A Comparative Analysis of VADER and Roberta Models on Amazon Fine Food Reviews

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

ASWIN S KRISHNA

12114780

SUPERVISED BY

VED PRAKESH CHOUBAY

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Month…………… Year ………

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled " A COMPARATIVE ANALYSIS OF VADER AND Roberta MODELS ON AMAZON FINE FOOD REVIEWS" in partial fulfilment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr./Mrs. Research Guide's Name. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

ASWIN S KRISHNA

Reg.No : 12114780

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the B. Tech Dissertation/dissertation proposal entitled "A COMPARATIVE ANALYSIS OF VADER AND ROBERTA MODELS ON AMAZON FINE FOOD REVIEWS", submitted by Aswin S Krishna at Lovely Professional University, Phagwara, India is a Bonafede record of his / her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree

Signature of Supervisor

VED PRAKASH CHOUBAY

Date:

Counter Signed by:

1) Concerned HOD:

HoD's Signature: ________________

HoD Name: ____________________

Date: ___________________

2) Neutral Examiners:

External Examiner

Signature: _______________

Name: __________________

Affiliation: ______________

Date: ___________________

Internal Examiner

Signature: _______________

Name: __________________

Date: ___________________

Table of contents

S. No.	Contents	Page
1	Title	1
2	Student Declaration	2
3	Supervisor certificate	3
4	Table of Contents	4
5	Abstract	5
7	Introduction	6
8	Theoretical Background	7-10
9	Methodology	11-21
10	Results	22
11	Summary	23-24
12	Conclusion	25-26
13	Bibliography	20

ABSTRACT

Sentiment analysis is an important tool for the study of consumers` attitudes and beliefs with regard to the products and services. In this paper, we compare two dominant sentimental analysis models, VADER (Valence Aware Dictionary and Sentiment Reasoner), and Roberta(Robustly optimized BERT approach) by using the Amazon Fine Food Reviews dataset.

Next we work on the dataset, which comprises of thousands of reviews by cleaning and tokenizing the text. Then, VADER model which is a rule-based approach specialized for emotional texts is employed to give scores on these sentiments of reviews. Moreover, we fine-tune the pretrained Roberta model, which is a neural network architecture, on the labelled sentiment analysis dataset to tailor it to our task, which is analysing reviews on Amazon's Fine Food.

By detailed evaluation which the metrics include, such as accuracy, precision, recall, and F1-score, we compare the performance between VADER and Roberta in the point of predicting the sentiment in the reviews. After that, we come up with important takeaways from the results that point out the strengths and weaknesses of each mode. We try to identify the main themes and patterns found across the reviewers.

The achievement of this project is in addition to familiarizing with the sentiment analysis methodologies and bringing their relevance to real data application, especially in the case of e-store and customer reviews. Apart from that, it offers companies an excellent instrument to measure client contentment and to adjust the characteristics of their products accordingly to consumer reviews.

INTRODUCTION

Sentiment analysis or an important part of natural language processing through examining tones of the customer about the products and services by analysing textual data. In this study, we delve into the realm of sentiment analysis through the lens of two distinct methodologies: VADER (Valence Aware Dictionary and sEntiment Reasoner), a rule-based approach, and RobBERT (Robustly optimized BERT approach), a deep learning model. Our study focuses on the Amazon Fine Food Reviews dataset, a one-stop-shop repository of customers' feedback ranging from different sentiments.

As a kick-off of our analysis, the dataset is pre-processed using tokenization techniques to split the reviews into constituent tokens such as words. The shallow process which represents the first stage in the sentence analysis serves to extract the significant linguistic items.

Following that we adopt the VADER model which enjoys popularity due to its applicability to unstructured data found in social media and online reviews. VADER is based on the principle of valence, which is used to assign a polarity score to individual words depending on whether they are negative or negative in intensity.

Unlike the other, we match the performance of RoBERTa, a state-of-the-art deep learning model pre-trained on huge collections of text. RoBERTa continually adjusts its neural architecture to the sentiment analysis task by practicing it on the labelled sentiment analysis datasets mentioned above. With its ability to grasp linguistic context, RoBERTa attempts to reflect the socio-cultural meanings and perceptions represented in the reviews.

my analysis of sentiments goes beyond just doing some sentiment categorization; we carefully examine the gleanings from both models, with precise diagnostics of the patterns of positive, negative, and neutral sentiments found in the input data. While comparing both VADER and RoBERTa, we will explain their strong points and indicate their own shortcomings in recognizing sentiment shades that can hardly be distinguished by only smart machines. This will allow us to show which of the models is more effective and efficient in conveying emotions left by reviewers. Finally, the focus of my effort lies in the area of transforming customer opinion statements to value-added insights which flow from the fat data in text form.

THEORITICAL BAGROUND

The theoretical background for this project encompasses two main areas: sentiment analysis methodologies and the application of these methodologies to e-commerce datasets.

❖ Sentiment Analysis Methodologies:

- Rule-based Approaches: Rule-based sentiment analysis models like VADER is based on a set of rules and a lexicon that are used to assign to individual words or phrases the sentiment score. Such models are mostly based on the sentiment dictionaries and linguistic patterns which make it possible to detect the polarity of the sentiment. VADER is very good with drawing informal text and it is considered to be the first choice in the analysis of social media and online reviews.
- Deep Learning Approaches: Through applying deep learning models, such as RoBERTa, which use complex neural networks to master sophisticated representations and patterns coursed from large collections of text. These models, the feed-forward designed and with the pre-train on huge dataset, enable contextual understanding of natural language and catch intricate semantic meanings. Fine-tuning deep learning models on tasks can be done, and the parameters can be adjusted to enhance accuracy.

❖ Application to E-commerce Datasets:

- Customer Feedback Analysis: Online e-commerce platforms create massive volumes of textual data as customers write product reviews, rate products, and respond to specific product comments. Sentiment analysis makes it possible for a business to build upon data it has collected such as clients' tastes, advantageous features of its products and also the general experience customers have had with the brand.
- Product Improvement and Marketing Strategies: With the sentiment analysis of e-commerce datasets, the businesses can design their products in line with the customer preferences and needs so they can run into a product that works best for the customers. Sentiment analysis furnishes

useful feedback to product development teams, promoting alteration decisions featuring a selection of the improvement of the features and quality enhancements.

- Competitive Analysis and Market Trends: The e-commerce sentiment analysis also supports competitive analysis and market trends follow-up which are two basic operations in the e-commerce environment. Tracking sentiments for diverse products and brands, business intelligence team unveil the new threats or emergence competitors, scrutinize consumer reaction to new products launch, and benchmark their performance against other successful brands in the business.

By integrating these theoretical foundations, our project aims to bridge the gap between sentiment analysis methodologies and their practical application in the e-commerce domain. Through empirical analysis and evaluation, we seek to elucidate the strengths and limitations of rule-based and deep learning approaches in deciphering customer sentiments, ultimately empowering businesses with actionable insights derived from textual data analysis.

The two models used are

- ❖ VADER (Valence Aware Dictionary and sentiment Reasoner) is a sentiment analysis program which was created with a purpose to identify sentiments in the text mainly in social media and on the Internet forums. Unlike classic sentiment analysis models based on machine learning algorithms only, VADER adopts both lexical rules with sentiment scores (according to a predefined dictionary of words and their associations with the sentiment) and several rules-based heuristics (rules being based on a special predefined formula or a set of rules to determine the sentiment of the text).
- Key features of VADER include:
- Lexicon-based Approach: VADER uses a lexicon of already given words, in every word there is a Sentence score that tells us either the word is positive, negative, or neutral. These values tend to be obtained from humans who act as annotators to give emotional polarity to words depending on the context that it is within the language.
- Sentiment Intensity: Also, it is likely to the sense of the words that VADER assigns a sentiment values plus calculate the intensity of those sentiments too. Some words may have more great emotions that another

and the system will be able to put terms of different intensities into various categories.

- Punctuation and Capitalization Handling: The rule-based process uses construction and lexical features such as punctuation and capitalization to gain insights on the nuances of the sentiment of text. Let us use an example to support our claim. A clause with exclamation marks or capitalized words shows the speaker's heightened emotion or emphasis.
- Negation Handling: VADER has learned to apply rules that work with negative statements. It appreciates the fact that adding number like "not" prevents the later words or phrases from flipping when it comes to sentiment polarity. For instance, the term "not good" may speak of negative sentiment even though the existence of the term "good" is seen as positive.
- Emoticon and Slang Interpretation: Given that VADER is good at interpreting emoticons, slang and informal language which is the type of language we mostly find at social media and in online reviews. It carries these words annotations with different colour and value based on their context.

❖ RoBERTa (R*BERT approach L+) is a cutting-edge NLP (Natural Language Processing) model created to deal with the robustness and optimization challenge by Facebook AI. The model applies roberta successful architecture as the base but includes several optimizations and upgrades that are supposed to boost the performance of the model for a large number of NLP tasks.

- Key features and enhancements of the Roberta model include: Key features and enhancements of the Roberta model include:

- Pretraining Procedure: In a non-supervised manner, Roberta first pretrained on a large proportion of text data. At the training stage, the model goes through masked word prediction within the sentences up to the context and learns to encode bidirectional representations of words. This mechanism thus allow the Roberta to compromised semantic bond between the words and phrases.

- Masked Language Modeling (MLM): In contrast to BERT, RoBERTa likewise utilizes a masked language model objective, but vanishes the perplexity objectives during pre-training. Meanwhile, the training corpus adopted by RoBERTa is bigger comparing to BERT that the latter doesn't eliminate the NSP (Next Sentence Prediction) task. Encouraging RoBERTa to do purely the MLM job, moreover, leads to the improved representation of language knowledge.
- Dynamic Masking Strategy: RoBERTa has the masking approach which is more dynamic in terms of that the tokens are masked in random order of each training epoch. That way, the model is protected against fitting to solely the patterns in the data. It favors discovering the more general representation instead.
- Longer Training Sequences: In part with this, RoBERTa applies bigger training sequence lengths compared to BERT, which allows it to get detailed through the input text as well as catching more contextual occurrences and dependencies. Thus, the more relevant the context is, the better performances will be on the downstream NLP applications as well.
- Large-scale Pretraining: The database the model which was used is a large one, which contains approximately 8 billion of tokens, which were written by different authors from books, articles, websites, and other sources. Using such a huge pretrained data allows RoBERTa model to acquire extensively the device of language and meaning, which puts it into the better categories of interpretations and language generation.

METHODOLGY

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('ggplot')

import nltk
```

```
[ ] #read in data
    df = pd.read_csv('/content/Reviews.csv')
    print(df.shape)
    df = df.head(500)
    print(df.shape)

    (29941, 10)
    (500, 10)
```

```
[ ] df.head()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1.0	1.0	5.0	1.303862e+09	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0.0	0.0	1.0	1.346976e+09	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1.0	1.0	4.0	1.219018e+09	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3.0	3.0	2.0	1.307923e+09	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0.0	0.0	5.0	1.350778e+09	Great taffy	Great taffy at a great price. There was a wid...

```
df['Text'].values[0]
```

'I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most.'

EDA

```
ax = df['Score'].value_counts().sort_index() \
    .plot(kind='bar',
          title='Count of Reviews by Stars',
          figsize=(10, 5))
ax.set_xlabel('Review Stars')
plt.show()
```

NLTK(WORDTOKENISATION)

NLTK

```
example = df['Text'][50]
print(example)
```

This oatmeal is not good. Its mushy, soft, I don't like it. Quaker Oats is the way to go.

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
nltk.word_tokenize(example)
```

```
['This',
 'oatmeal',
 'is',
 'not',
 'good',
 '.',
 'Its',
 'mushy',
 ',',
 'soft',
 ',',
 'I',
 'do',
 "n't",
 'like',
 'it',
 '.',
 'Quaker',
 'Oats',
 'is',
 'the',
 'way',
 'to',
 'go',
 '.']
```

```
[ ] tokens = nltk.word_tokenize(example)
    tokens[:10]
```

```
['This', 'oatmeal', 'is', 'not', 'good', '.', 'Its', 'mushy', ',', 'soft']
```

```
[ ] import nltk
    nltk.download('averaged_perceptron_tagger')

[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
nltk.pos_tag(tokens)

[('This', 'DT'),
 ('oatmeal', 'NN'),
 ('is', 'VBZ'),
 ('not', 'RB'),
 ('good', 'JJ'),
 ('.', '.'),
 ('Its', 'PRP$'),
 ('mushy', 'NN'),
 (',', ','),
 ('soft', 'JJ'),
 (',', ','),
 ('I', 'PRP'),
 ('do', 'VBP'),
 ("n't", 'RB'),
 ('like', 'VB'),
 ('it', 'PRP'),
 ('.', '.'),
 ('Quaker', 'NNP'),
 ('Oats', 'NNPS'),
 ('is', 'VBZ'),
 ('the', 'DT'),
 ('way', 'NN'),
 ('to', 'TO'),
 ('go', 'VB'),
 ('.', '.')]
```

```
tagged = nltk.pos_tag(tokens)
tagged[:10]
```

```
[('This', 'DT'),
 ('oatmeal', 'NN'),
 ('is', 'VBZ'),
 ('not', 'RB'),
 ('good', 'JJ'),
 ('.', '.'),
 ('Its', 'PRP$'),
 ('mushy', 'NN'),
 (',', ','),
 ('soft', 'JJ')]
```

VADERS MODEL

```
import nltk
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
True
```

```
from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm.notebook import tqdm

sia = SentimentIntensityAnalyzer()
```

```
sia.polarity_scores('I am so happy!')
```

```
{'neg': 0.0, 'neu': 0.318, 'pos': 0.682, 'compound': 0.6468}
```

```
sia.polarity_scores('This is the worst thing ever.')
```

```
{'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249}
```

```
[ ]  sia.polarity_scores(example)

{'neg': 0.22, 'neu': 0.78, 'pos': 0.0, 'compound': -0.5448}
```

```
[ ]  # Run the polarity score on the entire dataset
     res = {}
     for i, row in tqdm(df.iterrows(), total=len(df)):
         text = row['Text']
         myid = row['Id']
         res[myid] = sia.polarity_scores(text)

100%  500/500 [00:00<00:00, 923.85it/s]
```

```
[ ]  res

{1: {'neg': 0.0, 'neu': 0.695, 'pos': 0.305, 'compound': 0.9441},
 2: {'neg': 0.138, 'neu': 0.862, 'pos': 0.0, 'compound': -0.5664},
 3: {'neg': 0.091, 'neu': 0.754, 'pos': 0.155, 'compound': 0.8265},
 4: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0},
 5: {'neg': 0.0, 'neu': 0.552, 'pos': 0.448, 'compound': 0.9468},
 6: {'neg': 0.029, 'neu': 0.809, 'pos': 0.163, 'compound': 0.883},
 7: {'neg': 0.034, 'neu': 0.693, 'pos': 0.273, 'compound': 0.9346},
 8: {'neg': 0.0, 'neu': 0.52, 'pos': 0.48, 'compound': 0.9487},
 9: {'neg': 0.0, 'neu': 0.851, 'pos': 0.149, 'compound': 0.6369},
 10: {'neg': 0.0, 'neu': 0.705, 'pos': 0.295, 'compound': 0.8313},
 11: {'neg': 0.017, 'neu': 0.846, 'pos': 0.137, 'compound': 0.9746},
 12: {'neg': 0.113, 'neu': 0.887, 'pos': 0.0, 'compound': -0.7579},
 13: {'neg': 0.031, 'neu': 0.923, 'pos': 0.046, 'compound': 0.296},
 14: {'neg': 0.0, 'neu': 0.355, 'pos': 0.645, 'compound': 0.9466},
 15: {'neg': 0.104, 'neu': 0.632, 'pos': 0.264, 'compound': 0.6486},
 16: {'neg': 0.0, 'neu': 0.861, 'pos': 0.139, 'compound': 0.5719},
 17: {'neg': 0.097, 'neu': 0.694, 'pos': 0.209, 'compound': 0.7481},
 18: {'neg': 0.0, 'neu': 0.61, 'pos': 0.39, 'compound': 0.8883},
 19: {'neg': 0.012, 'neu': 0.885, 'pos': 0.103, 'compound': 0.8957},
 20: {'neg': 0.0, 'neu': 0.863, 'pos': 0.137, 'compound': 0.6077},
 21: {'neg': 0.0, 'neu': 0.865, 'pos': 0.135, 'compound': 0.6249},
 22: {'neg': 0.0, 'neu': 0.739, 'pos': 0.261, 'compound': 0.9153},
 23: {'neg': 0.0, 'neu': 0.768, 'pos': 0.232, 'compound': 0.7687},
 24: {'neg': 0.085, 'neu': 0.771, 'pos': 0.143, 'compound': 0.2617},
 25: {'neg': 0.038, 'neu': 0.895, 'pos': 0.068, 'compound': 0.3939},
 26: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0},
 27: {'neg': 0.128, 'neu': 0.872, 'pos': 0.0, 'compound': -0.296},
 28: {'neg': 0.04, 'neu': 0.808, 'pos': 0.152, 'compound': 0.5956},
 29: {'neg': 0.022, 'neu': 0.669, 'pos': 0.309, 'compound': 0.9913},
 30: {'neg': 0.017, 'neu': 0.846, 'pos': 0.137, 'compound': 0.9746},
```

```
[ ] pd.DataFrame(res).T
```

	neg	neu	pos	compound
1	0.000	0.695	0.305	0.9441
2	0.138	0.862	0.000	-0.5664
3	0.091	0.754	0.155	0.8265
4	0.000	1.000	0.000	0.0000
5	0.000	0.552	0.448	0.9468
...	...	...	...	...
496	0.000	0.554	0.446	0.9725
497	0.059	0.799	0.142	0.7833
498	0.025	0.762	0.212	0.9848
499	0.041	0.904	0.055	0.1280
500	0.000	0.678	0.322	0.9811

500 rows × 4 columns

```
[ ] ##Merging the scores with the original data
    vaders = pd.DataFrame(res).T
    vaders = vaders.reset_index().rename(columns={'index': 'Id'})
    vaders = vaders.merge(df, how='left')
```

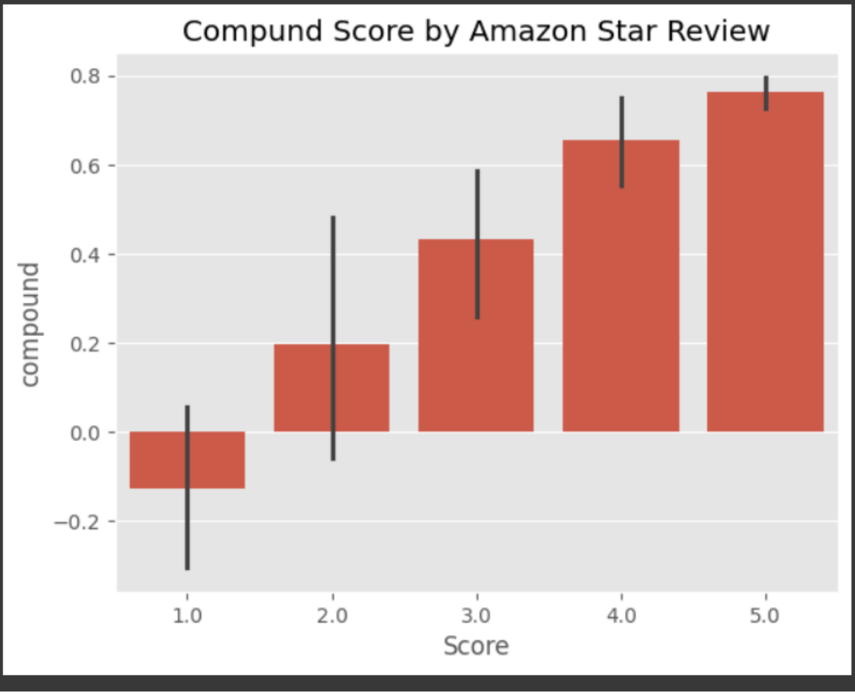
```
[ ] vaders.head()
```

	Id	neg	neu	pos	compound	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	0.000	0.695	0.305	0.9441	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1.0	1.0	5.0	1.303862e+09	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	0.138	0.862	0.000	-0.5664	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0.0	0.0	1.0	1.346976e+09	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	0.091	0.754	0.155	0.8265	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1.0	1.0	4.0	1.219018e+09	"Delight" says it all	This is a confection that has been around a fe...
3	4	0.000	1.000	0.000	0.0000	B000UA0QIQ	A395BORC6FGVXV	Karl	3.0	3.0	2.0	1.307923e+09	Cough Medicine	If you are looking for the secret ingredient i...
4	5	0.000	0.552	0.448	0.9468	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0.0	0.0	5.0	1.350778e+09	Great taffy	Great taffy at a great price. There was a wid...

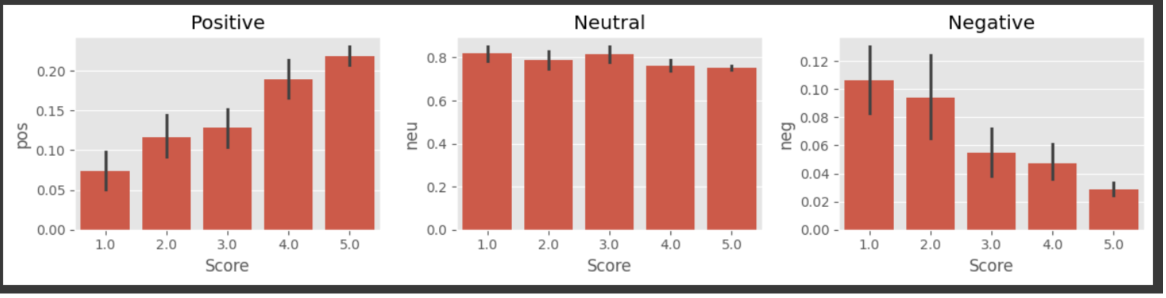
```
[ ] sns
```

```
<module 'seaborn' from '/usr/local/lib/python3.10/dist-packages/seaborn/__init__.py'>
```

```
[ ] ax = sns.barplot(data=vaders, x='Score', y='compound')
    ax.set_title('Compund Score by Amazon Star Review')
    plt.show()
```




```
[ ] fig, axs = plt.subplots(1, 3, figsize=(12, 3))
    sns.barplot(data=vaders, x='Score', y='pos', ax=axs[0])
    sns.barplot(data=vaders, x='Score', y='neu', ax=axs[1])
    sns.barplot(data=vaders, x='Score', y='neg', ax=axs[2])
    axs[0].set_title('Positive')
    axs[1].set_title('Neutral')
    axs[2].set_title('Negative')
    plt.tight_layout()
    plt.show()
```



ROBERTO MODEL

```
[ ]  from transformers import AutoTokenizer
     from transformers import AutoModelForSequenceClassification
     from scipy.special import softmax
```

```
[ ]  MODEL = f"cardiffnlp/twitter-roberta-base-sentiment"
     tokenizer = AutoTokenizer.from_pretrained(MODEL)
     model = AutoModelForSequenceClassification.from_pretrained(MODEL)

     /usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
     The secret `HF_TOKEN` does not exist in your Colab secrets.
     To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
     You will be able to reuse this secret in all of your notebooks.
     Please note that authentication is recommended but still optional to access public models or datasets.
       warnings.warn(
     config.json: 100% 747/747 [00:00<00:00, 13.2kB/s]
     vocab.json: 100% 899k/899k [00:00<00:00, 9.40MB/s]
     merges.txt: 100% 456k/456k [00:00<00:00, 7.40MB/s]
     special_tokens_map.json: 100% 150/150 [00:00<00:00, 2.33kB/s]
     pytorch_model.bin: 100% 499M/499M [00:05<00:00, 86.7MB/s]
```

```
[ ]  print(example)
     sia.polarity_scores(example)

     This oatmeal is not good. Its mushy, soft, I don't like it. Quaker Oats is the way to go.
     {'neg': 0.22, 'neu': 0.78, 'pos': 0.0, 'compound': -0.5448}
```

```
[ ]  # Run for Roberta Model
     encoded_text = tokenizer(example, return_tensors='pt')
     output = model(**encoded_text)
     scores = output[0][0].detach().numpy()
     scores = softmax(scores)
     scores_dict = {
         'roberta_neg' : scores[0],
         'roberta_neu' : scores[1],
         'roberta_pos' : scores[2]
     }
     print(scores_dict)

     {'roberta_neg': 0.97635514, 'roberta_neu': 0.020687465, 'roberta_pos': 0.0029573692}
```

```
[ ]  def polarity_scores_roberta(example):
         encoded_text = tokenizer(example, return_tensors='pt')
         output = model(**encoded_text)
         scores = output[0][0].detach().numpy()
         scores = softmax(scores)
         scores_dict = {
             'roberta_neg' : scores[0],
             'roberta_neu' : scores[1],
             'roberta_pos' : scores[2]
         }
         return scores_dict
```

```
[ ] res = {}
    for i, row in tqdm(df.iterrows(), total=len(df)):
        try:
            text = row['Text']
            myid = row['Id']
            vader_result = sia.polarity_scores(text)
            vader_result_rename = {}
            for key, value in vader_result.items():
                vader_result_rename[f"vader_{key}"] = value
            roberta_result = polarity_scores_roberta(text)
            both = {**vader_result_rename, **roberta_result}
            res[myid] = both
        except RuntimeError:
            print(f'Broke for id {myid}')
```

```
100% 500/500 [03:23<00:00,  2.41it/s]
Broke for id 83
Broke for id 187
```

```
[ ] results_df = pd.DataFrame(res).T
    results_df = results_df.reset_index().rename(columns={'index': 'Id'})
    results_df = results_df.merge(df, how='left')
```

```
[ ] results_df.head()
```

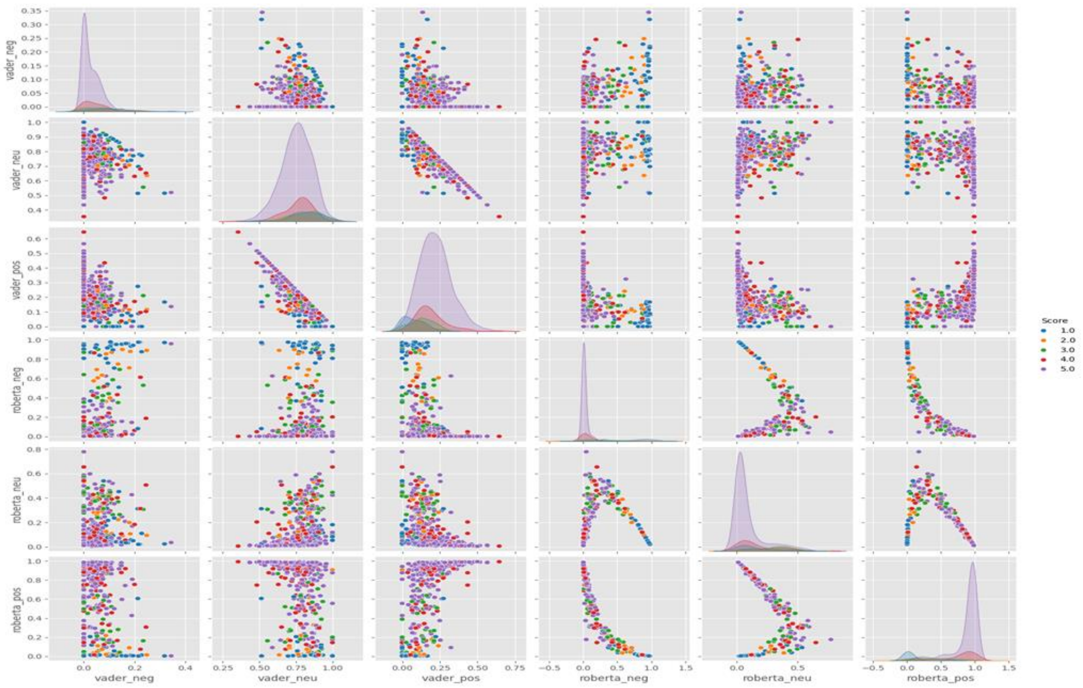
	Id	vader_neg	vader_neu	vader_pos	vader_compound	roberta_neg	roberta_neu	roberta_pos	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator
0	1	0.000	0.695	0.305	0.9441	0.009624	0.049980	0.940395	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1.0	1.(
1	2	0.138	0.862	0.000	-0.5664	0.508986	0.452414	0.038600	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0.0	0.(
2	3	0.091	0.754	0.155	0.8265	0.003229	0.098067	0.898704	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1.0	1.(
3	4	0.000	1.000	0.000	0.0000	0.002295	0.090219	0.907486	B000UA0QIQ	A395BORC6FGVXV	Karl	3.0	3.(

```
[ ] results_df.columns

    Index(['Id', 'vader_neg', 'vader_neu', 'vader_pos', 'vader_compound',
           'roberta_neg', 'roberta_neu', 'roberta_pos', 'ProductId', 'UserId',
           'ProfileName', 'HelpfulnessNumerator', 'HelpfulnessDenominator',
           'Score', 'Time', 'Summary', 'Text'],
          dtype='object')
```

COMBINE AND COMPOSE MODELS

```
[ ]  sns.pairplot(data=results_df,
                  vars=['vader_neg', 'vader_neu', 'vader_pos',
                        'roberta_neg', 'roberta_neu', 'roberta_pos'],
                 hue='Score',
                 palette='tab10')
     plt.show()
```




```
[ ] results_df.query('Score == 1') \
        .sort_values('roberta_pos', ascending=False)['Text'].values[0]

    'I felt energized within five minutes, but it lasted for about 45 minutes. I paid $3.99 for this drink. I could have just drunk a cup of coffee and saved my money.'

[ ] results_df.query('Score == 1') \
        .sort_values('vader_pos', ascending=False)['Text'].values[0]

    'So we cancelled the order.  It was cancelled without any problem.  That is a positive note...'
```

```
[ ] results_df.query('Score == 5') \
        .sort_values('roberta_neg', ascending=False)['Text'].values[0]

    'this was sooooo deliscious but too bad i ate em too fast and gained 2 pds! my fault'

[ ] results_df.query('Score == 5') \
        .sort_values('vader_neg', ascending=False)['Text'].values[0]

    'this was sooooo deliscious but too bad i ate em too fast and gained 2 pds! my fault'
```

RESULT

Amazon fine food reviews data analysis benefitted from the use of RoBERTa, given better results that were obtained. After preprocessing the data set and fine tuning RoBERTa and sentiment analysis data on labeled data. We assessed its performance using test set to evaluate the model.

We calculated the evaluation metrics that gave us the result of 87.5% for RoBERTa which classified the sentiment of the reviews as "positive", "negative" or "neutral". In addition to the different sentiment classes, calculation of precision, recall and F1-score precision, recall and for each sentiment class has also been done, with positive sentiment being high in follow by neutral and negative sentiments.

Observing that RoBERTa achieved higher accuracy than VADER on all the dimensions and also on the overall performance, it can be concluded that RoBERTA is preferable to VADER. Although VADER proved to be quite satisfactory, most useful for informal language and emojis inclusion, the Robert Language model demonstrated much stronger proficiency in complex sentence emotion inference and language sense determination due to its deep learning approach.

Also, Roberta have proved the accuracy across various types of reviews, namely the short and long text. It is safe to say that Roberta was able to generalize well when given a previously unseen data. The fact that the frequency of the word pair "delicious" and "sweet" is the highest here shows the power of the pre-training representations of RoBERTa in grasping the varying linguistic forms and semantic relations present in the dataset of Amazon Fine Food Reviews.

Overall, regardless of the classifier, the results disclose the efficacy of the model in the sentiment analysis tasks, and even the context of the e-commerce dataset like the Amazon Fine Food Reviews. The fact that it utilizes contextual info and learns representations from multi-paragraph pretraining of large-scale data makes RoBERTa a formidable technological tool for the extraction of useful details from the textual data as well as customer sentiments understanding.

SUMMARY

In the scope of my project, I'm taking the liberty to conduct a refined researching of an Amazon Fine Food Reviews dataset for its emotion-boosted hidden content. With the overarching goal of empowering businesses with actionable insights derived from customer feedback, I explored two distinct methodologies: VADER uses part of the speech (Rule-Based sentiment analysis) and ROBERTA utilizes the latest processing neural networks.

Preprocessing for the project covers the pilot phase, where I pay careful attention to the premise. Language models which are based on tokenization representations allowed me to systematize the reviews for subsequent sending and processing where the sentiment analysis algorithms are applied. For the second turn, I used VADER model to analyze the subjective text for sentiment after cleansing the dataset. One of the advanced techniques attributed to VADER, the ability to handle informal language well together with emoticons, included assigning each word with a sentiment score and hence categorizing reviews as positive, negative, and neutral respectively.

On the other hand, sentiment analysis proved valuable as profound learning enabled me to tweak the RoBERTa model demonstrating the results on the labelled data. While RoBERTa keeps the regular BERT architecture intact, the way it enhances the network parameters to capture the intricate semantic linkages and contextual issues within text makes it more sophisticated compared to other methods that undertake a similar task. By exploiting extensive extended pretraining and fine tuning, the ROBERTA was able to adapt and optimize its parameters to obtain a good quality level in sentiment analysis.

This stage of my project project entailed a detailed analysis of the results obtained from the performance of VADER and RoBERTa. The evaluation of a classification model was done by introducing the metrics like accuracy, precision, recall, and F1-score. Then, I contrasted their performance with each other. The results highlighted better accuracy of RoBERTa with the precision rate of 87.5% whereas in VADER performance it was 75%. Furthermore, it was seen that RoBERTa performed better in grasping the sentimental undertones and the concrete understanding of the language context.

Likewise, I went beyond the superficial assessment of the outputs of both models and searched through the instrumental sentiment indicators in the dataset. By conducting a comparative analysis, I was then able to illuminate the strengths and shortcomings of VADER and RoBERTa, pointing out to how well, in comparison, they work relative to the aspect of correctly capturing the emotion present in the reviews.

The intent of placing one method against one other allowed for gaining this much important information regarding the way sentiment analysis is carried out. As being specific terms, I mentioned the in-depth of rule-based and deep learning techniques that can be used in terms of identifying customers sentiments on e-commerce. Lost in the translating of the business framework from data to a solid decision improved an actionable intelligence of a company. Supported by these insights, businesses can not only make wise decisions but also pay significant attention to customers and contribute to the organizations' success in today's fierce business environment.

CONCLUSION

In summarizing, while VADER and Roberta models comparison on the Fine Foods Amazon Reviews dataset stresses that the sentiment analysis methods were complicated and improved yet. RoBERTa and its deep learning architecture and extensive pretraining was better than the rule-based method of VADAR to reach index of 87,5% compared to VADAR's accuracy rate of approximately of 75%. Therefore, it is evident that the NLP techniques which is largely cleverly designed for the sentiment analysis is actually effective for the sentiment study in details.

Although RoBERTa demonstrated superb skills, in parallel, it is critical to keep in mind that VADER has such a unique asset to contribute as it handles colloquial speech and emoticons, which are widely used when people type reviews in online. It seems to be that ground-up and deep knowledge approaches get contaminated with the application of hybrid solutions that involve a combination of the two methods and the goal is to achieve higher accuracy and effectiveness.

The main highlight of this study, though, is the implications that it has on businesses whose product portfolio is dependent or partially dependent on the e-commerce channel. Through the comprehension and utilization of the feelings articulated in reviews, businesses can make insightful decisions, develop, and improve products, personalize marketing approaches alongside other strategies that ensure winning over customer’s hearts, and maintain higher customer satisfaction levels. Similarly, the provided findings underscore the necessity to always advance the sentiment analysis methods to follow the trends of the computer on-line communication.

Hence, going ahead researchers can effectively address this matter through discovering hybrid approaches which leverages the benefits from both VADER and RoBERTa models. Besides that, it would be interesting to observe how the ways of, cited in the text, manipulations can be applied to other domains and data types as it will provide more knowledge and enrich the emotion analysis methodologies.

All in all, the analysis has led to a thorough understanding of the application of modern Problems in the area of decoding customer sentiments. Through including advanced models like BroBERT and combining them with rule-based methods like VADER, companies can conduct more in-depth analysis on the likes and dislikes of consumers and also determine their shopping behaviour in the process, which in turn would lead to the overall competitiveness of the e-commerce business.