Logo

Description automatically generated

Department of

**School of Engineering Technology and Applied Science (SETAS)**

COMP262 – NLP & Recommender System

**Project Report - Sentiment Analysis Model​ Phase #1**

Group 5:

Mahpara Rafia Radmy #301176893

Manipal Sidhu #300859319

Ronald Saenz Huerta #301218602

Kanishka Dhir #301220757

Kaushalkumar Pandya #301217897

Prepared for:

Professor Mayy Habayeb

Table of Contents

[1. Data Exploration 2](#_Toc329826332)

[1.1. Dataset Info 3](#_Toc1757128624)

[1.2. Descriptive Statistics 3](#_Toc1218862189)

[1.3. Correlation Analysis 4](#_Toc1565043833)

[1.4. Distribution Of Ratings and Reviews 5](#_Toc2009142367)

[1.5. Conclusion 6](#_Toc813146008)

[2. Dataset Pre-processing 6](#_Toc1207897581)

[2.1. Basic Dataset Pre-processing 7](#_Toc1291428776)

[2.2. Dataset Pre-processing for TextBlob model 9](#_Toc1797483531)

[2.3. Dataset Pre-processing for VADER model 11](#_Toc1358939769)

[3. Models 12](#_Toc669286571)

[3.1. TextBlob 12](#_Toc1331920002)

[3.1.1. Assumptions/Heuristics/algorithms used 12](#_Toc391848227)

[3.1.2. How it works 12](#_Toc513216070)

[3.1.3. External Datasets 13](#_Toc778843874)

[3.2. Valence Aware Dictionary and Sentiment Reasoner (VADER) 14](#_Toc1806736182)

[3.2.1. Assumptions/Heuristics/algorithms used 14](#_Toc1131609986)

[3.2.2. How it works 14](#_Toc2141428601)

[3.2.3. External Datasets 16](#_Toc515552106)

[4. Testing results summary 16](#_Toc788646772)

[4.1. TextBlob 17](#_Toc1110157593)

[4.2. Vader 17](#_Toc697578761)

[4.3. Model Comparison 17](#_Toc1728152950)

[5. Final Conclusion 19](#_Toc240010153)

[6. References 20](#_Toc1070278626)

[7. Appendix 1: Project plan 21](#_Toc509835489)

[8. Appendix 2: Meeting Register 22](#_Toc608304301)

# Data Exploration

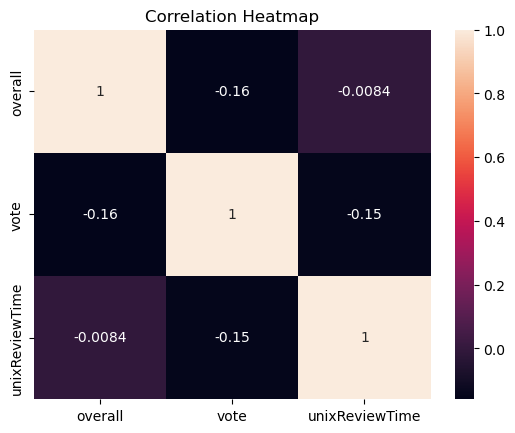
## Dataset Info

The dataset under consideration comprises customer reviews for gift cards. It encapsulates a variety of information pertaining to reviews, reviewers, and products. It encompasses 2,972 reviews for 148 unique gift card products and provides a comprehensive insight into customer sentiments and opinions. Each review was contributed by one of the 458 distinct reviewers. On an average, products have received a high rating of approximately 4.89 with 2,838 reviews being verified and each review fetching around 5.16 votes. The dataset is organized across 12 different columns, each serving a specific purpose. The “overall” column reflects the numerical rating given to a product, while “verified” indicates whether a review is authenticated. The dataset also records textual details with columns like “reviewerID” capturing the unique identifier of a reviewer, and “asin” pinpointing the specific gift card product. Other descriptors such as “reviewerName”, “reviewText”, “summary”, “image”, “style”, and “reviewTime” further enrich the dataset by offering nuanced insights into the reviewer’s identity, sentiments, and the timing of the review. Lastly, the “unixReviewTime” column provides a timestamp for each review, and the “vote” column quantifies the popularity or impact of a review through the number of votes it has garnered.

## Descriptive Statistics

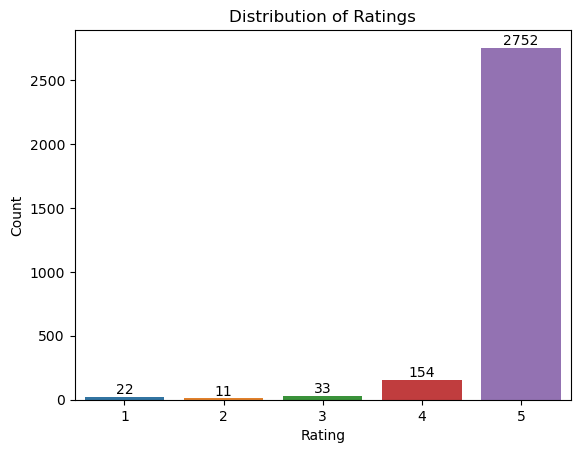
The descriptive statistics shed light on the dataset's attributes. With 2,972 reviews captured, the average rating sits at a commendable 4.88 on a 5-point scale. Reviews span across various timestamps, with "unixReviewTime" falling between 1.33e+09 and 1.53e+09. Notably, out of all reviews, 208 have been actively engaged with, averaging 5.16 votes each.

## Correlation Analysis

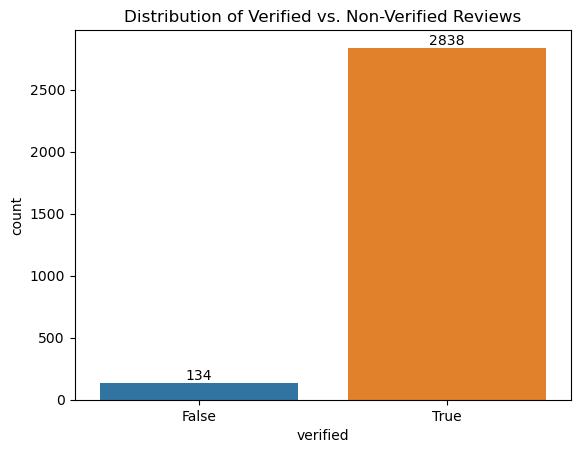


The heatmap shows the correlation between three columns (overall, vote, unixReviewTime). Specifically, there’s a mild negative association between ‘overall’ ratings and ‘vote’, denoted by a -0.16 correlation. Furthermore, the connection between ‘overall’ ratings and ‘unixReviewTime’ is very slight almost negligible at -0.0084. Lastly, ‘vote’ and ‘unixReviewTime’ also exhibit a negative relationship with a correlation of -0.15, hinting that as the review time progresses, there might be a slight decrease in the votes.

## Distribution Of Ratings and Reviews



The graph shows how people rated from 1 to 5. Most people, about 93% (2,752 out of 2,972), gave a 5-star rating, showing they were very happy. Only about 0.74% (22 out of 2,972) gave a 1-star rating, 0.37% (11 out of 2,972) gave a 2-star, and 1.11% (33 out of 2,972) gave a 3-star rating. A slightly larger group, about 5.18% (154 out of 2,972), gave a 4-star rating. So, the majority had a great experience. The average rating or score in the "overall" column is approximately 4.89.



The graph displays the distribution of verified versus non-verified reviews. About 95.5% (2,838 out of 2,972) of the reviews are verified, showing a high level of authenticity or validation in the review process. The remaining 4.5% (134 out of 2,972) of the reviews are not verified.

## Conclusion

The data exploration reveals a predominantly positive sentiment from the reviewers. The combination of a high average rating with the substantial proportion of verified reviews suggests that the feedback is both positive and credible. The product, service, or content under review seems to be well-received by its users.

# Dataset Pre-processing

The data pre-processing is a crucial step in NLP projects to clean, transform, and organize the data with the goal to prepare for further analysis. The dataset is related to reviews from Amazon Gift Cards which were written by users with their personal opinions. However, the dataset could contain noise in the form of special characters, blank spaces, digits, emoticons, emojis, URLs, and other irrelevant elements, which can impact the performance of NLP, Machine Learning, or AI models. Therefore, it is necessary to conduct comprehensive dataset preprocessing to prepare the data for modeling.

## Basic Dataset Pre-processing

Firstly, it is necessary to make a basic Dataset Pre-processing to clean, transform, and organize the data. This step is crucial for modeling because it is necessary to decide what columns/features should be dropped, modified, merged, or added.

For this project, there are some steps to perform the dataset before modeling:

1. Removed all non-verified records

For this step, it was necessary to check the original dataset which contains a column called “verified” which is related to the authenticity of the user review. It could help to determine if the user is a real person or a robot. In the data exploration, the distribution of Verified and Non-Verified Reviews shows 134 reviews that are not verified.

1. Dropped unwanted columns

To perform the sentiment analysis processing, it is necessary to drop unnecessary columns that do not contribute anything. After to analyze the dataset, there are some columns that should be dropped such as 'verified', 'reviewerID', 'asin', 'reviewerName', 'reviewTime', 'style', 'unixReviewTime', 'vote', and 'image'.

1. Dropped duplicates

To maintain data integrity, it is necessary to eliminate duplicate records. There are 892 duplicated records that should be eliminated from the dataset.

1. Labeled the data based on the value of “rating of the product”

After to check the original dataset, it contains a column called “overall” which is associated to the rating of the product. It is a form to categorize each review using the rating. Consequently, it was necessary to assign a sentiment analysis, using the following logic: ratings 4 and 5 as “Positive”, rating 3 as “Neutral”, and ratings 1, and 2 as “Negative”

1. Merge both columns “reviewText” and “summary” into a new column called “text”

After to check the original dataset, it contains two columns called “reviewText” and “summary” that have text related to the user review. The consolidation of both columns in a new column called “text” help to simplified subsequent text analysis processes, fostering a more streamlined analytical workflow.

## Dataset Pre-processing for TextBlob model

For TextBlob model, it was necessary to make some dataset pre-processing to perform the dataset.

For this project, there are some steps to perform the dataset before modeling:

1. Exclusion of Special Characters and Digits

This step is necessary to streamline the data, ensuring that the model focused only in the linguistic that could carry sentiment. Numbers or special characters could be irrelevant and cause some noise for modeling.

1. Trimming trailing whitespaces

To avoid some noise into the data, it is necessary to trim trailing whitespaces. This step is necessary to maintain consistency.

1. Punctuation removal

To avoid some noise into the data, it is necessary to remove punctuation. The punctuation could sometimes distort the meaning of the sentence, and this step fosters a cleaner analysis. For that reason, this step is crucial to improve the accuracy of the sentiment analysis.

1. URL elimination

Web links do not contribute to sentiment analysis, for that reason it is necessary to remove from the text with the goal to improve our data for modeling.

1. Removal of Stop Words

In sentiment analysis, some words such as “a”, “the”, and “is” have limited semantic meaning and could be disregarder. For that reason, is necessary to eliminate common and non-informative words know such as “stop words”.

1. Expanding contractions

To perform modeling some expansions of the contractions should be used to enable the understanding of the complete word. With expanding contractions, it could ensure that words like “can’t” were transformed into “cannot”.

1. Tokenization

To make it more amenable for analysis, it was necessary to use tokenization which is crucial to breaks down text into its basic elements.

1. Rejoining Tokens

Post tokenization, the individual token should be reassembled into coherent text.

After to make data pre-processing, the text is ready to sentiment analysis. Lemmatization and lowercase are not necessary to apply because TextBlob was embedded those steps.

## Dataset Pre-processing for VADER model

For VADER model, it is not necessary to do data pre-processing because it is specifically designed for sentiment analysis for social media posts. It is well-suited for processing short and informal textual data. It can effectively analyze the sentiment intensity and polarity of the text. VADER provides a proper handling of sentences with:

* Typical negations
* Use of contractions as negations
* Use of punctuation as signal of increment of the sentiment intensity
* Use of word-shape (ALL CAPS) as signal of emphasis
* Use of degree modifiers as alteration of the sentiment intensity
* Use of the slang words
* Use of emoticons and emojis
* Use of the initialisms and acronyms.

For that reason, the pre-processing was not required.

# Models

## TextBlob

TextBlob is a Python library for basic natural language processing (NLP) tasks. It offers a simple API for common tasks like tokenization, part-of-speech tagging, noun phrase extraction, and sentiment analysis. TextBlob Built on the NLTK and another package called Pattern, TextBlob provides an easy-to-use interface in NLP.

### Assumptions/Heuristics/algorithms used

The TextBlob by default uses the Pattern library for its sentiment analysis. Pattern’s sentiment analysis is built on large dataset annotated with polarity and subjectivity. Sentiment of a text is calculated based on the words it contains and their respective subjectivity and polarity.

Alternate to pattern we can use Naive Bayes classifier trained using NLTK on a movie review corpus. This method uses the probabilities of observing specific words given their sentiment labels.

We assume that polarity of text in range of –0.2 to +0.2 will have sentiment neutral whereas polarity of text greater than +0.2 will have positive and less than –0.2 will have negative sentiment.

### How it works

TextBlob is a Python library for processing textual data, and it provides a simple API for diving into common natural language processing (NLP) tasks.

TextBlob first Tokenize and preprocesses data (removing stop words, lowercasing etc.), and then TextBlob can tag each token with its corresponding Part-Of-Speech like noun, verb, etc. We can also call this as a feature extraction in which we convert the tokenized words into features that model can understand.

The sentiment property of the api/library returns polarity and subjectivity.

Polarity ranges from –1.0 to +1.0 while subjectivity ranges from 0 to +1.0. Polarity measures the emotion. Where +1.0 refers to positive and –1.0 refers to negative. While subjectivity refers to opinion or views which needs to analyze in given context where 0 is very objective and +1.0 is very subjective. A subjective instance may or may not carry any emotion.

### External Datasets

TextBlob relies on external datasets and resources for various functionalities like:

* **Sentiment Analysis->** TextBlob uses the ‘**Pattern’** library for sentiment analysis which comes with a built-in sentiment lexicon. This lexicon is used to determine polarity (positivity/negativity) and subjectivity of a text. It is not exactly a dataset but rather a collection of words and their associated sentiment scores.
* **POS Tagging and Noun Phrase Extraction->** For these functionalities, TextBlob leverages corpora and trained models from the Natural Language Toolkit (NLTK). Specifically, it typically uses the Penn Treebank dataset for POS tagging.
* **Tokenization->** TextBlob uses NLTK’s tokenization methods, while not directly relying on a specific dataset but have been informed and refined by numerous corpora.
* **Translation and Language Detection->** TextBlob offloads these tasks to the Google Translate API. This is not directly about an external dataset, but it is worth noting since the translation capability relies on an external service.

## Valence Aware Dictionary and Sentiment Reasoner (VADER)

VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon-based sentiment analysis tool that uses a pre-built dictionary of words and their associated sentiment scores to determine the sentiment of a given piece of text. Its lexicon is specifically tuned to handle informal language and features such as slang, emoticons, and capitalization commonly used in social media text, news articles, blogs etc.

### Assumptions/Heuristics/algorithms used

We have Assume that valance score of a text ranging from –0.2 to +0.2 will be count as a neutral sentiment. Valance score Greater than +0.2 and less than –0.2, we will count as a positive and negative sentiment respectively.

### How it works

VADER relies on a predefined dictionary (or lexicon) that maps words and other numerous lexical features common to sentiment expression in microblogs.

These features include:

* + A full list of Western-style emoticons (ex - :D and :P )
  + Sentiment-related acronyms (ex- LOL and ROFL)
  + Commonly used slang with sentiment value (ex- Nah and meh)

The valence scores in VADER's lexicon range from -4 (most negative) to +4 (most positive). The sentiment score of text is calculated as a sum of intensity score of words in the text. Words that are neutral or have no clear sentiment typically have a score close to 0.

To calculate the composite sentiment score for an entire piece of text, VADER doesn't just sum up the valence scores of individual words. Instead, it incorporates various heuristics and rules, including:

* + Adjusting scores for booster words (e.g., "very" or "extremely") that can amplify the sentiment of a neighboring word.
  + Handling negations that can reverse the sentiment (e.g., "not good" is negative despite the word "good" being positive).
  + Accounting for the effects of punctuation, capitalization, and other linguistic cues.

After processing the text and applying these rules, VADER produces a compound sentiment score that ranges from -1 (most negative) to +1 (most positive). This score offers a holistic view of the text's overall sentiment. In addition to the compound score, VADER also provides individual scores for the positive, neutral, and negative sentiments present in the text.

### External Datasets

VADER has its own dataset called vader\_lexicon.txt which is validated by multiple independent human judges. VADER incorporates a "gold-standard" sentiment lexicon that is especially attuned to microblog-like contexts.

Other than it, VADER also uses external datasets like

* nytEditorialSnippets\_GroundTruth.txt
* nytEditorialSnippets\_anonDataRatings.txt
* movieReviewSnippets\_GroundTruth.txt
* movieReviewSnippets\_anonDataRatings.txt
* amazonReviewSnippets\_GroundTruth.txt
* amazonReviewSnippets\_anonDataRatings.txt
* tweets\_GroundTruth.txt,tweets\_anonDataRatings.txt

# Testing results summary

The summary analysis based on the testing outcomes of the two approaches—Vader and Textblob that we used to develop our model is presented below.

## TextBlob

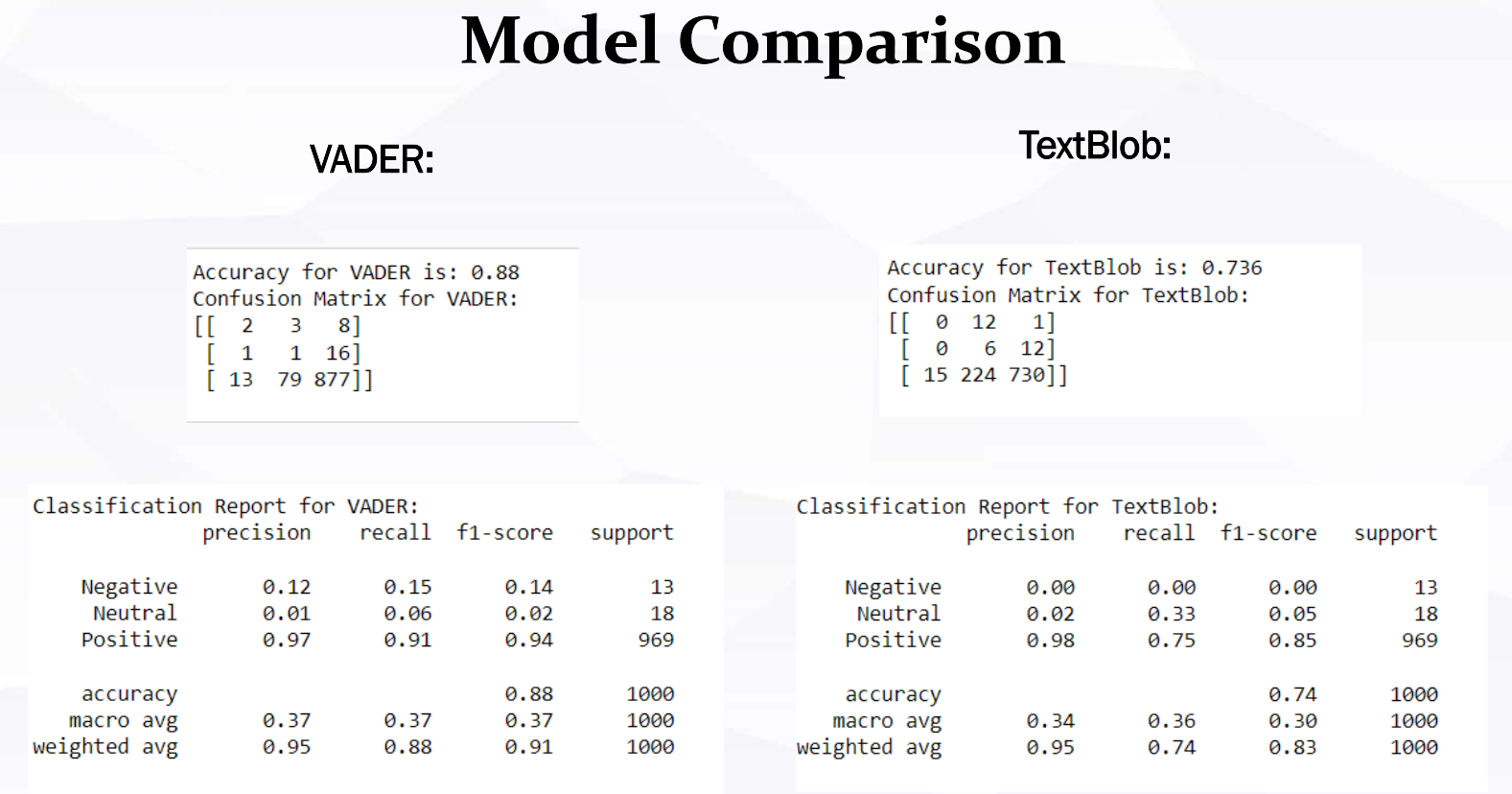
The TextBlob Model had an accuracy of about 73%, with a precision of 98% for positive sentiment, 2% for neutral sentiment and 0% for negative sentiment. With a weighted average of 95%.

## Vader

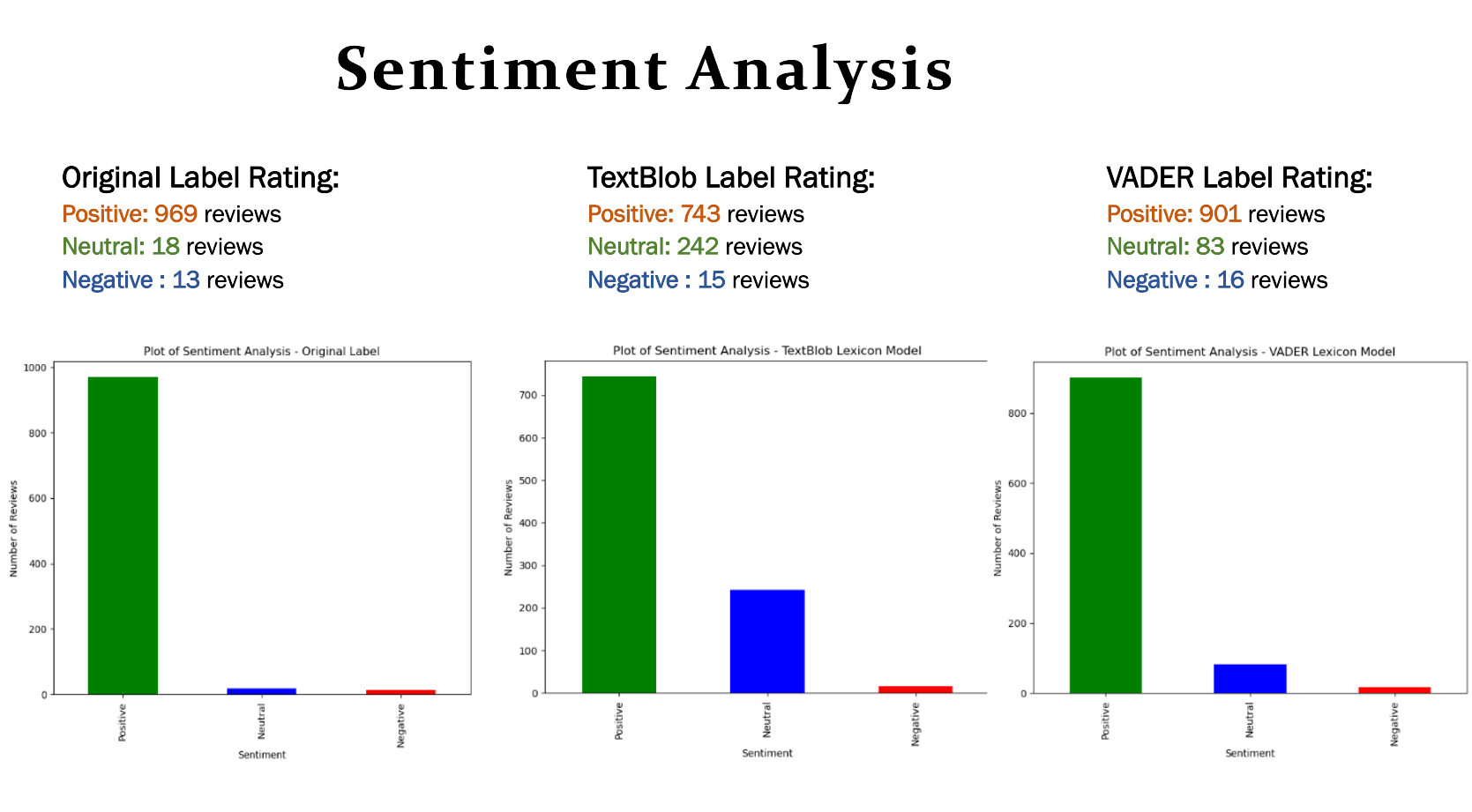
The VADER model had an accuracy of about 88%, with precision of 97% for positive sentiment, 1% for neutral sentiment and 12% for negative sentiment. With a weighted average of 95%

## Model Comparison

Overall, the Vader model performed the best out of two models, with the highest accuracy and f1-score for positive sentiment. The Textblob model does not perform as well as the Vader model, but it still has decent accuracy. One thing we should note is that both models have identical weighted avg score for precision of about 95%.



In the Original Label Rating, a vast majority of reviews, 969 to be precise and categorized as positive with a minimal 18 as neutral and 13 as negative. The TextBlob Lexicon Model has identified 743 reviews as positive, 242 as neutral and 15 as negative. Comparatively, the VADER Lexicon Model labels 901 reviews as positive, 83 as neutral and 16 as negative. Both the TextBlob and VADER models depict a higher number of neutral reviews compared to the Original Label.



# Final Conclusion

The data exploration reveals a predominantly positive sentiment from the reviewers. The combination of a high average rating with the substantial proportion of verified reviews suggests that the feedback is both positive and credible. The product, service, or content under review seems to be well-received by its users.

In data preprocessing we include steps regarding handling missing data, removing duplicates, removing stop-words, combining review text and summary etc. for textblob, whereas we only use clean text which is combination of review text and summary for VADER as it is. We dropped columns which are not useful for sentiment analysis like Style, vote, image, and reviewerName, etc.

Then we perform sentiment analysis using textblob and VADER by categorizing reviews into positive, negative, and neutral categories based on their ratings.

When comparing the sentiment analysis capabilities of VADER and TextBlob, VADER clearly emerges as the superior model. It consistently outperforms TextBlob across key metrics, particularly in accuracy and in identifying positive sentiments. While both models have their merits, for tasks demanding higher precision and recall, VADER appears to be the more reliable choice.

# References

* Project, U. C. R. (n.d.). Amazon Review Data (2018). <https://nijianmo.github.io/amazon/index.html>
* Tutorial: Quickstart - TextBlob 0.16.0 documentation (n.d.). <https://textblob.readthedocs.io/en/dev/quickstart.html>
* Cjhutto. (n.d.). CJHUTTO/Vadersentiment: Vader sentiment analysis. GitHub. <https://github.com/cjhutto/vaderSentiment>
* Vader sentiment analysis: A complete guide, algo trading and more. Quantitative Finance & Algo Trading Blog by QuantInsti. <https://blog.quantinsti.com/vader-sentiment/>
* Todi, M. (2019, September 23). Sentiment analysis using the vader library. Medium. <https://medium.com/analytics-vidhya/sentiment-analysis-using-the-vader-library-a91a888e4afd>

# Appendix 1: Project plan

The purpose of this project is to create a sentiment analysis model that can classify customers textual reviews on amazon data as positive, negative or neutral sentiment. Sentiment analysis is an important part of NLP that enables machines to understand human emotions conveyed in text. It aids businesses in gauging customer feedback, helps in automating responses to user queries, social media monitoring etc.

The goal of this project is to present an in-depth overview of the essential steps involved in creating a sentiment analysis model utilizing the Lexicon method.

* Data exploration is done in the first section. a vital first step in comprehending the information provided. It helps us make sense of the data provided and decides what needs to be done for future tasks.
* The second step is Data preprocessing. This step is necessary so that data is clean, consistent and ready for further processing.
* Model training and assessment constitute the last step, whereby two lexicon methods (Vader, TextBlob, and Sentiwordnet models) out of three should be chosen.
* Phase 1 ends with the selection of the two best models, which will then be compared to each other based on accuracy and other performance measures.

# Appendix 2: Meeting Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Time | Attendance | topic | Assignment/ Discussion |
| 10-04-2023 | 12pm | Mahpara  Manipal  Ronald  Kanishka | Data Exploration | Performed Data Exploration. |
| 10-11-2023 | 12pm | Mahpara  Manipal  Ronald  Kanishka | Preprocessing | Discussed preprocessing the data. |
| 10-28-2023 | 8pm | Mahpara  Manipal  Ronald  Kanishka  Kaushalkumar | Build Model | Divided task to learn about lexicon models and discuss about it |
| 10-30-2023 | 12PM | Mahpara  Manipal  Ronald  Kanishka  Kaushalkumar | Discussed improvements in model. | Work on building models. |
| 10-31-2023 | 9PM | Mahpara  Manipal  Ronald  Kanishka  Kaushalkumar | Build Project report | Work on Project report |