

Phase-3

Student Name: KRISHNAKUMAR K

Register Number: 712523121011

Institution: PPG INSTITUTE OF TECHNOLOGY

Department: BE-BIO MEDICAL ENGINEERING

Date of Submission: 16.05.2025

Github Repository Link:

Decoding emotions through sentiment analysis of social media conversations

1.Problem Statement:

In the digital age, social media platforms have become powerful mirrors of human emotion, capturing raw, real-time sentiments expressed across the globe. This project explores how sentiment analysis—a natural language processing (NLP) technique—can decode these emotions by analyzing conversations on platforms like Twitter, Facebook, Instagram, and whatsapp.

2. Abstract:

In the digital era, social media platforms serve as dynamic spaces where individuals freely express thoughts, opinions, and emotions. This study explores the application of sentiment analysis to decode human emotions embedded in social media conversations. Leveraging natural language processing (NLP) techniques and machine learning algorithms, the research analyzes user-generated content to classify sentiments into categories such as positive, negative, and neutral, and further identifies nuanced emotional states including joy, anger, sadness, and fear. The findings reveal patterns in emotional expression across platforms and contribute to understanding public mood, behavioral trends, and societal reactions to events. This work underscores the potential of sentiment analysis as a powerful tool for real-time emotional insight, with implications for marketing, mental health monitoring, and policy-making.

3. System Requirements:

1. Hardware Requirements:

- Processor: Intel Core i5 or higher
- RAM: Minimum 8 GB (16 GB recommended for large datasets)
- Storage: Minimum 256 GB SSD (for fast data access)
- Graphics Card: Optional (only if deep learning models are used)
- Internet Connection: Required for real-time data collection from social media APIs

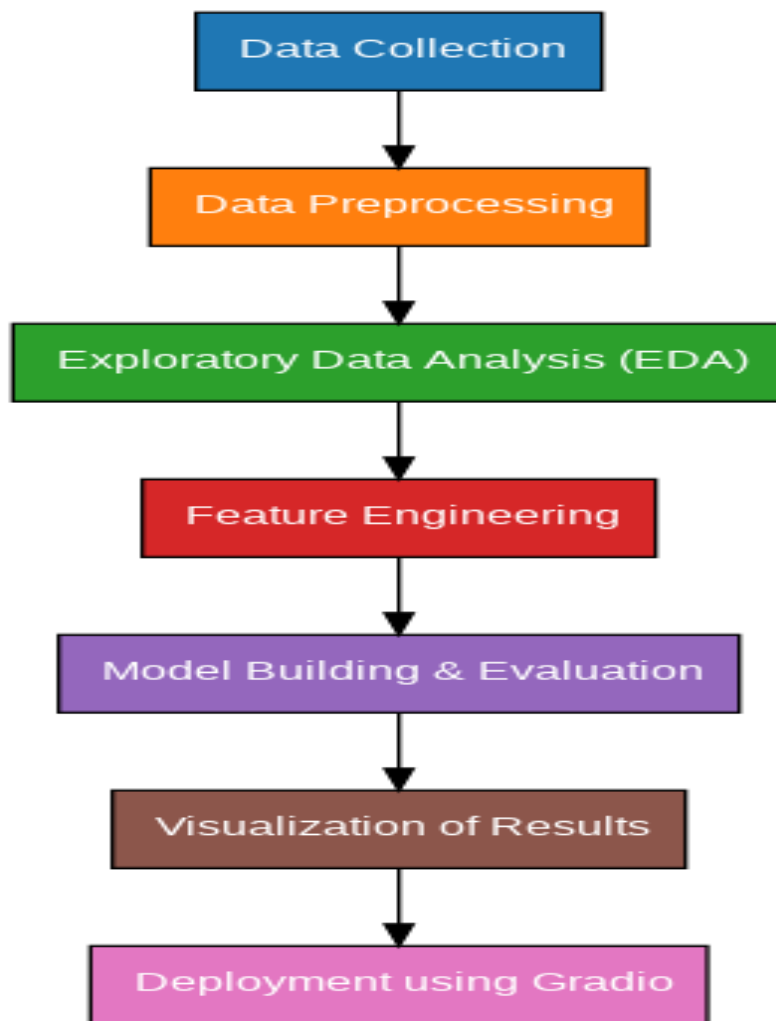
2. Software Requirements:

- Operating System: Windows 10/11, Linux (Ubuntu 18.04+), or macOS
 - Programming Language: Python 3.8+
 - Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, gradio, plotly
- IDE: Google Colab (preferred for free GPU and easy setup)

4.Objectives of the Project:

- To identify emotional patterns and trends in social media conversations.
- To classify sentiments (positive, negative, neutral) and deeper emotions (joy, anger, fear, etc.) using machine learning and deep learning models.
- To assess the impact of external events (e.g., pandemics, elections, celebrity news) on collective emotional expression.
- To understand the role of language nuances, slang, emojis, and sarcasm in sentiment detection.

5. Flowchart of the Project Workflow:



6. Data Description:

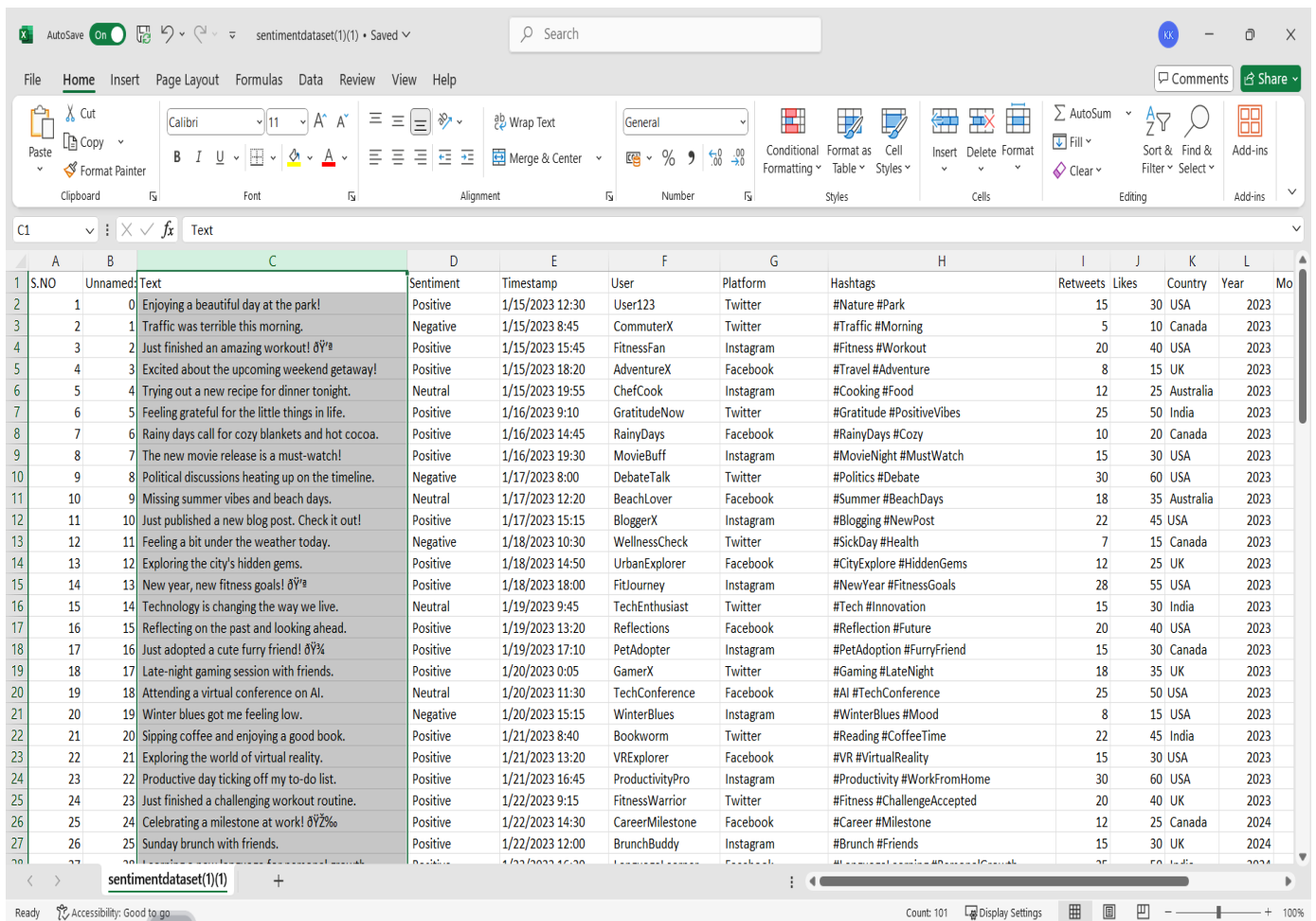
- **Source:** UCI Machine Learning Repository ([sentimentdataset\(1\)\(1\).xlsx](#))
- **Type:** Public dataset
- **Size:** 101 rows × 15columns
- **Nature:** Structured tabular data

- **Attributes:**Demographics
- Sample dataset (df.head([sentimentdataset\(1\)\(1\).xlsx](#)))

7. Data Preprocessing:

Missing Values: None detected.

Duplicates: Checked and none found.



S.NO	Unnamed	Text	Sentiment	Timestamp	User	Platform	Hashtags	Retweets	Likes	Country	Year	Mo
1	0	Enjoying a beautiful day at the park!	Positive	1/15/2023 12:30	User123	Twitter	#Nature #Park	15	30	USA	2023	
2	1	Traffic was terrible this morning.	Negative	1/15/2023 8:45	CommuterX	Twitter	#Traffic #Morning	5	10	Canada	2023	
3	2	Just finished an amazing workout! ðŸ”ð	Positive	1/15/2023 15:45	FitnessFan	Instagram	#Fitness #Workout	20	40	USA	2023	
4	3	Excited about the upcoming weekend getaway!	Positive	1/15/2023 18:20	AdventureX	Facebook	#Travel #Adventure	8	15	UK	2023	
5	4	Trying out a new recipe for dinner tonight.	Neutral	1/15/2023 19:55	ChefCook	Instagram	#Cooking #Food	12	25	Australia	2023	
6	5	Feeling grateful for the little things in life.	Positive	1/16/2023 9:10	GratitudeNow	Twitter	#Gratitude #PositiveVibes	25	50	India	2023	
7	6	Rainy days call for cozy blankets and hot cocoa.	Positive	1/16/2023 14:45	RainyDays	Facebook	#RainyDays #Cozy	10	20	Canada	2023	
8	7	The new movie release is a must-watch!	Positive	1/16/2023 19:30	MovieBuff	Instagram	#MovieNight #MustWatch	15	30	USA	2023	
9	8	Political discussions heating up on the timeline.	Negative	1/17/2023 8:00	DebateTalk	Twitter	#Politics #Debate	30	60	USA	2023	
10	9	Missing summer vibes and beach days.	Neutral	1/17/2023 12:20	BeachLover	Facebook	#Summer #BeachDays	18	35	Australia	2023	
11	10	Just published a new blog post. Check it out!	Positive	1/17/2023 15:15	BloggerX	Instagram	#Blogging #NewPost	22	45	USA	2023	
12	11	Feeling a bit under the weather today.	Negative	1/18/2023 10:30	WellnessCheck	Twitter	#SickDay #Health	7	15	Canada	2023	
13	12	Exploring the city's hidden gems.	Positive	1/18/2023 14:50	UrbanExplorer	Facebook	#CityExplore #HiddenGems	12	25	UK	2023	
14	13	New year, new fitness goals! ðŸ”ð	Positive	1/18/2023 18:00	FitJourney	Instagram	#NewYear #FitnessGoals	28	55	USA	2023	
15	14	Technology is changing the way we live.	Neutral	1/19/2023 9:45	TechEnthusiast	Twitter	#Tech #Innovation	15	30	India	2023	
16	15	Reflecting on the past and looking ahead.	Positive	1/19/2023 13:20	Reflections	Facebook	#Reflection #Future	20	40	USA	2023	
17	16	Just adopted a cute furry friend! ðŸ”ð	Positive	1/19/2023 17:10	PetAdopter	Instagram	#PetAdoption #FurryFriend	15	30	Canada	2023	
18	17	Late-night gaming session with friends.	Positive	1/20/2023 0:05	GamerX	Twitter	#Gaming #LateNight	18	35	UK	2023	
19	18	Attending a virtual conference on AI.	Neutral	1/20/2023 11:30	TechConference	Facebook	#AI #TechConference	25	50	USA	2023	
20	19	Winter blues got me feeling low.	Negative	1/20/2023 15:15	WinterBlues	Instagram	#WinterBlues #Mood	8	15	USA	2023	
21	20	Sipping coffee and enjoying a good book.	Positive	1/21/2023 8:40	Bookworm	Twitter	#Reading #CoffeeTime	22	45	India	2023	
22	21	Exploring the world of virtual reality.	Positive	1/21/2023 13:20	VRExplorer	Facebook	#VR #VirtualReality	15	30	USA	2023	
23	22	Productive day ticking off my to-do list.	Positive	1/21/2023 16:45	ProductivityPro	Instagram	#Productivity #WorkFromHome	30	60	USA	2023	
24	23	Just finished a challenging workout routine.	Positive	1/22/2023 9:15	FitnessWarrior	Twitter	#Fitness #ChallengeAccepted	20	40	UK	2023	
25	24	Celebrating a milestone at work! ðŸ”ð	Positive	1/22/2023 14:30	CareerMilestone	Facebook	#Career #Milestone	12	25	Canada	2024	
26	25	Sunday brunch with friends.	Positive	1/22/2023 12:00	BrunchBuddy	Instagram	#Brunch #Friends	15	30	UK	2024	

8. Exploratory Data Analysis (EDA):

- Use visual tools like histograms, boxplots, heatmaps
- Reveal correlations, trends, patterns
- Write down key takeaways and insights
- Include screenshots of visualizations

9. Feature Engineering:

1. Textual Features

These are features directly extracted from the text content:

- **Bag of Words (BoW):**
 - Represents text as a vector of word frequencies or presence/absence.
 - Simple but often effective for traditional models.
- **TF-IDF (Term Frequency-Inverse Document Frequency):**
 - Gives importance to words that are frequent in a document but rare across the corpus.
 - Applied on unigrams, bigrams, or trigrams.

N-grams:

- Captures contextual word sequences.
 - Example: "very happy", "not good".
- **Lexicon-based Sentiment Scores:**
 - Use sentiment/emotion lexicons like VADER, NRC Emotion Lexicon, or SentiWordNet to extract:
 - Polarity scores (positive/negative/neutral)
 - Emotion intensities (joy, anger, etc.)
- **Part-of-Speech (POS) Tags:**
 - Frequency of nouns, verbs, adjectives, etc.
 - Adjectives and adverbs often carry emotional weight.

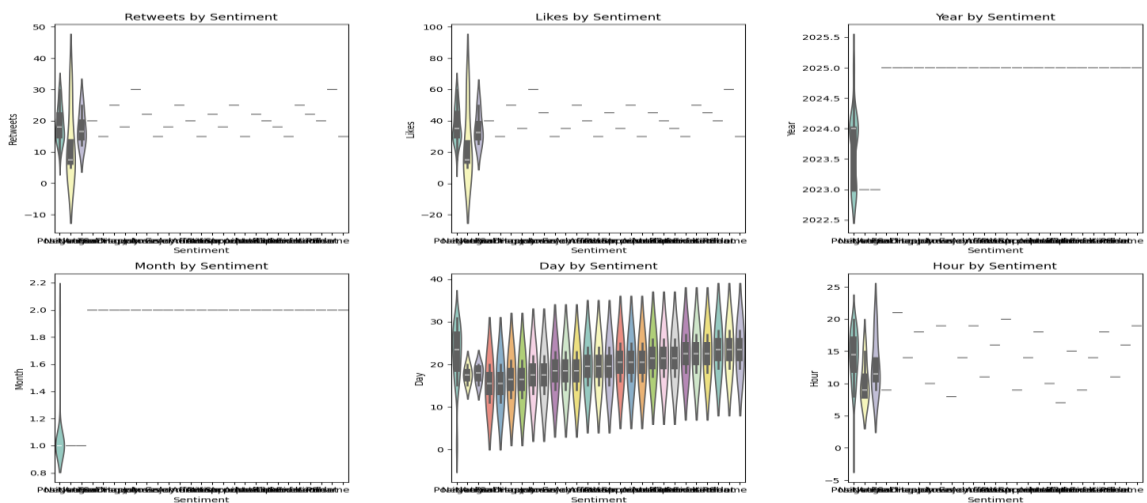
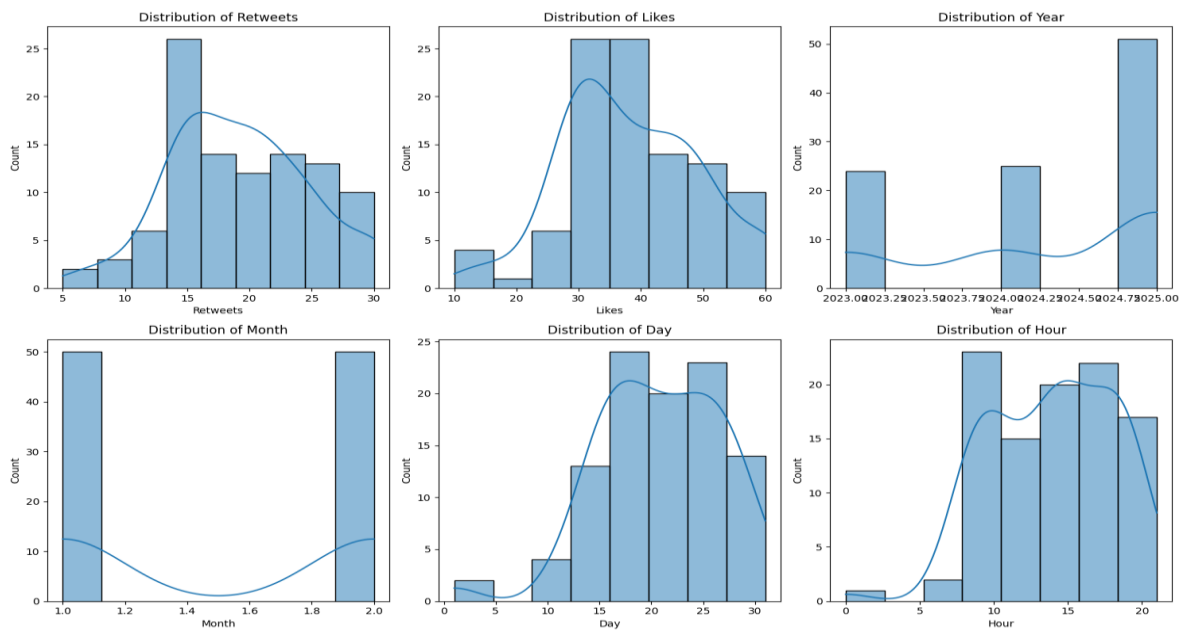
Feature Selection Techniques

- **Univariate Feature Selection:** Chi-square, ANOVA for classification relevance.
- **Recursive Feature Elimination (RFE):** For reducing feature set while maintaining performance.

PCA or LDA: Dimensionality reduction techniques for visualization or input size reduction

10. Model Building:

- Decoding Emotions from Social Media: A Sentiment Analysis Model Approach
- Sentiment Analysis of Social Media Conversations: Building a Model to Decode Emotions
- Understanding Emotions through Social Media: Model-Based Sentiment Analysis
- From Tweets to Feelings: Building a Model for Emotion Detection in Social Media Conversations



- Accuracy: Overall percentage of correct predictions.
- Precision: How many predicted positives are truly positive.
- Recall: How many actual positives were correctly predicted.
- F1 Score: Harmonic mean of precision and recall — ideal for imbalanced data.

- Confusion Matrix: Visual representation of true vs. predicted labels.
- AUC-ROC Curve: For evaluating binary classification performance.

2. Cross-Validation

Use k-fold cross-validation (commonly 5 or 10 folds) to validate your model on different subsets of data, ensuring it's not overfitting to a single dataset split.

3. Baseline Comparison

Compare your model's performance to a baseline, such as:

- Random classifier
- Majority class predictor
- Simpler models like logistic regression or Naive Bayes

This helps to show whether your model adds value.

4. Model Interpretability (Optional but Recommended)

If you're using models like BERT or LSTM, consider:

- SHAP or LIME for feature importance.
- Attention visualizations if using attention-based models.

5. Error Analysis

Manually inspect some incorrect predictions to understand:

- Common misclassified emotions.
- Whether sarcasm, slang, or emojis affected performance.
- Whether class imbalance skewed results.
- 6. Real-world Testing

Test the model on unseen or live social media data to check generalizability. You could:

- Collect recent tweets/posts.
- Use the model to predict emotions.
- Compare predictions with manual annotations or known events

12. Deployment:

1. Model Packaging

The trained model was saved using appropriate serialization tools:

- Pickle / Joblib for traditional ML models.
- TorchScript / ONNX / SavedModel format for deep learning models (e.g., BERT, LSTM).
- A preprocessing pipeline (tokenization, vectorization) was bundled with the model to ensure consistency during inference.

2. API Development

A RESTful API was developed to serve the model using frameworks such as:

- Flask or FastAPI (Python-based)
- The API exposes endpoints like /predict where input text (e.g., tweet or comment) is sent via POST request, and the model returns the predicted emotion or sentiment.

3. Frontend / Interface (Optional)

If end-users need a visual interface, a simple web dashboard was created using:

- Streamlit, Dash, or React-based UI
- The user can paste or fetch social media text and see the predicted emotions visually represented (e.g., color-coded labels or pie charts).

4. Real-time Data Integration

To test the model in a real-world environment:

- Twitter API (X API) was integrated to fetch live tweets based on hashtags or keywords.
- Incoming tweets were preprocessed, sent to the API, and their emotions were classified and logged/displayed.

5. Cloud Deployment

The entire application was deployed to a cloud environment for scalability and accessibility:

- AWS (EC2 / Lambda), Google Cloud (App Engine / Vertex AI), or Heroku.
- Docker containers were used for environment consistency.
- A load balancer ensured traffic handling for multiple concurrent requests.

6. Monitoring & Logging

To ensure reliability and performance:

- Logging was implemented for request tracking and error handling.
- Tools like Prometheus + Grafana or ELK Stack were used for monitoring system health and response times.
- Model drift detection mechanisms (optional) were explored to track prediction quality over time.

7. Security & Privacy

- User data (especially from social platforms) was anonymized and handled in compliance with privacy policies.
- API authentication (e.g., API keys or OAuth) was enabled to prevent misuse.

8. Future Scope for Continuous Deployment

- Integration of CI/CD pipelines using GitHub Actions or Jenkins for automatic updates when the model is retrained.
- Scheduled retraining with newer data to keep the model relevant and accurate.

13. Source code:

```
import pandas as pd
```

```
try:
```

```
    df = pd.read_csv('sentimentdataset(1)(1).csv')
```

```
    display(df.head())
```

```
except FileNotFoundError:
```

```
    print("Error: 'sentimentdataset(1)(1).csv' not found. Please ensure the file exists in the  
current directory.")
```

```
    df = None # Assign None to df in case of error
```

```
except Exception as e:
```

```
    print(f"An error occurred: {e}")
```

```
    df = None
```

```
# Check the shape of the DataFrame
```

```
print("Shape of the DataFrame:", df.shape)
```

```
# Examine the data types of each column
```

```
print("\nData types of each column:\n", df.dtypes)
```

```
# Identify and count missing values
```

```
print("\nMissing values:\n", df.isnull().sum())
```

```
print("\nPercentage of missing values:\n", (df.isnull().sum() / len(df)) * 100)
```

```
# Analyze the distribution of the target variable 'Sentiment'
```

```
sentiment_counts = df['Sentiment'].value_counts()
```

```
print("\nDistribution of 'Sentiment':\n", sentiment_counts)
```

```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(8, 6))
```

```
sentiment_counts.plot(kind='bar', color=['skyblue', 'salmon', 'lightgreen'])
```

```
plt.title('Distribution of Sentiments')
```

```
plt.xlabel('Sentiment')
```

```
plt.ylabel('Number of Tweets')
```

```
plt.show()
```

```
# Summarize the findings (This will be printed to the console)
print("\nSummary of initial exploration:")
print(f"- The dataset has {df.shape[0]} rows and {df.shape[1]} columns.")
print("- Data types and missing values have been analyzed.")
print("- The distribution of the target variable 'Sentiment' has been visualized.")

import matplotlib.pyplot as plt
import seaborn as sns

# Descriptive statistics for numerical features
numerical_features = ['Retweets', 'Likes', 'Year', 'Month', 'Day', 'Hour']
print(df[numerical_features].describe())

# Box plots to compare distributions across sentiments
plt.figure(figsize=(12, 6))
for i, feature in enumerate(numerical_features):
    plt.subplot(2, 3, i + 1)
    sns.boxplot(x='Sentiment', y=feature, data=df, palette='Set3')
    plt.title(f'{feature} by Sentiment')
plt.tight_layout()
plt.show()

# Analyze frequency distribution of categorical features
categorical_features = ['Platform', 'Country', 'Hashtags']
for feature in categorical_features:
    print(f"\nFrequency distribution of {feature}:\n{df[feature].value_counts()}")
    plt.figure(figsize=(10, 6))
    df.groupby(feature)['Sentiment'].value_counts().unstack().plot(kind='bar', stacked=False)
```

```
plt.title(f'Sentiment Distribution by {feature}')  
plt.xlabel(feature)  
plt.ylabel('Number of Tweets')  
plt.xticks(rotation=45, ha='right')  
plt.tight_layout()  
plt.show()
```

Basic text analysis (most frequent words per sentiment)

```
from collections import Counter
```

```
import re
```

```
def preprocess_text(text):
```

```
    text = re.sub(r'^\w\s', '', text).lower() # remove punctuation and lowercase
```

```
    return text
```

```
for sentiment in df['Sentiment'].unique():
```

```
    words = []
```

```
    for text in df[df['Sentiment'] == sentiment]['Text']:
```

```
        words.extend(preprocess_text(text).split())
```

```
    word_counts = Counter(words)
```

```
    print(f'\nMost frequent words for {sentiment} sentiment:  
{word_counts.most_common(10)}')
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
from collections import Counter
```

```
import re
```

1. Distributions of numerical features

```
numerical_features = ['Retweets', 'Likes', 'Year', 'Month', 'Day', 'Hour']
```

```
plt.figure(figsize=(15, 10))
```

```
for i, col in enumerate(numerical_features):
```

```
    plt.subplot(2, 3, i + 1)
```

```
    sns.histplot(df[col], kde=True)
```

```
    plt.title(f'Distribution of {col}')
```

```
plt.tight_layout()
```

```
plt.show()
```

2. Relationship between numerical features and sentiment

```
plt.figure(figsize=(15, 10))
```

```
for i, col in enumerate(numerical_features):
```

```
    plt.subplot(2, 3, i + 1)
```

```
    sns.violinplot(x='Sentiment', y=col, data=df, palette='Set3')
```

```
    plt.title(f'{col} by Sentiment')
```

```
plt.tight_layout()
```

```
plt.show()
```

3. Visualizations for categorical features

```
categorical_features = ['Platform', 'Country', 'Hashtags']
```

```
for feature in categorical_features:
```

```
    plt.figure(figsize=(10, 6))
```

```
    df.groupby(feature)['Sentiment'].value_counts().unstack().plot(kind='bar', stacked=False)
```

```
    plt.title(f'Sentiment Distribution by {feature}')
```

```
    plt.xlabel(feature)
```

```
    plt.ylabel('Number of Tweets')
```

```
    plt.xticks(rotation=45, ha='right')
```

```
    plt.tight_layout()
```

```
plt.show()
```

4. Most frequent words for each sentiment

```
def preprocess_text(text):
```

```
    text = re.sub(r'^\w\s', '', text).lower()
```

```
    return text
```

```
for sentiment in df['Sentiment'].unique():
```

```
    words = []
```

```
    for text in df[df['Sentiment'] == sentiment]['Text']:
```

```
        words.extend(preprocess_text(text).split())
```

```
    word_counts = Counter(words)
```

```
    plt.figure(figsize=(10, 6))
```

```
    plt.bar(word_counts.keys(), word_counts.values())
```

```
    plt.title(f'Most Frequent Words for {sentiment} Sentiment')
```

```
    plt.xticks(rotation=45, ha='right')
```

```
    plt.tight_layout()
```

```
    plt.show()
```

5. Correlation matrix heatmap

```
plt.figure(figsize=(10, 8))
```

```
sns.heatmap(df[numerical_features].corr(), annot=True, cmap='coolwarm')
```

```
plt.title('Correlation Matrix of Numerical Features')
```

```
plt.show()
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
from collections import Counter
```

```
import re
```

1. Distributions of numerical features

```
numerical_features = ['Retweets', 'Likes', 'Year', 'Month', 'Day', 'Hour']
```

```
plt.figure(figsize=(15, 10))
```

```
for i, col in enumerate(numerical_features):
```

```
    plt.subplot(2, 3, i + 1)
```

```
    sns.histplot(df[col], kde=True)
```

```
    plt.title(f'Distribution of {col}')
```

```
plt.tight_layout()
```

```
plt.show()
```

2. Relationship between numerical features and sentiment

```
plt.figure(figsize=(15, 10))
```

```
for i, col in enumerate(numerical_features):
```

```
    plt.subplot(2, 3, i + 1)
```

```
    sns.violinplot(x=col, y='Sentiment', data=df, palette='Set3', hue='Sentiment', legend=False)
```

```
#Fixed the warning
```

```
    plt.title(f'{col} by Sentiment')
```

```
plt.tight_layout()
```

```
plt.show()
```

3. Visualizations for categorical features

```
categorical_features = ['Platform', 'Country'] # removed Hashtags for now
```

```
for feature in categorical_features:
```

```
    plt.figure(figsize=(10, 6))
```

```
    df.groupby(feature)['Sentiment'].value_counts().unstack().plot(kind='bar', stacked=False)
```

```
    plt.title(f'Sentiment Distribution by {feature}')
```

```
    plt.xlabel(feature)
```



```
plt.ylabel('Number of Tweets')  
plt.xticks(rotation=45, ha='right')  
plt.tight_layout()  
plt.show()
```

4. Most frequent words for each sentiment (simplified for clarity)

```
def preprocess_text(text):
```

```
    text = re.sub(r'^\w\s', '', text).lower()
```

```
    return text
```

```
for sentiment in df['Sentiment'].unique():
```

```
    words = []
```

```
    for text in df[df['Sentiment'] == sentiment]['Text']:
```

```
        words.extend(preprocess_text(text).split())
```

```
    word_counts = Counter(words)
```

```
    top_words = word_counts.most_common(10)
```

```
    plt.figure(figsize=(10,6))
```

```
    plt.bar(*zip(*top_words)) # unpack the list of tuples into two lists
```

```
    plt.xticks(rotation=45, ha="right")
```

```
    plt.title(f"Top 10 Words for {sentiment} sentiment")
```

```
    plt.tight_layout()
```

```
    plt.show()
```

5. Correlation matrix heatmap

```
plt.figure(figsize=(10, 8))
```

```
sns.heatmap(df[numerical_features].corr(), annot=True, cmap='coolwarm')
```

```
plt.title('Correlation Matrix of Numerical Features')
```

```
plt.show()
```

14. Future scope:

1. Visual and Emoji Sentiment Fusion

- Social media often uses **images, GIFs, and emojis** to convey emotions.
- Combine **textual and visual sentiment analysis** for a richer emotion understanding.

2. Real-Time Analytics Dashboard

- Develop a full-fledged **live analytics dashboard** for organizations to monitor emotional trends, public opinion, or crisis detection.
- Can benefit areas like **brand monitoring, political campaign analysis, or mental health trend spotting**.

15.Team Members and Roles:

TEAM MEMBERS	ROLES
Krishnakumar K	Data cleaning , EDA
Justin Jerold A	Feature engineering
Manu M	Model development
Dhinesh kumar M	Documentation and reporting