****

### AML 2304 – Natural Language Processing 01

**Project Report**

Date of Submission: December 12th 2023

**Instructor: Mr. Bhavik Gandhi**

**Group Members**

Gitik Kaushik (C0867079)

Harcharan Singh (C0863764)

Yajur Sethi (C0863424)

Tega Orido (C0857045)

Anargha Manoj (C0865188)

# Index

|  |  |
| --- | --- |
| **S.No.** | **Table of Content** |
| 1 | **Introduction** |
| 2 | **Objectives** |
| 3 | **Data Loading and Exploration** |
| 4 | **Data Preprocessing** |
| 5 | **Exploratory Data Analysis (EDA)** |
| 6 | **Natural Language Processing (NLP) with NLTK** |
| 7 | **Sentiment Analysis with VADER** |
| 8 | **Sentiment Analysis with RoBERTa** |
| 9 | **Sentiment Analysis on the Entire Dataset** |
| 10 | **Data Visualization** |
| 11 | **Examples of High and Low Sentiment Reviews** |
| 12 | **Hugging Face's Sentiment Analysis Pipeline** |
| 13 | **Conclusion** |
| 14 | **Future Enhancements** |
| 15 | **Acknowledgments** |
| 16 | **References** |
| 17 | **Code Availability** |

**Project Report: Sentiment Analysis on Amazon Reviews**

**1. Introduction:**

Sentiment analysis, also known as opinion mining, is a natural language processing (NLP) task that involves determining the sentiment expressed in a piece of text. In this project, our aim is to conduct sentiment analysis on a dataset of Amazon reviews. We utilize two different approaches: rule-based sentiment analysis with VADER and machine learning-based sentiment analysis with a pre-trained RoBERTa model.

**2. Objectives:**

Analyze the sentiment of Amazon reviews using both rule-based and machine learning-based methods.

Compare the results obtained from VADER and RoBERTa to understand their respective strengths and limitations.

Visualize sentiment distributions and explore relationships with review scores.

**3. Data Loading and Exploration:**

We start by loading the dataset (Reviews.csv) using the Pandas library and explore its structure to understand the available attributes.

python

Copy code

import pandas as pd

# Load dataset

df = pd.read\_csv('Reviews.csv')

# Explore dataset structure

print(df.info())

print(df.head())

**4. Data Preprocessing:**

We preprocess the data by selecting a subset of the dataset for analysis. Additionally, we handle any missing or irrelevant data.

python

Copy code

# Select a subset for analysis

df = df.head(500)

**5. Exploratory Data Analysis (EDA):**

Visualizing the distribution of review scores provides insights into the overall sentiment of the dataset.

python

Copy code

import matplotlib.pyplot as plt

import seaborn as sns

# Visualize review score distribution

plt.figure(figsize=(8, 5))

sns.countplot(x='Score', data=df)

plt.title('Distribution of Review Scores')

plt.show()

**6. Natural Language Processing (NLP) with NLTK:**

We demonstrate basic NLP techniques such as tokenization and part-of-speech tagging using NLTK.

python

Copy code

import nltk

from nltk.tokenize import word\_tokenize

from nltk.tag import pos\_tag

nltk.download('punkt')

nltk.download('averaged\_perceptron\_tagger')

# Tokenization and part-of-speech tagging

example\_text = df['Text'][50]

tokens = word\_tokenize(example\_text)

tagged = pos\_tag(tokens)

print(tokens[:10])

print(tagged[:10])

**7. Sentiment Analysis with VADER:**

We utilize VADER, a rule-based sentiment analysis tool from NLTK, to calculate sentiment scores for each review.

python

Copy code

from nltk.sentiment import SentimentIntensityAnalyzer

# Sentiment analysis with VADER

sia = SentimentIntensityAnalyzer()

vader\_scores = df['Text'].apply(lambda x: sia.polarity\_scores(x))

df['vader\_compound'] = vader\_scores.apply(lambda x: x['compound'])

**8. Sentiment Analysis with RoBERTa:**

We employ a pre-trained RoBERTa model from Hugging Face's Transformers library for machine learning-based sentiment analysis.

python

Copy code

from transformers import AutoTokenizer, TFAutoModelForSequenceClassification

import tensorflow as tf

from scipy.special import softmax

# Load RoBERTa model and tokenizer

MODEL = "cardiffnlp/twitter-roberta-base-sentiment"

tokenizer = AutoTokenizer.from\_pretrained(MODEL)

model = TFAutoModelForSequenceClassification.from\_pretrained(MODEL)

# Example sentiment analysis with RoBERTa

encoded\_text = tokenizer(example\_text, return\_tensors='pt')

input\_ids = tf.convert\_to\_tensor(encoded\_text['input\_ids'])

attention\_mask = tf.convert\_to\_tensor(encoded\_text['attention\_mask'])

output = model(input\_ids=input\_ids, attention\_mask=attention\_mask)

# Convert output to probabilities

scores = output[0][0].numpy()

scores = softmax(scores)

**9. Sentiment Analysis on the Entire Dataset:**

Apply both VADER and RoBERTa sentiment analysis to the entire dataset and combine the results.

python

Copy code

res = {}

for i, row in tqdm(df.iterrows(), total=len(df)):

text = row['Text']

myid = row['Id']

res[myid] = {'vader': sia.polarity\_scores(text), 'roberta': polarity\_scores\_roberta(text)}

**10. Data Visualization:**

Visualize relationships between sentiment scores and review scores using pair plots.

Python

Copy code

sns.pairplot(data=df,

vars=['vader\_compound', 'roberta\_pos', 'roberta\_neu', 'roberta\_neg'],

hue='Score',

palette='tab10')

plt.show()

**11. Examples of High and Low Sentiment Reviews:**

Explore examples of reviews with high and low sentiment scores from both VADER and RoBERTa.

python

Copy code

high\_sentiment\_vader = df.query('Score == 5').sort\_values('vader\_compound', ascending=False)['Text'].values[0]

low\_sentiment\_roberta = df.query('Score == 1').sort\_values('roberta\_neg', ascending=False)['Text'].values[0]

**12. Hugging Face's Sentiment Analysis Pipeline:**

Demonstrate the simplicity and efficiency of Hugging Face's sentiment analysis pipeline.

python

Copy code

from transformers import pipeline

sent\_pipeline = pipeline("sentiment-analysis")

# Example usage of sentiment analysis pipeline

result = sent\_pipeline('I love sentiment analysis!')

**13. Conclusion:**

This project provides a comprehensive analysis of sentiment in Amazon reviews using both rule-based (VADER) and machine learning-based (RoBERTa) approaches. The comparison between these methods offers valuable insights. Visualizations enhance the understanding of sentiment distributions and their relationships with review scores.

**14. Future Enhancements:**

Explore other pre-trained models for sentiment analysis.

Perform sentiment analysis on larger datasets for a broader understanding.

Implement fine-tuning of models on domain-specific data for improved accuracy.

**15. Acknowledgments:**

We acknowledge the NLTK, Hugging Face Transformers, and other open-source libraries that contributed to the success of this project.

**16. References:**

NLTK Documentation: https://www.nltk.org/

Hugging Face Transformers Documentation: <https://huggingface.co/transformers/>

**17. Code Availability:**

The complete code for this project is available ON https://github.com/Harcharan1997/NLP-Project