Sentiment analysis

Machine learning & NLP

Dr. Esraa Afify
Eng. Mostafa Gamal

Team

- Elsayed Mohamed 205038
- Abdelrahman Amer 205055
- Nourhan Fareed 205077

Illustrations by Pixeltrue on icons8

Abstract

This presentation focuses on sentiment analysis, specifically building a text classification model for emotion prediction based on textual data.

The dataset used in this project consists of text samples labeled with 9 emotions into 3 classes: negative, neutral, and positive.

The presentation covers various steps involved in the project, including data preprocessing, balancing the classes, building a pipeline with TF-IDF vectorization and Naive Bayes classifier, hyperparameter tuning using randomized search, training the model, evaluating its performance, visualizing the results, and saving the pipeline for future use.

The presentation also includes the development of a graphical user interface (GUI) using Streamlit.

Introduction

Text classification is a common task in natural language processing (NLP) that involves categorizing text into 3 classes into negative, neutral, and positive.

Emotion classification is one such application of text classification, where the goal is to identify the emotions expressed in textual data.

We focus on building a text classification model to predict emotions based on text samples.

We explore the process step-by-step, starting from data loading and visualization to data preprocessing, model training, evaluation and GUI.

Text data and known labels

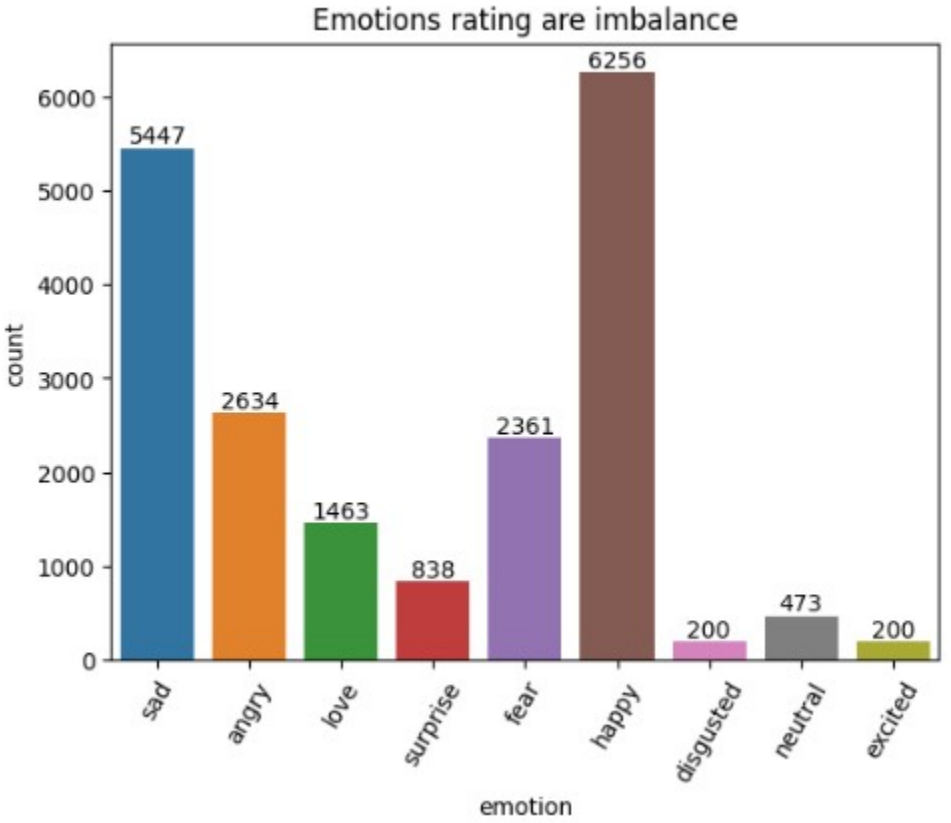
Reading the dataset and visualizing it using pandas, matplotlib, and seaborn libraries.

Displaying its information and shape and description.

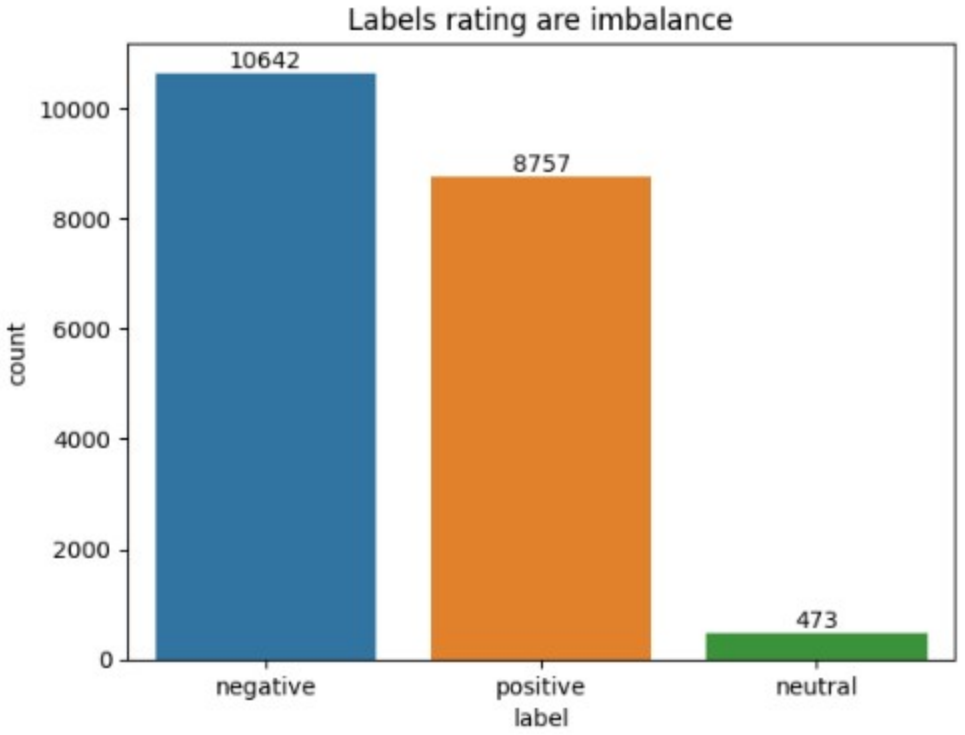
The missing data is handled by dropping the rows with missing values.

Solve imbalance between classes is visualized using a countplot.

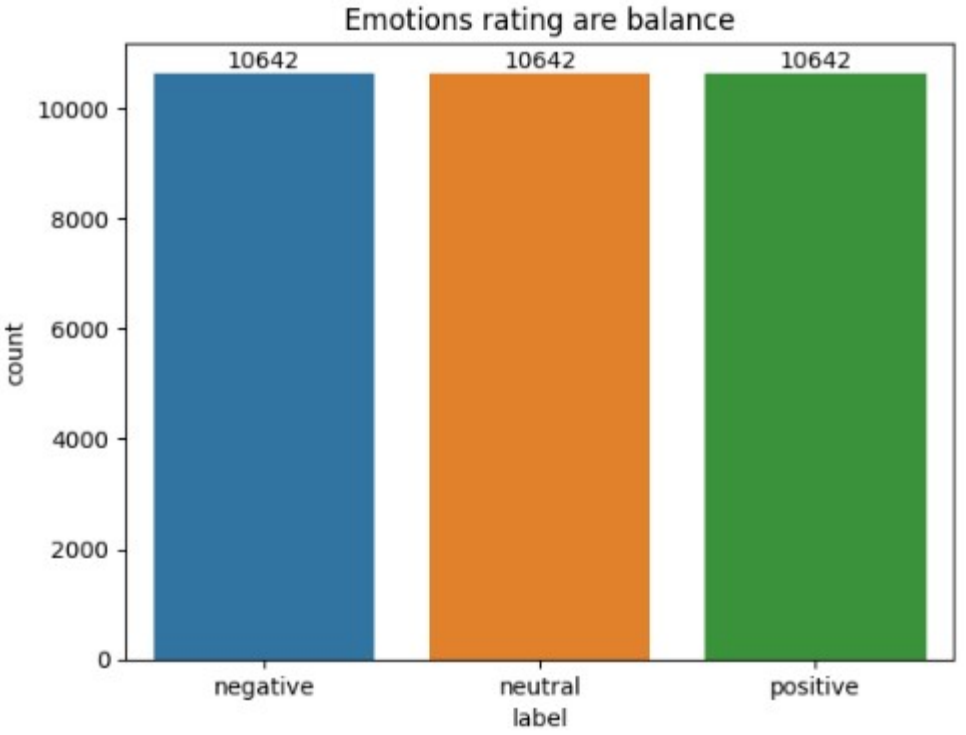
9 Classes!


Emotions rating are imbalance
count
emotion
0
1000
2000
3000
4000
5000
6000
5447
2634
1463
838
2361
6256
200
473
200
sad
angry
love
surprise
fear
happy
disgusted
neutral
excited

Merge to 3 Classes!



Resample method!



Data preprocessing

The project highlights the importance of data preprocessing in cleaning and preparing textual data for analysis:

+ Noise removal:
    - stop words

+ Tokenization :
    - word tokenize

+ Word normalization:
    - PorterStemmer

    - WordNetLemmatizer.

Build Pipeline

- The pipeline is built to simplify the training and evaluation process.

+ The pipeline includes the
    - TfidfVectorizer for feature extraction
    - the Naive Bayes classifier for training.

- RandomizedSearchCV is used to find the best hyperparameters for the pipeline

Training machine learning model

+ The Naive Bayes model is trained on the training data using the pipeline with the best hyperparameters.

Evaluation

+ The performance of the model is evaluated using

- accuracy score:

93%

- classification report:

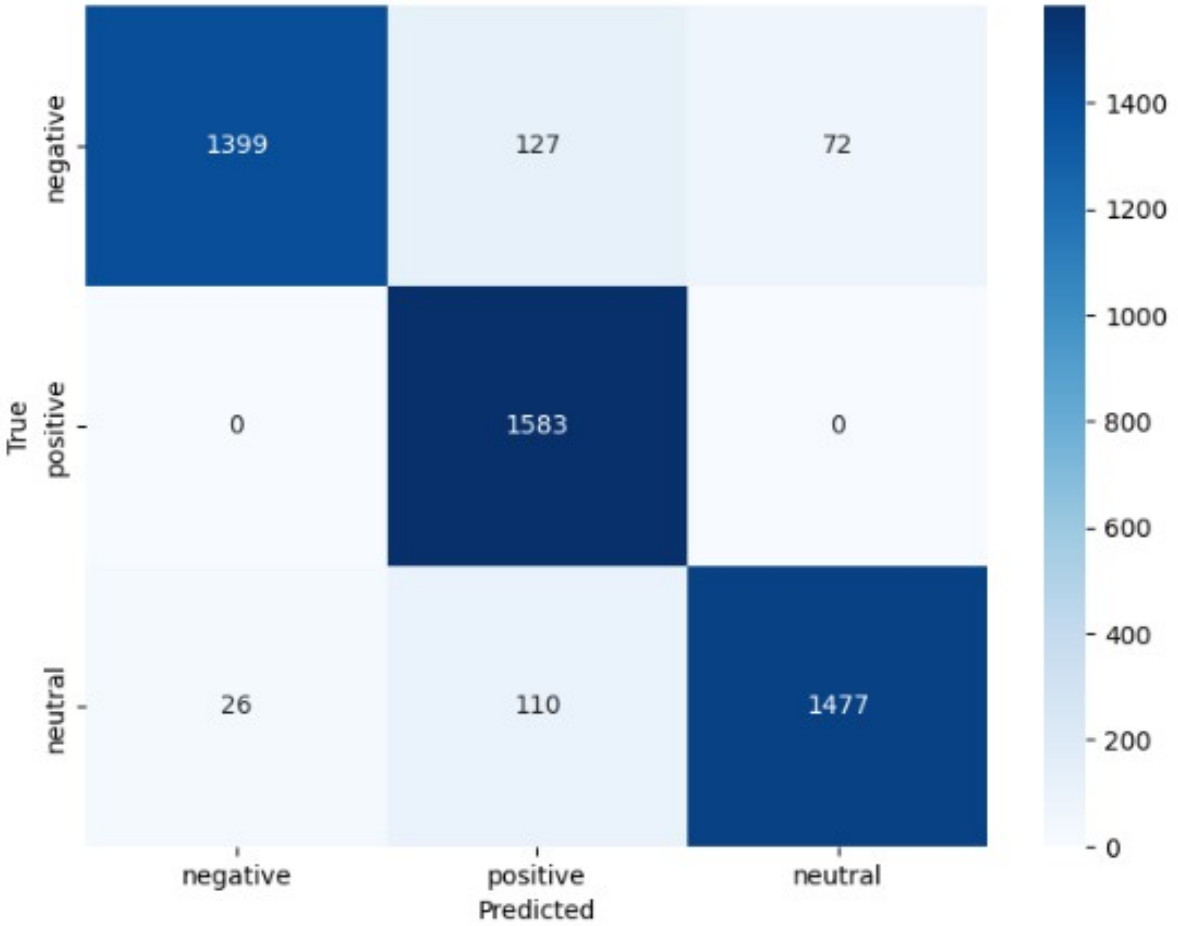
	precision	recall	f1-score	support
-1	0.98	0.88	0.93	1598
0	0.87	1.00	0.93	1583
1	0.95	0.92	0.93	1613

Visualization

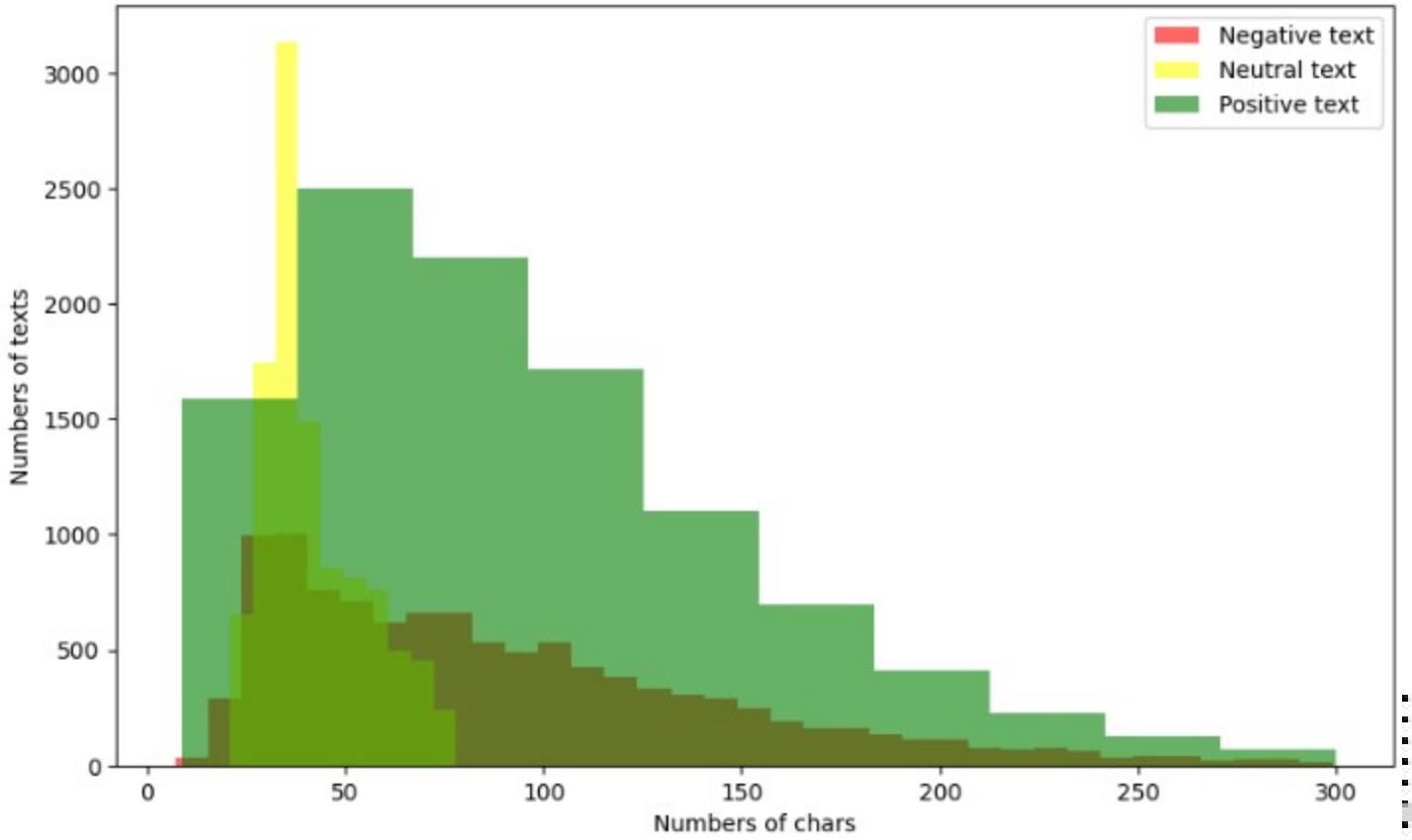
- A heatmap of the confusion matrix is plotted to visualize the performance of the model.

- confusion matrix

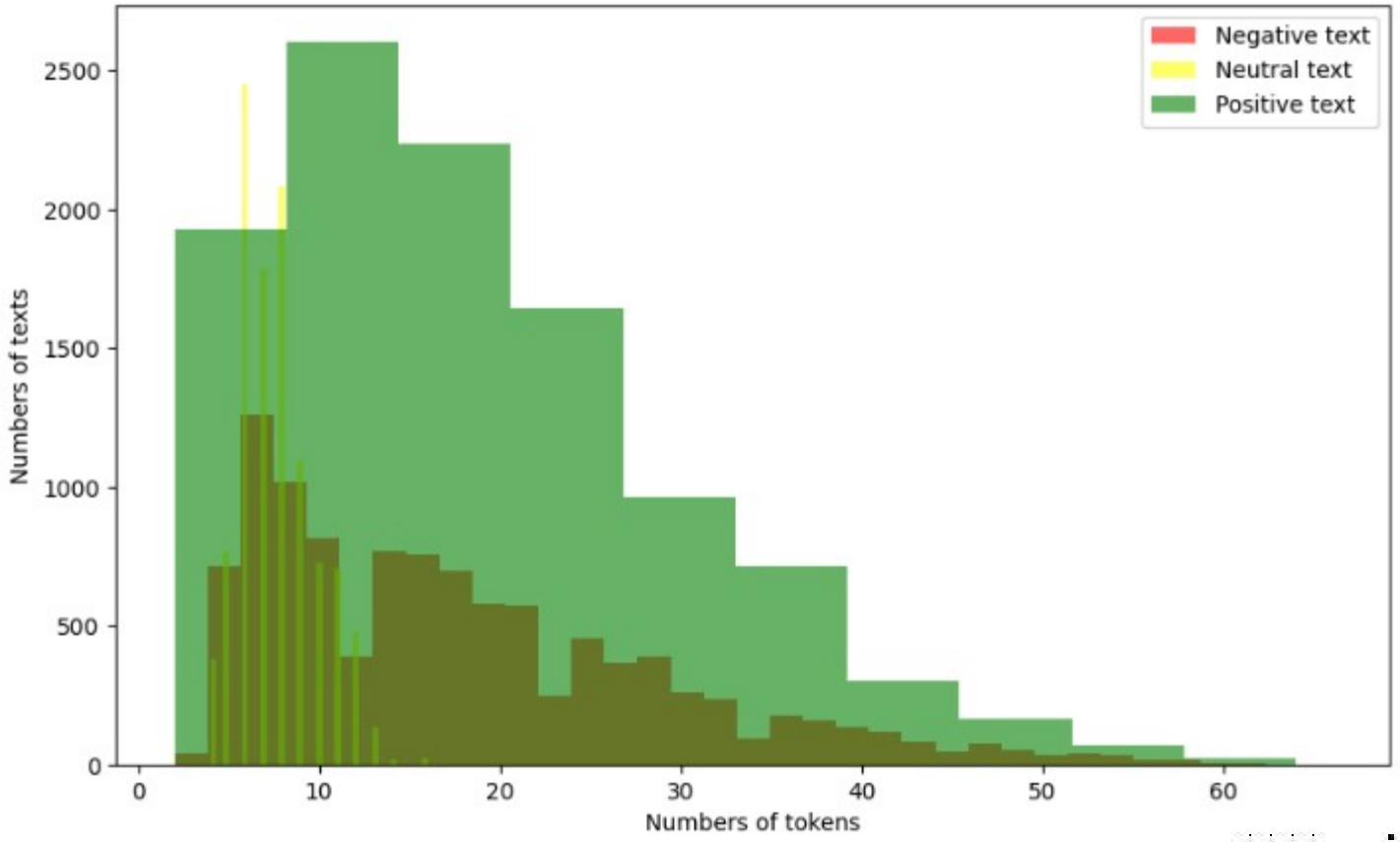
Confusion matrix

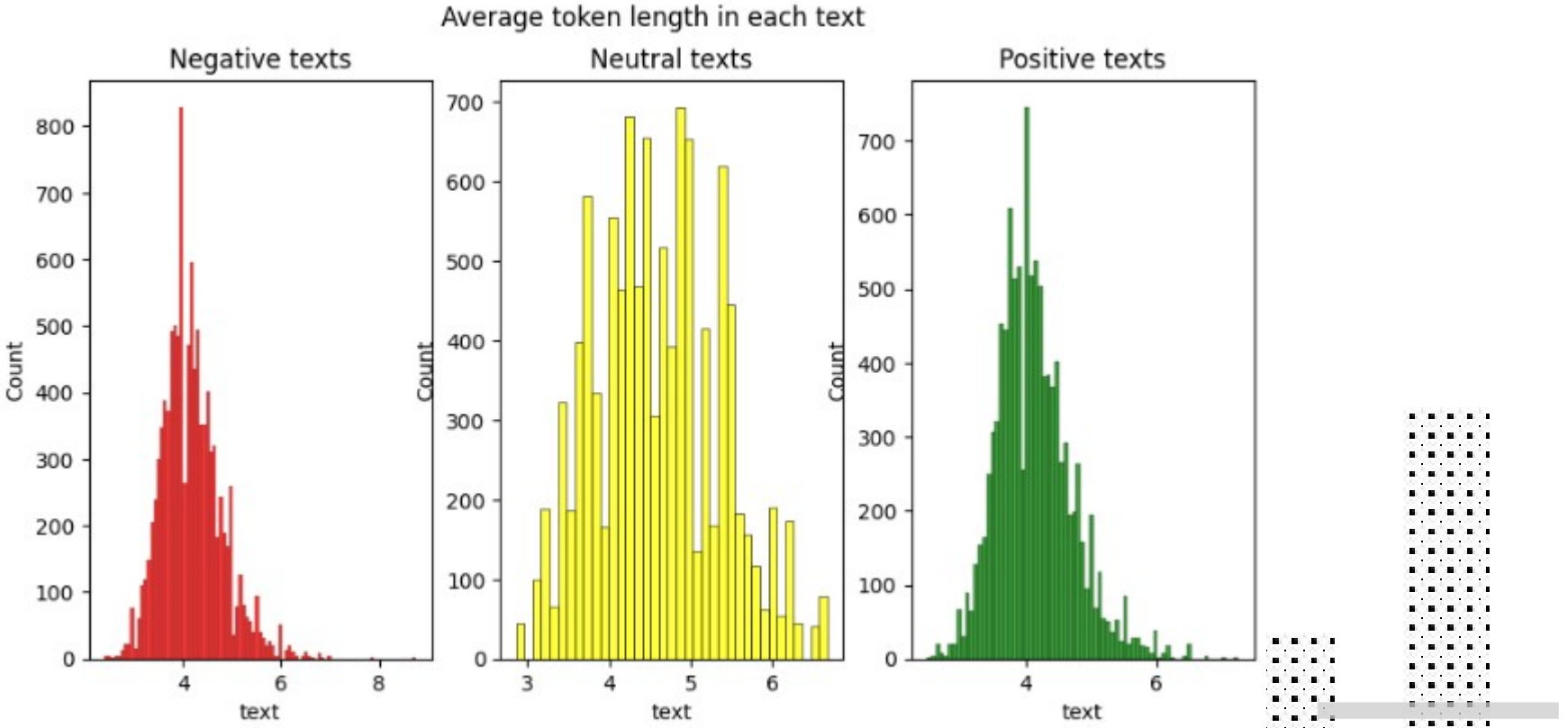


Number of chars



Number of tokens




Average token length in each text
Negative texts
Neutral texts
Positive texts
Count
text
0
100
200
300
400
500
600
700
800
3
4
5
6
8

Save the pipeline

- The trained pipeline is saved using the joblib library for future use.

Build GUI

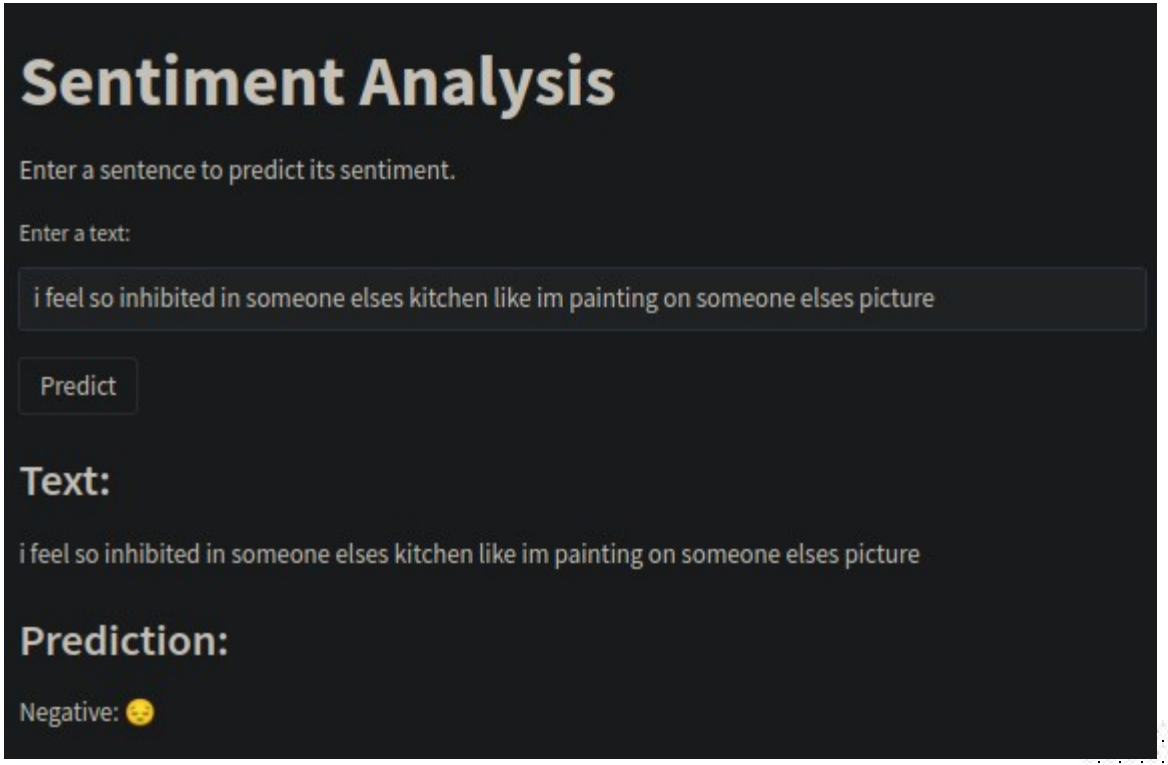
- This part involves building a graphical user interface (GUI) using the Streamlit library.

- Load model from the saved pipeline.

- A function is defined to preprocess the input text.

- The GUI allows users to input text and get predictions from the model.

Build GUI

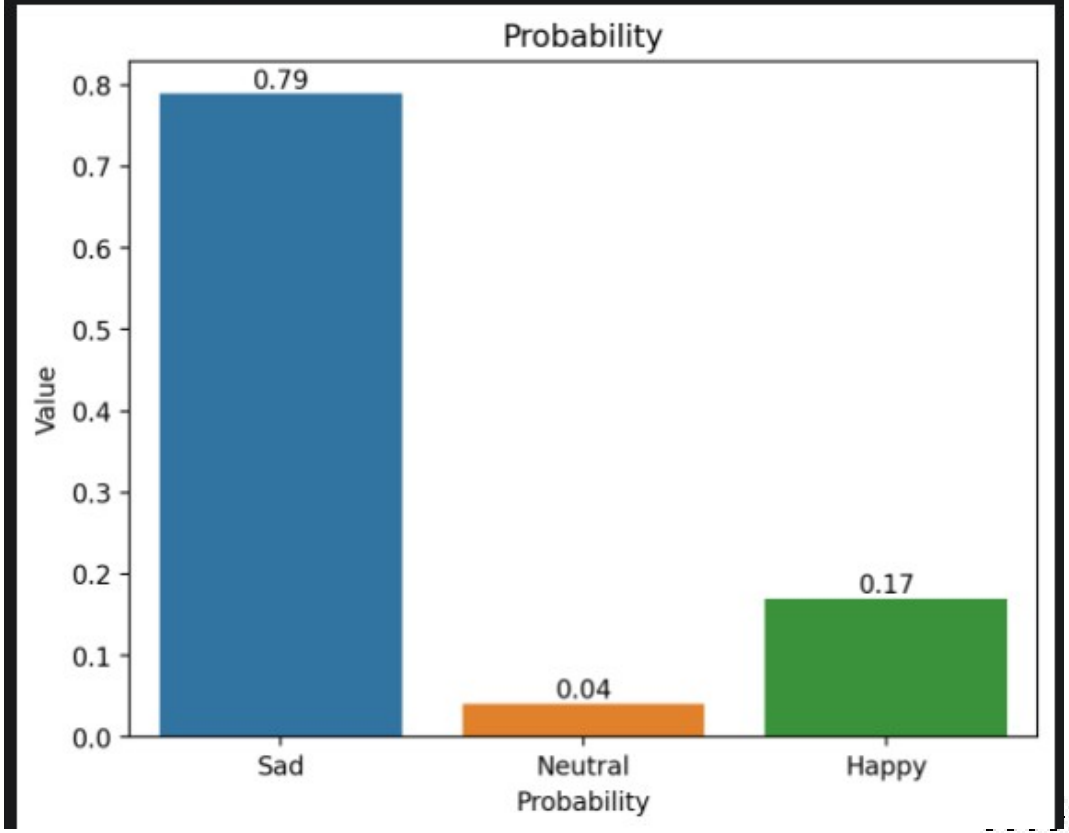

Sentiment Analysis
Enter a sentence to predict its sentiment.
Enter a text:
i feel so inhibited in someone elses kitchen like im painting on someone elses picture
Predict
Text:
i feel so inhibited in someone elses kitchen like im painting on someone elses picture
Prediction:
Negative:

Probability:

Probability of Sad: 0.79

Probability of Neutral: 0.04

Probability of Happy: 0.17



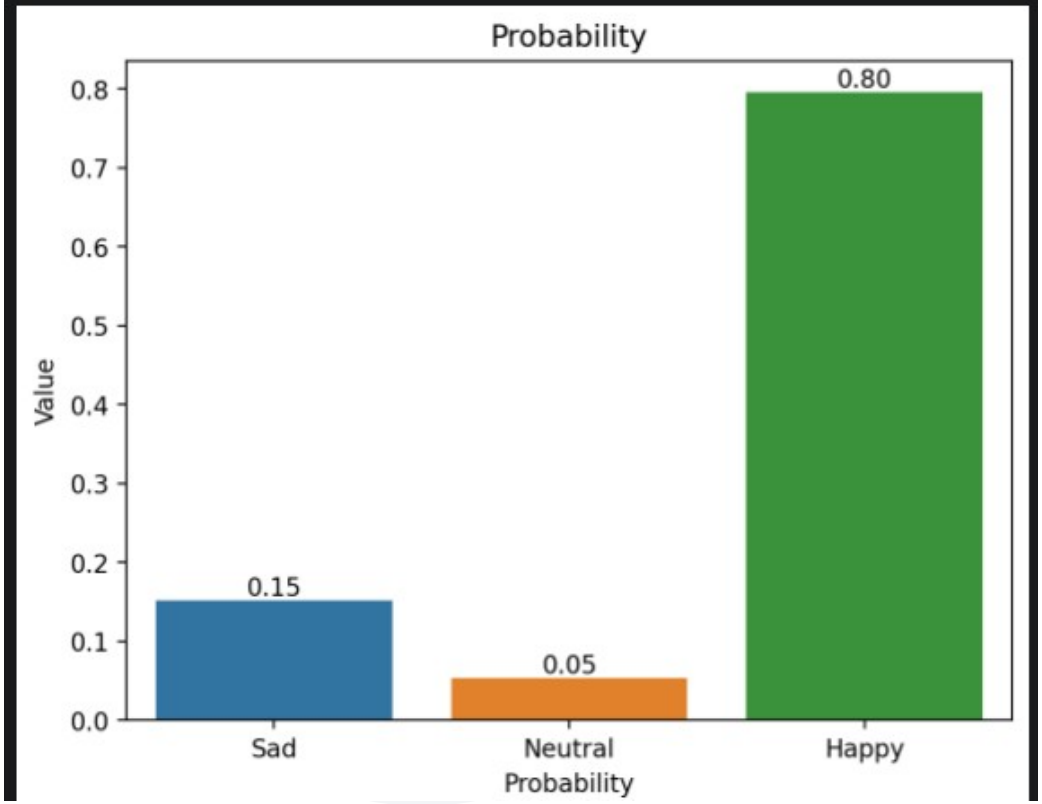

Sentiment Analysis
Enter a sentence to predict its sentiment.
Enter a text:
i have been feeling the need to be creative
Predict
Text:
i have been feeling the need to be creative
Prediction:
Positive: 😂

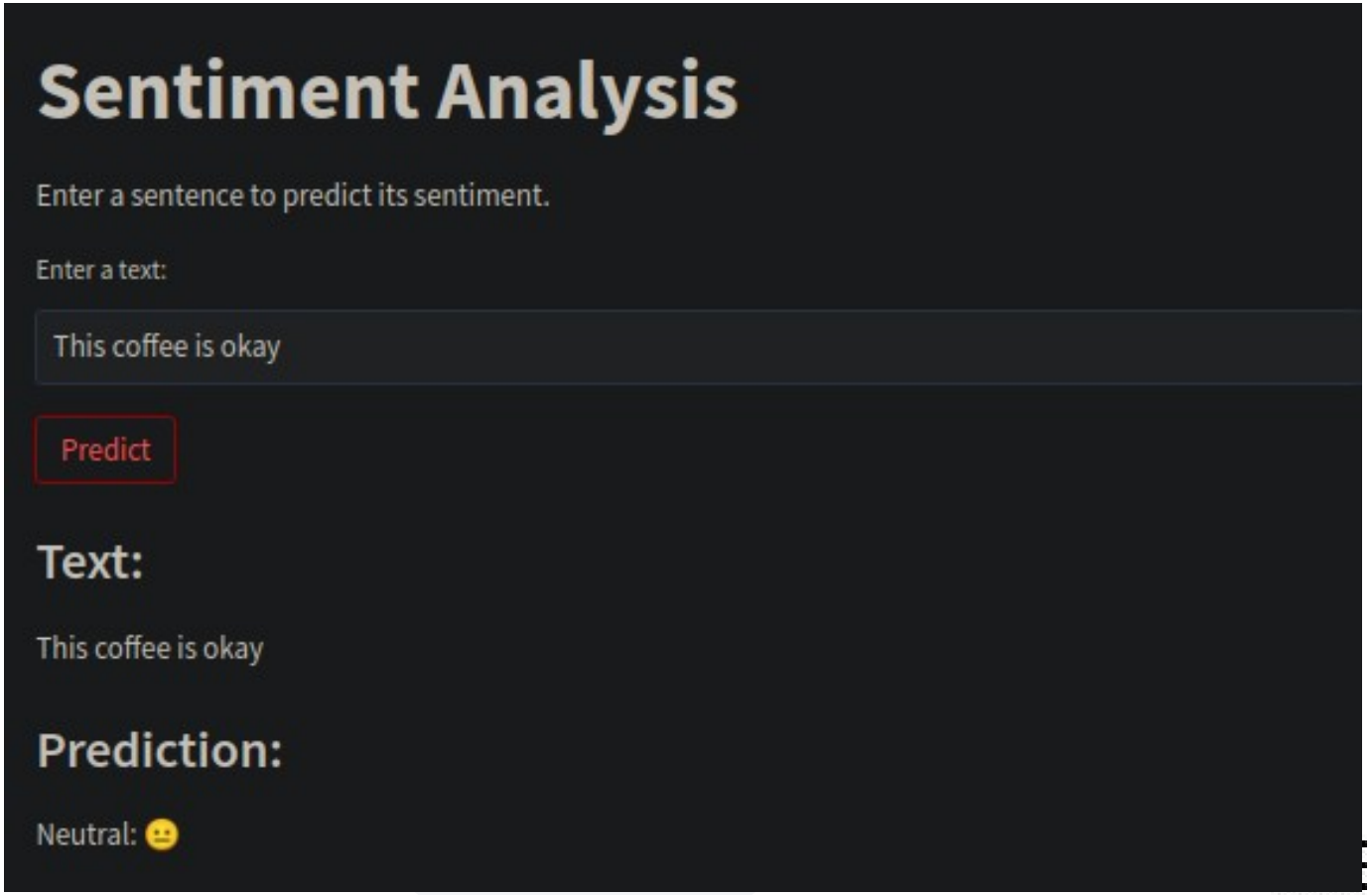
Probability:

Probability of Sad: 0.15

Probability of Neutral: 0.05

Probability of Happy: 0.80



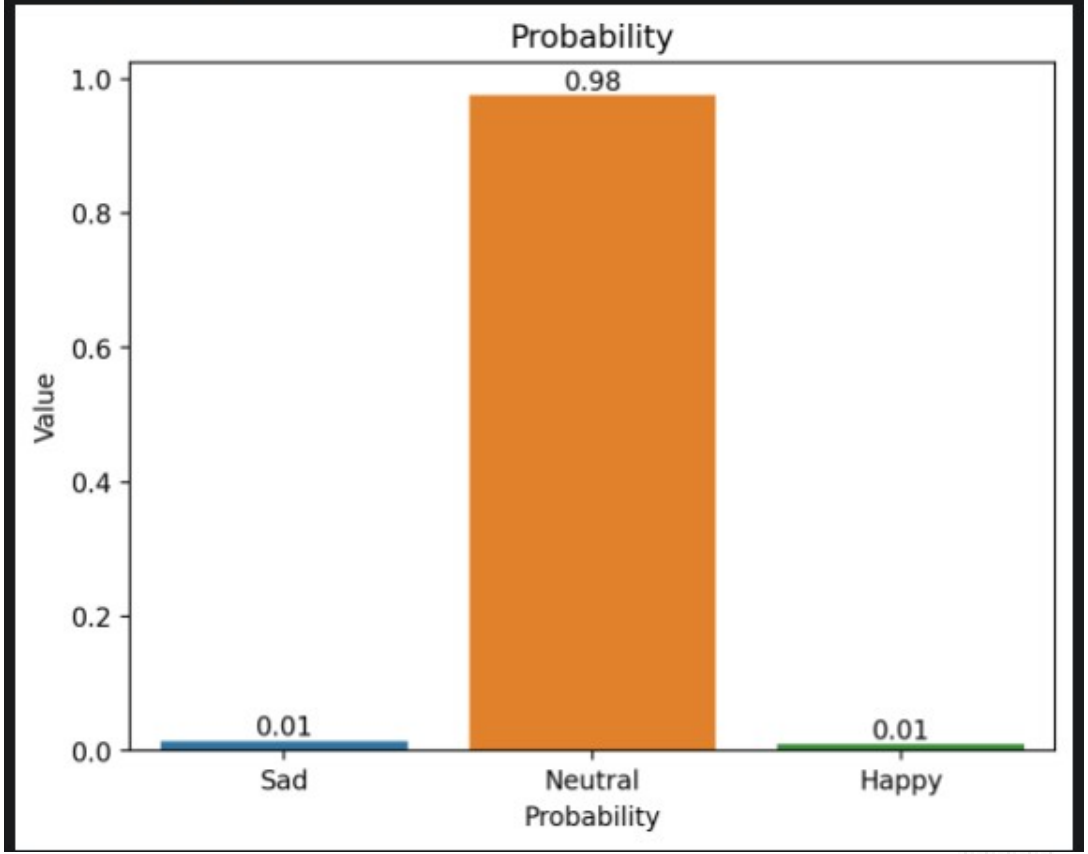

Sentiment Analysis
Enter a sentence to predict its sentiment.
Enter a text:
This coffee is okay
Predict
Text:
This coffee is okay
Prediction:
Neutral: 😐

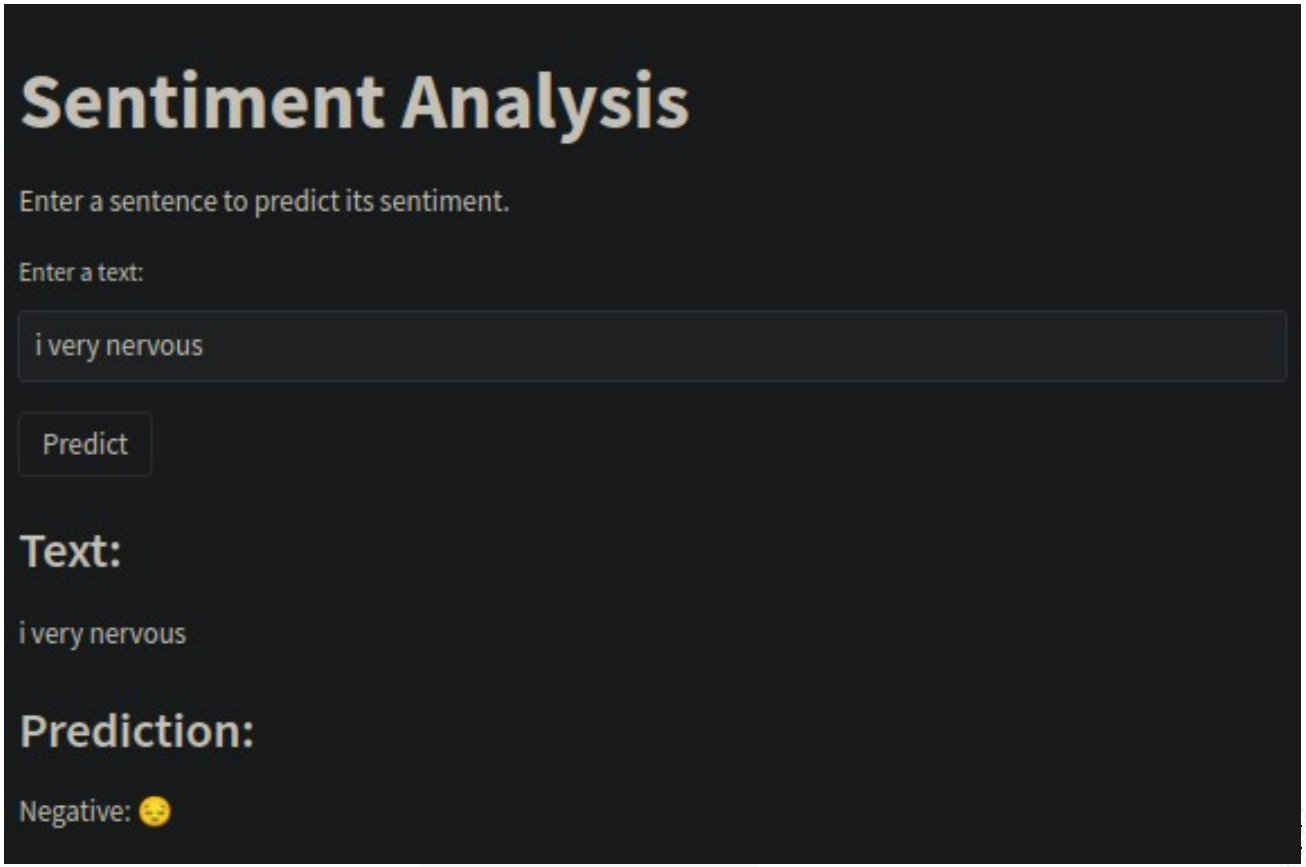
Probability:

Probability of Sad: 0.01

Probability of Neutral: 0.98

Probability of Happy: 0.01



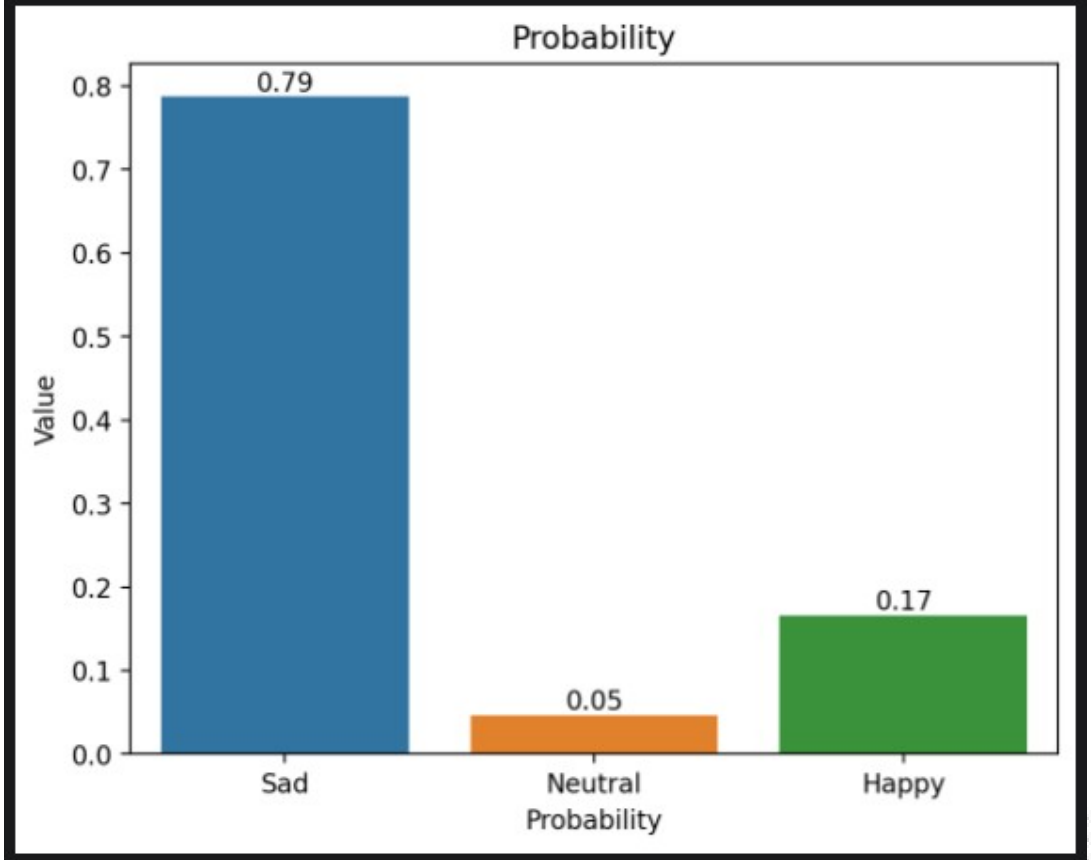

Sentiment Analysis
Enter a sentence to predict its sentiment.
Enter a text:
i very nervous
Predict
Text:
i very nervous
Prediction:
Negative: 😔

Probability:

Probability of Sad: 0.79

Probability of Neutral: 0.05

Probability of Happy: 0.17



Conclusion

This project contributes to the field of natural language processing by providing insights into sentiment analysis and emotion prediction. The demonstrated techniques and methodologies can be applied in various domains, such as social media sentiment analysis, customer feedback analysis, and market research.

Future research can focus on exploring advanced techniques and incorporating additional features to further improve the model's performance and expand its applicability

Photo by Dave Hoefler on Unsplash