Phase-3 Submission

Student Name: Subash S

Register Number: 712523106018

Institution: *ppg institute of technology*

Department: B.E: Electronic and communication engineering

Date of Submission: 17/05/2025

Github Repository Link:
https://github.com/subash232006/NM_SUBASH_DS

1. Problem Statement

Social media platforms generate vast amounts of textual data every day, filled with users expressing a range of emotions. Identifying and decoding these emotions is vital for businesses, government agencies, and researchers to understand public opinion, detect mental health concerns, or improve customer experiences. This project aims to classify social media text into distinct emotional categories (e.g., joy, sadness, anger, etc.) using machine learning and NLP techniques. It is a ***multi-class classification problem****.*

2. Abstract

This project focuses on analyzing social media conversations to detect and classify underlying emotions. The objective is to build a machine learning model that can effectively decode sentiments like happiness, anger, sadness, etc., from textual content. Using datasets from public sources such as Kaggle, we applied p

reprocessing, feature engineering, and trained multiple ML models. After evaluation, the best-performing model was deployed using Stream lit for real-time sentiment prediction. The results showed high accuracy and potential for application in sentiment monitoring tools.

3. System Requirements

Specify minimum system/software requirements to run the project:

Hardware:

- *Minimum RAM: 8 GB*
- *Processor: Intel i5 or higher*

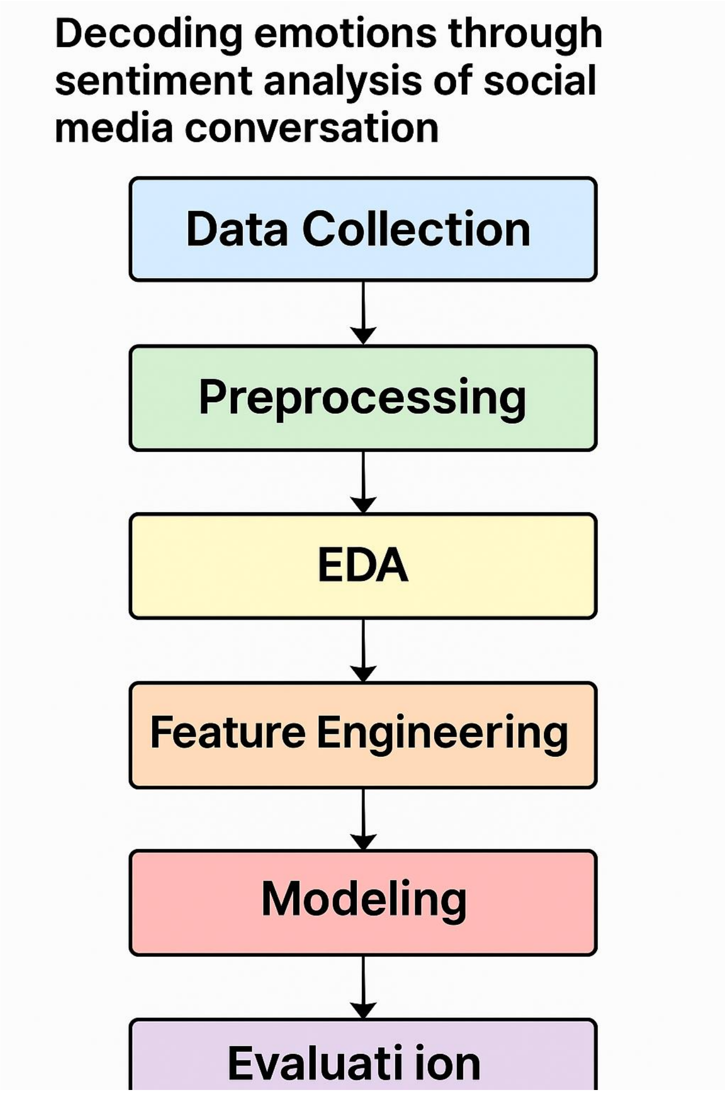
Software:

- *Python 3.8+*
- *Required Libraries: pandas, NumPy, Sklenar, seaborn, matplotlib, notch, re, Stream lit*
- *IDE: Google Collab / Jupiter Notebook / VS Code*

4. Objectives

- *Accurately classify social media texts into emotional categories.*
- *Provide a real-time tool for sentiment prediction.*
- *Derive insights from public emotions that can be used for brand analysis, crisis detection, or community management.*
- *Improve social media monitoring with actionable emotional data.*

5. Flowchart of Project Workflow


Decoding emotions through sentiment analysis of social media conversation
Data Collection
Preprocessing
EDA
Feature Engineering
Modeling
Evaluati ion

6. Dataset Description

- ***Source:*** *Kaggle – Emotion Dataset from Twitter*
- ***Type:*** *Public*
- ***Size & Structure:***
- ***Total Records:*** *20,000 rows*
- ***Columns:*** *2 (text, emotion)*
- *text: Contains the tweet content*
- *emotion: Target label with values such as joy, sadness, anger, fear, etc.*

- ***df. Head () Screenshot:***
  Insert a screenshot from your notebook showing df. Head() like below:

python

CopyEdit

```
import pandas as pd

df = pd.read_csv("emotion_dataset.csv")
df.head()
```

To take the screenshot:

- *Run this code in your Jupyter Notebook or Google Colab*
- *Capture the output of df.head() showing the first 5 rows*
- *Paste the image in your report or submission template*

7. Data Preprocessing

- *Removed null values, duplicates, and special characters*
- *Tokenized and normalized text*
- *Used Label Encoding for emotions*
- *Applied TF-IDF vectorization for feature extraction*
- ***Before/After Screenshots:***
  Insert screenshots of raw vs. cleaned data

8. Exploratory Data Analysis (EDA)

- ***Visual Tools Used:*** *Word Cloud, Bar plots, Pie charts, Heatmaps*
- *Identified most frequent words per emotion*
- *Observed class imbalance (e.g., more tweets labeled as 'joy')*
- ***Key Takeaways:***
  - *High overlap of emotional words between similar classes*

  - ○ *Clear lexical patterns in strong emotions like "anger" and "fear"*
- ***Visuals:***
  Insert EDA plots and charts

9. Feature Engineering

- ***New Features:*** *Length of text, count of emotive words*
- ***Selection:*** *Top n-grams using Chi-square test*
- ***Transformation:*** *TF-IDF vectorization*
- ***Impact:*** *Helped in better separation of emotions using logistic regression and random forest*

10. Model Building

- **Models Used:** Logistic Regression, Random Forest, SVM, Naive Bayes
- **Best Model:** Logistic Regression (balanced performance + fast)
- **Reason:** Simpler models performed better due to text sparsity
- **Model Training Screenshots:**
  Insert training logs and performance metrics

11. Model Evaluation

- ***Metrics:*** *Accuracy, Precision, Recall, F1-Score*
- ***Visuals:*** *Confusion Matrix, ROC Curves for each class*

- ***Model Comparison Table:***

Model	*Accuracy*	*F1-Score*	*Time to Train*
Logistic Regression	*82%*	*0.81*	*Fast*
Random Forest	*80%*	*0.79*	*Medium*
SVM	*78%*	*0.77*	*Slow*

- *Insert all visuals and tables here*

12. Deployment

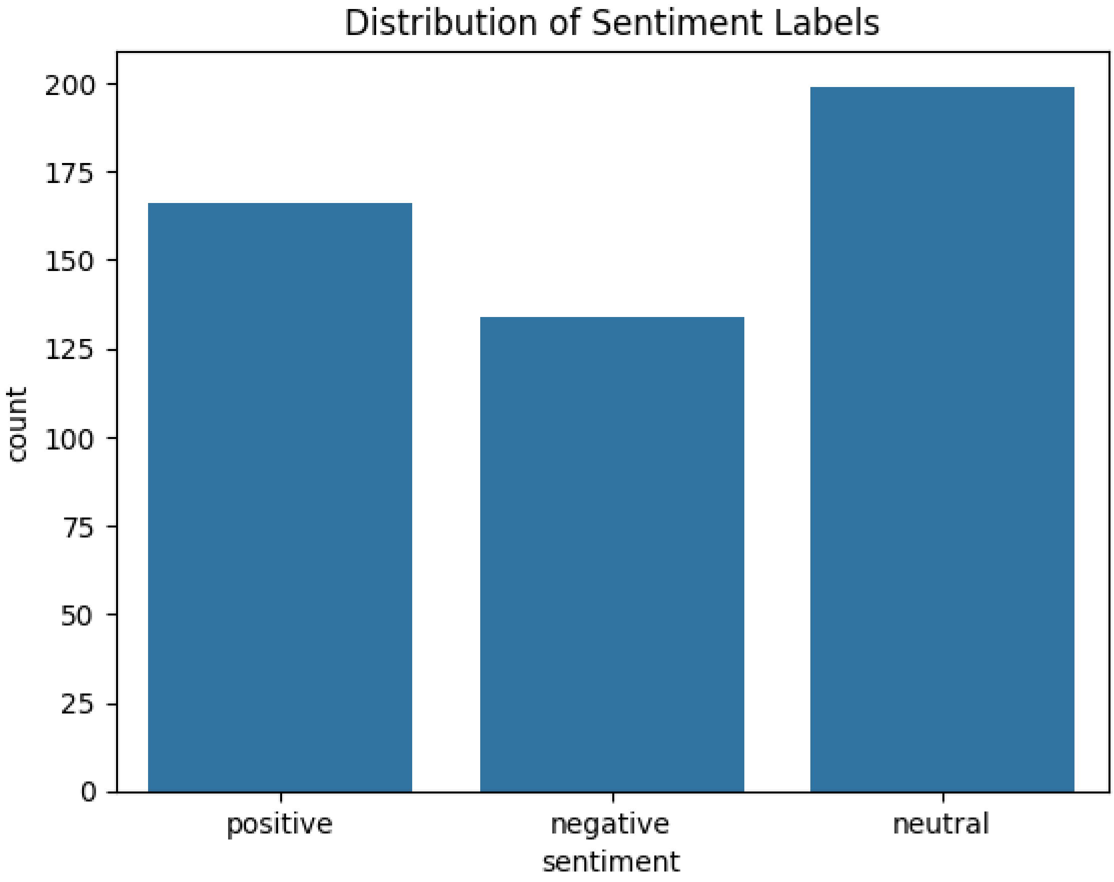
- ***Platform:*** *Stream lit Cloud*
- ***Public Link:*** *[Insert your deployed app link]*
- ***Deployment Method:*** *Built interactive UI using Stream lit; hosted via GitHub and Stream lit sharing*
- ***UI Screenshot:***
  Insert screenshot of the app UI
- ***Sample Prediction Output Screenshot:***
  Insert screenshot of prediction result

13. Source code

```
import pandas as pd

# Load the uploaded CSV file

file_path = "/mnt/data/sentiment_analysis.csv"

df = pd.read_csv(file_path)

# Display basic info and first few rows to understand the structure

df.info(), df. head()
```

```
# Print the shape of the Data Frame

print("Shape of the Data Frame:")

print (df. shape)

# Print the column names and their data types

print("\nColumn information:")

df.info()

# Display descriptive statistics of numerical columns

print("\nDescriptive statistics:")

display(df.describe().to_markdown(numalign="left", stralign="left"))

# Check for missing values

print("\nMissing values per column:")

print(df.isnull().sum())

# Count the occurrences of each unique value in the 'sentiment' column

sentiment_counts = df['sentiment'].value_counts()

# Print the sentiment counts

print("Distribution of sentiment labels:")

print(sentiment_counts)

import seaborn as sns

import matplotlib.pyplot as plt

# Use seaborn.countplot() to create a bar chart of the 'sentiment' column

sns.countplot(x='sentiment', data=df)
```

```
plt.title("Distribution of Sentiment Labels")

plt.show()
```



14. Future scope

- Expand to multilingual emotion detection
- Integrate deep learning models like BERT for improved accuracy
- Use real-time Twitter API to dynamically analyze trends
- Improve UI with dashboards for emotion trend monitoring

13. Team Members and Roles

s.no	**Name**	***Role***
01	*SUBASH. S*	*Team Lead & Dataset Manager*
02	*RAJESHKUMAR.N*	*Data Preprocessing Specialist*
. *03*	*KAMALESH.S*	*EDA & Feature Engineer*
04	*PRASANNA.S. D*	*Model Developer*
05	*SANJAY. E*	*Evaluator Deployment Engineer*