A Comprehensive Sentiment Analysis for Amazon's Appliance Product Reviews

Project Team members: Debanik Chakraborty, Mahsa Nafar Sefiddashti, Moumita Sen

Abstract:

In today's world, online shopping is becoming increasingly popular as it's time saving, informed and convenient. Initially renowned for its extensive book collection, Amazon has diversified its offerings to include electronics, home appliances, and various consumer products. Presently, it boasts a vast inventory comprising millions of products. The surge in E-commerce has underscored the importance of understanding customer needs and opinions, leading to the emergence of a crucial facet in online shopping -User Reviews. These reviews encapsulate the suggestions and opinions of customers, serving as invaluable insights for others contemplating a purchase. Due to this escalating significance of customer reviews, sentiment analysis is also becoming an area of interest for the E-commerce sites.

The main objective of this project is to score the Amazon Appliance product reviews using two lexicon-based (VADER, RoBERTa) and one machine learning (Google API) sentiment analysis models to evaluate based on three criteria (Relationship to True Sentiment score, Relationship to Ratings and Precision, Recall, F1 scores) and decide the most appropriate model. The paper finds RoBERTa as the best model and recommends that for future sentiment analysis of Amazon Appliance product users reviews. The paper applied Proportionate Stratified Sampling in making dataset concise and also in obtaining a sample for evaluation purpose. True Sentiment score was given manually reviewing the reviews for the sample. The paper also conducts product type specific analysis and tries to find out if there is any correlation between price vs sentiment score and price vs ratings, however, it concludes there is no significant relationship between price and reviews and ratings of customers. Finally, the paper tries to find out the product type having the most negative reviews and 'Range Knobs' is the product type with maximum negative reviews. The paper recommends this product for improvement as it is receiving almost 40% negative reviews, whereas the whole dataset has 23% negative reviews.

1. Introduction:

Amazon is one of the most popular e-commerce sites where you can find a lot of different things from many different brands. Appliance products are widely sold on Amazon, and customers leave positive, negative, neutral, and irrelevant reviews along with ratings that reflect public opinion and customer satisfaction levels regarding appliance products. Sentiment analysis converts this qualitative data into quantitative scales, helping the company identify, understand, and evaluate customer satisfaction.

Sentiment analysis is the process of determining the general emotional tone or attitude expressed by the author within a given text, typically in the context of a review. There are mainly three types of models used for conducting sentiment analysis: Lexicon-based models, Machine learning models, and Hybrid models. Lexicon-based models use a dictionary of words with predefined labels of positive, negative, or neutral sentiment. This is also known as rule-based sentiment analysis since it relies on a set of rules created by language experts. In our study, we will include two lexicon-based models: VADER (Valence Aware Dictionary and Sentiment Reasoner), RoBERTa and Google API which is a pre-trained model by machine learning. All of which are natural language processing (NLP) algorithms used for sentiment analysis. These techniques generally calculate the polarity score of the texts by matching words with their specific dictionary and considering other factors such as negation, context, phrases, and intensity.

1.1 Project Objective:

- To find and recommend the most appropriate model for sentiment analysis of future Amazon Appliance products' reviews.
- Find the correlation between price and sentiment score/ price and rating for each product type.
- Recognizing the product type that needs improvement and giving recommendations on that.

2. Related Literature:

(Huang & Rashid, 2021) performed a sentiment analysis on a user review data of 3000 records from famous Ecommerce site Amazon. This dataset on electronic products was trimmed into 10 variables where the significant columns were-Summary (title of review), Review text (the actual content), Rating (1-5 stars) and Helpfulness (number of people who found it helpful). The authors used Monkey Learn API and trained their model using more than 150 comments which were marked as positive, negative, and neutral. Later, whenever a new review was given into the model, it predicted the type of the comment (positive, negative, or neutral) with certain percentage of confidence. For the explanatory data analysis part, some interesting correlation graphs were done. The scatter plot between price and helpful reviews showed that price was negatively correlated to the number of helpful reviews (higher price triggers negative opinion). They also conducted scatter plots between rating and helpful reviews, chi square test for different product types, word cloud of the reviews and clarified the conclusion. The paper could fulfill its first objective to classify the positive and negative

reviews of the customers over different electronic products using a correlation between rating for the product and the hopeful numbers. However, it could not get anything different from the null hypothesis for different categories within the Amazon electronics product.

(Hutto & Gilbert, 2014) articulated the first paper to introduce, develop, validate, and evaluate the VADER (Valence Aware Dictionary for sentiment Reasoning), which is one of the most popular lexicon-based API techniques now-a-days. Parsimonious Model means the model which can analyze and make decisions based on as few variables as possible. VADER was compared with some prominent lexicon-based sentiment analysis techniques (LIWC, GI, SWN and Hu- Liu04), some machine learning text mining techniques (SVM and NB) and individual human interpretation (20 people). Social Media Text (4,200 tweets), Amazon Reviews (3,708 reviews), Movie Reviews (10,605 reviews) and NY Times Editorials (5,190 articles reviews) were used as dataset in this comparison. The overall precision, recall and F1 score was much better for VADER. In Social Media reviews VADER's F1 score was better than even human interpreters, which is a remarkable standing of this paper ultimately giving them the ground to declare it as a gold standard.

(Hussain, Dhanda, & Verma, 2023) used 500,000 Amazon product reviews (mainly-electronics, books and clothing) with 9 columns as their dataset. They followed stopword removal, tokenizing, feature extraction and feature selection to clean and prepare the data for analysis and it was partially trained with 5 machine learning models (Naive Byes, Logistic Regression, Decision Tree, Random Forest and SVM) and partially kept for testing. SVM came out to be the winner using accuracy, precision, recall and F1 score as the evaluation metrics of the models and later VADER and RoBERTa Models (lexicon-based) were applied on the same dataset and accuracies were compared SVM. The paper mainly concluded that SVM (machine learning technique) is better performing in opinion mining than lexicon-based techniques like VADER or RoBERTa.

(M D, C, & Ganesh, 2016) made a review paper defining the processes of three different sentiment analysis approaches. These are machine learning approaches (SVM, N-gram, Naive Bayes, ME Classifier, Multilingual, Feature Driven sentiment Analysis), Rule based Approach and Lexicon based Approach. The paper concludes with classifying the three approaches based on unsupervised and supervised learning scope and discussing their advantages and disadvantages. It also compares the different machine learning approaches based on their advantages and disadvantages.

Diekson et al. (2022) analyzed 1200 tweets to measure the satisfaction of customers with the Traveloka application's services. They collected data by narrowing down their focus on tweets from Indonesia, containing either "Traveloka" or "Traveloka eats". Data was transformed to a set of numerical vector data. Having split the data into two sub-sets of training set and test set, three methods of classification (Support Vector Model (SVM), Logistic Regression, and Naive Bayes) were used in this study and the results show that SVM was the most accurate one among the three.

The study by Afifah et al. (2021) analyzed the reviews of a mobile application called Telemedicine on Google Play. Telemedicine is a popular health application in Indonesia. They collected 12,969 reviews within a period of nine months (from Jan. 1st to Sep. 30th, 2021). After cleaning the data, Python Sastrawi library was used for data processing such as stopword removal, tokenization, normalization and stemming. For converting raw data to

machine- readable data, they used Term Frequency Inverse Document Frequency (TF-IDF). The algorithm used for classification of imbalanced data was Extreme Gradient Boosting. The study found that the main complaints were about payment, place, service, and system.

In 2021, Xiao et al. performed sentiment classification on amazon product review datasets. Three datasets which contained customer's rating and reviews were gathered from Amazon Simple Storage Service (Amazon S3). To analyze the user's opinions to get polarity and subjectivity they utilized TextBlob which is natural language processing library in Python. Also, the algorithm of TF-IDM was employed for word frequency analysis associated with the star ratings of each product. Based on the results customers' primary concerns revolve around quality, service, and pricing.

Nguyen et al. (2018) conducted a comprehensive study on sentiment analysis within the context of product reviews. The study involved a comparison of six algorithms to assess their effectiveness. These algorithms were categorized into three machine learning methods (Logistic Regression, Support Vector Machine, and Gradient Boosting) and three lexicon-based methods (Valence Aware Dictionary and Sentiment Reasoner, Pattern, and SentiWordNet). The dataset used in this research consisted of 43,620 product reviews sourced from 1,000 unique products. Four metrics, namely accuracy, precision, recall, and F1-score, were measured and compared to evaluate the performance of each method. The findings indicate that machine learning models demonstrate superior performance compared to lexicon-based approaches. The assessment of classifier accuracy reveals that, among the three machine learning algorithms, LR outperforms both SVM and Gradient Boosting. Additionally, among the lexicon-based models, VADER, SentiWordNet, and Pattern exhibit the highest accuracy, respectively.

Tang, T., Huang, L., Chen, Y. and Ieee (2020) evaluated four Chinese sentiment analysis APIs, including Alibaba Cloud, Tencent Cloud, Baidu Cloud, and JD Cloud. These APIs were tested using authentic online reviews collected from Ctrip.com, a major online travel agency in China. The study aimed to assess the accuracy and performance of these mainstream sentiment analysis tools, revealing that all four APIs exhibited accuracy rates of around 50% with none exceeding 60%. The primary reasons for misinterpretations were attributed to segmentation ambiguity and context ambiguity, highlighting the challenges in accurately understanding Chinese text for sentiment analysis.

Kothalawala, M., & Thelijjagoda, S. (2020) conducted research to perform aspect-based sentiment analysis on customer reviews of hair care products. The data set used here consisted of over 400,000 online reviews for approximately 4,500 mobile phones from Amazon.com and stored the datasets in CSV format. They developed a system to extract aspect-wise polarity from consumer reviews, addressing the need for manufacturers to understand consumer opinions and consumers to make informed purchase decisions. They collected and pre-processed review data, breaking down opinion units within sentences, and used a Support Vector Machine (SVM) to classify aspects and polarity. The system effectively presented aspect-wise polarity results, but there were challenges in correctly identifying certain aspects, such as "formula." While the system proved valuable, it can be improved with dynamic aspect detection and more training data for increased accuracy.

Mahgoub, A., Atef, H., Nasser, A., Yasser, M., Medhat, W. M., Darweesh, M. S., & El-Kafrawy, P. M. (2022, October) analyzed comprehensive research to evaluate customer

sentiment in the context of 1,500 Amazon electronics reviews from Egyptian customers. They employed both the TextBlob and BERT sentiment analysis models to determine the sentiment of these reviews, presenting their results through data visualization. Their findings indicated a predominance of positive sentiment, with an average satisfaction rate of 47%. The evaluation also revealed a significant performance advantage for BERT over TextBlob, with an accuracy difference of 15% to 25%, highlighting the superiority of word embedding models in sentiment analysis for both Arabic and English reviews.

A. M. Rajeswari, M. Mahalakshmi, R. Nithyashree, and G. Nalini (2020) conducted research to evaluate sentiment analysis on various online reviews using a hybrid approach combining a lexicon-based method (SentiWordNet) with machine learning algorithms. They addressed the limitation of binary sentiment classification by introducing a third "neutral" class. Their results demonstrated that using the lexicon approach and feature extraction using TF-IDF significantly improved accuracy, with logistic regression outperforming other algorithms, achieving an accuracy increase of approximately 6% to 10%. This approach presents a promising solution to improve sentiment analysis by accommodating neutral opinions in review classification.

3. Methodology

3.1 Data Overview

The provided data set contains information related to Appliances product reviews (Jianmo et al,2019). Here's a breakdown of the columns and a short description of the data:

1. Serial No: A serial number or index for each row in the dataset.
2. Overall: The overall rating given to the product.
3. Vote: A column that indicates the number of votes the review received.
4. Verified: A boolean value indicating whether the review is verified.
5. ReviewTime: The date when the review was posted.
6. ReviewerID: An identifier for the reviewer.
7. ASIN: Amazon Standard Identification Number for the product being reviewed.
8. ReviewerName: The name of the reviewer.
9. ReviewText: The text of the review.
10. Summary: A short summary or title of the review.
11. ReviewTime: The review time when it is posted.
12. Category: A list of categories or tags associated with the product.
13. Product Type: The specific type of product.
14. Brand: The brand of the product.
15. Main_cat: The main category to which the product belongs.
16. Price: This column associated price for the product.

This dataset is related to reviews of various product types within the Appliances Category on Amazon. It includes information about the product, reviewer details, and the content of the reviews.

3.2 Data Processing

3.2.1 Data Loading and Cleaning

We had two JSON files. We joined them based on ‘asin’ column and made a combined dataset. For generating the combined data set, we load the JSON files of review and meta table and selected 'category', 'brand', 'main_cat', 'price'- these columns and converted in CSV file as our final dataset that is “ApplianceDataset.csv”. This data set contains 90,739 rows. The implemented codes are in “Part-0 Codes for joining to create final data.py” file.

After getting the previous data, the next step involves the data cleaning process. Joining the datasets resulted in some duplications. Duplicate and null rows on the "review Text" column were removed, resulting in the elimination of 15,300 entries. We also removed null values from columns such as "reviewerName," "summary," and "brand". Subsequently, only reviews marked as "verified" users were retained, resulting in the removal of 4,699 unverified reviews. The dataset was then updated to reflect these changes, containing 70,739 rows. But the dataset was still too large for the class project, so we needed to make it smaller. In order to make it smaller we used proportionate stratified sampling method to prevent any bias.

3.2.2 Proportionate Stratified Sampling to get the concise dataset

To create a proportionate stratified sample with approximately 11,000 rows, 85% of the data was randomly dropped from each stratum, defined by the combination of "product type" and "ratings." The resulting dataset that is “ConciseApplianceDataset.csv”, containing 10,610 rows. The representativeness of the new dataset was validated by comparing the percentages of each product type, ratings, and their combinations with the original dataset. We implemented the wholes process in the file “Part-1 Data Cleaning and Getting 10k representative rows.py”.

The detail process of proportionate stratifies is that we created a list of the 42 product types, and then looped through each product type form that list, to identify the rows with that specific product type and rating 1, then we found some indices to be dropped randomly (85%) from those specific rows. Finally, we dropped those indices, and we repeated this process for rating 2,3,4 and 5 for that specific product. After running the entire loop, we got our proportionate stratified concise dataset having 15% of 70,739 rows or 10,610 rows. We checked the percentage of ratings (1-5), product types count percentage, and each product’s each rating count percentages as well and find that, they are pretty much like those of original dataset.

3.3 Sentiment Analysis Using Google API, VADER and RoBERTa Model

Our goal is to compare the model’s sentiment score with true score interpreted by human and then validate which model works well. For human interpretation we evaluated the reviewsText column manually and provided the true score (‘Human Sentiment Score’ column) and true review type (‘Review Type’ column). It was difficult for us to implement this manual process on whole data set, so again we applied proportionate stratified sampling on final dataset to get the small concise dataset which results in 1993 rows. Subsequently, we tested our models on this dataset and identified the model that performed better.

We used the below score range for scoring:

Negative	Neutral	Positive
-1 to -0.25	-0.25 to +0.25	0.25 to 1

3.3.1 Proportionate Stratified Sample Dataset for Human Interpretation and Model Comparison:

In the picture below it is visualized that how we reach to the sample dataset. We followed the same mechanism in sample dataset creation as we did in obtaining concise dataset Which is explained above. Each rating from each product type considered as stratum like for product A the rating 1 is stratum same as for product B rating 3 or etc., we took 20% of data from each stratum and maintaining the proportion of the stratum of the final dataset. We finally got the file "Finalsample_file.csv" with the 20% of the dataset. Please refer to file "Part-2 Obtaining the Sample dataset.py" for codes related to this part.

3.3.2 Sentiment Analysis with Google API

Google APIs are Application Programming Interfaces (APIs) developed by Google that provides communication with different Google services. These APIs make possible programmatic access to Google's vast data and functionality. (Google Cloud APIs, n.d.)

In this step we implemented sentiment analysis with google API. The Google API performs sentiment analysis on a CSV file. It begins by reading the input file into a data frame, replacing the null values. The script them after that defines a function to analyze sentiment, and applies this function to the 'reviewText' column, adding 'Sentiment_Score' and 'Sentiment_Magnitude' columns to the data frame. Subsequently, it calculates and adds normalized sentiment scores using the hyperbolic tangent function. The final data frame is saved to a new file named "SampleAppliance_Output.csv." Please refer to ‘Part-3 Google API Scoring on Sample Dataset.py’.

3.3.3 VADER Sentiment Analysis

Hutto & Gilbert (2014) VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool specifically designed for handling informal language, slang, and sentiment expressions commonly found in social media.

As the codes for this part are presented in the file “Part-4 VADER and ROBERTA Scoring on Sample dataset. Py”, in this step we need to implement VADER model on the sample. The first step is to import the needed libraries like pandas, numpy, matplotlib. Then we load the data into the data frame.Then we import necessary NLTK libraries and functions like punkt, averaged_perceptron_tagger, maxent_ne_chunker, words, SentimentIntensityAnalyzer. Also we imported tqdm.notebook for progress Bar Tracker for looping. After that, because VADER has the bag of words approach and calculates the sentiment score for each word individually, we need to combine all of the scores in a review to reach an overall score for each review as positive, negative or neutral. Because the VADER does not account for the relationship between the words, we needed to remove the stop words like “and”, “the”, etc. The next step is to create a function using SentimentIntensityAnalyzer() to feed text to derive VADER score. VADER sentiment analysis was applied to the "reviewText" column, providing compound scores for each review. The results were appended to the existing dataset.

3.3.4 RoBERTa Sentiment Analysis

ROBERTA stands for A Robustly Optimized BERT Pretraining Approach, and it is a large language model (LLM). It is based on the BERT architecture, but it uses several training improvements that result in a better performance on different natural language processing tasks (Liu, et al, 2019). A RoBERTa model, was utilized to evaluate sentiment scores. The RoBERTa scores were incorporated into the dataset alongside the VADER scores.

As the last model, we used RoBERTa model for sentiment analysis. Please refer to the file "Part-4 VADER and ROBERTA Scoring on Sample dataset. Py" to understand this part and the loop. We imported AutoTokenizer, AutomodelForSequenceClassification libraries from Transformers package and softmax library (applied to outputs to scale down to 0 and 1) to conduct this RoBERTa code implementation on our sample. As RoBERTa model works on Transfer Learning, we imported a model, 'cardiffnlp/twitter-roberta-base-sentiment', which is trained on bunch of twitter comments that were labled. We tokenized the model for pulling the trained weights for twitter project. Applying the model tokenizer to an example and detaching the scores from an array, we checked if it's working properly. The final scores (negative, neutral, and positive) were saved into a dictionary. After successful testing, we created a function "Polarity score Roberta (with example or text in the argument).

3.3.5 The Looping Process to implement both VADER and ROBERTA on sample:

First, we created an empty dictionary 'result'. Secondly, we created a loop which would consider all the rows in data frame, and for each row it would take each review as variable 'text', would take associated 'Serial No' as 'myid' variable. Thirdly, it would apply VADER polarity score function to each 'text' and rename vader score columns. Fourthly, we would apply the previously created RoBERTa polarity score function on each 'text'. On fifth stage, it would attach VADER & ROBERTA scores in a dictionary('both') and on the final stage it would arrange the result from 'both' dictionaries based on id of each row in 'result' dictionary. We added a 'try and except' in the loop because RoBERTa faced RunTime Error for reviews which had more than 726, that is the word limit of this model. So, this tried the rows which don't face RunTime error and exempted the rows facing these errors (printed the serial number of rows facing RunTime error).

Finally, we get the VADER and RoBERTa scoring done for 1984 rows (9 rows had runtime error due to lengthy reviews). Converting to Data frame and transposing the result dictionary, we joined the scores to our original data frame, on Serial no as key. We dropped the rows for which we could not get the RoBERTa scores (9 rows). After calculating the Roberta compound score for each row, we exported the output to 'Finalsample_file.csv'.

3.3.6 Compound Sentiment Score:

While implementing google API and Roberta models to analysis the sentiment we faced an issue that these models won't give us compound sentiment score, we got sentiment score and magnitude for google API and negative positive and neutral score for Roberta. But we need the compound scores to compare these models. To address this issue, we normalized the scores.

According to Hong (2023), for google API we implemented,

Compound Score: $\tanh\left(\frac{\text{Sentiment\_Magnitude}}{\text{Sentiment\_Score}}\right)$

Here for Google API, we applied tanh function, it is a non-linear function that used to model the input values of sentiment score and magnitude to a range in between -1 and 1. We implemented our code for Google API in the file “Part-3 Google API Scoring on Sample Dataset.py”.

According to Charles (2022) we implemented this derived formula for Roberta,

Compound Score: roberta_neg+2×roberta_neu+3×roberta_pos−2

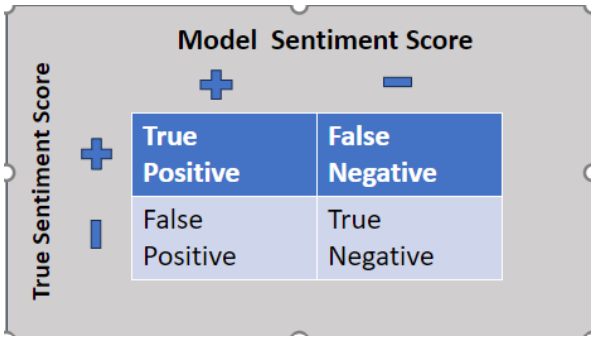
In this formula we got weighted combination involving three variables: "roberta_neg," "roberta_neu," and "roberta_pos." Each variable corresponds to a numerical value linked to sentiment categories, presumably denoting negative, neutral, and positive sentiments, respectively. It is assigned weights 1, 2, and 3 for negative, neutral, and positive sentiments, respectively. Additionally, the sum is subject to a negative weight of 2. This formulation implies a mathematical representation for consolidating sentiment scores, wherein greater importance is attributed to positive sentiments (with a weight of 3) in comparison to neutral sentiments (with a weight of 2) and negative sentiments (with a weight of 1). The subtraction of 2 may indicate a penalty or adjustment factor. The result value from this equation serves as a composite sentiment score, reflecting the overall sentiment based on the weighted contributions of the three sentiment categories. Codes are provided in RoBERTa in “Part-3 Google API Scoring on Sample Dataset.py” file.

3.4 Evaluation Criteria for Model:

There are three criteria’s that we are considering for assessing the performance of the model.

Relationship with the True Sentiment Score: here we compare the distribution of the models and find correlation of the models.
Relationship with the Ratings: We have used score vs rating bar plot for analysis and correlation of ratings.
Precision, Recall, and F1 Scores: This metric measures the model's ability to minimize false positive and false negative.

To identify precision and recall we need to calculate the counts of reviews which are True positives, False positives, True negatives, and False negatives for each model. True positive and True negative are where the model accurately predicts actual sentiment labels, while False Positice and False Negative are misclassifications (Nguyen et al ,2018). We can look at the metrics below and understand True and False positive negative precisely.



Precision assesses the classifier's ability to correctly identify positive sentiment reviews. Recall gauges its effectiveness in identifying negative sentiment reviews. F1 score combines both precision and recall providing a single evaluation metric (Nguyen et al ,2018).

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive(FP)}}$$

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative(FN)}}$$

$$\text{F1 Score} = \frac{2 * \text{True Positive (TP)}}{2 * \text{True Positive (TP)} + \text{False Positive} + \text{False Negative(FN)}}$$

The model with higher Precision indicates it has the least False Positive, and the model with higher Recall has the least False Negative. F1 Score is the harmonic mean of precision and recall, which finds a balance between these two metrics.

We used the 'Review Type' column and Model score columns (Roberta compound score, vader compound score and Google API scores) for identifying the True/False Positive and Negative counts. For example, to get the False positive for VADER model, we conditioned our data frame for only the rows with 'Review Type= Negative' (Actually Negative) and 'vader compound >0.25' (positive on VADER model) and found the length of this data frame. Codes are provided in the "Part 5 Evaluation of the models by 3 criteria.py" file for further details.

4. Result

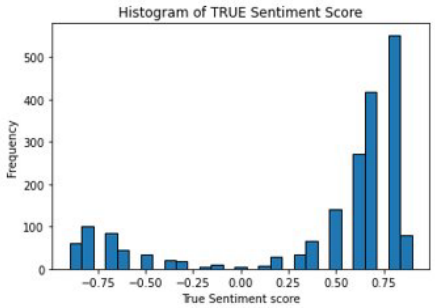
4.1 Deciding the Most Appropriate Model based on sample dataset:

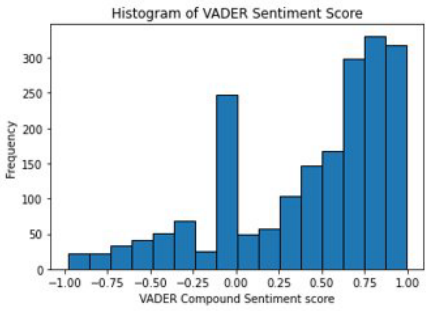
For this part, please refer to "Part-5 Evaluation of models by 3 criteria.py" file where we imported 'Finalsample_file.csv' which had all three models scoring done on the sample dataset of 1993 rows. We did our evaluation of all three models here.

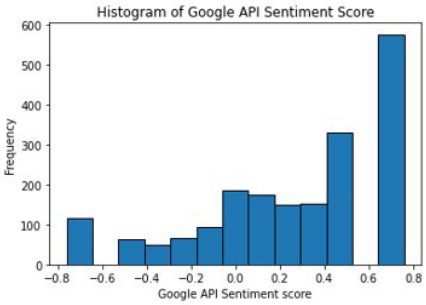
4.1.1 Relationship with True Sentiment Score

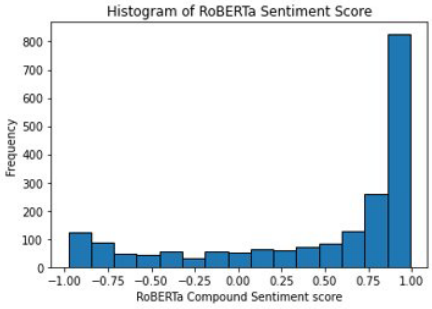
The relationships between True Sentiment scores and model's sentiment scores were explored through histograms, revealing how similar or dissimilar it is from true scores. We also did correlation of the model scoring to true scores.

1. Histograms comparison: We created histograms on ‘Human Sentiment Score’ column to see the distribution of reviews’ true sentiment scores. Similar histograms were created on ‘vader compound’, ‘roberta compound’, ‘Google API Scores’ columns to compare with the distribution of true sentiment scores.









Correlation to True Sentiment Scores

Model	Correlation
VADER	0.55
RoBERTa	0.82
Google API	0.69

As it is shown in the picture, in ‘Histogram of VADER Sentiment Score’ the distribution has a certain peak in the neutral negative zone (0 to -0.25), closer to 0, whereas in ‘Histogram of TRUE Sentiment Score’, there are limited data points in the neutral zone. In the ‘Histogram of Google API Sentiment Score’ we have distribution for 0.6 and -0.6 missing, which might be an after effect of normalization. However, ‘Histogram of RoBERTa Sentiment Score’ is very consistent and has the most similarity to the ‘Histogram of TRUE Sentiment Score’. So, we can conclude that distribution-wise, VADER compound and Google API sentiment scores on the reviews are different from True sentiment scores, however, RoBERTa compound score is very similar.

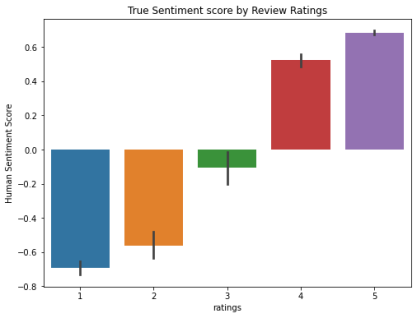
2. Correlation to True Sentiment Score: We can see at the table ‘Correlation to True Sentiment Scores’, VADER is having 0.55 which is moderately positive correlation, RoBERTa is having 0.82 which is the strong positive correlation, and Google API is having 0.69, which is slightly below the strong positive correlation.

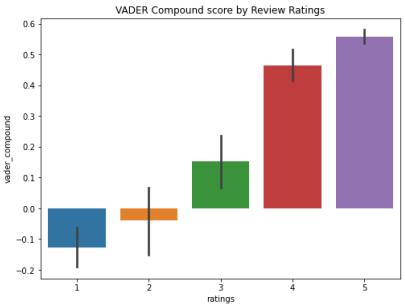
As RoBERTa is having the maximum correlation to True sentiment score and most similarity to its distribution, we can conclude that RoBERTa model is having the strongest relationship to True Sentiment Score among all three models.

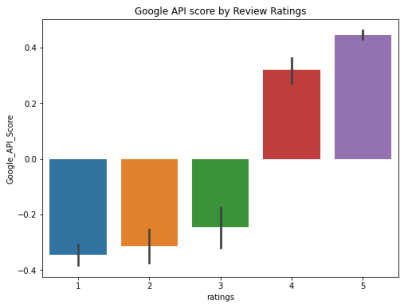
4.1.2 Relationship with Ratings

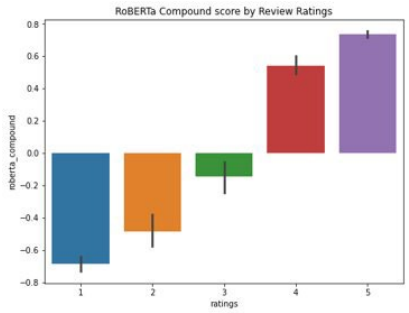
The relationships between sentiment scores and review ratings were explored through bar plots, revealing how sentiment varies across different ratings. We also did correlation of the model's scoring to ratings.

1. Score vs Rating bar plot comparison: Each bar in this graph represents the distribution of review ratings from 1 to 5 based on sentiment score, range from -1 to 1. The line on the bars shows the range of sentiment score for each rating. We created bar plots on 'Review Ratings' column to see the distribution of reviews' true sentiment scores. Similar bar plots were created on 'vader compound', 'roberta compound', 'Google API Scores' columns to compare the distribution of review ratings based on model's sentiment scores.









Correlation to Ratings

Model	Correlation
VADER	0.5
RoBERTa	0.78
Google API	0.67

If we compare all the model's bar plot with true score bar plot, we can see Roberta score is having more similar distribution with true sentiment score, we can conclude that distribution-wise, RoBERTa distribution is very similar.

2. Correlation to Ratings: We can see at the table 'Correlation to Ratings', VADER is having 0.5 which is moderately positive correlation, RoBERTa is having 0.78 which is the strong positive correlation, and Google API is having 0.67, which is slightly below the strong positive correlation.

As RoBERTa is having the maximum correlation to Rating score and most similarity to its distribution, we can conclude that RoBERTa model is having the strongest relationship to Rating Score among all three models.

4.1.3 Precision, Recall, and F1 Scores

We can find counts of the reviews with True Positive, True Negative, False Positive and False Negative reviews for each model on the left side table. Then we calculated the Precision, Recall and F1 score for each model which we can see in the 'Precision, Recall and F1 Score' table.

Counts of True and False Positive & True and False Negative

Model	True Positive	False Positive	True Negative	False Negative
VADER	1249	93	158	65
RoBERTa	1359	20	305	54
Google API	1071	6	199	34

Precision, Recall and F1 Score

Model	Precision	Recall	F1-Score
VADER	0.93	0.95	0.94
RoBERTa	0.99	0.96	0.97
Google API	0.99	0.97	0.98

VADER has the least Precision, Recall and F1 Score among all models. RoBERTa and Google API models are having same Precision (0.99) which means that they are performing equally in terms of minimizing the False positive reviews, however, Google API model (0.97) is having 1% higher Recall than RoBERTa model (0.96), meaning Google API is performing slightly better than RoBERTa in term of having least number of False Negative reviews. Due to difference in Recall, the F1 score for Google API (0.98) is 1% higher than that of RoBERTa(0.97). So based on Recall and F1 Score, Google API is performing a little bit better than RoBERTa.

4.1.4 The Decision:

We declared RoBERTa as the most appropriate model for further analysis, due to following reasons,

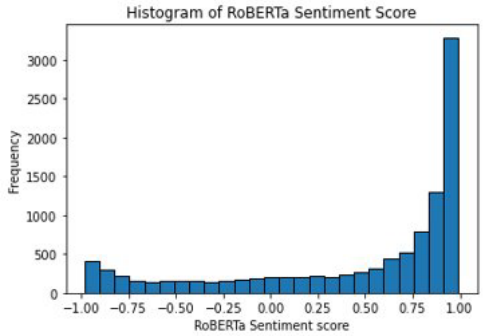
1. On evaluation criteria 1 and 2, RoBERTa is having most similar relationship with True Sentiment score and coding.
2. On evaluation criteria 3, Google API had 1% higher Recall (0.97 vs 0.96) and F1 Score (0.98 vs 0.97) than Roberta Model. This 1% marginal Recall or F1 Score is very insignificant, and we can conclude RoBERTa and Google API models are performing almost similarly in terms of minimizing the False positive and False negatives.
3. Google API has some limitations, like inconsistency in distribution (missing data points with 0.6 and -0.6 scores) for our sample, and google credits are required to perform it.

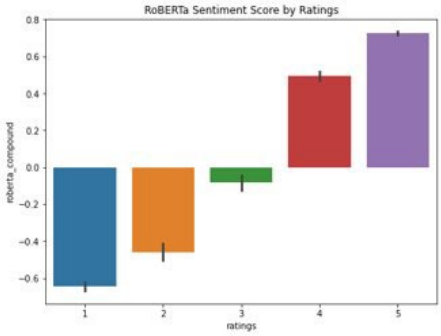
So, we recommend RoBERTa Model for future analysis of Amazon Product reviews based on our evaluation and analysis. This is how we reached our first and main objective.

4.2 Implementation of RoBERTa on Concise Dataset of 10,610 rows:

We implemented the RoBERTa Sentiment score coding on our concise dataset (10,610), following a similar process that we used while applying this on sample and we got 10,565 rows out of 10,610, with 45 rows having RunTime errors. We created the output csv file "Coded_whole_dataset.csv" with RoBERTa scoring for 10,565 rows. Please refer to "Part-6 RoBERTa code implementation of whole dataset.py" file for this part.

Please refer to 'Part-7 Further Evaluation for Recommedation.py' where we load the output csv file. If we look at the of 'Histogram of RoBERTa Senitment Score' and 'RoBERTa Sentiment Score by Rating' bar plot for entire dataset, we can see these are pretty much consistent and similar to those of True sentiment scores from sample. We created a word cloud with the most frequent words from positive and negative reviews. We also found the percentages of Positive Reviews (77%) and Negative Reviews (23%) of the entire 10,565 rows. We also got a strongly positive correlation of RoBERTa score to Ratings (0.76) which was much like what we had in sample between RoBERTa score and Ratings (0.78).





4.3 Product Type Specific Analysis:

Now we run the loops for each of the 42 product types and try to find if there is relationship between Price vs Sentiment Score and Price vs Ratings. The outcomes are going to be similar as Sentiment Score and Ratings normally have a strong correlation. We also tried to find the

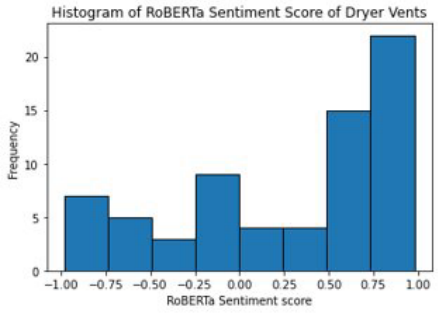
product types with a significant number of rows having the most negative reviews. Codes are provided in “Part-7 Further Evaluation for Recommedation.py” file.

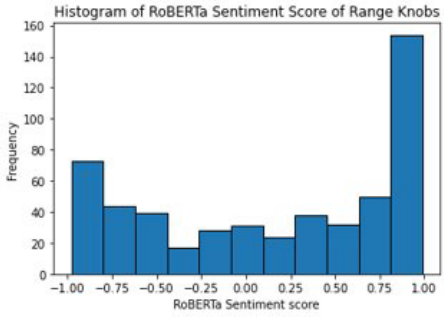
4.3.1 Price vs Sentiment Analysis and Price vs Rating Correlations:

Here we find the correlation between price and Sentiment Score, and between Price and Ratings for each product. We fixed a correlation threshold of 0.3 and -0.3, to get the product types having correlation more than 0.3 and less than -0.3, as |0.3| is slightly more than the weak positive or negative correlation. We only get Single Wall Oven as the only product type with moderate positive correlation between Price and Sentiment score (0.46), and Price vs Ratings (0.50). But this product type only has 13 reviews, due to insufficient rows we cannot consider it as a significant result. So, as we don’t find any other product types with correlation between price and these two things more than 0.3 or less than -0.3, we can conclude that for almost all the product types, Price is not Significantly Related to Sentiment Score and Ratings. This is how we reached our second objective.

4.3.2 Product Type with Highest Negative Reviews:

Earlier we got 23% of our entire dataset had negative reviews. These reviews even include negative neutral reviews as we kept it for broad range analysis (to see even including positive and negative neutrals how many are positive and negative). For each product type, we fixed 34% (1/3) as our threshold for minimum negative reviews percentage, so that we can obtain the products which have more than 1/3 of their reviews as negative. Thus, we obtained ‘Build in Dishwasher’ with 39% negative reviews, but had only 23 reviews, which is why it was considered insignificant.





We also obtained ‘Dryer Vents’ having 35% negative reviews out of its 69 reviews and Range Knobs having 39% negative reviews out of its 530 reviews. However, the ‘Histogram of Dryer Vents’ shows a spike in neutral negative scores (0 to -0.25), which means that it does not have significant portion of data on negative side (-0.25 to -1). Only, ‘Histogram of Range Knobs’ shows that it has significant reviews on negative range (-0.25 to -1). That’s why we can conclude that ‘Range Knobs’ is having the maximum negative sentiment from Public, and we recommend it for improvement to Amazon. This is how we reach our third and final objective for the project.

5. Conclusion

The comprehensive methodology in this paper involved data cleaning, proportionate stratified sampling, sentiment analysis using multiple methods, and thorough evaluation. The results and visualizations provided insights into the performance of sentiment analysis models and their relationships with human sentiment scores and review ratings. The evaluation metrics shed light on the precision, recall, and F1 scores, offering a holistic understanding of the sentiment analysis models' effectiveness.

In short, the paper accomplished all its three objectives. RoBERTa model had similar precision recall and F-1 score as Google API, however, as it had the strongest relationship to True sentiment score and Ratings than the VADER and Google API. So RoBERTa was declared as the most appropriate sentiment analysis model out of these three for future sentiment analysis of Appliance product reviews, which is how the paper completes its main objective. The paper finds that no significant relationship exists between the price of an appliance product and reviews/ratings of public. The paper finds 'Range Knobs' as the product type having maximum negative public sentiment and recommends it for improvement.

Future studies can be brand specific analysis, or review analysis by time. The scope of sentiment analysis and so wide and more studies and implications on this area can help the E-Commerce sites, brands, and companies to understand customers and improve their supply chain, research on products, and alter their business strategies.

References:

Afifah, K., Yulita, I. N., & Sarathan, I. (2021, October). Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier. In 2021 International Conference on Artificial Intelligence and Big Data Analytics (pp. 22-27). IEEE.

A.M. Rajeswari, M. Mahalakshmi, R. Nithyashree, and G. Nalini (2020). "Sentiment analysis for predicting customer reviews using a hybrid approach," in 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), pp. 200–205, Cochin, India,2020.

Charles, C. (2022, August 15). Sentiment analysis using snscrape and roberta. https://medium.com/mlearning-ai/tweets-sentiment-analysis-with-roberta-1f30cf4e1035

Dyson, Z. A., Prakoso, M. R. B., Putra, M. S. Q., Syaputra, M. S. A. F., Achmad, S., & Sutoyo,R. (2023). Sentiment analysis for customer review: Case study of Traveloka. Procedia Computer Science, 216, 682-690.

Hong, Won Kee. (2023). Understanding artificial neural networks: Analogy to the biological neuron model. In *Artificial Intelligence-Based Design of Reinforced Concrete Structures*

(pp. 7-13). Woodhead Publishing.
https://www.sciencedirect.com/topics/engineering/hyperbolic-tangent

Huang, C.y., & Rashid, A. (2021). Sentiment Analysis on Consumer Reviews of Amazon Products. International Journal of Computer Theory and Engineering, Vol 13, No. 2.

Hussain, S., Dhanda, N., & Verma, R. (2023). Sentiment Analysis of Amazon Product Reviews. International Conference on Communication and Electronics Systems. Shenzhen: IEEE.

Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. International AAAI Conference on Weblogs and Social Media. Ann Arbor: Association for the Advancement of Artificial Intelligence.

Jianmo Ni, Jiacheng Li, Julian McAuley Empirical Methods in Natural Language Processing (EMNLP), 2019 https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/

Kothalawala, M., & Thelijjagoda, S. (2020). Aspect-based sentiment analysis on hair care product reviews. In: 2020 International Research Conference on Smart Computing and Systems Engineering (SCSE), IEEE. https://doi.org/10.1109/SCSE49731.2020.9313040

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Mahgoub, A., Atef, H., Nasser, A., Yasser, M., Medhat , W. M., Darweesh, M. S., & El-Kafrawy, P. M. (2022, October). Sentiment Analysis: Amazon Electronics Reviews Using BERT and Textblob. In 2022 20th International Conference on Language Engineering (ESOLEC) (Vol. 20, pp. 6-10). IEEE.

M D, D., C, S., & Ganesh, A. (2016). Sentiment Analysis: A Comparative Study On Different Approaches. Fourth International Conference on Recent Trends in Computer Science & Engineering. (pp. 44-49). Chennai: ScienceDirect.

Nguyen, H., Veluchamy, A., Diop, M., & Iqbal, R. (2018). Comparative study of sentiment analysis with product reviews using machine learning and lexicon-based approaches. SMU Data Science Review, 1(4), 7.

Tang, T., Huang, L., Chen, Y. and Ieee (2020), “Evaluation of Chinese sentiment analysis APIs based on online reviews”, IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Electr Network, pp. 923-927.

Xiao, Y., Qi, C., & Leng, H. (2021, March). Sentiment analysis of Amazon product reviews based on NLP. In 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE) (pp. 1218-1221). IEEE.

Group Contribution

Our team, consisting of three members, exhibited exceptional collaboration and commitment throughout our Python course project. We convened weekly meetings on Tuesdays from 3:00 PM to 4:30 PM, fostering open discussions, idea exchanges, and strategic planning. From the project's inception—choosing the topic, finding the dataset, drafting the proposal, conducting the manual review of 2000 rows, coding implementation, PowerPoint preparation, to final report writing—we collaborated seamlessly. Our interactions were marked by mutual respect, effective communication, and a shared vision. Each member's dedication and expertise contributed significantly, fostering a cohesive and productive team dynamic that ultimately led to the successful achievement of our objectives.