Sentiment Analysis Using Natural Language Processing (NLP)

Prepared by: Ruwan Pathiranage Sanduni Nisansala

Internship ID: CV/A1/45373

Tool Used: Python (VS Code)

Task: Sentiment Analysis using NLP

1. Introduction

This project focuses on performing Sentiment Analysis using Natural Language Processing (NLP) techniques.
The main goal is to analyze textual data (such as customer feedback, reviews, or social media comments) and classify each text entry as Positive, Negative, or Neutral.

By automating this process, businesses and organizations can quickly understand public opinion, improve services, and make data-driven decisions.

2. Objectives

- To preprocess text data using tokenization, stopword removal, and stemming.
- To perform sentiment analysis using the TextBlob library.
- To visualize sentiment distribution using bar charts.
- To generate word clouds representing the most frequent words in each sentiment category.
- To save the results and visualizations for reporting and analysis.

3. Tools and Technologies

Tool / Library	Purpose
Python	Programming language used for implementation
Pandas	Data manipulation and cleaning
NLTK	Natural language processing (tokenization, stopwords, stemming)
TextBlob	Sentiment polarity analysis
Matplotlib	Data visualization (bar charts)
WordCloud	Visualization of word frequency by sentiment

4. Dataset Overview

Dataset Name: 3) Sentiment dataset.csv

Number of Columns: 15

Key Columns: Text, Sentiment, Timestamp, User, Platform, Hashtags, Likes, Country, Year, Month, Day, Hour

Preview of Dataset:

Text	Sentiment	Timestamp	Platform
Enjoying a beautiful day at the park!	Positive	2024-03-15 12:00	Twitter
Traffic was terrible this morning.	Negative	2024-03-15 08:00	Twitter
Just finished an amazing workout! 💪	Positive	2024-03-15 15:00	Twitter

Excited about the upcoming weekend getaway!	Positive	2024-03-15 18:00	Twitter
Trying out a new recipe for dinner tonight.	Positive	2024-03-15 19:00	Twitter

5. Methodology

Step 1 – Importing Libraries and Loading Dataset

The necessary Python libraries such as pandas, nltk, TextBlob, matplotlib, and wordcloud were imported.

NLTK resources like punkt, punkt_tab, and stopwords were downloaded to enable tokenization and text cleaning.

```
import pandas as pd
import nltk
from textblob import TextBlob
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('stopwords')
```

The dataset was loaded using Pandas and inspected for missing values and duplicated rows.

There are no missing values or duplicated rows in this dataset used here.

Step 2 – Text Preprocessing

Text preprocessing was performed to clean and prepare text data for analysis. Operations included:

- Converting all text to lowercase
- Removing punctuation
- Tokenization using word_tokenize()
- Removing stopwords (common words like *the*, *is*, *in*)
- Applying stemming to reduce words to their root forms

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
import string


def preprocess_text(text):
    text = str(text).lower()
    text = text.translate(str.maketrans('', '', string.punctuation))
    tokens = word_tokenize(text)
    filtered_words = [word for word in tokens if word not in stop_words]
    stemmed_words = [stemmer.stem(word) for word in filtered_words]
    return ' '.join(stemmed_words)
```

Step 3 – Sentiment Analysis

Each preprocessed sentence was analyzed using **TextBlob**.
TextBlob assigns a **polarity score** between -1 and +1:

- **Positive Sentiment:** Polarity > 0
- **Negative Sentiment:** Polarity < 0
- **Neutral Sentiment:** Polarity = 0

```
def get_sentiment(text):
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity
    if polarity > 0:
        return 'Positive'
    elif polarity < 0:
        return 'Negative'
    else:
        return 'Neutral'
```

Step 4 – Visualization

1. **Sentiment Distribution:**

   A bar chart was created using Matplotlib to show the number of positive, negative, and neutral texts.

2. **Word Clouds:**

   Word clouds were generated for each sentiment category to visualize the most frequently used words.

6. Results

6.1 Preprocessed Text Sample

Original Text	Cleaned Text
Enjoying a beautiful day at the park!	enjoy beauti day park
Traffic was terrible this morning.	traffic terribl morn
Just finished an amazing workout!	finish amaz workout
Excited about the upcoming weekend getaway!	excit upcom weekend getaway
Trying out a new recipe for dinner tonight.	tri new recip dinner tonight

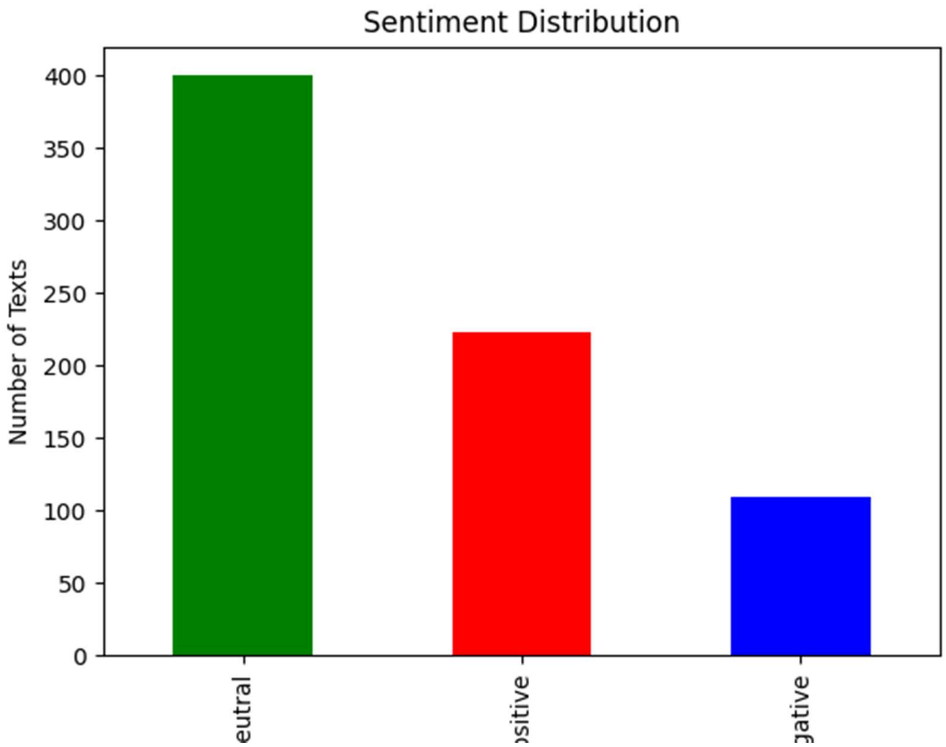
6.2 Sentiment Classification

Text	Sentiment
Enjoying a beautiful day at the park!	Positive
Traffic was terrible this morning.	Neutral
Just finished an amazing workout!	Neutral
Excited about the upcoming weekend getaway!	Neutral
Trying out a new recipe for dinner tonight.	Positive

6.3 Visualizations

1. **Sentiment Distribution Bar Chart**

- File Saved As: sentiment_distribution.png.
- This chart represents the total number of positive, negative, and neutral texts.

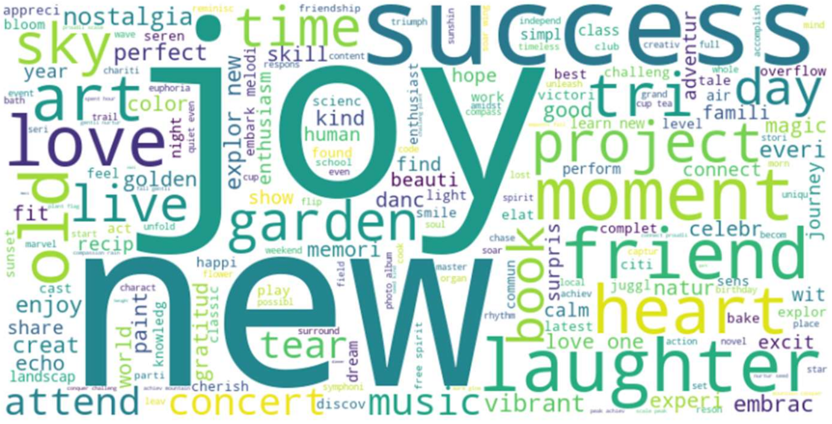


2. **Word Clouds**

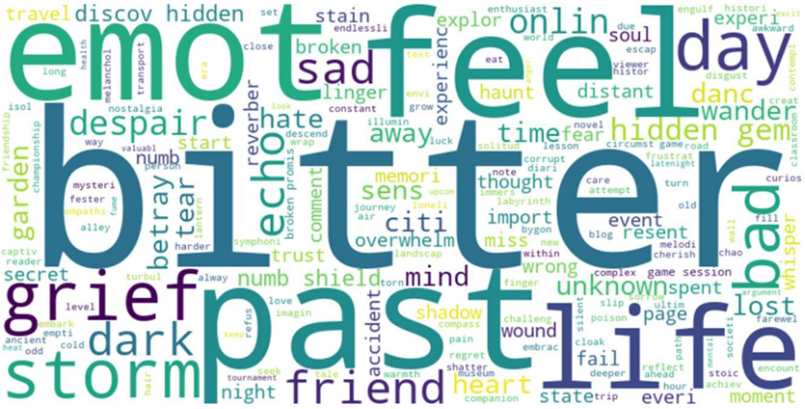
Sentiment	**File Name**
Positive	wordcloud_positive.png
Negative	wordcloud_negative.png
Neutral	wordcloud_neutral.png

- Each word cloud highlights the most common words for that particular sentiment.

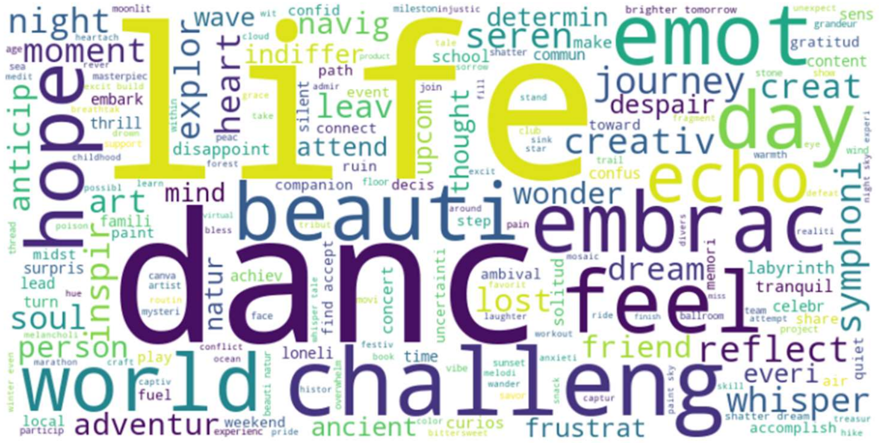
Word Cloud for Positive Sentiment



Word Cloud for Negative Sentiment



Word Cloud for Neutral Sentiment



7. Output Files Generated

File Name	Description
sentiment_analysis_results.csv	Contains the original text, cleaned text, and predicted sentiment
sentiment_distribution.png	Bar chart of sentiment counts
wordcloud_positive.png	Word cloud for positive texts
wordcloud_negative.png	Word cloud for negative texts
wordcloud_neutral.png	Word cloud for neutral texts

8. Conclusion

This project successfully implemented sentiment analysis using NLP techniques in Python. Through preprocessing, polarity-based classification, and visualizations, textual data was efficiently analyzed and categorized.

The approach demonstrates how businesses can utilize automated sentiment detection to gain insights into customer opinions and improve decision-making processes.