Title: - Document Sentiment Analysis Using Polarity Words

Aim/Objective: - The objective of this experiment is to classify documents as positive, negative, or neutral based on the occurrence of predefined positive and negative words. The sentiment analysis will be conducted by applying a simple rule-based classifier to assess the sentiment polarity of each document.

Software Required:

- Python programming environment (Jupyter Notebook, Google Colab, or any Python IDE)
- Libraries: pandas, matplotlib

Hardware Required:

- 4GB RAM
- Intel i3 or higher / AMD equivalent
- GPU for accelerated computations (if using deep learning frameworks)
- 120 GB SSD

Theory:

Sentiment analysis refers to the process of identifying and extracting subjective information from text data. It is a common application in natural language processing (NLP). A common method of sentiment classification is based on polarity words: predefined lists of words classified as either positive or negative. The polarity of the document is determined by counting how many positive and negative words it contains.

Positive Polarity: A document is considered positive if the number of positive words exceeds the number of negative words.

Negative Polarity: A document is considered negative if the number of negative words exceeds the number of positive words.

Neutral Polarity: A document is considered neutral if the number of positive and negative words are equal.

Procedure

1. Define two lists of words: one for positive words and another for negative words.
2. Read the dataset (reviews.csv) that contains the documents to be classified.
3. Implement a classifier function that processes each document:
    - Split the document into words.
    - Count the occurrences of positive and negative words.

    - Classify the document as "Positive," "Negative," or "Neutral" based on the counts.

4. Apply the classifier to each document in the dataset and store the sentiment result.
5. Visualize the distribution of sentiments using a bar chart.

Observations:

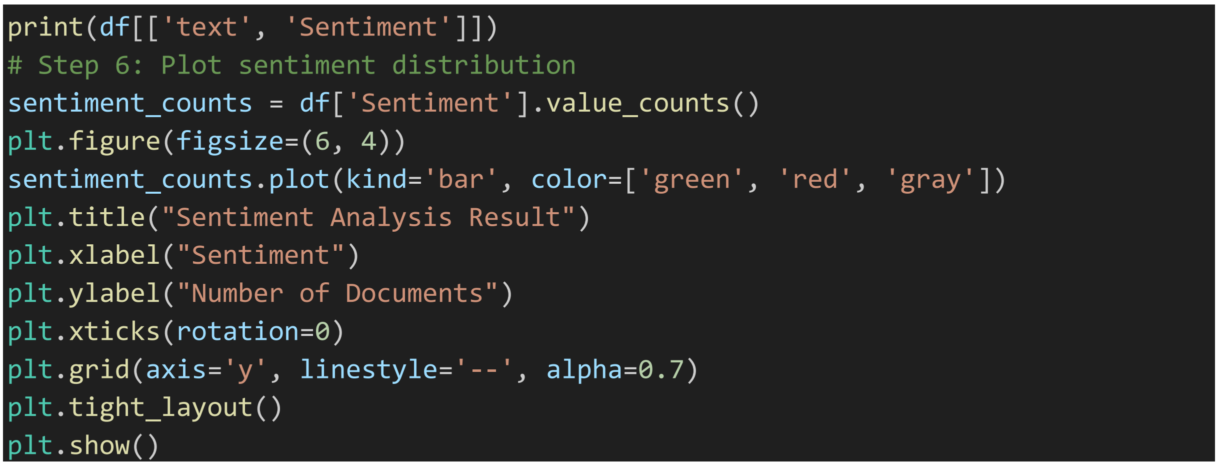
After applying the sentiment classification, the following observations were noted:

- Sentiments were classified as "Positive," "Negative," or "Neutral."
- The bar chart visually represented the sentiment distribution across the documents.

Code:

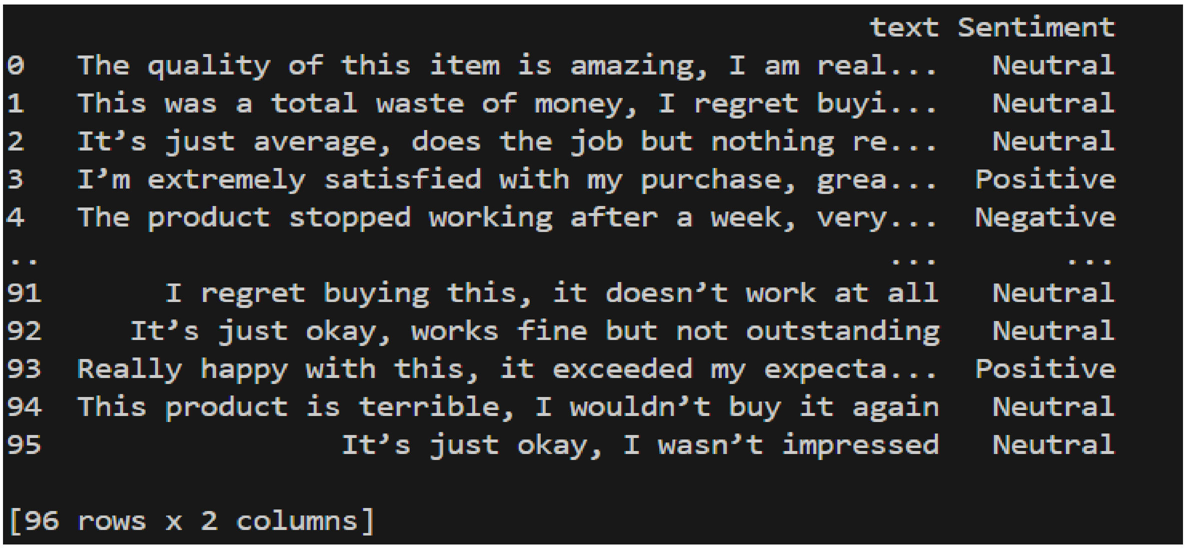
```
import pandas as pd
import matplotlib.pyplot as plt
# Step 1: Define positive and negative word lists
positive_words = ["good", "great", "happy", "love", "excellent", "fantastic",
"awesome", "pleased", "enjoy"]
negative_words = ["bad", "terrible", "sad", "hate", "poor", "awful", "worst",
"angry", "disappointed"]
# Step 2: Read CSV file
df = pd.read_csv("reviews.csv")  # Make sure 'reviews.csv' has a column named
'text'
# Step 3: Define sentiment classifier
def classify_sentiment(text):
    tokens = text.lower().split()
    pos = sum(1 for word in tokens if word in positive_words)
    neg = sum(1 for word in tokens if word in negative_words)
    #11th 12th alternative method/explanation
    #for word in tokens:
    #if word in positive_words:
    #    positive_count += 1
    #elif word in negative_words:
    #    negative_count += 1

    if pos > neg:
        return "Positive"
    elif neg > pos:
        return "Negative"
    else:
        return "Neutral"
# Step 4: Apply classifier to each document
df['Sentiment'] = df['text'].apply(classify_sentiment)
# Step 5: Print classified sentiments
```



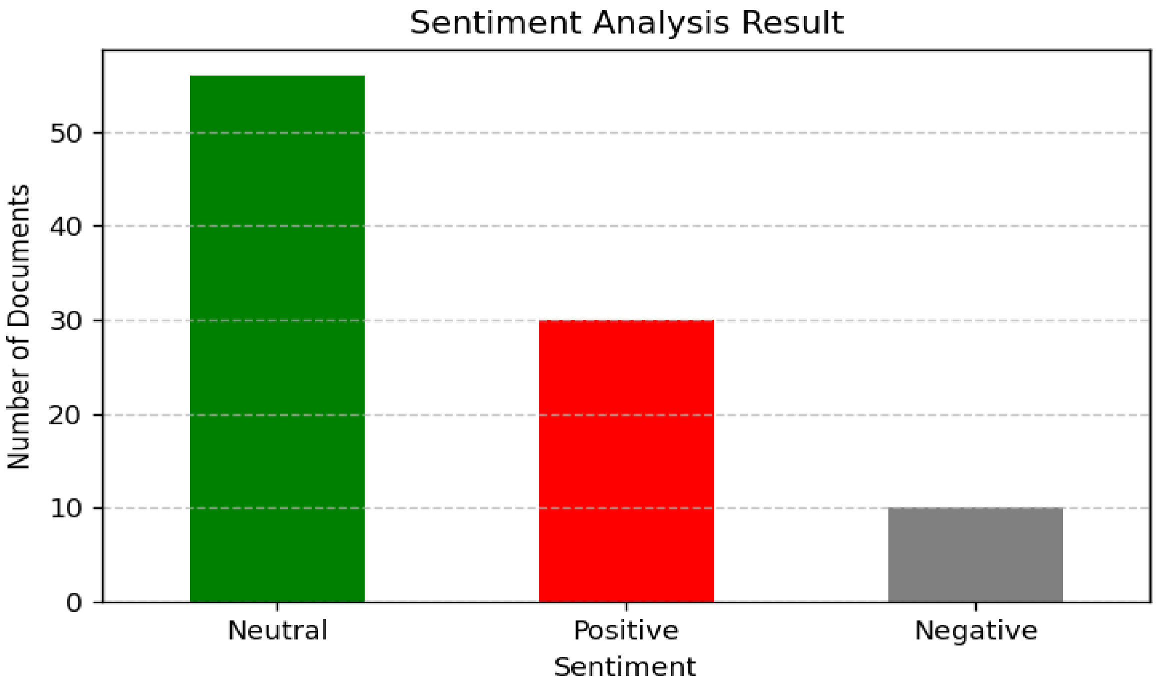
```
print(df[['text', 'Sentiment']])
# Step 6: Plot sentiment distribution
sentiment_counts = df['Sentiment'].value_counts()
plt.figure(figsize=(6, 4))
sentiment_counts.plot(kind='bar', color=['green', 'red', 'gray'])
plt.title("Sentiment Analysis Result")
plt.xlabel("Sentiment")
plt.ylabel("Number of Documents")
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Output:



```
                                                 text Sentiment
0   The quality of this item is amazing, I am real...   Neutral
1   This was a total waste of money, I regret buyi...   Neutral
2   It’s just average, does the job but nothing re...   Neutral
3   I’m extremely satisfied with my purchase, grea...  Positive
4   The product stopped working after a week, very...  Negative
..                                                ...       ...
91       I regret buying this, it doesn’t work at all   Neutral
92     It’s just okay, works fine but not outstanding   Neutral
93  Really happy with this, it exceeded my expecta...  Positive
94  This product is terrible, I wouldn’t buy it again   Neutral
95                  It’s just okay, I wasn’t impressed   Neutral

[96 rows x 2 columns]
```



Conclusion:

This experiment successfully demonstrated the use of polarity words for sentiment classification. Documents were effectively classified as positive, negative, or neutral based on the presence of words from predefined lists. Although this method is relatively simple, it provides an understanding of sentiment analysis and can be further extended to more advanced techniques like machine learning-based classifiers for better accuracy.