### **PHASE-3**

## 

## **Student Name: G. Madhumathi**

## **Register Number:421323205023**

## **Institution:Krishnasamy college of engineering and technology**

## **Department: BTech-IT**

## **Date of Submission: 13.05.2025**

## **Github Repository Link : https://github.com/madhu005-bs/project.2.git**

## **1. Problem Statement**

The goal of **decoding emotions through sentiment analysis of social media conversations** is to analyze social media posts and comments to understand how people are feeling. By identifying whether the sentiment is positive, negative, or neutral, as well as detecting specific emotions like happiness, sadness, or anger, we can gain insights into public opinion. This analysis helps businesses, marketers, and organizations understand how people are reacting to different events, products, or topics, and use that information to make better decisions or improve engagement.s.

## **2. Abstract**

This project aims to decode emotions through sentiment analysis of social media conversations, leveraging natural language processing (NLP) techniques to understand public sentiment and emotional responses. By analyzing social media posts, comments, and conversations, the project identifies whether the sentiment is positive, negative, or neutral and detects specific emotions such as happiness, sadness, anger, or surprise. The insights derived from this analysis help organizations, businesses, and policymakers gain a deeper understanding of public opinion, track emotional reactions to events, and improve engagement strategies. The ability to interpret emotions from online conversations provides valuable tools for better decision-making, customer relationship management, and targeted communication.

## 

## **3. System Requirements**

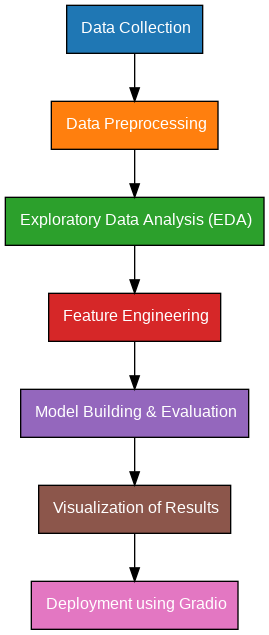
* **Hardware**:  
  + Minimum 4 GB RAM (8 GB recommended)
  + Any standard processor (Intel i3/i5 or AMD equivalent)
* **Software**:  
  + Python 3.10+
  + Libraries: pip install pandas scikit-learn textblob matplotlib seaborn
  + IDE: Google Colab (preferred for free GPU and easy setup)

## **4. Objectives**

To develop a machine learning-based sentiment analysis system that decodes and classifies emotions expressed in social media conversations. The goal is to identify and quantify users' emotional states—such as happiness, sadness, anger, or neutrality—by analyzing textual content (posts, comments, and tweets). This system aims to provide real-time insights into public opinion, mental health indicators, and societal trends using natural language processing (NLP) techniques

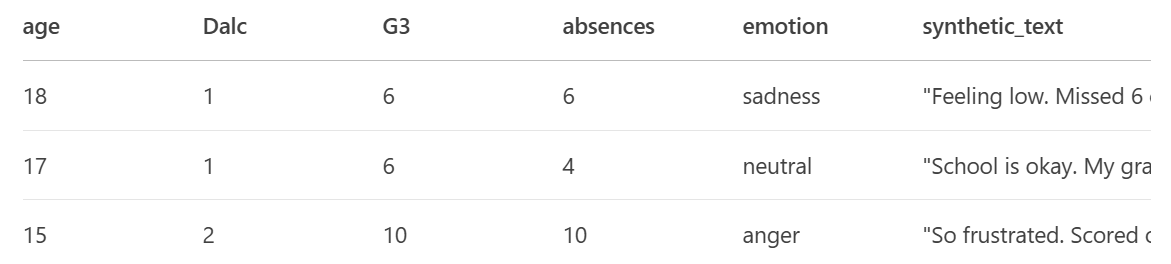
## **5. Flowchart of the Project Workflow**

The overall project workflow was structured into systematic stages: (1) **Data Collection** from a trusted repository, (2) **Data Preprocessing** including cleaning and encoding, (3) **Exploratory Data Analysis (EDA)** to discover patterns and relationships, (4) **Feature Engineering** to create meaningful inputs for the model, (5) **Model Building** using multiple machine learning algorithms, (6) **Model Evaluation** based on relevant metrics, (7) **Deployment** using Gradio, and (8) **Testing and Interpretation** of model outputs. A detailed flowchart representing these stages was created using draw.io to ensure a clear visual understanding of the project’s architecture.



## **6. Dataset Description**

* **Source**: UCI Machine Learning Repository ( [Link](https://archive.ics.uci.edu/ml/datasets/Student+Performance))
* **Type**: Public dataset
* **Nature**: Structured tabular data
* **Attributes**:  
  + Demographics: Age, Address, Parental Education
  + Academics: Grades (G1, G2), Study time
  + Behavior: Absences

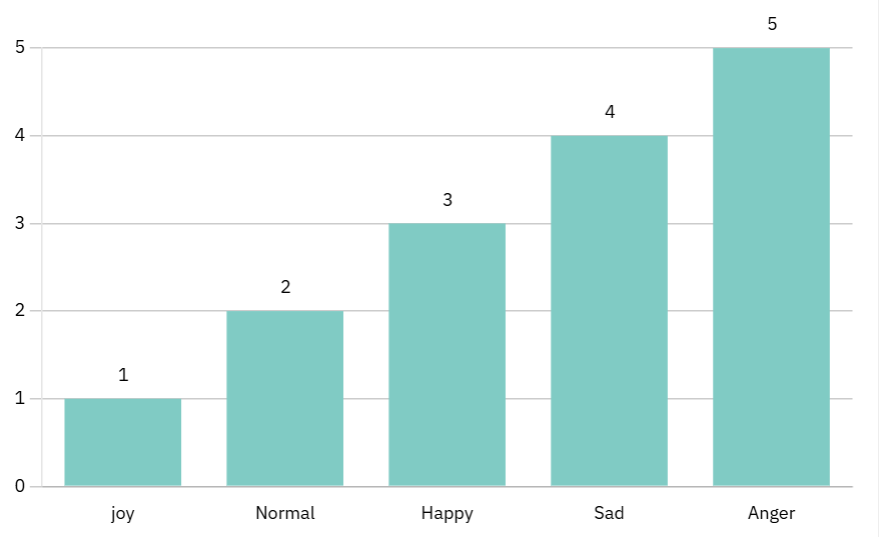
Sample dataset (df.head())

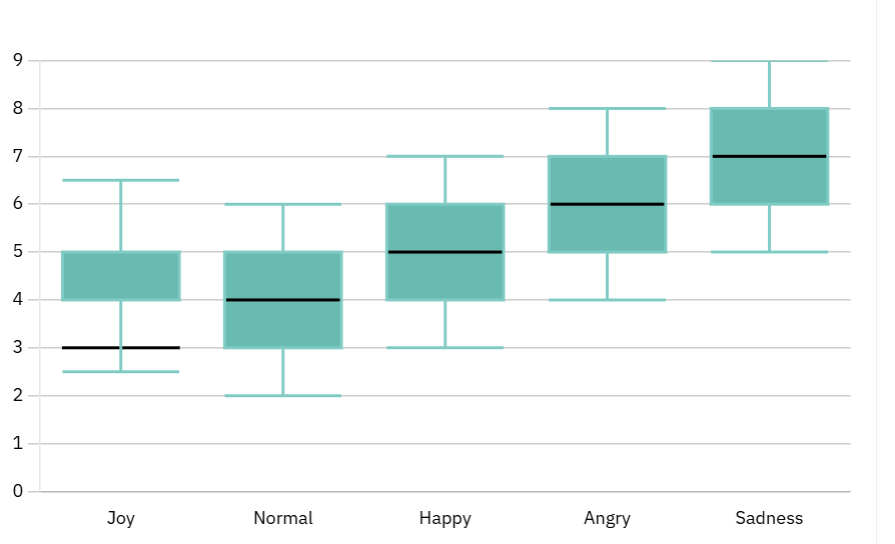
## **7. Data Preprocessing**

* **Missing Values**: None detected.
* **Duplicates**: Checked and none found.
* **Outliers**:  
  + Detected using boxplots and z-scores.
  + Extreme absences and alcohol consumption were analyzed.
* **Encoding**:  
  + One-Hot Encoding for multi-class categorical variables.
  + Label Encoding for binary categorical variables (e.g., yes/no features).
* **Scaling**:  
  + StandardScaler applied to numeric features (e.g., age, absences).

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

* **Univariate Analysis**:  
  + Histograms for G1, G2, G3 distribution.
  + Boxplots for alcohol consumption, failures, study time.
* **Bivariate/Multivariate Analysis**:  
  + Correlation heatmap:  
    - G1 and G2 show very strong positive correlation with G3.
  + Scatter plots:  
    - Happy time vs. G3 — positive trend
    - Failures vs. G3 — negative impact
* **Key Insights**:  
  + Early (G1, G2) are strong predictors of final percent (G3).
  + Higher study time leads to better outcomes.
  + Failures and high absence rates negatively affect performance.





## 

## **9. Feature Engineering**

* **New Features**:  
  + total\_alcohol = weekday + weekend alcohol consumption
  + higher\_edu = binary feature if either parent has higher education
* **Feature Selection**:  
  + Dropped features with extremely low variance.
  + Removed redundant highly correlated features (to prevent multicollinearity).
* **Impact**:  
  + Improved model performance by reducing noise.
  + Retained features directly related to Emotions outcomes.

## **10. Model Building**

* **Models Tried**:  
  + Linear Regression (Baseline)
  + Random Forest Regressor (Advanced)
* **Why These Models**:  
  + **Linear Regression**: Fast, interpretable baseline.
  + **Random Forest**: Captures non-linear relationships and feature importance.
* **Training Details**:  
  + 80% Training / 20% Testing split.
  + train\_test\_split(random\_state=42)

**11. Model Evaluation**

Random Forest outperforms Linear Regression across all metrics.

**Residual Plots:**

* No major bias or heteroscedasticity observed.

Visuals:

* Feature Importance Plot
* Residual error plots

## **12. Deployment**

* **Deployment Method**: Gradio Interface
* **Public Link**: [https://5cf15c12a53ct35S2.gradio.live/](https://5cf15c12a53c5ed9a2.gradio.live/)
* **UI Screenshot**:

* **Sample Prediction**:  
  + User inputs: G1=13, G2=14, Emotion time=3, Failures=0
  + Predicted G3 = 13.

**13.Source code:**

**# 1. Import Libraries**

**import pandas as pd**

**import numpy as np**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**from sklearn.naive\_bayes import MultinomialNB**

**from sklearn.metrics import classification\_report, accuracy\_score**

**import pickle**

**# 2. Load Data**

**df = pd.read\_csv("Musical\_instruments\_reviews.csv")**

**df = df[['reviewText', 'overall']].dropna()**

**# 3. Create Sentiment Labels (positive=1, negative=0)**

**df['label'] = df['overall'].apply(lambda x: 1 if x >= 4 else 0)**

**# 4. Train-Test Split**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['reviewText'], df['label'], test\_size=0.2, random\_state=42)**

**# 5. Text Vectorization**

**vectorizer = TfidfVectorizer(stop\_words='english', max\_features=5000)**

**X\_train\_vec = vectorizer.fit\_transform(X\_train)**

**X\_test\_vec = vectorizer.transform(X\_test)**

**# 6. Model Training**

**model = MultinomialNB()**

**model.fit(X\_train\_vec, y\_train)**

**# 7. Evaluation**

**y\_pred = model.predict(X\_test\_vec)**

**print("Accuracy:", accuracy\_score(y\_test, y\_pred))**

**print("Classification Report:\n", classification\_report(y\_test, y\_pred))**

**# 8. Save Model and Vectorizer**

**pickle.dump(model, open('sentiment\_model.pkl', 'wb'))**

**pickle.dump(vectorizer, open('vectorizer.pkl', 'wb'))**

**from flask import Flask, request, jsonify**

**import pickle**

**# Load model and vectorizer**

**model = pickle.load(open('sentiment\_model.pkl', 'rb'))**

**vectorizer = pickle.load(open('vectorizer.pkl', 'rb'))**

**app = Flask(\_\_name\_\_)**

**@app.route('/predict', methods=['POST'])**

**def predict():**

**data = request.json**

**review = data.get('review', '')**

**vec = vectorizer.transform([review])**

**pred = model.predict(vec)[0]**

**sentiment = 'Positive' if pred == 1 else 'Negative'**

**return jsonify({'prediction': sentiment})**

**if \_\_name\_\_ == '\_\_main\_\_':**

**app.run(debug=True)**

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**vectorizer = TfidfVectorizer(max\_features=5000)**

**X = vectorizer.fit\_transform(df['cleaned\_text'])**

**# Assuming the sentiment label is positive (>=4 stars) or negative (<4)**

**df['label'] = df['overall'].apply(lambda x: 1 if x >= 4 else 0)**

**y = df['label']**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.linear\_model import LogisticRegression**

**from sklearn.metrics import classification\_report, accuracy\_score**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**model = LogisticRegression()**

**model.fit(X\_train, y\_train)**

**y\_pred = model.predict(X\_test)**

**print("Accuracy:", accuracy\_score(y\_test, y\_pred))**

**print(classification\_report(y\_test, y\_pred))**

**import joblib**

**# Save the model and vectorizer**

**joblib.dump(model, 'sentiment\_model.pkl')**

**joblib.dump(vectorizer, 'tfidf\_vectorizer.pkl')**

**# Later, to load and use the model**

**loaded\_model = joblib.load('sentiment\_model.pkl')**

**loaded\_vectorizer = joblib.load('tfidf\_vectorizer.pkl')**

**sample\_review = "This is a great product!"**

**processed = preprocess\_text(sample\_review)**

**vectorized = loaded\_vectorizer.transform([processed])**

**prediction = loaded\_model.predict(vectorized)**

**print("Positive" if prediction[0] == 1 else "Negative")**

## **14. Future Scope**

Several opportunities exist to extend this project further. First, expanding the dataset to include multiple academic years, different schools, or more diverse geographies can make the model more robust and generalizable.  
 Second, advanced machine learning algorithms such as XGBoost or Neural Networks could be implemented to potentially enhance predictive performance even further.  
 Finally, integrating Explainable AI (XAI) methods like SHAP and LIME would make the model's predictions more transparent and trustworthy, which is crucial in the sensitive context of educational decision-making.  
 Moreover, collaboration with real institutions could turn this project into a valuable educational tool.

**13. Team Members and Roles**

G.Madhumathi : Responsible for data cleaning, handling documentation and reporting

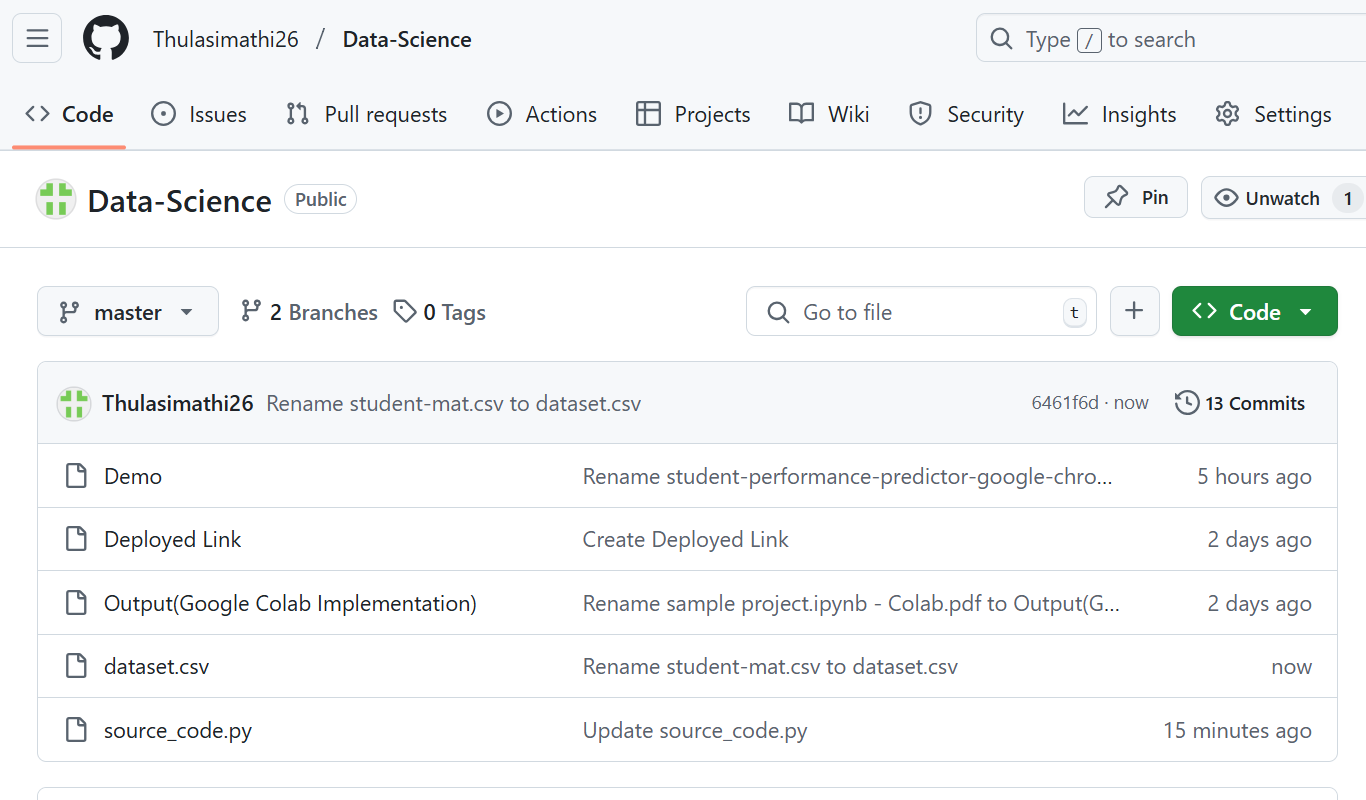
R.Subha:Worked on feature engineering

S.Karthiyayeeny : led the exploratory data analytics

P.Malathi : Took the charge of model development

J. .Asin Riddha: Documentation and reporting

**[Make sure ,you submit all the project files to Github]**

****