Academic Year: 2023-24 **Semester: VIII**
Class / Branch: BE Computer **Subject: Social Media Analytics Lab**

Name: Shivam Pandey
Date of performance: 03/04/2024
Date of Submission: 10/04/2024

Experiment No. 10

Aim: Develop social media text analytics models for improving existing product/ service by analyzing customers' reviews/comments.

Objective: Develop social media text analytics models to extract actionable insights from customer reviews and comments, aiming to enhance existing products or services. Utilize natural language processing and machine learning algorithms to analyze sentiment, identify areas for improvement, and optimize marketing strategies based on customer feedback.

Software used: Kaggle Notebook.

Theory:

Developing social media text analytics models for improving existing products or services through the analysis of customer reviews and comments is a strategic approach increasingly embraced by businesses aiming to enhance their offerings and customer satisfaction levels. In today's digitally-driven era, where consumers express their opinions and experiences openly on various social media platforms, leveraging advanced text analytics techniques becomes imperative for extracting actionable insights from this wealth of unstructured data.

At its core, social media text analytics involves the use of natural language processing (NLP) and machine learning algorithms to systematically analyze and extract meaningful information from textual data generated on social media platforms. This process encompasses several key steps, including data collection, preprocessing, feature extraction, sentiment analysis, and actionable insights generation.

The first step involves collecting customer reviews and comments from relevant social media channels, such as Facebook, Twitter, Instagram, and review platforms like Yelp or TripAdvisor. This data gathering process may involve web scraping techniques or utilizing application programming interfaces (APIs) provided by social media platforms.

Once the data is collected, preprocessing techniques are applied to clean and standardize the text, which may include tasks such as tokenization, stop word removal, stemming, and lemmatization. These preprocessing steps aim to improve the quality of the data and prepare it for further analysis.

Feature extraction techniques are then employed to transform the preprocessed text data into numerical representations that can be utilized by machine learning models. Common approaches include bag-of-words, term frequency-inverse document frequency (TF-IDF), and word embeddings such as Word2Vec or GloVe.

Sentiment analysis plays a crucial role in social media text analytics, enabling businesses to gauge the overall sentiment expressed in customer reviews and comments towards their products or services. By classifying sentiments as positive, negative, or neutral, organizations can identify areas of strength and improvement, as well as monitor changes in customer perceptions over time.
Finally, the insights generated from the analysis of social media text data can inform various aspects of product or service enhancement, including product feature prioritization, customer service improvements, marketing strategy optimization, and competitor benchmarking. By harnessing the power of social media text analytics, businesses can gain a deeper understanding of customer needs, preferences, and pain points, ultimately driving innovation and enhancing customer satisfaction.

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
```

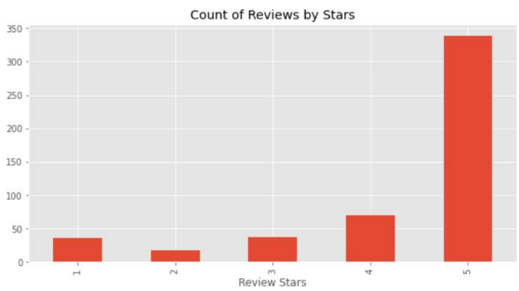
```
# Read in data
df = pd.read_csv('../input/amazon-fine-food-reviews/Reviews.csv')
print(df.shape)
df = df.head(500)
print(df.shape)
```

```
(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')
```

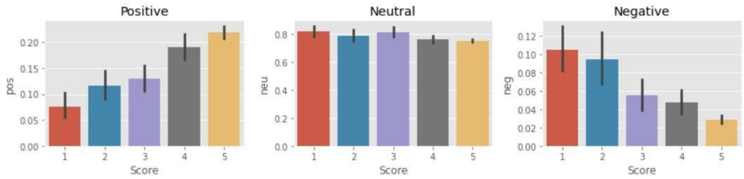
```
# Now we have sentiment score and metadata
vaders.head()
```

	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...

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

Plot VADER results

```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')
axs[1].set_title('Neutral')
axs[2].set_title('Negative')
plt.tight_layout()
plt.show()
```



Roberta Pretrained Model

```python
from transformers import AutoTokenizer
from transformers import AutoModelForSequenceClassification
from scipy.special import softmax
```

```python
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]
```

```python
# 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}')
```

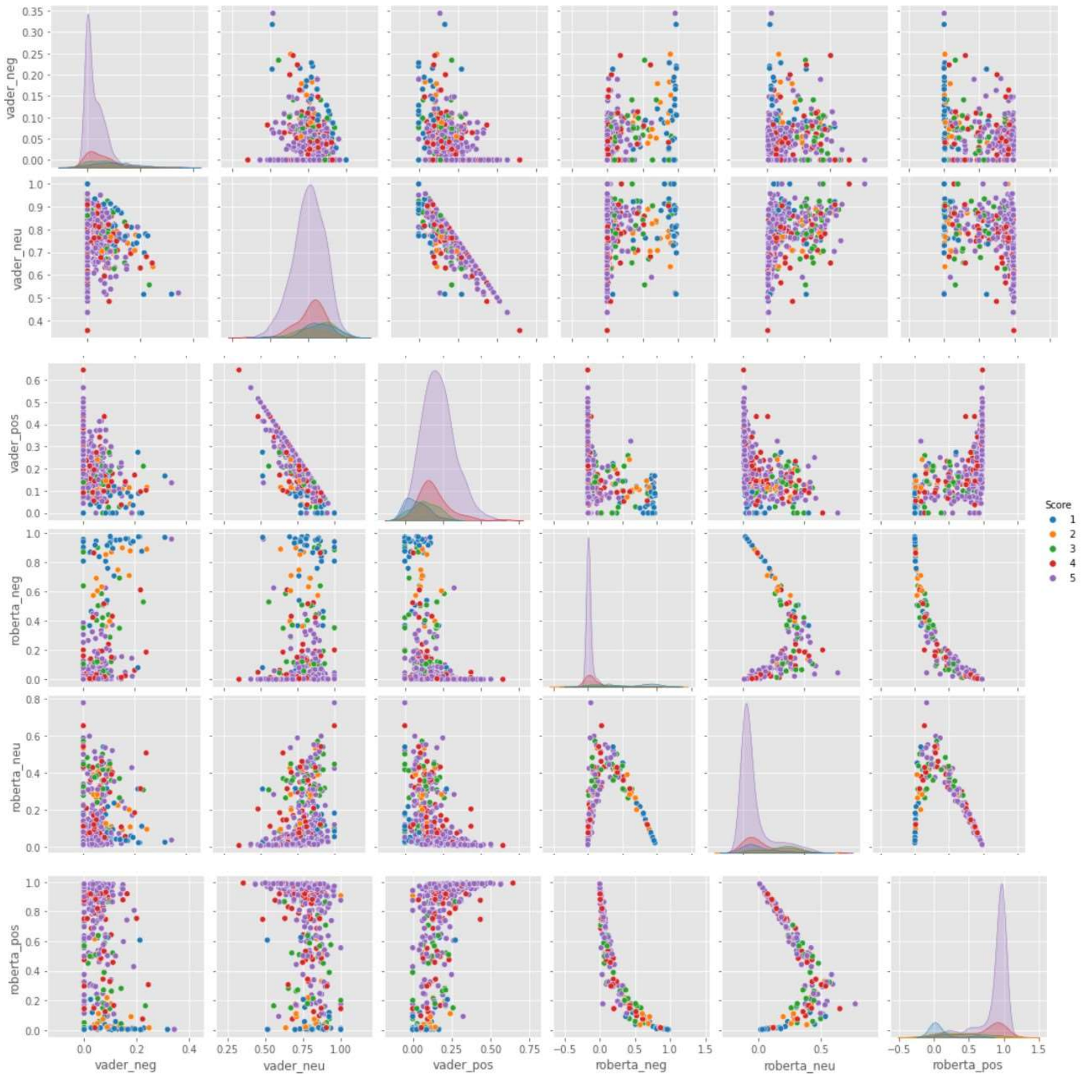
```
100% 500/500 [01:42<00:00, 3.62it/s]

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'
```

Certainly, here are some measures one can adopt to improve the product based on the analysis of negative comments:
Identify Common Pain Points: Analyze negative comments to identify recurring issues or pain points mentioned by multiple users. Addressing these common concerns can lead to significant improvements in product satisfaction.

Prioritize Feedback: Prioritize negative feedback based on severity and frequency. Focus on addressing critical issues that have a substantial impact on user experience before tackling less pressing concerns.

Engage with Users: Engage directly with users who leave negative comments to gather additional insights and clarify their concerns. This proactive approach demonstrates responsiveness and commitment to addressing customer needs.

Enhance Product Features: Use negative feedback to identify areas for feature enhancement or refinement. Incorporate user suggestions and feedback to make the product more robust and user-friendly.

Improve User Interface (UI) and Experience (UX): Assess negative comments related to UI/UX issues such as navigation difficulties, confusing layouts, or slow performance. Invest in UI/UX improvements to enhance the overall usability and satisfaction with the product.

Provide Clear Instructions or Documentation: Address complaints about confusion or difficulty in using the product by providing clear instructions, tutorials, or comprehensive documentation. Empowering users with the necessary guidance can alleviate frustration and improve the overall user experience.

Optimize Performance and Stability: Address negative feedback related to performance issues, bugs, or crashes promptly. Continuously optimize the product to ensure stability, responsiveness, and smooth operation across different devices and platforms.

Enhance Customer Support: Improve customer support channels and responsiveness to address user concerns effectively. Provide timely assistance, resolve issues promptly, and gather feedback to continuously improve support services.

Offer Product Training or Education: Address user frustration stemming from a lack of understanding or knowledge about the product by offering training sessions, webinars, or educational resources. Empowering users with the necessary skills and knowledge enhances their experience and satisfaction.

Implement Feedback Loop: Establish a feedback loop to gather ongoing feedback from users and monitor the impact of implemented improvements. Continuously iterate based on user input to ensure the product remains aligned with user expectations and needs.

Conclusion:

In conclusion, analyzing negative comments provides invaluable insights for improving product quality and enhancing customer satisfaction. By systematically addressing user concerns, prioritizing feedback, and implementing targeted measures, businesses can iteratively refine their products to better meet user needs and expectations. Engaging with users, enhancing features, optimizing performance, and providing clear guidance are among the key strategies for addressing negative feedback effectively. Establishing a feedback loop ensures continuous improvement and maintains alignment with evolving user preferences. Ultimately, leveraging negative comments as constructive feedback fosters product innovation strengthens customer relationships, and drives long-term success in the marketplace.