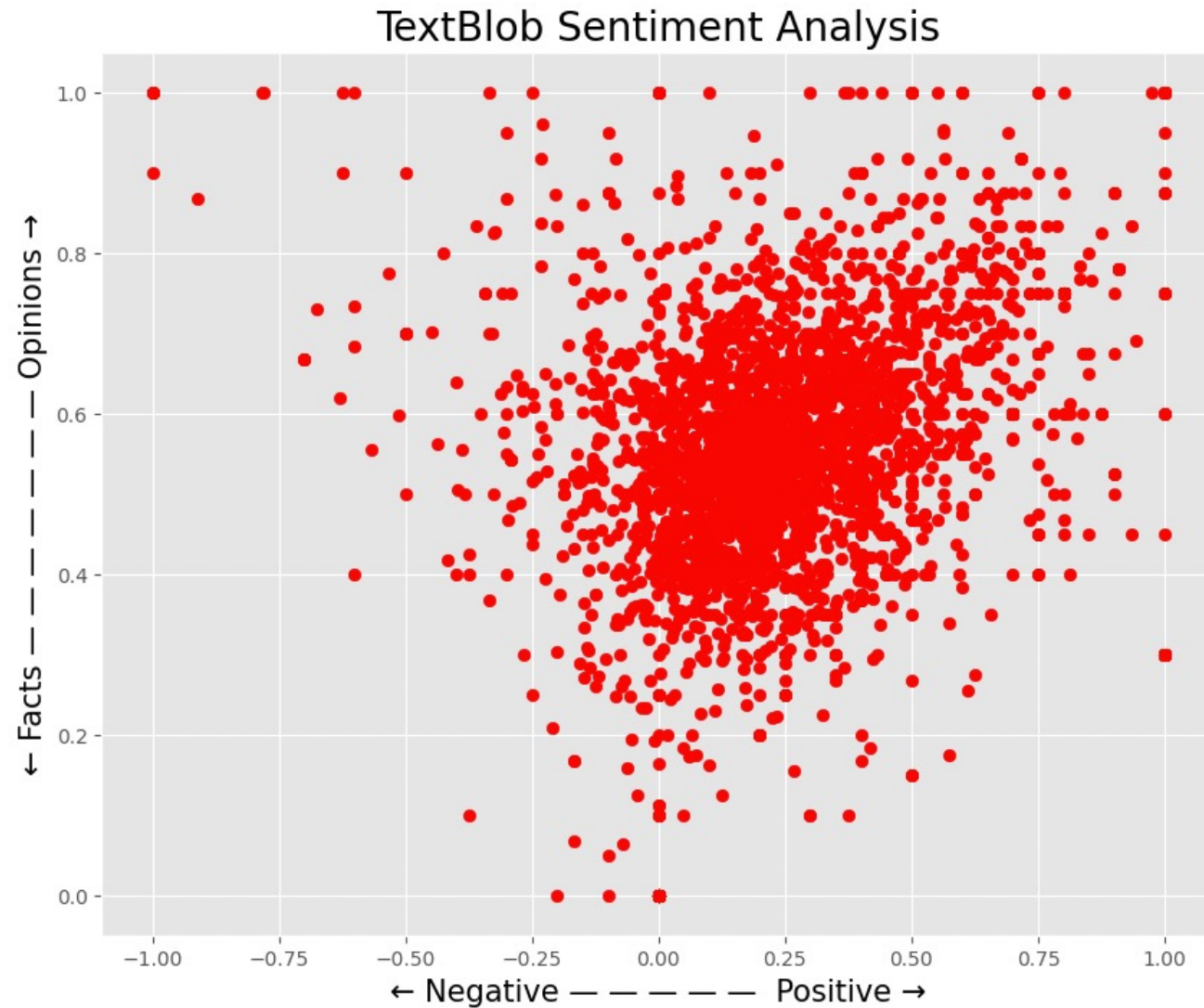
NLP Sentiment Analysis – TextBlob Model (Part I)

- TextBlob is a Python library for processing textual data. In addition to sentiment analysis, TextBlob also specializes on speech tagging, noun phrase extraction, classification, and more functions related to natural language processing.
- The output produced by this library is polarity and subjectivity:
 - Polarity score lies between (-1 to 1) where -1 identifies negative words (such as 'bad', 'awful', 'pathetic', etc) and 1 identifies positive words (like 'excellent', 'best', etc.).
 - Subjectivity score lies between (0 and 1), and it shows the amount of 'opinion'. If a sentence has high subjectivity (a value is closer to 1), it resembles that the text contains more opinion-based sentiment than facts.

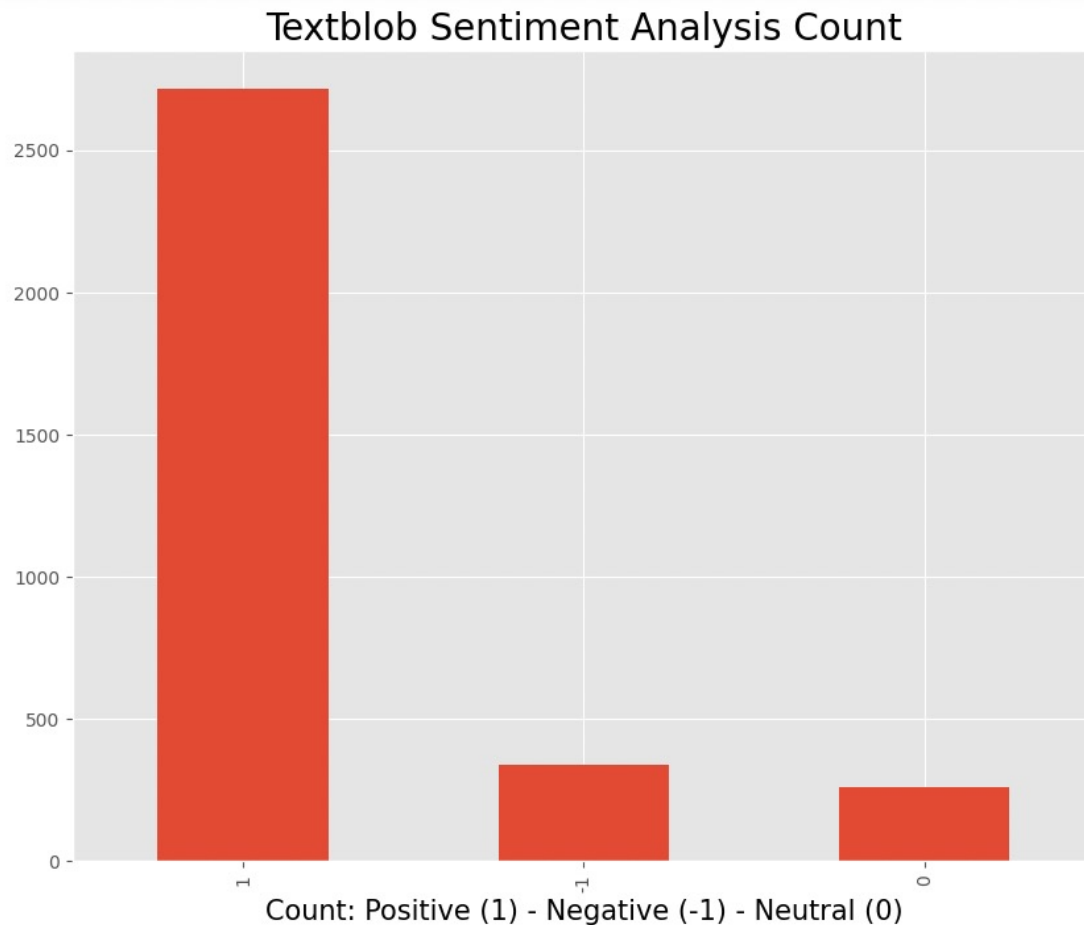
ID		Product_Name	Date	Rating_Score	textblob_polarity	textblob_subjectivity	textblob_analysis	Reviews	Product_ID
0	0	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-25	5.0	0.456250	0.628125	1	I love this. I make it for myself and my kids...	B0BRT77ZK8
1	1	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-02-06	5.0	0.000000	0.400000	0	Takes away lightheadedness and makes my husba...	B0BRT77ZK8
2	2	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-27	5.0	0.500000	0.750000	1	The chocolate tastes delicious! I drink it ev...	B0BRT77ZK8
3	3	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-01-27	5.0	0.406250	0.575000	1	I absolutely love this! My buddy gave me a fe...	B0BRT77ZK8
4	4	NatureWorks-HydroMATE-Electrolytes-Chocolate-C...	2023-02-18	4.0	0.171118	0.582585	1	I like to work out regularly. This includes w...	B0BRT77ZK8

Image of Jupyter Notebook showing a few TextBlob results from the author's script

NLP Sentiment Analysis – TextBlob Model (Part II)



NLP Sentiment Analysis – TextBlob Model (Part III)



```
textblob_analysis  Rating_Score
-1      339      1.0      74
0       261      2.0     108
1      2714      3.0     294
dtype: int64      4.0     738
                  5.0    2100
dtype: int64
```

- 'One' from the TextBlob analysis' perspective would be equivalent to 4 & 5-star comments (positive sentiment). Hence a total of 2838 reviews.
- 'Zero' would be equivalent to every 3-star reviews (neutral sentiment). Hence a total of 294 reviews.
- 'Minus One' would be equivalent to 1 & 2-star reviews (negative sentiment). Hence a total of 182 reviews.

NLP Sentiment Analysis – TextBlob Model (Part IV - Conclusion)

- Scatterplot conclusion: As expected, TextBlob predictions about whey protein products are biased towards positive comments/sentiments as well. However, most results from the sentiment scatterplot scored higher in subjectivity (meaning that most reviews are mostly opinion-based as expected).
- This model has a relatively low error percentage when comparing TextBlob's positive sentiment (1). However, a high error percentage when it comes to negative sentiment (-1), meaning that TextBlob's negative predictions are inaccurate and have low precision. Please see the percentages at the bottom of the slide.
- The percentage errors are the following using the error formula $((Estimated\ Number - Actual\ Number) / Actual\ number) \times 100$: (See previous charts as a reference)

$$E_{-1} = 1\text{-Star} + 2\text{-Star Comment Error} = ((339 - 182) / (182)) \times 100\% = 86.26\%$$

$$E_0 = 3\text{-Star Comment Error} = ((261 - 294) / (294)) \times 100\% = 11.22\%$$

$$E_1 = 4\text{-Star} + 5\text{-Star Comment Error} = ((2714 - 2838) / (2838)) \times 100\% = 4.37\%$$

Statistical Technique – Naïve Bayes Classifier (Part I)

- Naïve Bayes classifier will not be doing sentiment analysis based on a corpus of text directly. It will calculate the probability of customers liking current whey protein products from Amazon vs. the probability of NOT liking the products. This method classifies whether consumers will provide a positive comment OR a negative review (regardless of the number of stars given to the rated feedback).
- We 'feed' the algorithm / classifier with our scraped reviews' rating scores to refine our pre-existing belief – which is that most customers will probably like the current available whey protein products.
- Mathematically, the Bayes theorem is stated as: **$P(A|B) = P(B|A) * P(A) / P(B)$**
 - $P(A|B)$ is the probability of event A occurring, given that event B has occurred
 - $P(B|A)$ is the probability of event B occurring, given that event A has occurred
 - $P(A)$ is the probability of the event A
 - $P(B)$ is the probability of the event B

Statistical Technique – Naïve Bayes Classifier (Part II)

- Recall that previous models predicted 3-star comments differently - some 'neutral' reviews were more biased towards negative sentiment than positive. Hence - For us to test and train this statistical model on Python, we need to make 2 assumptions:
 - We assume 'positive' (1) sentiment is given when the review was given either 3, 4 or 5 stars, or 'negative' (0) sentiment when feedback was rated with 1 and 2 stars. Hence, we get a total of 3132 'positive' reviews and 182 'negative' reviews.
 - We assume 'positive' (1) sentiment is given when the review was given either 4 or 5 stars, or 'negative' (0) sentiment when feedback was rated with 1, 2 or 3 stars. Hence, we got a total of 2838 'positive' reviews and 476 'negative' reviews.

```
Rating_Score
1.0          74
2.0         108
3.0         294
4.0         738
5.0        2100
dtype: int64
```

Actual Results

Statistical Technique – Naïve Bayes Classifier (Part III)

```
In [7]: # Tfidf Vectorizer
stopset= set(stopwords.words('english'))
vectorizer= TfidfVectorizer(use_idf = True, \
                             lowercase = True, \
                             token_pattern = '[a-zA-Z0-9+#+-/*][^\s]', \
                             strip_accents = 'ascii', \
                             stop_words = stopset)

In [8]: # A) We assign new column with values of 'one' to 3, 4 & 5 star-comments, and a 'zero' to 1 & 2 star-comments
# B) We assign new column with values of 'one' to 4 & 5 star-comments, and a 'zero' to 1, 2 & 3 star-comments
df['sentiments'] = df['Rating_Score'].apply(lambda x: 0 if x in [1, 2] else 1)
#df['sentiments'] = df['Rating_Score'].apply(lambda x: 0 if x in [1, 2, 3] else 1)

In [9]: # In this case our dependant variable will be 'sentiments' as 0 (didn't liket)
# OR 1 (did like the product or are neutral)
y = df.sentiments.values
X = df.Reviews.values
encoder = LabelEncoder()
encoded_labels = encoder.fit_transform(y)

In [10]: # We split the data into 80% train and 20% test parts
X_train, X_test, y_train, y_test = train_test_split(X, encoded_labels, stratify = encoded_labels)

In [11]: # We store words in a dictionary called 'word_counts'. All the unique words in the corpus are stored in 'vocab'
vec = CountVectorizer(max_features = 3000)
X_train = vec.fit_transform(X_train)
vocab = vec.get_feature_names_out()
X_train = X_train.toarray()
word_counts = {}
for l in range(2):
    word_counts[l] = defaultdict(lambda: 0)
for i in range(X_train.shape[0]):
    l = y_train[i]
    for j in range(len(vocab)):
        word_counts[l][vocab[j]] += X_train[i][j]

In [12]: # We need to perform Laplace smoothing to handle words in the test set which are absent in the training set.
# We define a function 'laplace_smoothing' which takes the vocabulary and the raw 'word_counts' dictionary and
# returns the smoothened conditional probabilities.
def laplace_smoothing(n_label_items, vocab, word_counts, word, text_label):
    a = word_counts[text_label][word] + 1
    b = n_label_items[text_label] + len(vocab)
    return math.log(a/b)
```



Orange arrow shows the section of the Python code where the 'sentiment assumptions' are made (see slide 21 for more info.)

Statistical Technique – Naïve Bayes Classifier (Part IV - Conclusion)

- Results for the 1st assumption (case when our hypothesis states that 3-star ratings are biased towards 'positive' sentiment) reflect a prediction accuracy of **94.45%**, which means that our machine learning algorithm was able to predict a 'positive' sentiment with a high-level of accuracy. This is because statistically speaking, customers tend to give high rating scores to whey protein products at Amazon.
- Results for the 2nd assumption (case when our hypothesis states that 3-star ratings are biased towards 'negative' sentiment) reflect a prediction accuracy of **85.65%**. Accuracy decreased by 8.80% when making this assumption.
- Naïve Bayes classifier model is nothing more sophisticated than a simple probability model. Due to well-separated categories ('1' for positive and '0' for negative), the model can provide straightforward probabilistic prediction; in this situation – high accuracy translates as 'More than 85% of Amazon protein consumers will have a 'positive' sentiment towards this product'.

Business Analysis - Understanding Negative Sentiment (Part I)

- Our Machine Learning / Statistical models certainly predict with a high level of accuracy of positive sentiment due that most customers are satisfied with the current whey protein products available at Amazon. However, it is important to understand the root cause of 1 and 2-star comments or negative comments.
- How is this relevant? Data-driven insights will allow whey protein producers to understand the root-cause of the customers' negative sentiments. In addition, unrelated feedback or complaints may also be useful for monitoring the quality of other services associated with Amazon – such as shipping, package handling, etc.



Image by Nicolas Herrbach

Business Analysis - Understanding Negative Sentiment (Part II)

- After reviewing multiple comments labelled as 'negative' according to the previously mentioned classification models (except Naïve Bayes' classifier) , we noticed the most common words or sentences are related with:
 - Bad taste (Peanut Butter and 'Cookies n Cream' being the most common flavors)
 - Bad texture when mixed with liquids (mostly due to 'lumps')
 - Instances or occasional cases of 'bad batches'
 - Expensive relative to low quality
 - Nausea or stomach aches
 - Bad aftertaste
 - Excessive sugary or sweet flavors
 - Bad aroma
 - Quality decrease according to frequent buyers or consumers
 - Small or 'Incorrect' scoop sizing
 - Cases in which the package is damaged – seal is broken (specific issue)

Disclaimer: To protect the reputation and identity of popular brands and whey protein producers and/or sellers – no specific product names will be listed on this presentation (scraped reviews are posted on GitHub instead).

Business Analysis - Understanding Negative Sentiment (Part III)

```
In [92]: # TextBlob - 'Negative' / Polarity comment given by customer
textblob_neg_p = []
for i in range(0, 338):
    result = df2.query('textblob_analysis == -1') \
        .sort_values('textblob_polarity', ascending = False)['Reviews'].values[i]
    textblob_neg_p.append(result)
```

```
In [95]: print(textblob_neg_p[80]) # Bad Taste
```

I like [REDACTED] products. I enjoy their standard chocolate flavor a lot, so when I saw their chocolate peanut I thought it would be tasty. [REDACTED] I was wrong. The aftertaste is unbearable. Whoever approved this flavor deserves the death penalty.

```
In [99]: print(textblob_neg_p[300]) # Bad Taste # Bad Texture
```

Hard to choke down. Tastes so bad.

```
In [101]: print(textblob_neg_p[337]) # Bad Aftertaste
```

Aftertaste is terrible.

```
In [102]: print(textblob_neg_p[37]) # Bad Texture
```

Taste is good in a smoothie. My only small complaint is that it does not always mix well. It sometimes leaves deposits on the side of the blender. The compatible flavor outweighs that small issue.

```
In [107]: print(textblob_neg_p[29]) # Shipping Issue - Portion of product is missing
```

The tub doesn't come full FYI. 1/4 missing. My teenager almost got in trouble over the missing protein powder. I thought she had a protein shake party while I was at work.

```
In [108]: print(textblob_neg_p[215]) # Bad taste - Peanut Butter
```

Milk chocolate is good and I generally like gold standard protein but the peanut butter chocolate is disgusting. It doesn't even resemble the taste of peanut butter. There is not even a derivative of peanut butter in the ingredients. Just tastes like a chemically laden failed lab experiment to create a peanut butter flavor.

Image of multiple/random TextBlob-negative sentiment results on Jupyter Notebook

Business Analysis - Why is Negative Feedback Important? (Part IV)

- Negative experiences allows businesses to learn from their mistakes and to continuously improve the quality of their products and services.
- When it comes to whey protein products offered by Amazon, we learned the following from the Ai-driven sentiment analysis predictions:
 - Some artificial flavors simply don't blend in as sports supplements due to unpleasant taste and aroma – Probably flavors such as Peanut Butter, require further studies or changes in its formula or recipe.
 - Some brands' powder consistency has an unpleasant consistency or texture when stirring in liquid – Probably some ingredients need further investigation as well.
 - The most severe aspect that needs immediate fix would be the cases of broken seals, as this problem causes the product to spill or get contaminated during the shipping process. Although this issue is not too common – it is essential to verify if the products need better packaging, or if these require better handling during shipping processes.



Conclusion

- Machine Learning or Ai-powered sentiment analysis is an essential tool or component for businesses to understand how the public feels about their products or services.
- Web Scraping techniques can be applied to any website and product! Only if their respective online policies allow for data extraction.
- There are many Machine Learning models available for developers, however – for the sake and purpose of this project – only Vader, Roberta, Bert (Linear Regression included), TextBlob, and Naïve Bayes' algorithm were required for the completion of the analysis.
- Sentiment analysis allows businesses to use software that quickly and efficiently uses hundreds or thousands of data points to provide valuable insights.



Image by Swasthum.com

Reference Links

- <https://www.trueprotein.com.au/blogs/nutrition/what-is-whey-protein>
- <https://www.analyticsvidhya.com/blog/2021/06/vader-for-sentiment-analysis/>
- <https://www.geeksforgeeks.org/overview-of-roberta-model/>
- <https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/>
- <https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/>
- <https://www.analyticsvidhya.com/blog/2021/10/making-natural-language-processing-easy-with-textblob/>
- <https://www.datacamp.com/tutorial/naive-bayes-scikit-learn>
- <https://www.analytixlabs.co.in/blog/naive-bayes-machine-learning/>
- <https://www.amazon.com/>
- <https://www.analyticsvidhya.com/blog/2021/10/a-detailed-guide-on-web-scraping-using-python-framework/>