CODE:

```
import pandas as pd

import nltk

from nltk.sentiment.vader import SentimentIntensityAnalyzer


# Download NLTK resources

nltk.download('vader_lexicon')


# Initialize the VADER sentiment intensity analyzer

sid = SentimentIntensityAnalyzer()


# Function to calculate sentiment metrics for a given text

def calculate_sentiment_metrics(text):

    # Get the polarity scores for the text

    sentiment_scores = sid.polarity_scores(text)


    # Positive and negative count

    positive_count = sum(1 for score in sentiment_scores.values() if score > 0)

    negative_count = sum(1 for score in sentiment_scores.values() if score < 0)


    # Average positive and negative score

    avg_positive_score = sum(score for score in sentiment_scores.values() if score > 0) /
positive_count if positive_count > 0 else 0

    avg_negative_score = sum(score for score in sentiment_scores.values() if score < 0) /
negative_count if negative_count > 0 else 0


    # Total sentiment score
```

```
    total_score = sentiment_scores['compound']

    # Positive sentiment score
    positive_score = sentiment_scores['pos']

    # Negative sentiment score
    negative_score = sentiment_scores['neg']

    # Return sentiment metrics as a dictionary
    return {
        'Positive Count': positive_count,
        'Negative Count': negative_count,
        'Avg Positive': avg_positive_score,
        'Avg Negative': avg_negative_score,
        'Total Sentiment Score': total_score,
        'Positive Sentiment Score': positive_score,
        'Negative Sentiment Score': negative_score
    }

# Read the Excel file
df = pd.read_excel("C:\\Users\\Rama Krishna\\Desktop\\TextData.xlsx")

# Apply the function to calculate sentiment metrics for each tweet
sentiment_metrics = df['Text'].apply(calculate_sentiment_metrics)

# Convert the list of dictionaries to a DataFrame
```

```
sentiment_metrics_df = pd.DataFrame(sentiment_metrics.tolist())

# Concatenate the original DataFrame with the new sentiment metrics DataFrame
df = pd.concat([df, sentiment_metrics_df], axis=1)

# Save the DataFrame with sentiment metrics to a new Excel file
df.to_excel("C:\\Users\\Rama Krishna\\Documents\\rk865.xlsx", index=False)
```

ex:

Tokenization:

The tweet is tokenized into individual words: ["The", "movie", "was", "not", "good", ".", "It", "was", "actually", "quite", "bad", "."]

Sentiment Intensity Analysis:

VADER assigns polarity scores to each token:

"The": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

"movie": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

"was": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

"not": Positive=0.0, Negative=0.6, Neutral=0.4, Compound=-0.6

"good": Positive=0.7, Negative=0.0, Neutral=0.3, Compound=0.7

".": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

"It": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

"was": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

"actually": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

"quite": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

"bad": Positive=0.0, Negative=0.769, Neutral=0.231, Compound=-0.769

".": Positive=0.0, Negative=0.0, Neutral=1.0, Compound=0.0

Calculating Sentiment Metrics:

Positive Count: 2 (words with positive sentiment)

Negative Count: 2 (words with negative sentiment)

Average Positive Score: (0.7 + 0) / 2 = 0.35

Average Negative Score: (0.6 + 0.769) / 2 = 0.6845

Total Sentiment Score: Compound score of the tweet = -0.6 + 0.7 - 0.769 = -0.669

Positive Sentiment Score: Sum of positive polarity scores = 0.7

Negative Sentiment Score: Sum of negative polarity scores = 0.6 + 0.769 = 1.369

Output:

Sentiment Score: -0.669

Positive Count: 2

Negative Count: 2

Avg Positive: 0.35

Avg Negative: 0.6845

Total Sentiment Score: -0.669

Positive Sentiment Score: 0.7

Negative Sentiment Score: 1.369

These metrics provide a quantitative understanding of the sentiment expressed in the tweet, indicating its positivity, negativity, and overall sentiment polarity.

Compound score:

The compound score is calculated by taking the sum of all the lexicon ratings (positive, negative, neutral) and then normalizing it to be between -1 (most negative) and +1 (most positive). This is the score returned by VADER, which represents the overall sentiment of the text.

In the example we provided, the compound score is computed based on the individual positive and negative scores of each word in the text. Here's how it's calculated:

Compound Score = (Positive Score - Negative Score) / (Positive Count + Negative Count + Neutral Count)