Name: Dream Patel
Roll No.: 33
Experiment No. 5
Develop Content (text, emoticons, image, audio, video) based social media analytics model for business.
Date of Performance: 24/02/2024
Date of Submission: 28/02/2024

Experiment No. 05

Aim: Develop Content (text, emoticons, image, audio, video) based social media analytics model for business.

Objective: Implement sentiment analysis techniques to evaluate customer opinions and feedback regarding a product's satisfaction and market demand. Utilize Natural Language Processing algorithms to classify reviews and NPS survey responses, distinguishing between positive, negative, and neutral sentiments, as well as identifying underlying emotions

Software used: Kaggle Notebook.

Theory:
Sentiment analysis is the process of classifying whether a block of text is positive, negative, or neutral. The goal that Sentiment mining tries to gain is to analyze people's opinions in a way that can help businesses expand. It focuses not only on polarity (positive, negative & neutral) but also on emotions (happy, sad, angry, etc.). It uses various Natural Language Processing algorithms such as Rule-based, Automatic, and Hybrid.

let's consider a scenario, if we want to analyze whether a product is satisfying customer requirements, or is there a need for this product in the market. We can use sentiment analysis to monitor that product's reviews. Sentiment analysis is also efficient to use when there is a large set of unstructured data, and we want to classify that data by automatically tagging it. Net Promoter Score (NPS) surveys are used extensively to gain knowledge of how a customer perceives a product or service. Sentiment analysis also gained popularity due to its feature to process large volumes of NPS responses and obtain consistent results quickly.

This sentiment analysis project aims to analyze reviews of food products on Amazon using two techniques: VADER and a Roberta pre-trained model.

1. VADER (Valence Aware Dictionary and sEntiment Reasoner):
VADER is a lexicon-based sentiment analysis tool specifically attuned to sentiments expressed in social media contexts. It comes with pre-built sentiment lexicons that assign polarity scores (positive, negative, or neutral) to individual words.
Strengths:
1. Simple and easy to use.
2. Captures nuances of sentiment beyond basic positive/negative through polarity scores.
3. Accounts for sentiment intensifiers (e.g., "very," "extremely") and negation.

Weaknesses:
1. Relies on pre-defined lexicons, potentially missing domain-specific sentiment (e.g., "bland" for food reviews).
2. Limited ability to understand context and sarcasm.

2. Roberta Pretrained Model with Hugging Face Pipeline:

This approach utilizes a pre-trained transformer model called Roberta from the Hugging Face library. Transformer models are powerful deep learning architectures for natural language processing (NLP) tasks, including sentiment analysis. Pre-trained models like Roberta are trained on massive datasets, allowing them to capture complex relationships between words and sentiment.

Strengths:

1. Highly accurate and can learn context-dependent sentiment.
2. Adaptable to different domains with fine-tuning.
3. Handles complex language with a better understanding of sarcasm and negation.

Weaknesses:

1. Requires more computational resources compared to VADER.
2. Can be a "black box" - understanding model reasoning might be difficult.
3. Potential for bias if the pre-training data has inherent biases.

VADER offers a quick and interpretable analysis for sentiment overview. However, Roberta's deep learning capabilities might capture more nuanced sentiments specific to food reviews (e.g., subtle flavors, and texture descriptions).

Implementation and Output:

```
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('../input/amazon-fine-food-reviews/Reviews.csv')
print(df.shape)
df = df.head(500)
print(df.shape)
```

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

```
df.head()
```

	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. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

```
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()
```

```
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.
```

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

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

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

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

```
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
  ,/,
  I/PRP
  do/VBP
  n't/RB
  like/VB
  it/PRP
  ./.
  (ORGANIZATION Quaker/NNP Oats/NNPS)
  is/VBZ
  the/DT
  way/NN
  to/TO
  go/VB
  ./.)
```

VADER Seniment Scoring

```
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)
```

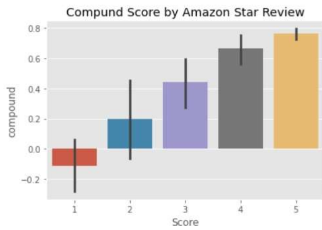
100% 500/500 [00:00<00:00, 860.30it/s]

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

```
# Now we have sentiment score and metadata
vaders.head()
```

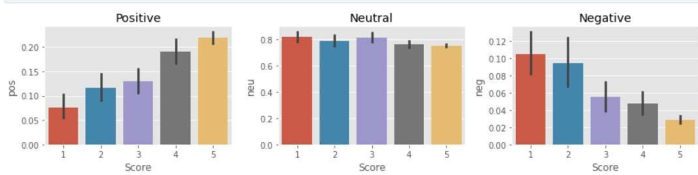
	Id	neg	neu	pos	compound	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	0.000	0.695	0.305	0.9441	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	0.079	0.853	0.068	-0.1027	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	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	1	4	1219017600	"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	3	2	1307923200	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	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

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



Plot VADER results

```
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()
```



Roberta Pretrained 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)
```

```
Downloading: 100% 747/747 [00:00<00:00, 23.7kB/s]

Downloading: 100% 878k/878k [00:00<00:00, 2.29MB/s]

Downloading: 100% 446k/446k [00:00<00:00, 818kB/s]

Downloading: 100% 150/150 [00:00<00:00, 5.03kB/s]

Downloading: 100% 476M/476M [00:22<00:00, 23.4MB/s]
```

```
# VADER results on example
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': 0.9763551, 'roberta_neu': 0.020687457, 'roberta_pos': 0.0029573673}
```

```
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
```

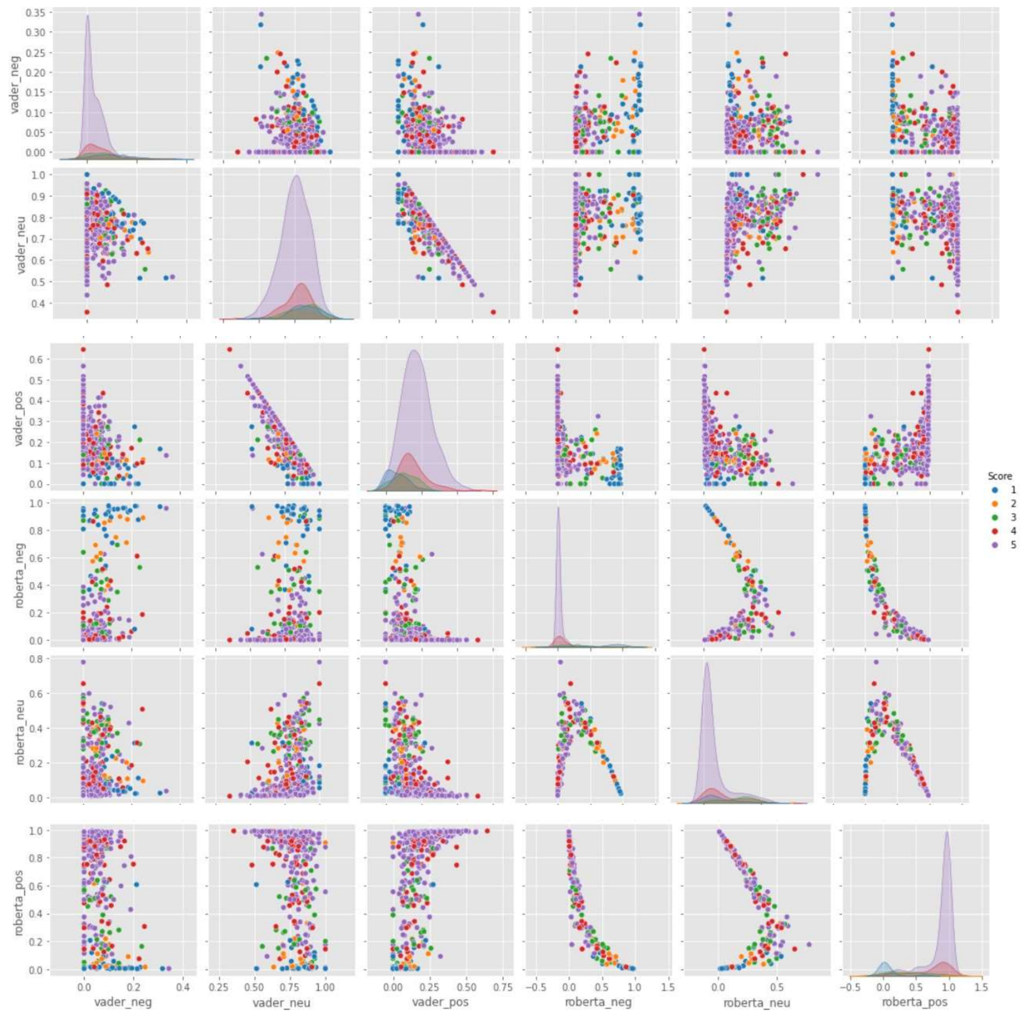
```
res = {}
for i, row in tqdm(df.iterrows(), total=len(df)):
    try:
        text = row['Text']
        myid = row['Id']
        vader_result = sia.polarity_scores(text)
        vader_result_rename = {}
        for key, value in vader_result.items():
            vader_result_rename[f"vader_{key}"] = 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 [01:42<00:00, 3.62it/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.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')
```

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



Lets look at some examples where the model scoring and review score differ the most.

```
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'
```

```
from transformers import pipeline

sent_pipeline = pipeline("sentiment-analysis")
```

```
Downloading: 100% 629/629 [00:00<00:00, 20.9kB/s]

Downloading: 100% 255M/255M [00:12<00:00, 23.0MB/s]

Downloading: 100% 48.0/48.0 [00:00<00:00, 1.42kB/s]

Downloading: 100% 226k/226k [00:00<00:00, 897kB/s]
```

```
sent_pipeline('I love sentiment analysis!')
```

```
[{'label': 'POSITIVE', 'score': 0.9997853636741638}]
```

```
sent_pipeline('Make sure to like and subscribe!')
```

```
[{'label': 'POSITIVE', 'score': 0.9991742968559265}]
```

```
sent_pipeline('booo')
```

```
[{'label': 'NEGATIVE', 'score': 0.9936267137527466}]
```

Conclusion:

In conclusion, the experiment employing both VADER with the Bag of Words approach and the Roberta Pretrained Model from Huggingface's Transformers library has showcased the versatility and effectiveness of different sentiment analysis techniques. VADER's rule-based approach provides a quick and efficient method for sentiment analysis, particularly suited for social media text, while the deep learning capabilities of Roberta offer an enhanced understanding of contextual nuances in language. By leveraging these techniques, businesses can gain valuable insights from Amazon food reviews, empowering them to make informed decisions to enhance customer satisfaction and product quality. Overall, this experiment underscores the importance of employing diverse methodologies in sentiment analysis for comprehensive understanding.