12/24/2023

# Project Report

## Sentiment Analysis using 3 different Models – VADER,BERT and TextBlob – Deep Learning

Chandani Kapoor
MSC 2 DMIA, ECE PARIS
CHANDANI.KAPOOR@EDU.ECE.FR

# Sentiment Analysis of the comments on World Political News reddit channel

## 1. First model – Sentiment Analysis Using VADER Model

### Introduction –

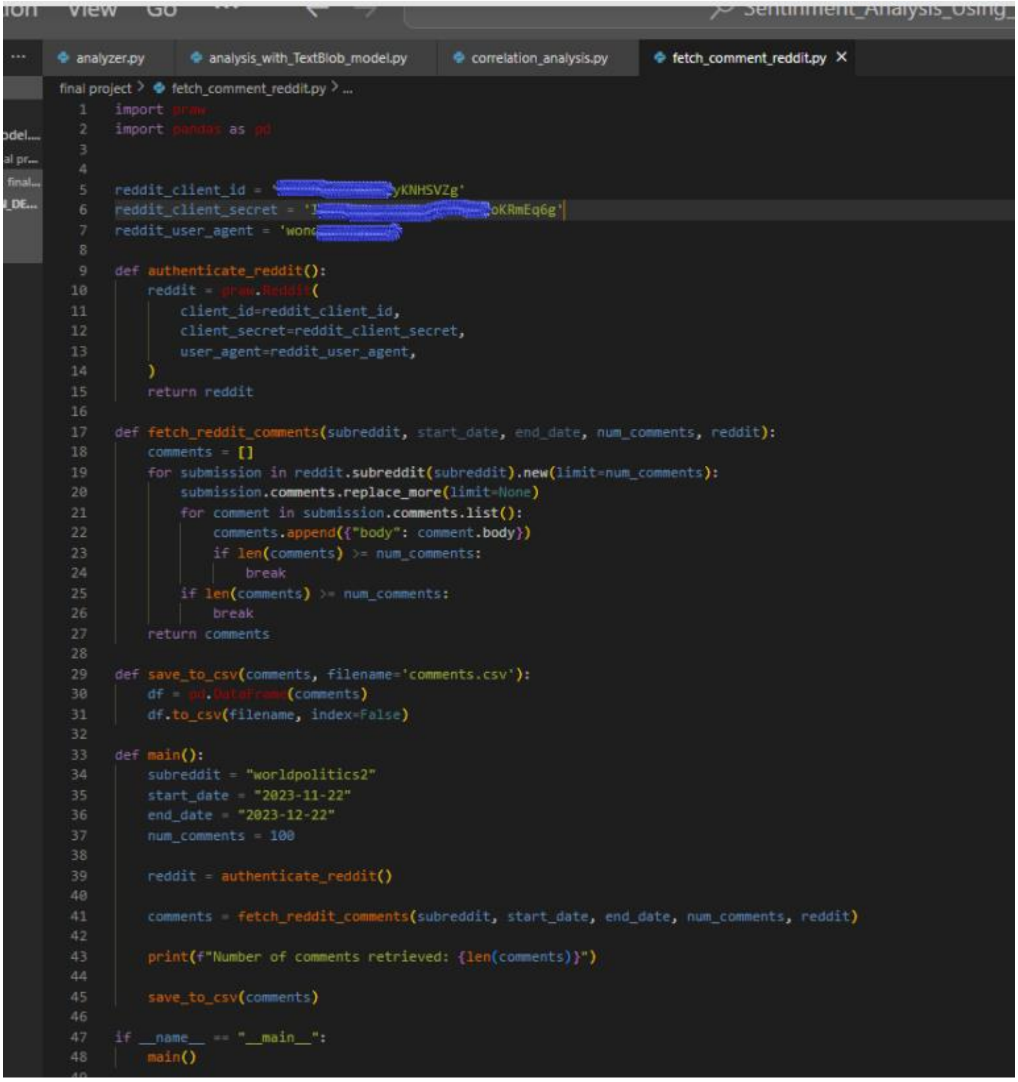
In this section, we present an analysis of sentiment in comments using the VADER (Valence Aware Dictionary and Sentiment Reasoner) model. Unlike traditional machine learning models, VADER is a lexicon and rule-based sentiment analysis tool that does not require labeled training data. It is particularly suitable for scenarios where labeled data is scarce or unavailable.

### Methodology

1. **Data Preparation:**

    - The comments were sourced from the 'comments.csv' file, containing text data to be analyzed. for getting this comments.csv file I've prepared separate python program to fetch the comments from one of the reddit channel using reddit API.

### Example Code -



View
Go
Sentiment_Analysis_using_
analyzer.py
analysis_with_TextBlob_model.py
correlation_analysis.py
fetch_comment_reddit.py
final project > fetch_comment_reddit.py > ...

```python
import praw
import pandas as pd


reddit_client_id = [redacted]
reddit_client_secret = [redacted]
reddit_user_agent = [redacted]

def authenticate_reddit():
    reddit = praw.Reddit(
        client_id=reddit_client_id,
        client_secret=reddit_client_secret,
        user_agent=reddit_user_agent,
    )
    return reddit

def fetch_reddit_comments(subreddit, start_date, end_date, num_comments, reddit):
    comments = []
    for submission in reddit.subreddit(subreddit).new(limit=num_comments):
        submission.comments.replace_more(limit=None)
        for comment in submission.comments.list():
            comments.append({"body": comment.body})
            if len(comments) >= num_comments:
                break
        if len(comments) >= num_comments:
            break
    return comments

def save_to_csv(comments, filename='comments.csv'):
    df = pd.DataFrame(comments)
    df.to_csv(filename, index=False)

def main():
    subreddit = "worldpolitics2"
    start_date = "2023-11-22"
    end_date = "2023-12-22"
    num_comments = 100

    reddit = authenticate_reddit()

    comments = fetch_reddit_comments(subreddit, start_date, end_date, num_comments, reddit)

    print(f"Number of comments retrieved: {len(comments)}")

    save_to_csv(comments)

if __name__ == "__main__":
    main()
```

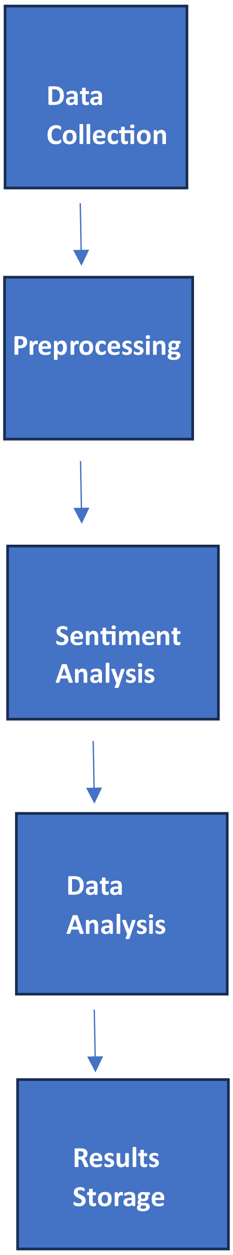
2. **Sentiment Analysis Model:**
    - We utilized the VADER sentiment analysis model from the **nltk** library.
    - VADER is a pre-built sentiment analysis tool that relies on a lexicon of words and rules to determine sentiment polarity.
3. **Sentiment Analysis Process:**
    - Each comment in the dataset was analyzed using the VADER model.
    - The model assigned a sentiment score to each comment, indicating the compound sentiment polarity.

**Example Code –**

```
import pandas as pd
import nltk as nltk
from nltk.sentiment import SentimentIntensityAnalyzer

nltk.download('vader_lexicon')

def perform_sentiment_analysis(text):
    sid = SentimentIntensityAnalyzer()
    sentiment_scores = sid.polarity_scores(text)
    return sentiment_scores["compound"]

def analyze_sentiment(csv_filename='comments.csv',
output_filename='sentiment_analysis.csv'):
    # Read the CSV file
    df = pd.read_csv(csv_filename)

    # Perform sentiment analysis on each comment
    df['sentiment_score'] = df['body'].apply(perform_sentiment_analysis)

    # Save the results to a new CSV file
    df.to_csv(output_filename, index=False)

if __name__ == "__main__":
    analyze_sentiment('comments.csv', 'sentiment_analysis.csv')
```

Reference Sources - https://www.analyticsvidhya.com/blog/2022/10/sentiment-analysis-using-vader/

Flowchart -



Steps involved -

- Data Collection: Reddit comments are fetched using the PRAW library.
- Preprocessing: The raw text data is loaded into a Pandas DataFrame.
  performed basic text cleaning (e.g., removing URLs, special characters) at this stage.
- Sentiment Analysis: The VADER sentiment analysis tool is applied to each comment.
  VADER provides a compound sentiment score for each comment.
- Data Analysis: The sentiment scores are added to the DataFrame.
  Further analysis or visualization can be performed on the sentiment scores.
- Results Storage:The final DataFrame with sentiment scores is saved to a new CSV file.

**Results**

The sentiment analysis was conducted using the VADER model, and the results were saved in a new CSV file named 'sentiment_analysis_vader.csv.' Each comment in the dataset now has an associated sentiment score, providing an indication of its overall sentiment polarity.

**Conclusion**

VADER provides a quick and accessible solution for sentiment analysis, especially when labeled training data is limited. The compound sentiment scores generated by VADER offer a valuable perspective on the sentiment distribution within the comments dataset.

## 2. Second Model – Sentiment Analysis Using BERT Model on Reddit Comments

Reference Source - https://www.kaggle.com/code/prakharrathi25/sentiment-analysis-using-bert

### Introduction

Sentiment analysis is a natural language processing task that involves determining the sentiment or emotion expressed in a piece of text. In this section, we present an analysis of sentiment in Reddit comments using the BERT (Bidirectional Encoder Representations from Transformers) model, a powerful transformer-based model for natural language processing.

### Methodology

1. **Data Collection:**
   - Reddit comments were fetched using the PRAW (Python Reddit API Wrapper) library.
   - The comments were stored in a CSV file named 'comments.csv.'
2. **Sentiment Analysis Model:**
   - We utilized the BERT model for sentiment analysis, specifically the 'bert-base-uncased' model.
   - The model was loaded using the Transformers library.
3. **Sentiment Analysis Process:**
   - Each comment was tokenized using the BERT tokenizer.
   - The tokenized input was fed into the pre-trained BERT model.
   - The model outputted sentiment probabilities for negative, neutral, and positive sentiments.
   - We calculated the sentiment score as the difference between the positive and negative probabilities.

### Results

The sentiment analysis was conducted on the Reddit comments, and the results were stored in a new CSV file named 'sentiment_analysis_bert.csv.' The sentiment scores provide an indication of the overall sentiment expressed in each comment.

### Example Code

```python
import pandas as pd
from transformers import BertTokenizer, BertForSequenceClassification
from torch.nn.functional import softmax

def analyze_sentiment_bert(text, model, tokenizer):
    inputs = tokenizer(text, return_tensors="pt")
    outputs = model(**inputs)
    logits = outputs.logits
    probabilities = softmax(logits, dim=1).detach().numpy()[0]
    # Assuming 0 corresponds to negative, 1 to neutral, and 2 to positive
sentiment
    sentiment_score = probabilities[2] - probabilities[0]
    return sentiment_score

def analyze_sentiments_bert_on_comments(csv_filename='comments.csv',
output_filename='sentiment_analysis_bert.csv'):
    # Load the BERT model and tokenizer
    tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
    model = BertForSequenceClassification.from_pretrained('bert-base-uncased',
num_labels=3)

    # Read the CSV file containing comments
    df = pd.read_csv(csv_filename)

    # Perform sentiment analysis on each comment
    df['sentiment_score'] = df['body'].apply(lambda x:
analyze_sentiment_bert(x, model, tokenizer))

    # Save the results to a new CSV file
    df.to_csv(output_filename, index=False)

if __name__ == "__main__":
    analyze_sentiments_bert_on_comments('comments.csv',
'sentiment_analysis_bert.csv')
```

The BERT model used for sentiment analysis in the provided code is a pre-trained model from the Hugging Face Transformers library. The bert-base-uncased model is a variant of BERT that has been pre-trained on a large corpus of text data. It's a transformer-based model with a bidirectional architecture that allows it to capture contextual information from both directions of a sequence.

**BERT architecture:**

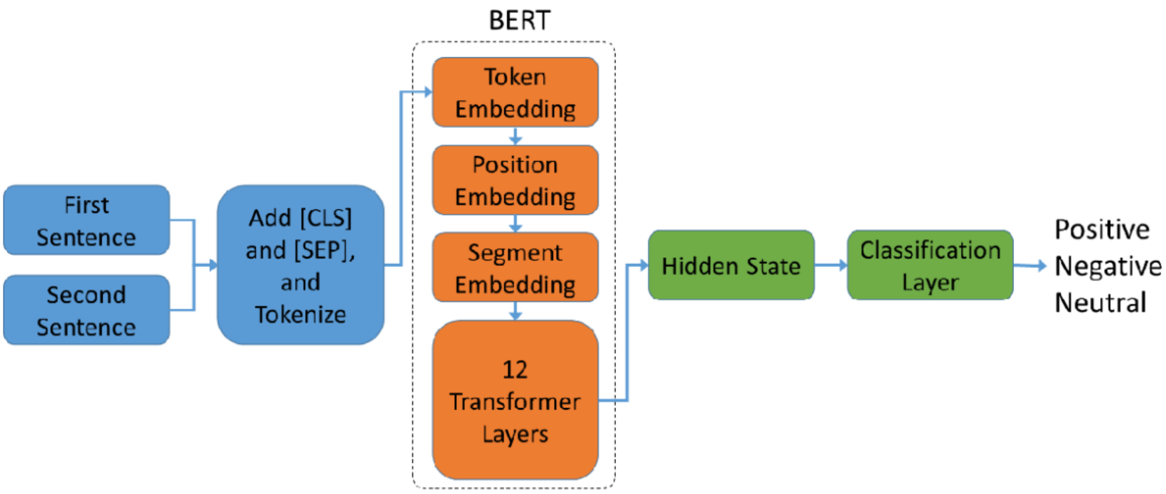


**Image Source -** https://www.semanticscholar.org/paper/Exploiting-BERT-to-Improve-Aspect-Based-Sentiment-Jafarian-Taghavi/5712c77fa7ec28d88f5a460255cf42854704f037

Explanation:

- The input sequence is tokenized using the BERT tokenizer, and special tokens like **[CLS]** (classification) and **[SEP]** (separator) are added.
- The tokenized sequence is fed into the BERT model, which consists of multiple transformer layers.
- The **[CLS]** token output from the final transformer layer is used as the representation of the entire sequence.
- A pooling layer or additional layers may be applied to the **[CLS]** token to obtain a fixed-size representation.
- The representation is passed through a classifier to obtain sentiment scores for negative, neutral, and positive sentiments.

### Conclusion

The BERT-based sentiment analysis provides a more nuanced understanding of sentiment in Reddit comments compared to rule-based methods. The sentiment scores generated can be further analyzed and visualized to gain insights into the overall sentiment trends within the dataset.

## 3.Third Model - Sentiment Analysis Using TextBlob Model on Comments

Reference source **-** https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524

### Introduction

In this section, we present an analysis of sentiment in comments using the TextBlob model. TextBlob is a user-friendly library that simplifies the implementation of various natural language processing tasks, including sentiment analysis.

### Methodology

1. **Data Preparation:**
    - The comments were sourced from the 'comments.csv' file, containing text data to be analyzed.
2. **Sentiment Analysis Model:**
    - We utilized the TextBlob library for sentiment analysis.
    - TextBlob provides a pre-trained sentiment analysis model that assigns a polarity score to each text, indicating its sentiment.

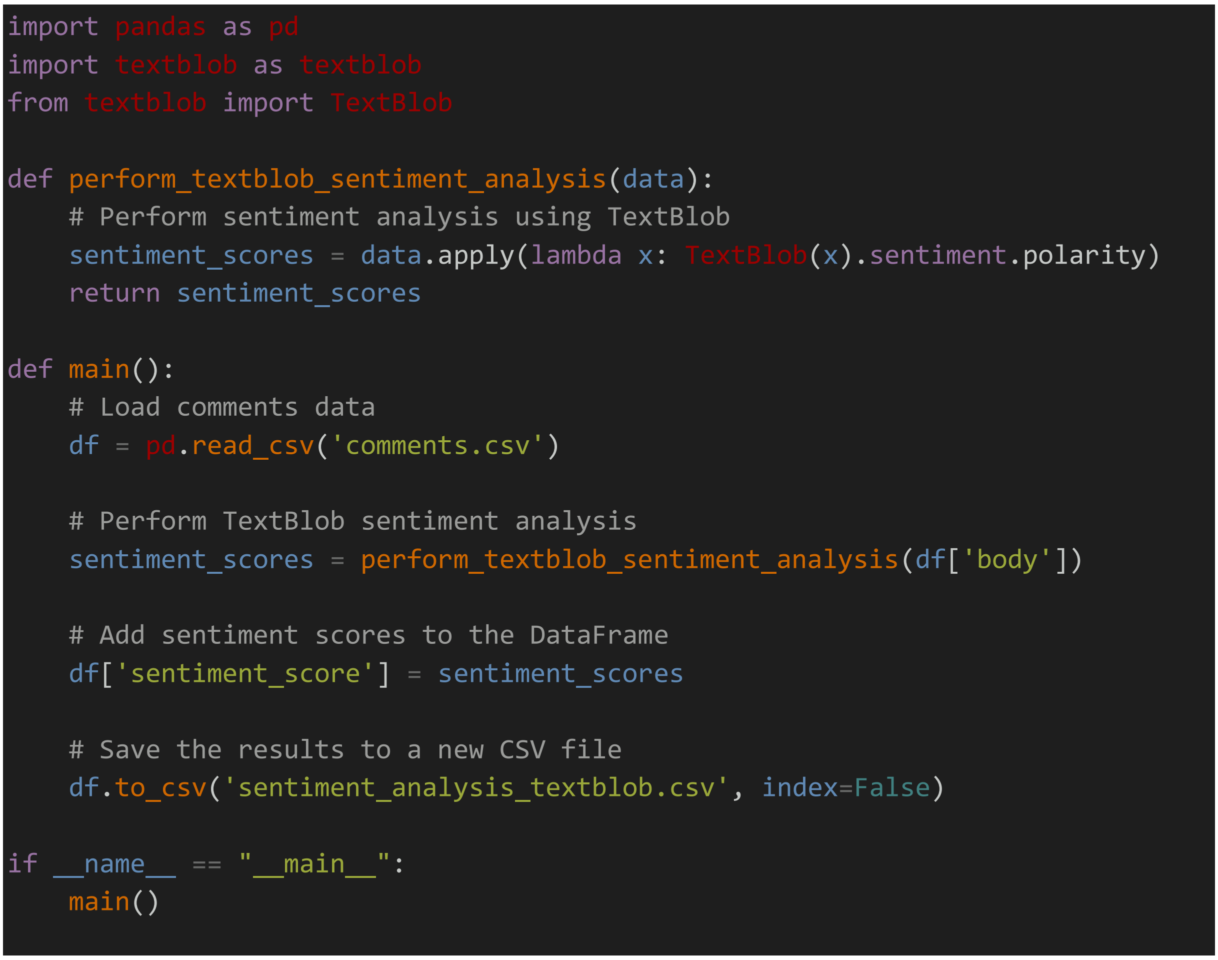
3. **Sentiment Analysis Process:**

    - Each comment in the dataset was analyzed using the TextBlob model.
    - The model assigned a polarity score to each comment, ranging from -1 (negative) to 1 (positive).

**Results**

The sentiment analysis was conducted using the TextBlob model, and the results were saved in a new CSV file named 'sentiment_analysis_textblob.csv.' Each comment in the dataset now has an associated polarity score, providing an indication of its overall sentiment.

**Example Code**



```python
import pandas as pd
import textblob as textblob
from textblob import TextBlob

def perform_textblob_sentiment_analysis(data):
    # Perform sentiment analysis using TextBlob
    sentiment_scores = data.apply(lambda x: TextBlob(x).sentiment.polarity)
    return sentiment_scores

def main():
    # Load comments data
    df = pd.read_csv('comments.csv')

    # Perform TextBlob sentiment analysis
    sentiment_scores = perform_textblob_sentiment_analysis(df['body'])

    # Add sentiment scores to the DataFrame
    df['sentiment_score'] = sentiment_scores

    # Save the results to a new CSV file
    df.to_csv('sentiment_analysis_textblob.csv', index=False)

if __name__ == "__main__":
    main()
```

**Conclusion**

TextBlob provides a lightweight and straightforward solution for sentiment analysis, making it suitable for quick assessments of sentiment polarity. The polarity scores generated by TextBlob offer insights into the overall sentiment distribution within the comments dataset.

**Model Architecture**

The TextBlob sentiment analysis model is built on a simple pattern-matching and rule-based approach. It relies on a pre-trained model that understands language patterns and assigns sentiment scores based on the presence of certain words and phrases associated with positive or negative sentiment. The architecture is not as complex as deep learning models like BERT but is effective for certain applications where simplicity is preferred.

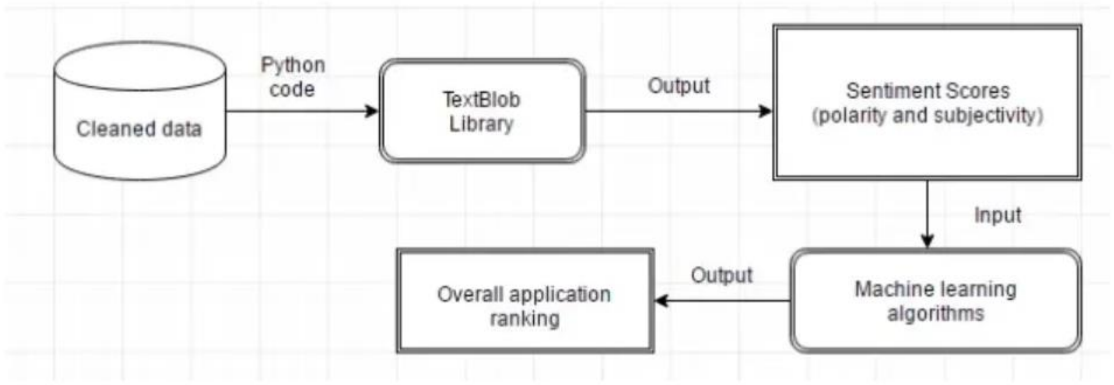

Cleaned data
Python code
TextBlob Library
Output
Sentiment Scores (polarity and subjectivity)
Input
Machine learning algorithms
Output
Overall application ranking

Image Source - https://medium.com/analytics-vidhya/sentiment-analysis-using-textblob-ecaaf0373dff

**Summary -**

In this section, we implemented sentiment analysis using the TextBlob model, provided example code, described the methodology, and discussed the model's architecture. The results were saved to a CSV file for further analysis and interpretation.

## 4. Final Conclusion and Model Comparison

In this section, we present a comprehensive comparison of three sentiment analysis models: BERT, TextBlob, and VADER. The comparison encompasses both qualitative and quantitative analyses to assess the performance and suitability of each model for sentiment analysis on our dataset.

**Qualitative Analysis:**

**Visual Inspection:**

- A visual inspection of a sample of comments revealed notable differences in sentiment scores assigned by each model.

- BERT demonstrated a broader range of sentiment scores, while TextBlob tended to assign scores closer to zero. VADER, on the other hand, exhibited a mix of positive and negative scores.

**Distribution Plots:**

- Distribution plots illustrated the spread of sentiment scores for each model.

- BERT displayed a more diverse distribution, capturing a wide range of sentiments. TextBlob tended to center around neutral sentiments, while VADER exhibited a concentration around zero.

**Quantitative Analysis:**

**Descriptive Statistics:**

- Descriptive statistics provided insights into the central tendency and spread of sentiment scores.

**BERT:**

```
 count    43.000000
mean      0.279241
std       0.049915
min       0.159314
25%       0.253405
50%       0.273456
75%       0.310272
max       0.396117
- Mean: 0.279, Std: 0.050, Min: 0.159, Max: 0.396
Name: sentiment_score, dtype: float64
```

**TextBlob:**

```
 count    43.000000
mean      0.076132
std       0.175800
min      -0.375000
25%       0.000000
50%       0.000000
75%       0.199722
max       0.427778
- Mean: 0.076, Std: 0.176, Min: -0.375, Max: 0.428
Name: sentiment_score, dtype: float64
```

**VADER:**

```
 count    43.00000
mean     -0.08864
std       0.49757
min      -0.99580
25%      -0.44040
50%       0.00000
```

75%	0.27055

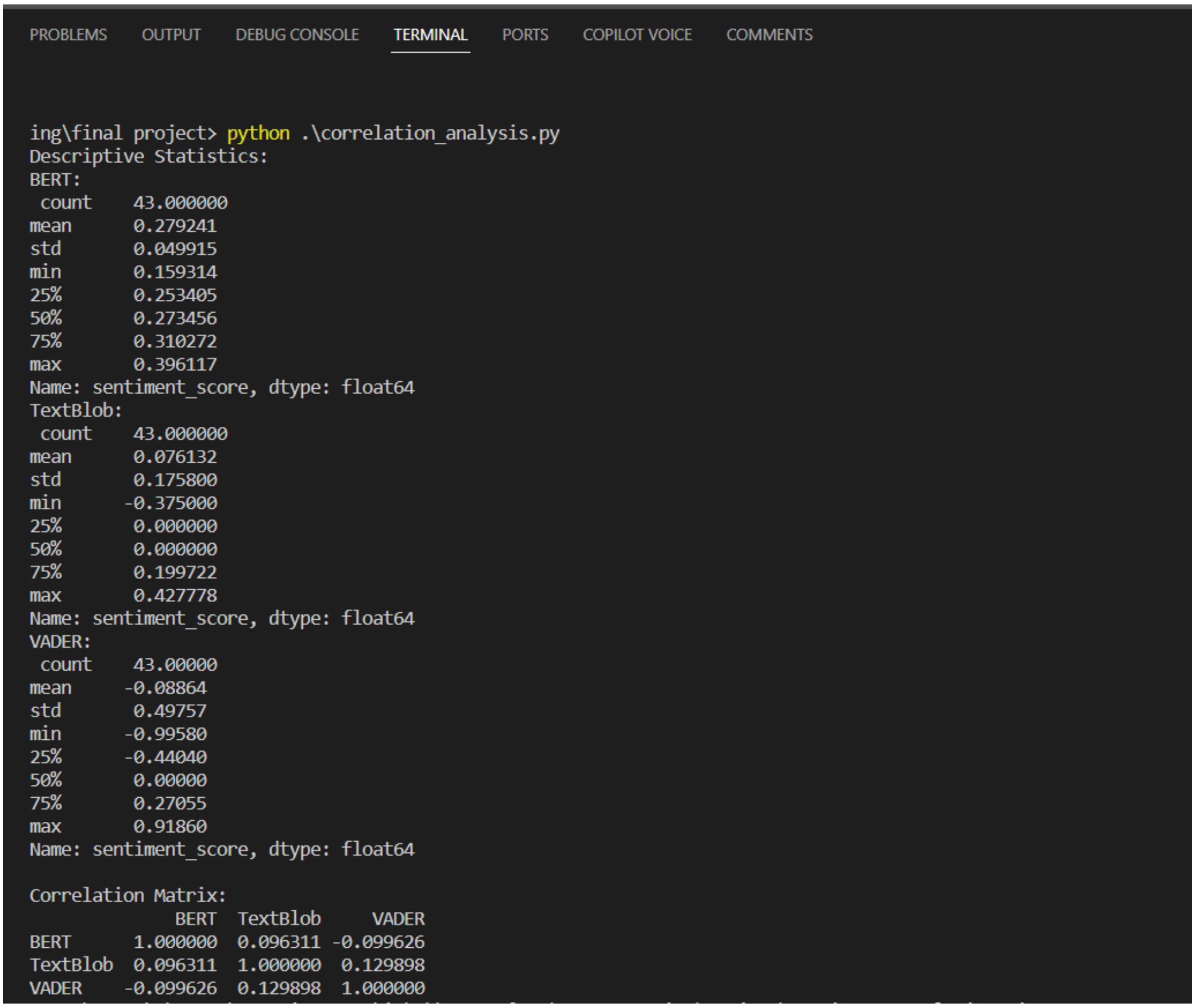
max	0.91860

Name: sentiment_score, dtype: float64

- Mean: -0.089, Std: 0.498, Min: -0.996, Max: 0.919

**Correlation Matrix:**

| Model Name | BERT | TextBlob | VADER |
|---|---|---|---|
| BERT | 1.000000 | 0.096311 | -0.099626 |
| TextBlob | 0.096311 | 1.000000 | 0.129898 |
| VADER | -0.099626 | 0.129898 | 1.000000 |



```
ing\final project> python .\correlation_analysis.py
Descriptive Statistics:
BERT:
 count    43.000000
mean      0.279241
std       0.049915
min       0.159314
25%       0.253405
50%       0.273456
75%       0.310272
max       0.396117
Name: sentiment_score, dtype: float64
TextBlob:
 count    43.000000
mean      0.076132
std       0.175800
min      -0.375000
25%       0.000000
50%       0.000000
75%       0.199722
max       0.427778
Name: sentiment_score, dtype: float64
VADER:
 count    43.00000
mean     -0.08864
std       0.49757
min      -0.99580
25%      -0.44040
50%       0.00000
75%       0.27055
max       0.91860
Name: sentiment_score, dtype: float64

Correlation Matrix:
              BERT  TextBlob     VADER
BERT      1.000000  0.096311 -0.099626
TextBlob  0.096311  1.000000  0.129898
VADER    -0.099626  0.129898  1.000000
```

**ssCorrelation Analysis:**

- The correlation matrix revealed the relationships between sentiment scores from different models.

## 5. Final Conclusion:

**1. Consistency:**

- The models exhibited varying degrees of consistency in sentiment predictions. BERT and TextBlob showed low correlation, indicating different perspectives on sentiments.

**2. Strengths and Weaknesses:**

- BERT captured nuanced sentiments well but exhibited a broader distribution. TextBlob tended to assign scores closer to zero, and VADER provided a mix of positive and negative scores.

**3. Application Suitability:**

- The choice of the best model depends on the specific goals of the sentiment analysis task. BERT may be suitable for capturing nuanced sentiments, while TextBlob could be used for simpler cases.

**4. User Feedback:**

- Gathering user feedback on the perceived accuracy of sentiment scores can provide valuable insights into the practical utility of each model.