**BUILDING A COMPREHENSIVE SENTIMENT ANALYSIS MODEL FOR AMAZON PRODUCT REVIEWS**

**Phase # 1 Project Report**

Submitted by:

Roland Joseph Baja

Christian Aduna

Submitted to:

Professor Rushdi Shams

COMP 262: Natural Language Processing and Recommender Systems

School of Engineering Technology and Applied Science

Centennial College

23 February 2024

**Table of Contents**

[1.0 INTRODUCTION 3](#_Toc159757916)

[2.0 DATASET EXPLORATION 3](#_Toc159757917)

[2.1 Counts and averages 3](#_Toc159757918)

[2.2 Distribution of the number of reviews across products 4](#_Toc159757919)

[2.3 Distribution of the number of reviews per product 5](#_Toc159757920)

[2.4 Distribution of reviews per user 6](#_Toc159757921)

[2.5 Checking for outliers 7](#_Toc159757922)

[2.6 Analyzing lengths 7](#_Toc159757923)

[2.7 Checking for duplicates 8](#_Toc159757924)

[3.0 DATASET PREPROCESSING 9](#_Toc159757925)

[3.1 Data labelling based on the value of “rating of the product” 9](#_Toc159757926)

[3.2 Choosing columns for sentiment analyzer 10](#_Toc159757927)

[4.0 MODELS (SENTIMENT ANALYSIS) LEXICON APPROACH 12](#_Toc159757929)

[4.1 Valence Aware Dictionary and Sentiment Reasoner (VADER) 12](#_Toc159757930)

[4.2 TextBlob 13](#_Toc159757931)

[4.3 Justification 15](#_Toc159757932)

[5.0 TESTING 18](#_Toc159757933)

[5.1 Valence Aware Dictionary and Sentiment Reasoner (VADER) 18](#_Toc159757934)

[5.2 TextBlob 19](#_Toc159757935)

[5.3 Comparison 20](#_Toc159757936)

[6.0 CONCLUSION 20](#_Toc159757937)

[7.0 ASSUMPTIONS 21](#_Toc159757938)

[8.0 REFERENCES 22](#_Toc159757939)

[9.0 APPENDIX 23](#_Toc159757940)

[9.1 Project plan 24](#_Toc159757941)

[9.2 Meeting register 25](#_Toc159757942)

# INTRODUCTION

In today's digital age, understanding customer sentiments towards products is paramount for businesses striving to enhance user experience and product quality. Welcome to our two-phase project assignment, where we delve into the intricate realm of sentiment analysis for product reviews. Our goal is to construct a robust sentiment analysis model leveraging both Lexicon and machine learning approaches, culminating in the development of a sophisticated recommender system.

Throughout this project, we harness the power of textual data from the Amazon product review dataset for appliances. This dataset serves as a rich repository of consumer feedback, offering invaluable insights into customer perceptions and preferences. By leveraging insights gleaned from customer feedback, we endeavor to tailor personalized recommendations, fostering enhanced consumer experiences and driving business growth.

# 2.0 DATASET EXPLORATION

## 2.1 Counts and averages

A screenshot of a calculator

Description automatically generated

A graph with blue bars

Description automatically generated

## 2.2 Distribution of the number of reviews across products

A graph with numbers and a line

Description automatically generated

A graph with blue and white lines

Description automatically generated

## 2.3 Distribution of the number of reviews per product

A graph with numbers and lines

Description automatically generated

A graph with blue and white bars

Description automatically generated

## 2.4 Distribution of reviews per user

A graph with blue and white lines

Description automatically generated

A graph with blue and white stripes

Description automatically generated

## 2.5 Checking for outliers

A screenshot of a test

Description automatically generated

## 2.6 Analyzing lengths

A graph of a number of columns

Description automatically generated

## 2.7 Checking for duplicates

A white text box with black text

Description automatically generated

# 3.0 DATASET PREPROCESSING

## 3.1 Data labelling based on the value of “rating of the product”

A pie chart with a green circle and blue triangle

Description automatically generated

## 3.2 Choosing columns for sentiment analyzer

The following columns are chosen for sentiment analyzer:

* 'reviewText' Column: This column contains the actual text of the reviews written by users. Sentiment analysis is typically performed on textual data to determine the sentiment or opinion expressed within the text. Analyzing the 'reviewText' column allows the sentiment analyzer to extract sentiment from the reviews themselves, providing valuable insights into customer opinions about the products.
* 'overall' Column: The 'overall' column likely represents the overall rating given to the product by users. While sentiment analysis primarily focuses on the textual content of reviews, incorporating the overall rating can provide a numerical indicator of sentiment. It serves as a reference point for sentiment analysis and can be used to validate the sentiment extracted from the text.
* 'summary' Column: The 'summary' column contains brief summaries or titles of the reviews. Although shorter than the full review text, these summaries can still convey important sentiment-bearing information. Sentiment analysis can be performed on the 'summary' column to capture concise sentiments expressed by users in the titles or summaries of their reviews.
* 'rating' Column: The 'rating' column is derived from the 'overall' column based on predefined criteria:

df['rating'] = df['overall'].apply(lambda x: 'Negative' if x <= 2 else ('Neutral' if x == 3 else 'Positive'))

It categorizes the overall ratings into three categories: 'Negative' for ratings less than or equal to 2, 'Neutral' for ratings equal to 3, and 'Positive' for ratings greater than 3. While not commonly used in sentiment analysis, the 'rating' column may provide additional insights or features that could enhance sentiment analysis by offering a simplified representation of sentiment categories.

In summary, by choosing these columns in the sentiment analysis process, the analyzer can leverage a combination of textual content, numerical ratings, and summary information to gain a comprehensive understanding of user sentiment towards the products. Each column provides a unique perspective that enriches the sentiment analysis process and contributes to more informed decision-making.A white background with black and red text

Description automatically generated

A screenshot of a review text

Description automatically generated

# 4.0 MODELS (SENTIMENT ANALYSIS) LEXICON APPROACH

## 4.1 Valence Aware Dictionary and Sentiment Reasoner (VADER)

Valence Aware Dictionary and Sentiment Reasoner (VADER) is a lexicon and rule-based sentiment analysis tool designed for analyzing sentiments in text data. It is specifically attuned to sentiments expressed in social media, providing a nuanced understanding of text sentiment even in the presence of slang, emoticons, and other informal language constructs.

VADER Assumptions/Heuristics/Algorithms/Packages Used

* Lexicon-based Approach: VADER utilizes a lexicon that contains a pre-defined set of words, along with their sentiment scores. These scores represent the intensity of positive, negative, and neutral sentiment associated with each word.
* Rule-based Heuristics: VADER employs a set of rules and heuristics to interpret the sentiment expressed in text data. These rules consider the context of words, phrases, and punctuation marks to infer sentiment polarity and intensity.
* Algorithmic Analysis: VADER combines lexicon scores with syntactic and semantic rules to compute the overall sentiment polarity of a given text snippet. It employs algorithms to weigh individual word scores, consider negations, account for capitalization and punctuation, and handle degree modifiers.
* Python Package: VADER is implemented as a Python package, making it easily accessible and integrable into various text processing pipelines and applications.

How VADER Works

VADER operates by analyzing text input and assigning sentiment scores to individual words, phrases, and sentences. The key components of its functionality include:

* Sentiment Lexicon: VADER's lexicon contains thousands of words with pre-assigned sentiment scores, ranging from highly negative to highly positive.
* Sentiment Intensity Calculation: VADER computes the overall sentiment intensity of a text by aggregating the sentiment scores of individual words while considering their context and syntactic structure.
* Handling of Polarity Shifts: VADER accounts for negations, amplifiers, and diminishers, adjusting the sentiment polarity accordingly. For instance, phrases like "not good" would be interpreted as negative sentiment despite the presence of the word "good."
* Valence Shifters: VADER identifies and handles valence shifters such as intensifiers and de-intensifiers, ensuring that the sentiment intensity is appropriately adjusted.

Fine-Tuning Steps

Fine-tuning VADER typically involves adjusting its lexicon or incorporating domain-specific terms and expressions to enhance its accuracy for specific contexts. The steps for fine-tuning may include:

* Lexicon Expansion: Adding domain-specific terms and phrases to the lexicon to improve coverage and accuracy for particular industries or domains.
* Lexicon Pruning: Removing or adjusting sentiment scores for words that might yield inaccurate results or bias the analysis.
* Rule Modification: Tweaking the rules and heuristics used by VADER to better align with the nuances of the target text data.
* Evaluation and Validation: Assessing the performance of VADER on a validation dataset and iteratively refining the model based on the results.

By fine-tuning VADER, analysts and developers can tailor its capabilities to better suit the requirements of specific sentiment analysis tasks and domains, thereby enhancing the accuracy and relevance of sentiment analysis results.

## 4.2 TextBlob

TextBlob is a Python library for processing textual data, providing a simple interface for tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. It leverages the NLTK (Natural Language Toolkit) and Pattern libraries, making it a powerful tool for natural language processing tasks.

TextBlob Assumptions/Heuristics/Algorithms/Packages Used

* Rule-based Sentiment Analysis: TextBlob's sentiment analysis module employs a rule-based approach to assess the sentiment polarity of text data. It uses a combination of lexical patterns, syntactic structures, and linguistic heuristics to determine sentiment.
* Pattern Library Integration: TextBlob integrates the Pattern library, which provides access to pre-trained sentiment analysis models and lexicons. These resources are utilized to compute sentiment scores for text input.
* Noun Phrase Extraction: TextBlob leverages its ability to extract noun phrases to identify and analyze sentiment-bearing phrases within text data. This feature helps in capturing the context and subjectivity of sentiment expressions.
* Python Package: TextBlob is implemented as a Python package, offering an intuitive API for text processing tasks. It is built on top of NLTK and Pattern, providing seamless integration with existing Python-based NLP workflows.

How TextBlob Works

TextBlob's sentiment analysis functionality is primarily based on the following key components:

* Pre-Trained Sentiment Classifier: TextBlob employs a pre-trained sentiment classifier that assigns polarity scores to text inputs. The classifier is trained on labeled datasets to recognize patterns indicative of positive, negative, and neutral sentiments.
* Pattern-based Analysis: TextBlob utilizes the Pattern library's sentiment analysis capabilities to compute sentiment scores for text. The sentiment scores range from -1 (negative) to 1 (positive), with 0 indicating neutral sentiment.
* Subjectivity Assessment: In addition to sentiment polarity, TextBlob assesses the subjectivity of text expressions. It distinguishes between objective and subjective statements, providing insights into the degree of opinion or bias present in the text.
* Noun Phrase Extraction: TextBlob identifies noun phrases within text data, allowing for a more granular analysis of sentiment at the phrase level. This feature enables the identification of sentiment-bearing entities and their contextual relevance.

Fine-Tuning Steps

Fine-tuning TextBlob for sentiment analysis involves customization and optimization to better align with specific use cases and domains. The following steps may be taken for fine-tuning:

* Custom Lexicon Integration: Incorporating domain-specific terms and expressions into TextBlob's sentiment lexicon to improve accuracy and coverage for specialized domains or industries.
* Classifier Training: Training custom sentiment classifiers using domain-specific labeled datasets to enhance the model's performance on specific types of text data.
* Rule Modification: Adjusting sentiment analysis rules and heuristics to accommodate linguistic variations, cultural nuances, or specific text data characteristics.
* Feature Engineering: Exploring additional features or linguistic patterns that may contribute to more accurate sentiment analysis, such as syntactic structures, word embeddings, or context-based features.
* Evaluation and Validation: Evaluating TextBlob's performance on representative datasets and iteratively refining the model based on validation results and user feedback.

By fine-tuning TextBlob's sentiment analysis capabilities, users can tailor its functionality to suit the requirements of diverse text processing tasks and achieve more accurate and contextually relevant sentiment analysis results.

## 4.3 Justification

Choosing VADER and TextBlob among other lexicon-based approaches for sentiment analysis in the context of product reviews involves considering their specific features, performance, and suitability for the task at hand:

* Accuracy and Performance: VADER and TextBlob are known for their relatively high accuracy and performance in sentiment analysis tasks, particularly for social media data and informal text like product reviews. Their lexicons are well-tuned to capture the nuances of sentiment expressions commonly found in such text, including slang, emoticons, and informal language constructs.
* Subjectivity Handling: Both VADER and TextBlob incorporate mechanisms to detect and handle subjectivity in text, which is crucial for understanding the nuanced opinions and attitudes expressed in product reviews. Their ability to distinguish between objective and subjective statements enhances the granularity and accuracy of sentiment analysis results.
* Ease of Use and Integration: VADER and TextBlob offer user-friendly APIs and seamless integration with Python-based NLP workflows, making them accessible to developers and analysts with varying levels of expertise. Their straightforward interfaces and comprehensive documentation facilitate rapid prototyping, experimentation, and deployment of sentiment analysis solutions.
* Community Support and Maintenance: VADER and TextBlob benefit from active community support, frequent updates, and ongoing maintenance, ensuring that they remain robust and reliable tools for sentiment analysis tasks. Their popularity and widespread adoption within the NLP community attest to their effectiveness and relevance in real-world applications.
* Domain Adaptability: While VADER and TextBlob are general-purpose sentiment analysis tools, they can be easily customized and adapted to specific domains or industries by incorporating domain-specific terms and expressions into their lexicons. This flexibility allows users to tailor the models to capture domain-specific sentiment patterns and terminology present in product reviews.
* Proven Track Record: VADER and TextBlob have been extensively tested and validated in various research studies and real-world applications, demonstrating their effectiveness across different domains and text genres. Their robust performance and reliability make them preferred choices for sentiment analysis tasks, including those involving product reviews.

SentiWordNet is a valuable resource for sentiment analysis, as it assigns sentiment scores to synsets (sets of synonyms) in WordNet, a lexical database of the English language. While SentiWordNet can be useful for certain sentiment analysis tasks, there are several reasons why it may not be the optimal choice for analyzing product reviews specifically:

* Limited Coverage: SentiWordNet's coverage may be limited, especially when it comes to domain-specific terms and expressions commonly found in product reviews. Product reviews often contain specialized vocabulary, slang, and industry-specific terminology that may not be adequately represented in SentiWordNet's lexicon.
* Granularity: SentiWordNet provides sentiment scores at the level of synsets, which represent sets of synonyms sharing a common meaning. While this granularity may be suitable for certain text genres, it may not capture the nuances of sentiment expressed in product reviews, which often involve detailed opinions about specific features, functionalities, and experiences.
* Contextual Sensitivity: SentiWordNet's sentiment scores are based on the semantic relationships between words in WordNet, without considering the contextual usage or syntactic structures of words in actual text. Product reviews frequently contain context-dependent sentiments, sarcasm, and figurative language, which may not be accurately captured by SentiWordNet's lexical associations alone.
* Subjectivity Handling: SentiWordNet may not effectively handle subjectivity in text, distinguishing between objective and subjective statements. Product reviews inherently involve subjective opinions and evaluations, which require nuanced analysis to understand the underlying sentiments and attitudes expressed by reviewers.
* Integration Complexity: Working with SentiWordNet may require significant manual effort and expertise to map words to synsets and compute sentiment scores based on synset information. Integrating SentiWordNet into sentiment analysis workflows for product reviews may involve additional preprocessing and post-processing steps to align with the specific characteristics of the review data.

Other lexicon-based approaches, such as VADER and TextBlob, offer advantages in terms of subjectivity handling, ease of use, pre-trained models, and community support, making them preferred choices for sentiment analysis tasks involving product reviews.

# 5.0 TESTING

## 5.1 Valence Aware Dictionary and Sentiment Reasoner (VADER)

A green circle with a green circle

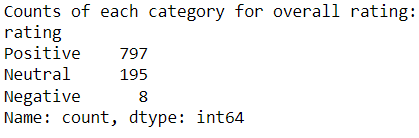
Description automatically generated

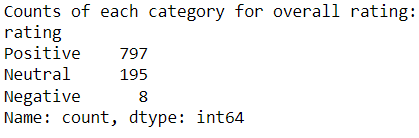
A white background with black text

Description automatically generated

A white background with black text

Description automatically generated





## 5.2 TextBlob

A green and grey circle with a green line

Description automatically generated

A white background with black text

Description automatically generated

A white background with black text

Description automatically generated

A white background with black text

Description automatically generated

## 5.3 Comparison

A screenshot of a computer

Description automatically generated

# 6.0 CONCLUSION

After modeling using the Lexicon approach, we can draw several conclusions about the dataset:

1. Overall Rating (overall):

* The average overall rating is approximately 4.49 out of 5, indicating that the majority of reviews are positive.
* The minimum rating is 1, and the maximum rating is 5, with 25%, 50%, and 75% quartiles ranging from 4 to 5, suggesting that most ratings are relatively high.

1. Review Timestamp (unixReviewTime):

* The dataset covers a range of review times, with a mean of approximately 1.45e+09.
* The minimum and maximum review times indicate the earliest and latest reviews in the dataset.

1. Text Length (text\_length):

* The average length of review texts is approximately 1479.76 characters.
* The minimum text length is 2 characters, while the maximum is 3932 characters, highlighting variability in review lengths.

1. VADER Sentiment Analysis:

* The average VADER sentiment score for review texts (reviewText\_vaderValue) is approximately 0.88, suggesting predominantly positive sentiment in the reviews.
* The average VADER sentiment score for summary texts (summary\_vaderValue) is approximately 0.43, indicating slightly lower positive sentiment compared to review texts.

1. TextBlob Sentiment Analysis:

* The average TextBlob polarity score for review texts (reviewText\_textblobPolarity) is approximately 0.16, indicating mildly positive sentiment.
* The average TextBlob polarity score for summary texts (summary\_textblobPolarity) is approximately 0.26, indicating slightly higher polarity compared to review texts.

Overall, the statistics suggest that the dataset consists of predominantly positive reviews, with varying lengths of review texts. Both VADER and TextBlob sentiment analysis tools indicate positive sentiment in the majority of reviews, with slight differences in polarity scores between review texts and summary texts.

# 7.0 ASSUMPTIONS

Several assumptions can be considered for the sentiment analysis project based on customer textual reviews:

* Reviews Reflect Genuine Customer Sentiments: The project assumes that the textual reviews provided by customers are genuine expressions of their sentiments towards the products. While this is generally true, there might be cases of fake or biased reviews that could affect the analysis.
* Reviews are Representative of Overall Customer Sentiment: It is assumed that the dataset of Amazon product reviews for appliances provides a representative sample of customer sentiments towards those products. However, it is important to acknowledge that the dataset might not cover the entire spectrum of customer opinions or demographics.
* Lexicon and Machine Learning Approaches Are Effective: The project assumes that both the Lexicon-based and machine learning-based approaches are effective methods for sentiment analysis. While these approaches are widely used, their accuracy and effectiveness might vary depending on the quality of the data and the complexity of the sentiment expressions.
* Review Data Quality and Consistency: The assumption is made that the review data is of sufficient quality and consistency for analysis. However, there may be instances of spelling errors, grammatical inconsistencies, or ambiguous sentiments that could challenge the analysis process.
* Relevance of Features for Recommender System: In the context of constructing a recommender system, it is assumed that features derived from the review data are relevant indicators of product preferences and user behavior. However, the relevance of these features may vary depending on the nature of the products and the preferences of the target audience.
* No External Factors Influencing Sentiments: The project assumes that sentiments expressed in the reviews are solely influenced by the quality and performance of the products themselves, and not by external factors such as promotional campaigns, competitor activities, or cultural biases. However, external factors can sometimes influence customer sentiments and should be taken into account during analysis.
* Availability of Sufficient Computational Resources: The project assumes access to sufficient computational resources for data processing, model training, and evaluation. However, resource constraints may impact the scalability and efficiency of the analysis, especially when dealing with large datasets or complex machine learning models.

By acknowledging and understanding these assumptions, the project team can better navigate the challenges and limitations inherent in sentiment analysis and make informed decisions throughout the project lifecycle.

# 8.0 REFERENCES

Cjhutto. (n.d.). *GitHub - cjhutto/vaderSentiment: VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains.* GitHub. <https://github.com/cjhutto/vaderSentiment>

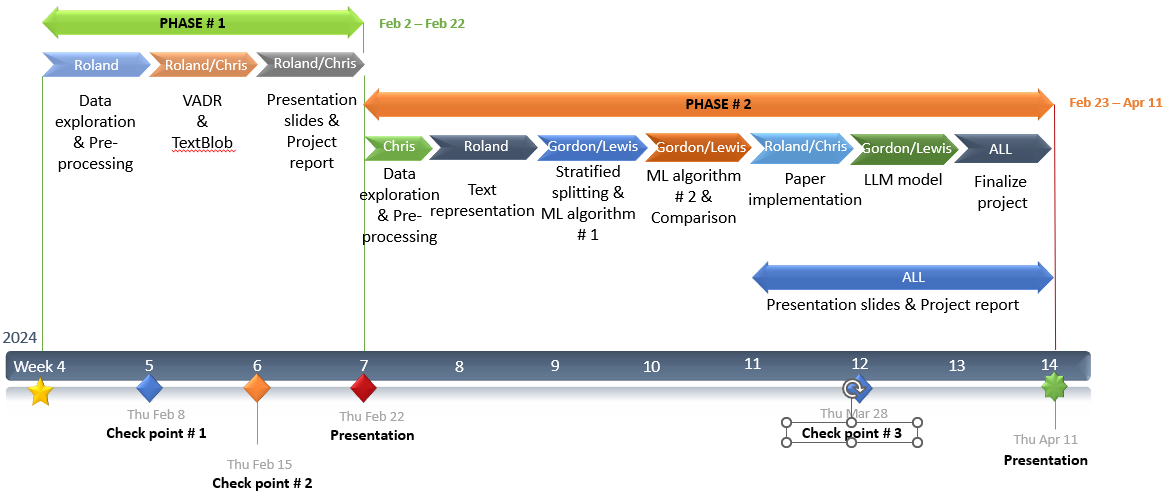
*Lexicon-based approaches: NLTK Vader or TextBlob?* (n.d.). MFIN7036. <https://insight-group.github.io/MFIN7036/sentiment-analysis-lexicon-based-nv-or-tb.html>

Ni, J. (n.d.). *Amazon review data*. <https://nijianmo.github.io/amazon/index.html>

*Tutorial: Quickstart — TextBlob 0.18.0.post0 documentation*. (n.d.). <https://textblob.readthedocs.io/en/dev/quickstart.html>

# 9.0 APPENDIX

## 9.1 Project plan



## 9.2 Meeting register

|  |  |
| --- | --- |
| COMP 262 Group 3 Meeting Log |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Attendance | Task | Member Assigned & Due Date |
| 2/8/2024 | Roland  Chris  Gordon | Create and join MS Teams chat  Invite and accept as GitHub collaborators  Complete tasks 1 – 4  Start VADER | Roland, Chris, Gordon – 2/10/2024  Roland, Chris, Gordon – 2/12/2024  Roland – 2/13/2024  Roland – 2/15/2024 |
| 2/13/2024 | Roland  Chris  Gordon | Complete VADER  TextBlob  Start presentation slides  Validation and comparison | Roland – 2/16/2024  Chris – 2/16/2024  Roland – 2/18/2024  Gordon – 2/18/2024 |
| 2/20/2024 | Roland  Chris  Gordon  Lewis | Accept GitHub invitation  Validation and comparison  Clean code  Continue presentation slides  Project report | Lewis – 2/20/2024  Gordon/Lewis – 2/21/2024  Roland – 2/21/2024  Lewis – 2/21/2024  Chris – 2/22/2024 |
| 2/21/2024 | Roland  Chris | Complete presentation slides  Complete project report  Finalize phase # 1 | Roland – 2/22/2024  Chris – 2/22/2024  Roland/Chris – 2/23/2024 |