A Project Report

On

PRODUCT REVIEW ANALYSIS FOR GENUINE RATING

Submitted in fulfillment of the

requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

DARBHANGA COLLEGE OF ENGINEERING

B.TECH

Session 2021-2025 in

Computer Science and Engineering

Submitted By

1. Divya Drishti 21105111017
2. Vishakha Bharti
3. Vishal Kumar 21105111031
4. Anshika Kumari 21105111021

Under the guidance of

Mrs. Vandana Kumari

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"**PRODUCT REVIEW ANALYSIS FOR GENUINE RATING**"** in partial fulfillment of the requirements for the award of the B. Tech. (Computer Science and Engineering) submitted in Darbhanga College Of Engineering, Darbhanga is an original work carried out during the period of January, 2025 to February, 2025, under the supervision of Mrs. Vandana Kumari, Department of Computer Science and Engineering, of Darbhanga College Of Engineering, Darbhanga

The matter presented in the thesis/project/dissertation has not been submitted by us for the award of any other degree of this or any other places.

Anshika Kumari (2110111021)
Divya Drishti (21105111017)
Vishakha Bharti (22)
Vishal Kumar (21105111031)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mrs. Vandana Kumari
Assistant Professor

CERTIFICATE

This is to certify that Project Report entitled “PRODUCT REVIEW ANALYSIS FOR GENUINE RATING” which is submitted by Anshika Kumari (2110511021), Divya Drishti (21105111017), Vishakha Bharti (22105111903), Vishal Kumar (21105111031) in partial fulfillment of the requirement for the award of degree B. Tech. in Department of CSE of School of Computing Science and Engineering Department of Computer Science and Engineering

Darbhanga College Of Engineering, Darbhanga is a record of the candidate's own work carried out by him/them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Program Chair **Signature of Dean**

Date: 04 February,2025

Place: Darbhanga

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to the Assistant Professor **Mrs. Vandana Kumari,** Department of Computer Science & Engineering, Darbhanga College of Engineering, Darbhanga for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

ABSTRACT

The web of communication has made expandable data units within the form of text response or self-expression and consumers often rely exclusively on such text, about the use of a particular product. Emotional analysis is used to accomplish the desired function. In this paper we have developed an Improved Emotional Analysis of Product Evaluation System that identifies hidden emotions within user text comments and uses Polarity Scores to rate product reviews as positive, negative or neutral. This sequential process aids in evaluating a product's eligibility for purchase. Product star rating will be evaluated based on the rating of all ratings. The task of the manager is to view the ideas of registered users, adjust the products and classify them based on product, size, color and category (men, women or children). Emotional analysis provides image representation of a product review that is categorized into positive or negative.

Table of Contents

CHAPTER 1

INTRODUCTION

Social media has seamlessly become a basic part of our daily livelihood, providing a channel for communication with friends, family, and individuals across both geographical and digital distances. Beyond mere interaction, it has evolved into a place where people put their opinions, particularly within the expansive landscape of E-commerce. The focus of this paper is to dig into the sentiments expressed by authors or clients in the context of reviews, whether they lean positively or negatively. The act of expressing thoughts and feelings in these reviews collectively forms what is commonly referred to as emotions.

Emotional Analysis, often termed Opinion Mining, stands as a specialized field within Natural Language Processing (NLP), employing frameworks tailored to discern and categorize sentiments embedded in textual content. As the volume of Internet-accessible data continues to surge, a parallel increase is witnessed in articles showcasing customer reviews across diverse online platforms, including research sites, forums, blogs, and social media. Consequently, customers routinely share their experiences, both positive and negative, through the medium of reviews. Prospective buyers heavily lean on the experiences shared by others before making informed purchasing decisions. Recognizing the pivotal role of feedback, businesses actively seek innovative solutions to harness the wealth of reviews available online.

Scientific Analysis Tool, a cutting-edge solution that offers a strategic approach to identify, extract, and interpret a large number of product reviews from diverse online sources, including E-commerce platforms, media outlets, and various review platforms. Whether you are a consumer in search of the best product, a business endeavoring to enhance your offerings, or a researcher delving into the dynamics of business, this tool stands as a valuable resource, providing insightful recommendations based on comprehensive analysis.

1.2 FORMULATION OF PROBLEM

In the era of e-commerce and digital interconnectedness, the surge in online product reviews brings a mix of advantages and challenges. While consumers turn to these reviews for well-informed purchase decisions, businesses leverage them to comprehend customer perspectives and enhance their product offerings. However, the enormity of reviews, spread across diverse platforms and languages, presents a notable hurdle.

The challenge unfolds on two fronts:

1. Data Overload: The daily generation of millions of product reviews overwhelms both individuals and businesses, making it a daunting task to efficiently navigate through this vast sea of unstructured data. Extracting meaningful insights and making informed decisions become arduous.
2. Sentiment Analysis and Trend Identification: The labor-intensive nature of extracting valuable information from reviews, including sentiment analysis, emerging trends, and competitive benchmarks, often falls short of delivering comprehensive and actionable insights.

To address this predicament, we have made a system combining pre trained models to offer the optimal remedy. Pre-trained language models are neural networks designed for various NLP tasks, utilizing a pre-train fine-tuning approach[13]. They are trained on vast text corpora and then fine-tuned for specific downstream tasks[14]. It streamlines the process of collecting, analyzing, and interpreting product reviews from diverse sources. By doing so, it empowers customers to make well-informed choices and enables businesses to promptly enhance their products and services based on customer feedback. This, in turn, elevates the overall customer experience and provides a competitive edge to the brand.

1.2.1 TOOLS USED:

Programming Language: Python 3.7.0

Hardware:

CPU - Intel(R) Core(TM) i5-10210U CPU

RAM - 8.00GB

System type -64-bit operating system, x64-based processor
Edition - Windows 10 Home Single Language

Libraries:

1. Pandas: Serving as a robust data manipulation library in Python, Pandas facilitates the reading and manipulation of datasets in a tabular format. This capability streamlines the analysis and processing of data.

2. Matplotlib and Seaborn: Matplotlib and Seaborn, both visualization libraries in Python, play a crucial role in creating various plots and charts. These visualizations are employed to represent the distribution of reviews, sentiment scores, and other pertinent information in a graphical format.

3. NLTK (Natural Language Toolkit): NLTK, a natural language processing library, offers tools for text processing, including tokenization, part-of-speech tagging, and sentiment analysis. Within this code, the SentimentIntensityAnalyzer from NLTK is utilized for VADER sentiment scoring.

4. Hugging Face Transformers: The Hugging Face Transformers library is employed for working with pre-trained language models. Specifically, the code leverages a pretrained Roberta model for conducting sentiment analysis. This library streamlines the integration of cutting-edge natural language processing models.

5. Scipy: As a scientific computing library in Python, Scipy is harnessed for the softmax function. In this context, it normalizes the output scores obtained from the Roberta model.

6. Tqdm: Tqdm is a library designed for incorporating progress bars into loops. In this code, Tqdm enhances the user experience by visually representing the progress when iterating through the dataset.

CHAPTER-2

2. Literature Survey

Web-based life has become part of everyone's daily life. Used to communicate with friends, family members, people who live far away and next to this, it allows people to express their opinions about things by commenting on the E commercial area. The purpose of this project is to express the feelings of the author or client within the nature of the audit; it will be good or bad for something. These thoughts or feelings that are expressed by individuals are called emotions. Emotional Analysis also referred to as Opinion Mines is an in-house field of Natural Language Processing (NLP) that integrates frameworks that attempt to differentiate and differentiate exploration within content.

Some famous works and studies on product analysis are:

1. Amazon Customer Reviews (ACR) Project: Amazon has dedicated extensive efforts to enhance customer reviews to help customers to learn more about the product and decide whether it is the right product for them[5]. The company is actively working on developing algorithms and models geared towards identifying and mitigating fake reviews, ultimately aiming to elevate the overall quality of reviews on their platforms.

2. Yelp's Review Analytics Research: Yelp is a local business directory and forum to review products, services, or places[1]. Yelp's review data is used to determine user's sentiment or opinion about products, services, or places. Sentiments or opinions are classified into positive reviews, or negative reviews. Leveraging machine learning models, Yelp strives to filter out fake reviews, and their commitment to relevant research is evident in their ongoing initiatives.

3. Stanford Sentiment Analysis Dataset (SST): Stanford University's Sentiment Analysis Project has curated the SST Dataset, a compilation of movie reviews enriched with emotional subtleties. This dataset serves as a valuable resource for researchers involved in developing and assessing models focused on emotional intelligence.

4. SemEval - Sentiment Analysis on Twitter (SentiTUT): Within the SemEval competition, the "SentiTUT" project stands out as it seeks to provide global insights into sentiment analysis. By determining the sentiment of product reviews and tweets, this initiative contributes to the broader understanding of sentiment dynamics.

5. Fakespot Research: Fakespot, a dedicated company in the realm of fake review detection on e-commerce platforms, significantly contributes to the field by investigating the prevalence of fake reviews and the methodologies employed to detect them.

6. Review Meta: Review Meta is a project employing data analysis and machine learning techniques to scrutinize and filter product reviews on e-commerce platforms like Amazon. Their exploration includes an in-depth investigation into the impact of fake reviews on consumer purchasing decisions.

7. UCLA Fake Review Detection Dataset: The University of California, Los Angeles (UCLA) has released the Fake Review Detection Dataset, offering researchers a valuable dataset for developing and evaluating methods to identify fake reviews.

8. Image-Based Sentiment Analysis Research: Image sentiment concepts are ANPs i.e. Adjective Noun Pairs automatically discovered tags of web images which are useful for detecting the emotions or sentiments conveyed by the image[4]. A significant focus in recent studies revolves around aspect-based sentiment analysis, aiming to uncover sentiments related to specific aspects or product features. These studies often entail the development of content-specific annotations and machine learning models.

9. Internal Research on E-Commerce Platforms: Leading companies such as eBay, Walmart, and Alibaba invest in research to enhance product reviews. These companies leverage advanced algorithms to filter and analyze comments, ultimately improving user experience and trust.

10. Academic Research Articles: A plethora of research articles and studies have been published, covering various aspects of product reviews, including opinions, content analysis, factual findings, and the influence of reviews on purchasing decisions. Researchers from global universities and institutions actively contribute to expanding knowledge in this field.

11. These diverse activities and research endeavors collectively drive continuous improvement in product analysis, empowering businesses and consumers to make well-informed decisions within the system.

2.1 EXISTING SYSTEMS AND THEIR LIMITATIONS

1. Sentiment Analysis Techniques:

Exploration of sentiment analysis techniques has been a focal point in research, spanning from traditional machine learning approaches to advanced deep learning methods. Algorithms such as Support Vector Machines (SVMs), Naive Bayes, and Recurrent Neural Networks (RNNs) have undergone thorough investigation for their effectiveness.

Disadvantages in these techniques arise from challenges in handling context-sensitive sentiments, sarcasm, and nuanced language. Traditional methods often grapple with the intricacies of natural language, while deep learning approaches demand large labeled datasets for optimal performance.

2. Sentiment Analysis Based on Aspects:

Advancements in sentiment analysis have led to a focus on specific aspects of products beyond overall sentiment. Techniques involving facet extraction and sentiment analysis for each facet have been explored.

Challenges persist in accurately identifying relevant aspects and managing the intricacies of reviews discussing multiple facets. The absence of labeled datasets for specific domains and products further complicates the task.

3. Entity Recognition and Feature Extraction:

Researchers have delved into methods for identifying entities (products, brands) in reviews and extracting features or attributes discussed by users.

Challenges include the need for robust named entity recognition (NER) models and the complexity of distinguishing between features and feelings. The excessive usage of abbreviations with ambiguous meanings has driven some authors to focus explicitly on abbreviation resolving due to its importance[9].

4. Rating Prediction Models:

Studies have focused on predicting product ratings based on text reviews, incorporating sentiment analysis into machine learning models.

Challenges persist in accurately predicting nuanced ratings. Discrepancies between sentiment in the text and numerical ratings, along with variations in user rating scales, pose hurdles.

5. Bias and Fairness in Analysis:

Increasing attention is directed towards addressing bias and fairness in sentiment analysis models to avoid perpetuating stereotypes or unfair ratings.

Challenges include defining and identifying bias, particularly in subjective areas. Limited diverse and representative datasets can result in biased models.

6. Integration of User Feedback and Continuous Learning:

Research explores mechanisms for integrating user feedback to enhance model performance over time.

Challenges include managing user feedback at scale and ensuring integration avoids introducing biased feedback loops or compromising system performance.

7. Ethical Considerations:

Ethical dimensions in sentiment analysis, including privacy, consent, and responsible artificial intelligence, have gained prominence.

Challenges persist in balancing transparency with privacy concerns and developing universally applicable ethical guidelines, areas where ongoing research is crucial.

CHAPTER 3

SYSTEM DESIGN AND METHODOLOGY

3.1 PROPOSED SYSTEM

As most people need a review about the product beforehand spending their money on the product. So people get various reviews on the website but these updates are true either fraud is not detected by the user. They offer great reviews of many different products produced by their company. Users will not be able to find out if the review is true or fake. The system will receive false reviews. Users will watch various products and will update with the product. And the user will receive real updates about the product.

Datasets:

In our proposed system, data update sets for such products taken directly online from the amazon websitewww.amazon.com using the import.io tool After feeding I An online data set should download our favorite .csv file product, We can collect multiple user reviews of product and now these updates can be viewed at the discretion of the product.With the help of online analytics data sets, user ideas are like that collection and we can get user feedback products. After applying the data processing process and letting the words be deleted .Now the list of words will now be viewed to compare this list of words with a word bag containing Good ideas and Bad ideas. After this vision the mines we can find high quality pinioned products. Those products can be viewed in the browser.

VADER: It stands for Valence Aware Dictionary and Sentiment Reasoner, is a sentiment analysis tool designed to evaluate the emotional tone within a given text. Developed as a part of the Natural Language Toolkit (NLTK) in Python, VADER specializes in analyzing sentiments in social media text, informal language, and short sentences.

Operated on a rule-based framework, VADER utilizes a predefined lexicon containing words with assigned sentiment scores. Each word in this lexicon is associated with a polarity (either positive or negative) and an intensity score, indicating the strength of the sentiment. VADER goes beyond merely considering individual word sentiments; it also incorporates rules to handle sentiment modifiers, negations, and conjunctions.

VADER produces four distinct sentiment scores for a given text: positive, negative, neutral, and a compound score. The compound score reflects the overall sentiment, amalgamating individual scores while considering their respective intensities. The score scale ranges from -1 to 1, where -1 signifies highly negative sentiment, 0 indicates neutrality, and 1 represents highly positive sentiment. The key strength of VADER lies in its efficacy for sentiment analysis in short and informal texts, making it particularly valuable for processing social media content and customer reviews. Its rule-based nature facilitates quick and efficient sentiment assessment without the requirement for extensive training data. However, it's essential to note that, like any tool, VADER has its limitations, and its performance may vary depending on the type of text and context.

RoBERTa: It is an optimized variant of BERT (Bidirectional Encoder Representations from Transformers), is available in Hugging Face's Transformers library, representing a substantial leap in natural language processing (NLP) models. In the Hugging Face library, RoBERTa can be accessed through the transformers module, providing pre-trained models and tokenizers for straightforward integration into various NLP tasks.

Built on the transformer architecture, RoBERTa utilizes self-attention mechanisms to process input sequences bidirectionally, excelling at capturing nuanced contextual relationships in text. The Hugging Face's Transformers library simplifies the utilization of RoBERTa by offering a user-friendly interface to load pre-trained models, tokenize text data, and fine-tune the model for specific NLP applications such as sentiment analysis, text classification, or named entity recognition. The library's API streamlines the implementation of RoBERTa, enabling researchers, developers, and data scientists to leverage cutting-edge NLP capabilities without extensive coding. Hugging Face's implementation of RoBERTa encapsulates the power of state-of-the-art language models, making them accessible to a broader audience and facilitating efficient integration into various projects.

3.2 MODEL ARCHITECTURE:

I. Load and Preprocess Data:

The data preprocessing step is crucial in ensuring the accuracy of the sentiment analysis. The preprocessing step involves cleaning the raw text data by removing irrelevant information, such as stop words, punctuations, URLs, and hashtags[12]. Start by reading the dataset into a Pandas DataFrame, to manage processing, limit the dataset to the first 500 rows.. The code utilizes the Pandas library to read a CSV dataset, providing a structured format for further analysis. The decision to limit the dataset to 500 rows ensures computational efficiency and manageable exploration.

II. Exploratory Data Analysis (EDA):

Conduct Exploratory Data Analysis (EDA) to gain insights into the dataset's composition. Visualize the distribution of reviews across various star ratings using a bar plot.

Matplotlib and Seaborn are employed to create a bar plot, offering a graphical representation of the dataset's star rating distribution. This initial exploration sets the stage for a more in-depth understanding.

III. VADER Sentiment Scoring:

Implement sentiment analysis using the VADER (Valence Aware Dictionary and Sentiment Reasoner) tool from NLTK. This rule-based approach assigns sentiment scores (negative, neutral, positive, and compound) to each review.

The NLTK library's `SentimentIntensityAnalyzer` is applied to tokenize, tag parts of speech, and score sentiments. The resulting scores are integrated into the original dataset, enriching it with sentiment information.

IV. Visualize VADER Results:

Create visualizations to represent the sentiment distribution across different star ratings. This step aids in understanding the prevalence of sentiments within each rating category.

Matplotlib and Seaborn functions generate bar plots and pair plots, providing a comprehensive view of sentiment distributions. These visuals enhance interpretability and highlight potential patterns.

V. Roberta Pretrained Model:

Introduce a more advanced sentiment analysis approach using a pre-trained Roberta model from Hugging Face's Transformers library. Tokenize, process through the model, and extract sentiment scores.

Leveraging state-of-the-art language models, the code incorporates the Roberta model to obtain nuanced sentiment scores. This step showcases the integration of sophisticated machine learning techniques.

VI. Combine and Compare Results:

Unify the sentiment scores from both VADER and the Roberta model into a single DataFrame. Generate pair plots for visual comparison, shedding light on the strengths and weaknesses of each approach.

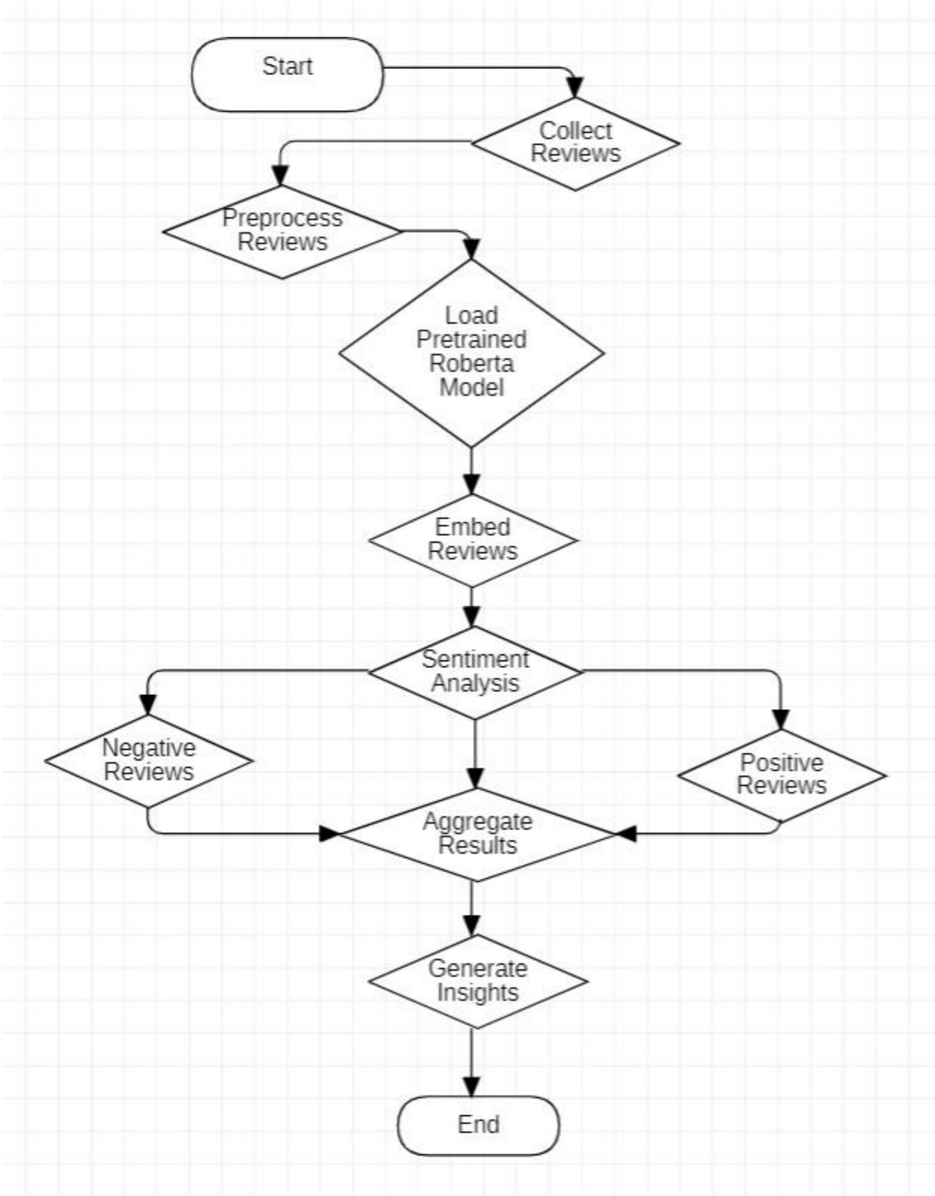
Pandas' `merge` function is employed to combine VADER and Roberta scores, enabling a side-by-side analysis. Pair plots offer a visual exploration of how the two models align or differ in sentiment predictions.

VII. Review Examples:

Move beyond numerical analysis by presenting specific examples where VADER and Roberta sentiment scores differ significantly. These examples serve as case studies, providing qualitative insights into potential divergences in sentiment predictions.

The code identifies and showcases instances where rule-based and machine learning models produce distinct sentiment predictions, adding a qualitative layer to the analysis.

3.3 SYSTEM DESIGN



- Collect Reviews: Gather product feedback from diverse sources, including e-commerce platforms, social media, and other relevant channels.

- Preprocess Reviews: Clean and refine the text data by performing tasks such as removing irrelevant characters, tokenization, and eliminating stopwords.

- Load Pretrained Roberta Model: Incorporate the pretrained Roberta model, previously trained on an extensive corpus of text data.

- Embed Reviews: Utilize the Roberta model to convert preprocessed text reviews into numerical vectors, a process known as embedding.

- Sentiment Analysis: Apply sentiment analysis using the loaded Roberta model to predict whether each review conveys a positive or negative sentiment.

- Post-process Results: Categorize sentiment predictions into distinct classes, such as positive or negative.

- Aggregate Results: Compute overall sentiment scores by considering individual predictions, potentially generating metrics like average sentiment.

- Generate Insights: Analyze the aggregated results to discern trends, key points, and valuable insights from the pool of product reviews.

- End: Conclude the process.

CHAPTER 4

IMPLEMENTATION AND RESULTS

Code on Jupyter Notebook:

jupyter sentiment-analysis-python-youtube-tutorial Last Checkpoint: 15 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run | Markdown

Step 1. Read in Data and NLTK Basics

In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('ggplot')

import nltk
```

In [7]:
```
# Read in data
df = pd.read_csv('Reviews.csv')
print(df.shape)
df = df.head(500)
print(df.shape)
```

```
(568454, 10)
(500, 10)
```

In [8]:
```
df.head()
```

Out[8]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M.	0	0	5	1350777600	Great taffy	Great taffy at a great price. There

Quick EDA

In [9]:
```
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()
```

Basic NLTK

In [10]:
```
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.
```

In [11]:
```
tokens = nltk.word_tokenize(example)
tokens[:10]
```

Out[11]:
```
['This', 'oatmeal', 'is', 'not', 'good', '.', 'Its', 'mushy', ',', 'soft']
```

In [12]:
```
tagged = nltk.pos_tag(tokens)
tagged[:10]
```

Out[12]:
```
[('This', 'DT'),
 ('oatmeal', 'NN'),
 ('is', 'VBZ'),
 ('not', 'RB'),
 ('good', 'JJ'),
 ('.', '.'),
 ('Its', 'PRP$'),
 ('mushy', 'NN'),
 (',', ','),
 ('soft', 'JJ')]
```

In [13]:
```
entities = nltk.chunk.ne_chunk(tagged)
entities.pprint()
```

```
(S
  This/DT
  oatmeal/NN
  is/VBZ
  not/RB
  good/JJ
  ./.
  Its/PRP$
  mushy/NN
  ,/,
  soft/JJ
```

Step 2. VADER Seniment Scoring

In [14]:
```
import nltk
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to C:\Users\Kumari
[nltk_data]     Prerna Sinha\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

Out[14]:
```
True
```

In [15]:
```
from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm.notebook import tqdm

sia = SentimentIntensityAnalyzer()
```

In [16]:
```
sia.polarity_scores('I am so happy!')
```

Out[16]:
```
{'neg': 0.0, 'neu': 0.318, 'pos': 0.682, 'compound': 0.6468}
```

In [17]:
```
sia.polarity_scores('This is the worst thing ever.')
```

Out[17]:
```
{'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249}
```

In [18]:
```
sia.polarity_scores(example)
```

Out[18]:
```
{'neg': 0.22, 'neu': 0.78, 'pos': 0.0, 'compound': -0.5448}
```

In [19]:
```
# 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, 1240.47it/s]

In [20]:
```
vaders = pd.DataFrame(res).T
```

```
In [20]: vaders = pd.DataFrame(res).T
         vaders = vaders.reset_index().rename(columns={'index': 'Id'})
         vaders = vaders.merge(df, how='left')
```

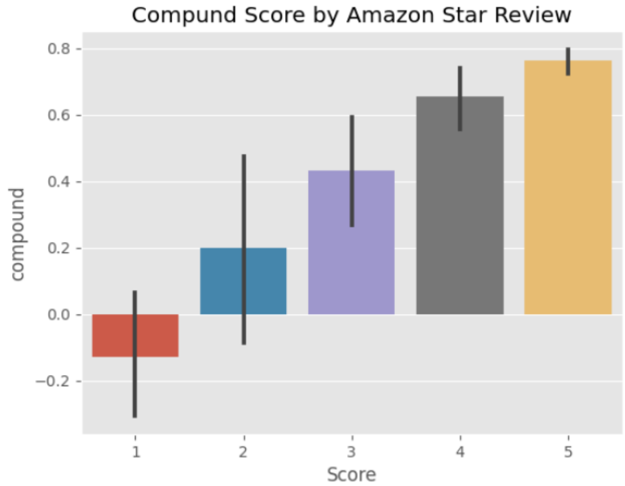
```
In [21]: # Now we have sentiment score and metadata
         vaders.head()
```

Out[21]:

	Id	neg	neu	pos	compound	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Sum
0	1	0.000	0.695	0.305	0.9441	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	C Dog
1	2	0.138	0.862	0.000	-0.5664	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	I Adve
2	3	0.091	0.754	0.155	0.8265	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"D say
3	4	0.000	1.000	0.000	0.0000	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	C Me
4	5	0.000	0.552	0.448	0.9468	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Grea

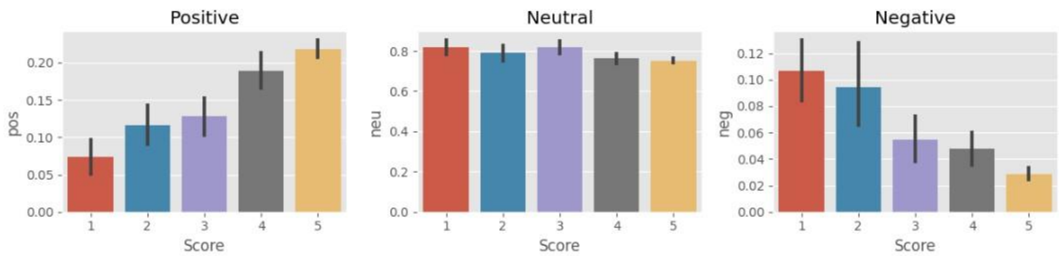
Plot VADER results

In [22]:
```python
ax = sns.barplot(data=vaders, x='Score', y='compound')
ax.set_title('Compund Score by Amazon Star Review')
plt.show()
```



In [23]:
```python
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')
```

```python
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()
```



Step 3. Roberta Pretrained Model

- Use a model trained of a large corpus of data.
- Transformer model accounts for the words but also the context related to other words.

In [24]:
```python
from transformers import AutoTokenizer
from transformers import AutoModelForSequenceClassification
from scipy.special import softmax
```

In [25]:
```python
MODEL = f"cardiffnlp/twitter-roberta-base-sentiment"
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)
```

In [26]:
```python
# VADER results on example
print(example)
sia.polarity_scores(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.
```

Out[26]:
```
{'neg': 0.22, 'neu': 0.78, 'pos': 0.0, 'compound': -0.5448}
```

In [27]:
```
# 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': 0.02068747, 'roberta_pos': 0.0029573706}
```

In [28]:
```
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
```

In [30]:
```
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
```

```
        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 [02:49<00:00, 2.94it/s]
```

```
Broke for id 83
Broke for id 187
```

In [31]:
```
results_df = pd.DataFrame(res).T
results_df = results_df.reset_index().rename(columns={'index': 'Id'})
results_df = results_df.merge(df, how='left')
```

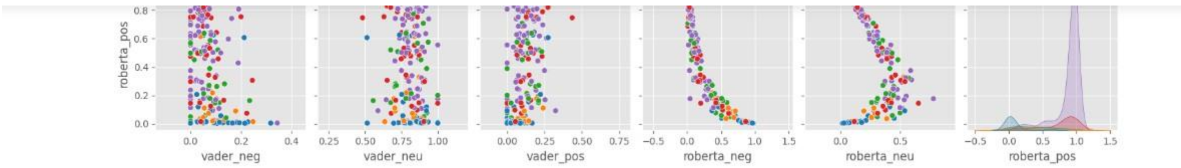
Compare Scores between models

In [32]:
```
results_df.columns
```

Out[32]:
```
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')
```

Step 4. Combine and compare

In [33]:
```
sns.pairplot(data=results_df,
             vars=['vader_neg', 'vader_neu', 'vader_pos',
                   'roberta_neg', 'roberta_neu', 'roberta_pos'],
            hue='Score',
            palette='tab10')
plt.show()
```



Step 5. Review Examples

- Positive 1-Star and Negative 5-Star Reviews

```
In [34]: results_df.query('Score == 1') \
             .sort_values('roberta_pos', ascending=False)['Text'].values[0]

Out[34]: '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.'

In [35]: results_df.query('Score == 1') \
             .sort_values('vader_pos', ascending=False)['Text'].values[0]

Out[35]: 'So we cancelled the order.  It was cancelled without any problem.  That is a positive note...'

In [29]: # nevative sentiment 5-Star view

In [36]: results_df.query('Score == 5') \
             .sort_values('roberta_neg', ascending=False)['Text'].values[0]

Out[36]: 'this was sooooo deliscious but too bad i ate em too fast and gained 2 pds! my fault'

In [37]: results_df.query('Score == 5') \
             .sort_values('vader_neg', ascending=False)['Text'].values[0]

Out[37]: 'this was sooooo deliscious but too bad i ate em too fast and gained 2 pds! my fault'
```

This Python code offers a holistic and sophisticated exploration of sentiment analysis techniques. By seamlessly integrating both rule-based and machine learning approaches, the code provides practitioners and enthusiasts with a versatile tool for understanding sentiment patterns in textual data. The combination of visualizations, pair plots, and detailed review examples enhances the interpretability of sentiment analysis results, positioning the code as an invaluable resource across various domains.

CHAPTER 5

CONCLUSION

Product reviews serve as a dynamic and adaptable tool within the ever-expanding realm of customer feedback and product evaluations. In an era dominated by data-centric decision-making, this system showcases the potential to redefine the landscape of product analysis, ushering in benefits for both consumers and businesses.

This study marks a progressive exploration into the realm of deceptive reviews. While pinpointing individual fraudulent submissions proved to be a challenging task, discerning patterns within groups of such reviews emerged as a comparatively manageable endeavor. The introduction of a scoring algorithm, coupled with the application of a pre-trained model, facilitates the analysis of databases, identifying clusters of potential false reviews. The study also proposes specific behaviors and features integral to the identification of deceptive reviews. Through rigorous analysis and testing on real-world databases, our findings yield crucial insights into the intricate domain of false reviews.

In summary, Product Analysis Analysis emerges as a transformative force in the landscape of data-driven decision-making. It envisions a future where analysis not only imparts knowledge but also serves as a catalyst for product refinement and heightened consumer awareness, fostering more informed choices. This system holds the promise of reshaping our understanding and utilization of product reviews, positioning them as indispensable assets for businesses, consumers, and researchers in an era defined by robust connectivity and business dynamics.

5.1 BENEFITS AND FUTURE DIRECTIONS

I. **In-Depth Understanding:**

   - The system offers a thorough understanding of customer sentiments by integrating both rule-based (VADER) and machine learning (Roberta) approaches. This dual-method strategy ensures a nuanced comprehension of sentiment nuances.

II. **Comparative Model Analysis:**

- By amalgamating sentiment scores from VADER and Roberta models, the code allows for a comparative analysis. This feature enables users to observe and analyze differences between rule-based and machine learning sentiment predictions.

III. **Visualized Data Insights:**

- The code employs visual elements like bar plots and pair plots, enhancing the interpretability of sentiment distributions across diverse star ratings. Visual representations contribute to a clearer understanding of the analyzed data.

IV. **Efficient Data Processing:**

- Designed for scalability, the code efficiently handles large datasets. It includes measures to limit data for manageable processing, ensuring accurate sentiment analysis without compromising efficiency.

V. **Utilization of Advanced Models:**

- Leveraging the Hugging Face's Transformers library, the code harnesses the power of a pretrained Roberta model. This choice ensures advanced language comprehension and context-aware sentiment predictions.

VI. **Illustrative Examples:**

- The code features specific examples where sentiment scores differ between VADER and Roberta models. These real-world instances serve as instructive case studies for interpreting the models' sentiment predictions.

VII. **Ethical Awareness:**

- Ethical considerations are acknowledged in the code, emphasizing the importance of fairness and bias in sentiment analysis. It actively contributes to ongoing research in responsible artificial intelligence and ethical sentiment analysis.

VIII. **Integration of External Models:**

- The code seamlessly integrates external sentiment analysis models through Hugging Face's library, demonstrating an openness to adopting cutting-edge models and staying abreast of advancements in the sentiment analysis field.

IX. Educational Value:

- With step-by-step explanations and comments, the code serves as an educational tool for those interested in sentiment analysis. It provides a practical demonstration of implementing sentiment analysis techniques in Python.

X. Holistic Sentiment Analysis Approach:

- By combining rule-based and machine learning models, the code embraces a holistic approach to sentiment analysis. This ensures a more nuanced and accurate interpretation of sentiments present in textual data.

REFERENCES

1. H. Faisol, K. Djajadinata and M. Muljono, "Sentiment Analysis of Yelp Review," 2020 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 2020, pp. 179-184, doi: 10.1109/iSemantic50169.2020.9234213.

2. Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

3. https://paperswithcode.com/datasets?task=sentiment-analysis

4. N. Mittal, D. Sharma and M. L. Joshi, "Image Sentiment Analysis Using Deep Learning," 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Santiago, Chile, 2018, pp. 684-687, doi: 10.1109/WI.2018.00-11. keywords: {Sentiment analysis;Deep learning;Biological neural networks;Feature extraction;Visualization;Analytical models;image sentiment analysis, deep learning, deep neural network, convolutional neural network, Region based CNN and Fast R-CNN.},

5. https://www.amazon.com/Good-Great-Some-Companies-Others/product-reviews/0066620996

6. Parul Pandey(2018), 'Simplifying Sentiment Analysis using VADER in Python(on Social Media Text)', *https://medium.com/*, Published in Analytics Vidhya: Publisher

7. https://www.geeksforgeeks.org/overview-of-roberta-model/

8. Appasani, Bhargav(2023), 'Applied Computational Intelligence and Soft Computing', *Horsa, Obsa Gelchu Tune*, *Kula Kekeba*, doi: 10.1155 /2023 /3462691, 1687-9724.

9. Perera N, Dehmer M, Emmert-Streib F. Named Entity Recognition and Relation Detection for Biomedical Information Extraction. Front Cell Dev Biol. 2020 Aug 28;8:673. doi: 10.3389/fcell.2020.00673. PMID: 32984300; PMCID: PMC7485218.

10. Khan, M. A., Khan, M. A., Ahmed, F., Mittal, M., Goyal, L. M., Hemanth, D. J., & Satapathy, S. C. (2020). Gastrointestinal diseases segmentation and classification based on duo-deep architectures. Pattern Recognition Letters, 131, 193-204.

11. Barik, K., Misra, S. Analysis of customer reviews with an improved VADER lexicon classifier. *J Big Data* 11, 10 (2024). https://doi.org/10.1186/s40537-023-00861-x

12. Tan, Kian Long, Chin Poo Lee, and Kian Ming Lim. (2023), "RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis" *Applied Sciences* 13, no. 6: 3915. https://doi.org/10.3390/app13063915

13. Elazar Y, Kassner N, Ravfogel S, Ravichander A, Hovy E, Schütze H, Goldberg Y (2021) Erratum: measuring and improving consistency in pretrained language models. Trans Assoc Comput Linguist 9:1407. https://doi.org/10.1162/tacl_x_00455

14. Soliman, A., Shaheen, S. & Hadhoud, M. Leveraging pre-trained language models for code generation. *Complex Intell. Syst.* (2024). https://doi.org/10.1007/s40747-024-01373-8