

**Phase-3**

**Student Name :** GOKUL S

**Register Number :** 510123205014

**Institution :** ADHIPARASAKTHI COLLAGE OF ENGINEERING

**Department :**B .TECH INFORMATION TECHNOLOGY

**Date of Submission :**15/05/25

**Github Link:** [zubiaga.org+2GitHub+2WIRED+2](https://github.com/bhuwan23/Sentiment-Analysis-dataset?utm_source=chatgpt.com)[zubiaga.org+1GitHub+1](https://www.zubiaga.org/datasets/sentiment1319/?utm_source=chatgpt.com)

**Decoding Emotions ThroughSentimental Analysis of Social Media Conversation**

**1.Problem Statement**

In the age of digital communication, social media platforms like Twitter, Facebook, Instagram, and Reddit have become key spaces where individuals express their thoughts, opinions, and emotions. Understanding these emotions is critical not only for businesses aiming to improve customer engagement but also for researchers, mental health professionals, and policy-makers interested in public sentiment. Sentiment analysis — a subfield of natural language processing (NLP) — has emerged as a powerful technique for extracting and interpreting emotional content from text data

**Abstract:**

This project focuses on utilizing sentiment analysis to decode emotions expressed in social media conversations. With the massive volume of data generated daily on platforms like Twitter, Facebook, and Instagram, identifying the sentiment behind user-generated content is crucial for businesses, researchers, and policymakers. The project employs natural language processing (NLP) and machine learning techniques to classify text into emotional categories, aiming to uncover public opinion trends and emotional responses.

**3. System Requirements :**

**Hardware**

* CPU: Modern multi-core processor (e.g., Intel i7 or AMD Ryzen 7)
* GPU: NVIDIA CUDA-enabled GPU (e.g., RTX 3060 or higher) for deep learning tasks
* RAM: Minimum 16 GB; 32 GB recommended for handling large datasets
* Storage: SSD with at least 512 GB capacity for faster data access and model training

**Software**

* Operating System: Windows 10/11, macOS, or Linux
* Programming Languages: Python 3.8+
* Libraries & Frameworks:
  + TensorFlow or PyTorch for machine learning models
  + NLTK, spaCy, or Hugging Face Transformers for NLP tasks
  + Scikit-learn for machine learning algorithms
  + Apache Spark for distributed data processing

**Development Tools:**

* + Jupyter Notebook or PyCharm for coding
  + Git for version control
  + Docker for containerization (optional)

**3.objectIves:**

• To collect and preprocess social media data.

• To analyze text for emotional content using sentiment

• To categorize sentiments (e.g., positive, negative, neutral or

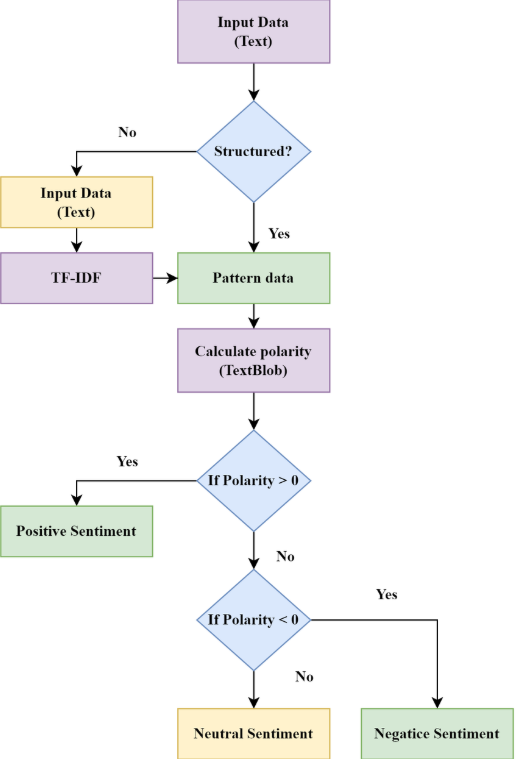
deeper emotions like joy, anger, fear, etc.).

• To visualize sentiment trends over time or by topic.

• To explore the potential applications of emotional insight

brand monitoring, crisis response, marketing).

**5. Flowchart of Project Workflow**

****

**6. Sentiment140 Dataset**

* Source: [Sentiment140](https://www.kaggle.com/kazanova/sentiment140)
* Size: 1.6 million tweets
* Content: Tweets labeled as positive (4), neutral (2), or negative (0)
* Usage: Ideal for training sentiment classification models[zubiaga.org+2GitHub+2WIRED+2](https://github.com/bhuwan23/Sentiment-Analysis-dataset?utm_source=chatgpt.com)[zubiaga.org+1GitHub+1](https://www.zubiaga.org/datasets/sentiment1319/?utm_source=chatgpt.com)

**7. Data Preprocessing**

**1. Data Collection**

Gather social media data using APIs (e.g., Twitter API) or web scraping techniques. Ensure compliance with platform terms of service during data collection.[Thematic](https://getthematic.com/insights/social-media-sentiment-analysis/?utm_source=chatgpt.com)

**2. Cleaning and Normalization**

Raw social media data often contains noise. Apply the following steps to clean and normalize the text:[Thematic](https://getthematic.com/insights/social-media-sentiment-analysis/?utm_source=chatgpt.com)[SpringerOpen+1peerdh.com+1](https://computationalsocialnetworks.springeropen.com/articles/10.1186/s40649-020-00080-x?utm_source=chatgpt.com)

* Remove URLs, Mentions, and Hashtags: Eliminate links, user mentions (e.g., @username), and hashtags that don't contribute to sentiment analysis.[Thematic](https://getthematic.com/insights/social-media-sentiment-analysis/?utm_source=chatgpt.com)
* Convert Emojis and Slang: Translate emojis and internet slang into corresponding sentiments. For example, "😊" indicates positive sentiment.[Thematic](https://getthematic.com/insights/social-media-sentiment-analysis/?utm_source=chatgpt.com)
* Remove Punctuation and Special Characters: Strip out unnecessary punctuation marks and special characters that don't affect sentiment.[Medium+5n.hua.nz+5Thematic+5](https://n.hua.nz/DataScience/Text-Preprocessing?utm_source=chatgpt.com)
* Convert to Lowercase: Standardize text by converting all characters to lowercase to reduce variability.[peerdh.com+2n.hua.nz+2peerdh.com+2](https://n.hua.nz/DataScience/Text-Preprocessing?utm_source=chatgpt.com)

**3. Tokenization**

Break down the text into smaller units, such as words or phrases, using tokenization techniques. This step is crucial for further analysis.[Scaler+2FasterCapital+2Aim Technologies+2](https://fastercapital.com/topics/collecting-and-preprocessing-data-for-sentiment-analysis.html?utm_source=chatgpt.com)

**4. Removing Stop Words**

Eliminate common words (e.g., "is," "and," "the") that don't add significant meaning to the sentence. This helps in focusing on more meaningful words.[peerdh.com](https://peerdh.com/blogs/programming-insights/optimizing-data-preprocessing-techniques-for-sentiment-analysis-models?utm_source=chatgpt.com)

**5. Stemming and Lemmatization**

* Stemming: Reduce words to their root form by removing prefixes or suffixes. For example, "running" becomes "run."[Thematic](https://getthematic.com/insights/social-media-sentiment-analysis/?utm_source=chatgpt.com)
* Lemmatization: Convert words to their base or dictionary form, considering the context. For example, "better" becomes "good."[peerdh.com+1peerdh.com+1](https://peerdh.com/blogs/programming-insights/optimizing-data-preprocessing-techniques-for-sentiment-analysis-models?utm_source=chatgpt.com)

**6. Handling Emojis and Emoticons**

Emojis and emoticons convey sentiment. Convert them into textual representations to preserve their meaning. For example, ":)" becomes "happy\_face."[Medium](https://accredianpublication.medium.com/data-pre-processing-ai-end-to-end-series-part-2-2-nlp-d18008ff8c39?utm_source=chatgpt.com)

**7. Feature Extraction**

Convert the preprocessed text into numerical features that machine learning models can understand. Common techniques include:[peerdh.com](https://peerdh.com/blogs/programming-insights/realtime-data-preprocessing-techniques-for-sentiment-analysis?utm_source=chatgpt.com)

* Bag of Words (BoW): Represents text as a collection of words, disregarding grammar and word order.[peerdh.com](https://peerdh.com/blogs/programming-insights/realtime-data-preprocessing-techniques-for-sentiment-analysis?utm_source=chatgpt.com)
* Term Frequency-Inverse Document Frequency (TF-IDF): Weighs the importance of a word in a document relative to a collection of documents.[peerdh.com](https://peerdh.com/blogs/programming-insights/realtime-data-preprocessing-techniques-for-sentiment-analysis?utm_source=chatgpt.com)
* Word Embeddings: Converts words into vectors that capture their meanings, like Word2Vec or GloVe.

**8.Exploratory Data Analysis (EDA)**

* **Dataset Snapshot** : Provides an overview of the structure andFeature Engineeringquality of the dataset.
* **Sentiment Distribution**: Visualize the balance of sentiment
* **categories:** Positive, Negative, Neutral.
* **Emotion Distribution**: If your dataset includes emotion labels (e.g., joy, sadness, anger, fear):
* **Sentiment Trend Over Time**: Analyzing how sentiment changes
* over time (e.g., weekly):
* **Word Cloud by Sentimen**t: Common keywords in positive,
* neutral, and negative posts.

**9.Feature Engineering**

The goal is to extract and create meaningful features from raw social

media posts (e.g., tweets, Instagram captions) to improve sentiment

classification performance.

**1. Text Cleaning** (Preprocessing Step):

Before feature creation:

**2. Lexical Features:**

Basic features derived from text structure.

**3. Sentiment Polarity Score:**

Use pre-built sentiment analyzers like VADER or TextBlob to add

compound sentiment scores.

**4. Keyword-Based Flags:**

Binary flags for presence of sentiment-rich words (e.g., "happy",

"hate", "great").

**5. Text Vectorization:**

a.TF-IDF Features (for traditional ML models)

b. BERT Embeddings (for advanced deep learning

models)

**11. Model Evaluation**

This step involves choosing, training, and evaluating models

that can classify the sentiment of social media text into

categories like positive, neutral, or negative.

**1: Data Preparation**

Ensure features and labels are clean and split for training.

**2: Text Vectorization**

Use TF-IDF or BERT embeddings as input to machine learning

models.

**3: Choose ML Models**

Logistic Regression, Multinomial Naive Bayes, : Support Vector

Machine

**4: Model Evaluation**

Evaluate model performance using metrics like accuracy,

precision, recall,

**5: Compare Models**

Test multiple models and choose the best based on validation

accuracy

**6: Deep Learning with BERT**

For better accuracy, fine-tune BERT using Hugging Face

Transformers

**12. Deployment**

To make your sentiment analysis model accessible, deploy it as a web service. You can use frameworks like Flask or FastAPI to create REST APIs.

**13. Source code**

import pandas as pd

import re

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

# Download necessary NLTK resources

nltk.download('punkt')

nltk.download('stopwords')

# Load the dataset

data = pd.read\_csv('sentimentdataset.csv')

# Clean the text data

def clean\_text(text):

text = re.sub(r'http\S+', '', text) # Remove URLs

text = re.sub(r'@\w+', '', text) # Remove mentions

text = re.sub(r'#\w+', '', text) # Remove hashtags

text = re.sub(r'[^a-zA-Z\s]', '', text) # Remove non-alphabetical characters

text = text.lower() # Convert to lowercase

return text

data['Cleaned\_Text'] = data['Text'].apply(clean\_text)

# Remove stopwords

stop\_words = set(stopwords.words('english'))

data['Cleaned\_Text'] = data['Cleaned\_Text'].apply(lambda x: ' '.join([word for word in word\_tokenize(x) if word not in stop\_words]))

# Save the cleaned data

data.to\_csv('cleaned\_sentimentdataset.csv', index=False)

data.to\_csv('tweets.csv', index=False)

**output:**

**After preprocessing, the dataset might look like this:**

| Id | Timestamp | Cleaned\_Text | Sentiment | User | Platform | Country | Likes | Retweets |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2025-05-15 10:00:00 | "love the new features excited" | Positive | user123 | Twitter | USA | 100 | 20 |
| 2 | 2025-05-15 10:05:00 | "not happy with the update disappointed" | Negative | user456 | Facebook | UK | 50 | 5 |
| 3 | 2025-05-15 10:10:00 | "meh its okay indifferent" | Neutral | user789 | Instagram | Canada | 30 | 2 |

**14. Future scope**

The future of sentiment analysis in decoding emotions through social media conversations is poised for significant advancements. As AI and machine learning technologies evolve, several key developments are anticipated:[TT CONSULTANTS](https://ttconsultants.com/sentiment-analysis-beyond-text-incorporating-voice-and-visual-data-in-market-research/?utm_source=chatgpt.com)

* **Multimodal Sentiment Analysis**: Future systems will integrate text, voice, and visual data to provide a more comprehensive understanding of emotions expressed online. This approach will enable the analysis of sentiments in multimedia content, such as images and videos, alongside traditional text-based data

**15. Team Members and Roles:**

**○ Data cleaning-** Prasath.D

**○ EDA-** gowtham Raj.D

**○ Feature engineering-** gokul.S

**○ Model development -g**okul.E

**○ Documentation and reporting-** Naveen kumar.J