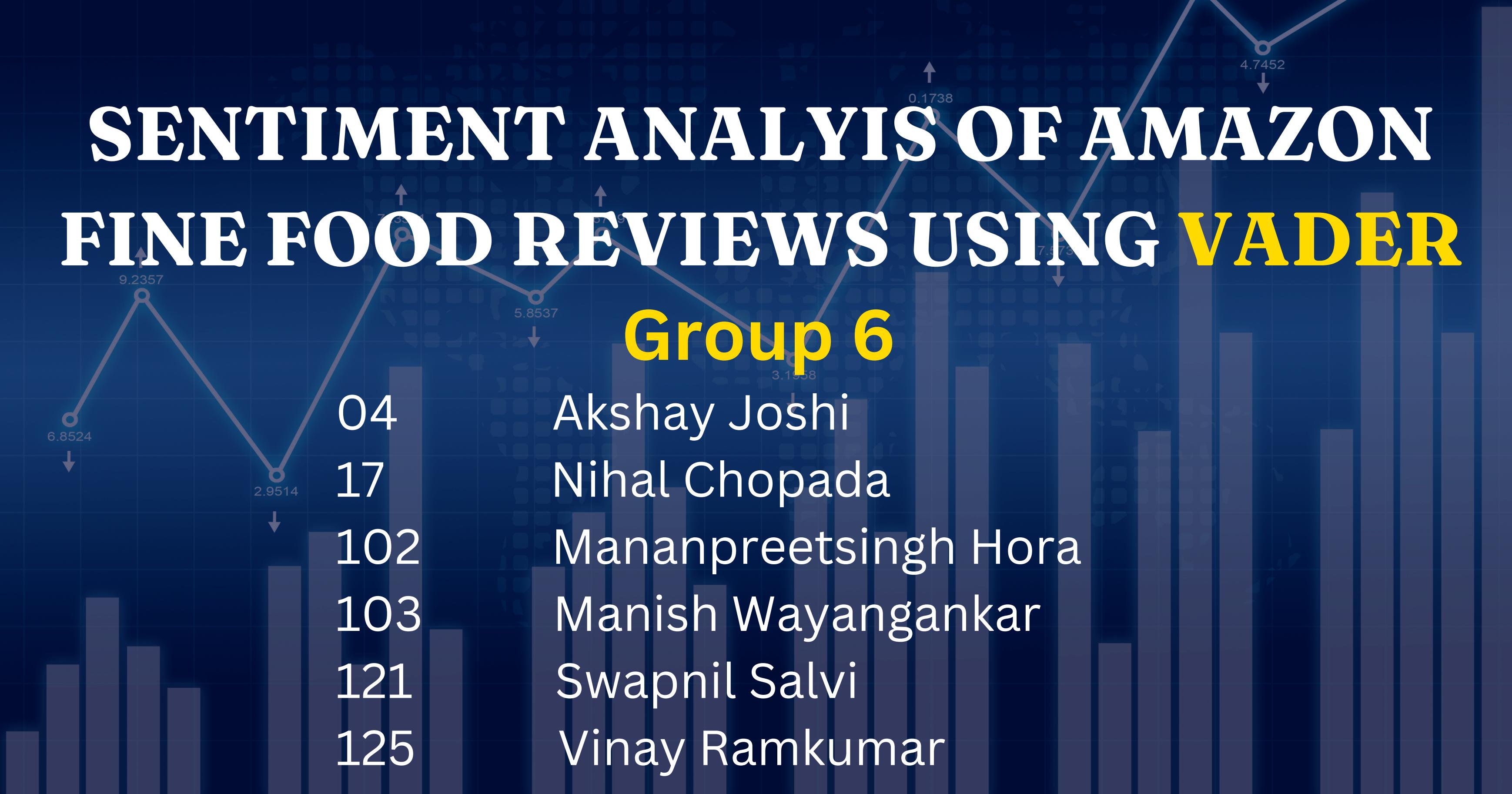


TEXT ANALYTICS

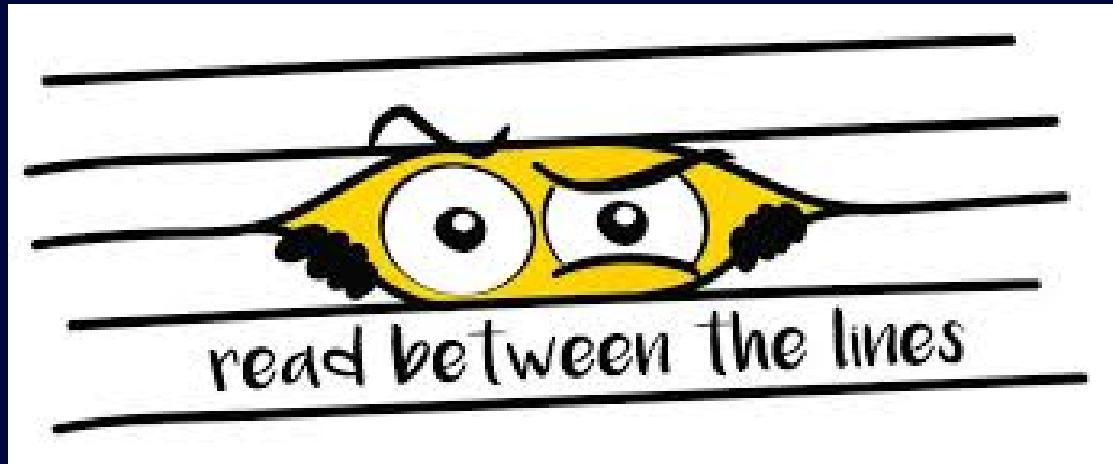
SENTIMENT ANALYSIS OF AMAZON FINE FOOD REVIEWS USING VADER

Group 6

Akshay Joshi
Nihal Chopada
Mananpreetsingh Hora
Manish Wayangankar
Swapnil Salvi
Vinay Ramkumar



WHAT IS SENTIMENT ANALYSIS ???



Secret sauce behind understanding how people really feel about your brand, product, or campaign.



TOOLS OF THE TRADE:

- VADER: Quick social media sentiment checks
- Transformers like BERT: Next-gen AI that reads text with near-human understanding
- SaaS Powerhouses: Google NLP and Azure Text Analytics for scalable, enterprise-grade sentiment analysis

WHY IT MATTERS:

- Real Time Audience Pulse
- Tailored Marketing
- Sharper Product Development



WHAT IS VADER ??

VALENCE AWARE DICTIONARY FOR SENTIMENT REASONING

VADER is a lexicon and rule-based sentiment analysis tool designed to handle sentiment in text, particularly suited for social media text due to its emphasis on the sentiment intensity of informal language.

WHY VADER ??

LEXICON
AND RULE-
BASED
APPROACH

HANDLING
OF
INFORMAL
LANGUAGE

NO
TRAINING
REQUIRED

HANDLES
INTENSIVE
MODIFIERS

SPEED



COMPARISON WITH OTHERS

| Approach | Data Requirement | Performance | Speed | Computational Cost | Use Case |
|-------------------------------------|-------------------------|---|-----------------------|-----------------------------------|---|
| VADER | Small Minimal Labelling | Good for social media, lacks deep context | Very Fast (Real Time) | Low(CPU) | Social media, quick insights |
| Machine Learning (SVM, Naive Bayes) | Large Labelled Datasets | High accuracy but less context-aware | Moderate | Moderate (Requires Training Time) | Traditional text classification |
| Deep Learning (LSTM, BERT) | Large Labelled Datasets | Best for nuanced, context-rich sentiment | Slow (requires GPU) | High (GPU Needed) | Complex language understanding, reviews |

DATA PREPARATION: IMPORTING KEY LIBRARIES AND LOADING THE DATASET

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import nltk  
  
plt.style.use('ggplot')
```

- pandas: A tool that helps in handling and analyzing data.
- numpy: Helps with numerical calculations.
- matplotlib and seaborn: Tools for making graphs and visualizing data.
- nltk: A tool for working with text data, especially useful for analyzing text like customer reviews.
- plt.style.use('ggplot'): This changes the style of the graphs to look nice and clean. It's just a visual choice.

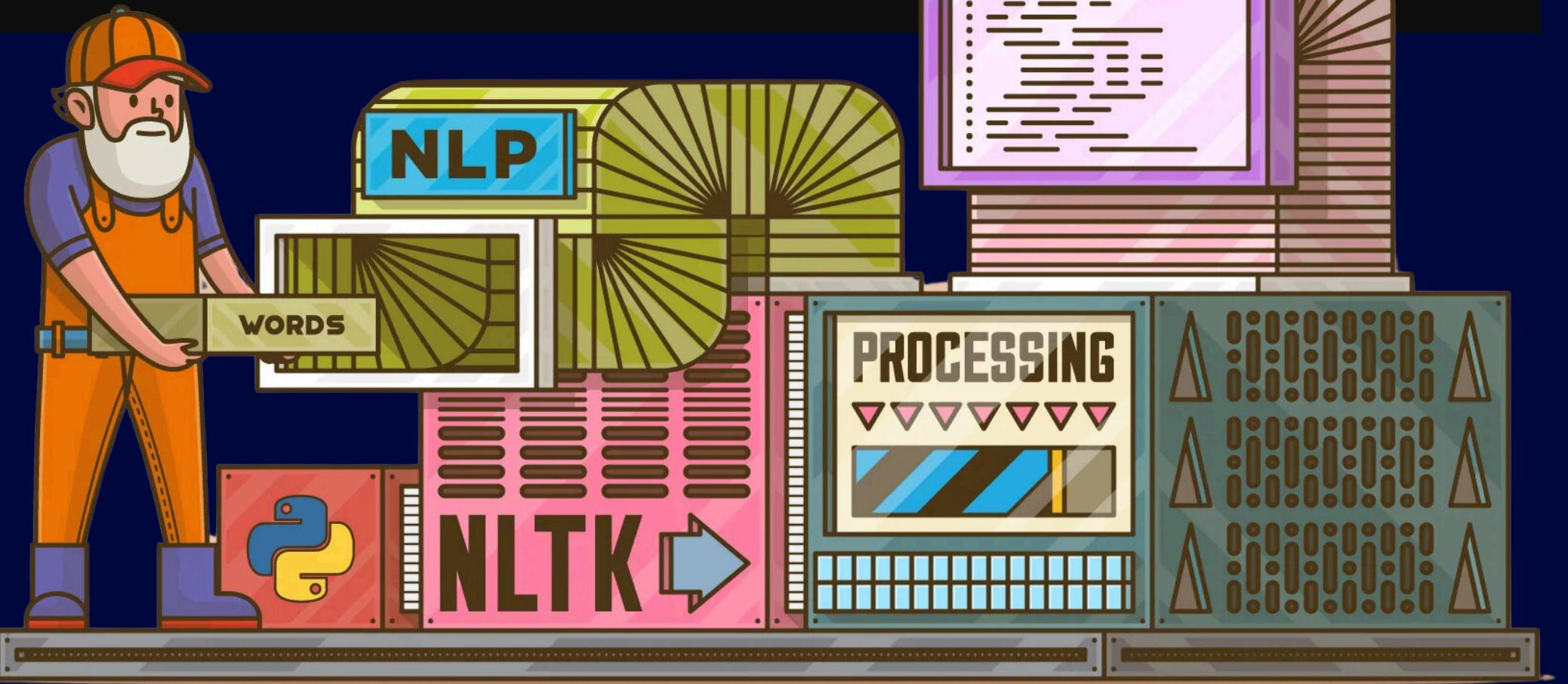
LOADING THE DATA

```
df = pd.read_csv('/Reviews.csv')  
print(df.shape)  
df = df.head(500)  
print(df.shape)  
df.head()
```

- df.shape: Shows number of rows and columns in the data. It's like checking the size of the table.
- df.head(500): First 500 rows (reviews) of the data.
- df.head(): This shows the first few rows of the table, allowing you to check how the data looks.

Download NLTK resources

```
nltk.download('averaged_perceptron_tagger')  
nltk.download('words')  
nltk.download('maxent_ne_chunker')  
nltk.download('punkt')
```



1 **averaged_perception_tagger**
Helps to identify the parts of speech in a sentence (e.g., noun, verb, adjective).

2 **words:**
Contains a list of words that the library uses for text processing.

3 **maxent_ne_chunker:**
Helps in identifying named entities (e.g., names of people, organizations) in a sentence.

4 **punkt:**
Helps in splitting the text into sentences and wo

VISUALIZING REVIEW COUNTS

```
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()
```

- **df['Score'].value_counts():**
Counts how many reviews have each star rating (1 to 5 stars).
- **plot(kind='bar'):** Bar chart showing how many reviews are rated 1 star, 2 stars, etc.
- **ax.set_xlabel('Review Stars'):**
Labels the x-axis as "Review Stars."
- **plt.show():** Displays the graph.



TOKENIZING TEXT

```
example = df['Text'][50]
tokens = nltk.word_tokenize(example)
tokens[:10]
```

- df['Text'][50]: **Review text** from the 50th review.
- nltk.word_tokenize: **Breaks the review** into individual words (called tokens).
- tokens[:10]: Shows the first 10 words of the review

POS TAGGING

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

Output ==>

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

NAMED ENTITY RECOGNITION (NER)

```
entities = nltk.chunk.ne_chunk(tagged)
entities.pprint()
```

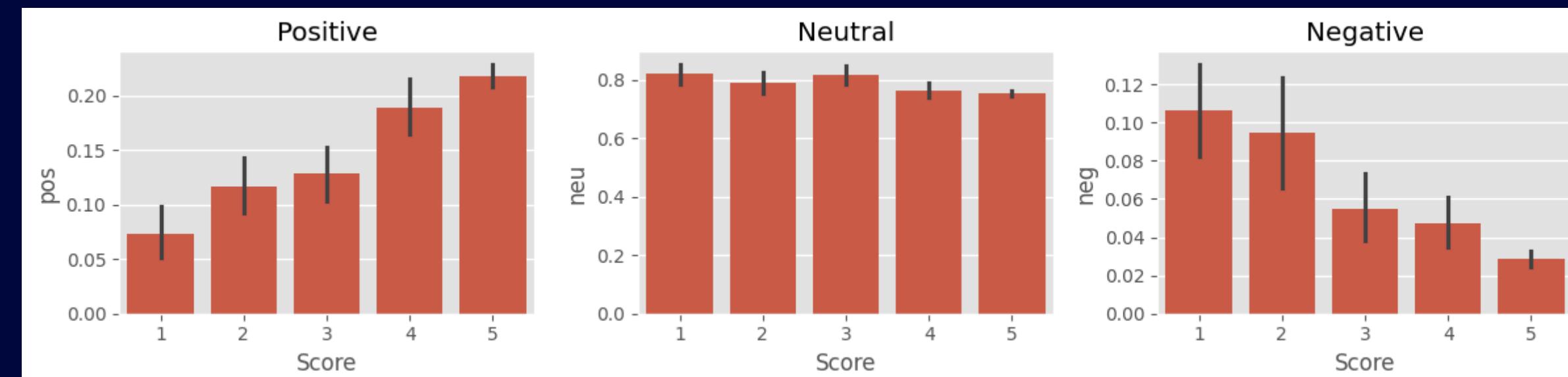
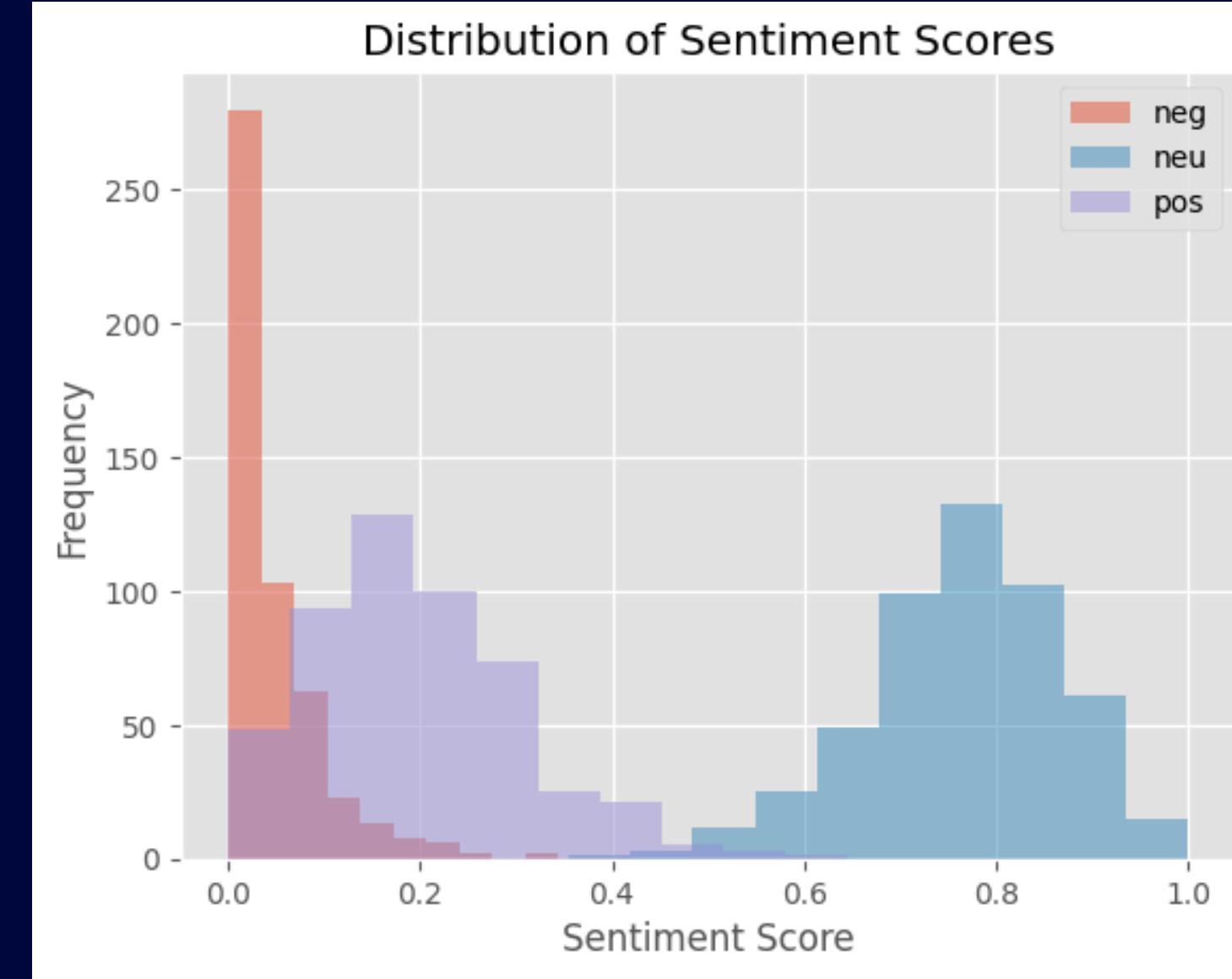
Named Entity Recognition (NER) helps in **identifying specific important entities in a review**. For example, a review might mention a product or a company, and this information can be useful for deeper analysis.

SENTIMENT SCORING USING VADER

VADER is specifically designed for analyzing the sentiment of social media and product reviews. It gives scores for positive, negative, and neutral sentiments, as well as an overall compound score, which summarizes the sentiment.

```
nltk.download('vader_lexicon')
from nltk.sentiment import SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()
sia.polarity_scores('I am so happy!')
sia.polarity_scores(example)
```

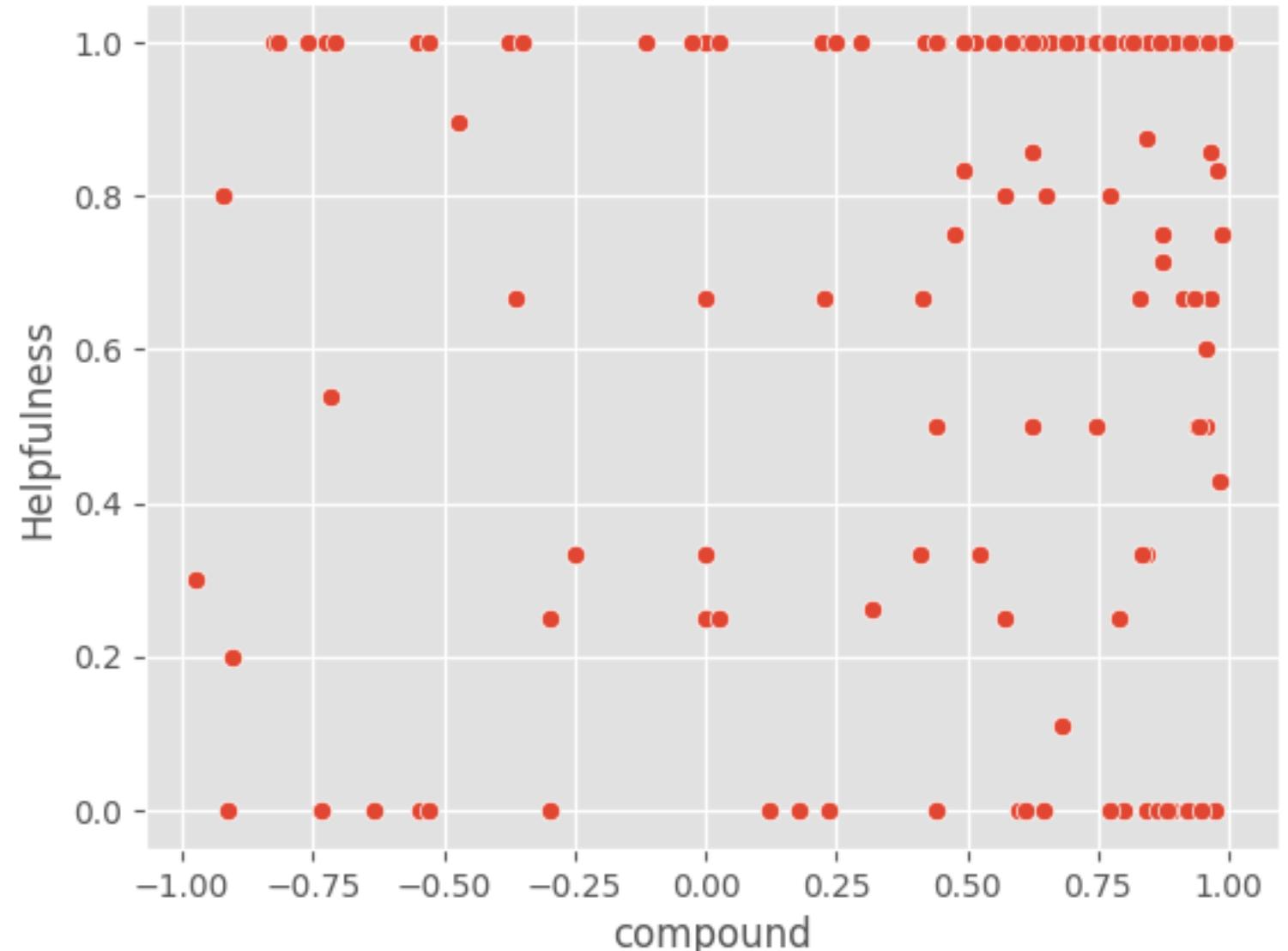
DISTRIBUTION OF SENTIMENT SCORES



SCATTER PLOT:

```
import matplotlib.pyplot as plt
# 1. Analyze the relationship between sentiment scores and review helpfulness.
# We can check if more helpful reviews tend to have higher positive sentiment scores or lower negative sentiment scores.
vaders['Helpfulness'] = vaders['HelpfulnessNumerator'] / vaders['HelpfulnessDenominator']
sns.scatterplot(data=vaders, x='compound', y='Helpfulness')
plt.title('Relationship between Compound Sentiment and Review Helpfulness')
plt.show()
```

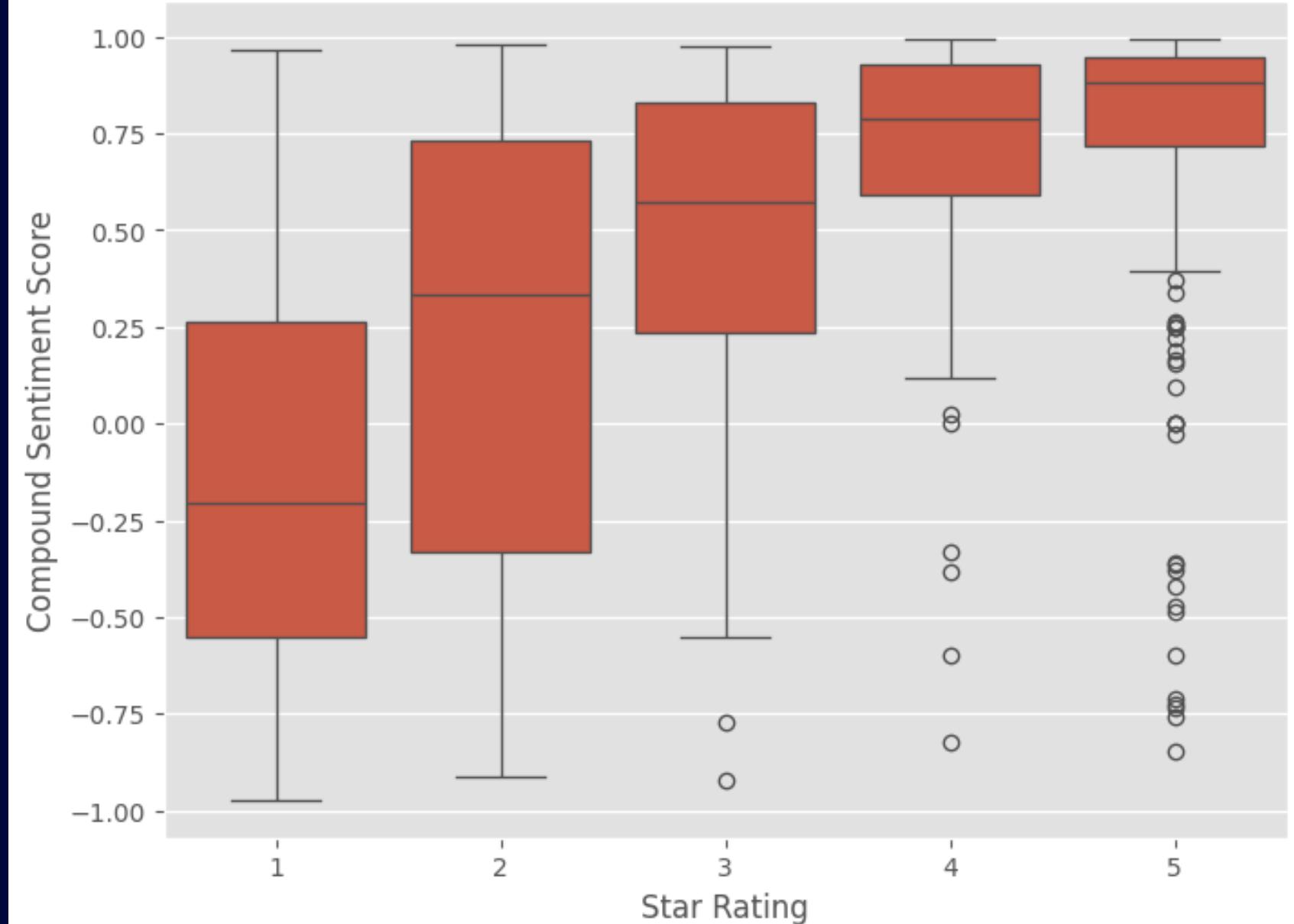
Relationship between Compound Sentiment and Review Helpfulness



BOX PLOT:

```
import matplotlib.pyplot as plt
# Analyze the distribution of sentiment scores across different star ratings.
# This can help identify if certain star ratings are associated with specific sentiment distributions.
plt.figure(figsize=(8, 6))
sns.boxplot(x='Score', y='compound', data=vaders)
plt.title('Distribution of Compound Sentiment Scores by Star Rating')
plt.xlabel('Star Rating')
plt.ylabel('Compound Sentiment Score')
plt.show()
```

Distribution of Compound Sentiment Scores by Star Rating



CONCLUSION & FINDINGS

Sentiment Distribution:

- Reviews vary widely in sentiment, spanning positive, neutral, and negative tones.
- Positive reviews (4 or 5 stars) typically show strong satisfaction with high compound and positive scores.
- Negative reviews (1 or 2 stars) often express dissatisfaction, with low compound and higher negative scores.
- Neutral sentiments, notably in 3-star reviews, indicate mixed feelings or indifference.

Product Perception:

- Positive sentiment correlates with comments on product quality and satisfaction (e.g., freshness, taste).
- Negative reviews commonly cite issues like delivery problems, packaging concerns, or misleading descriptions.

Overall Insights:

- Reviews skew positively, especially with higher star ratings (4 or 5 stars).
- Helpful reviews tend to feature extreme sentiments, while neutral reviews are perceived as less helpful.
- Areas for improvement, highlighted in negative reviews, focus on logistics and packaging



THANK YOU