



# Sentiment analysis

Machine learning & NLP

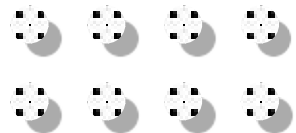
Dr. Esraa Afify  
Eng. Mostafa Gamal

# Team

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



Illustrations by [Pixeltrue](#) on [icons8](#)





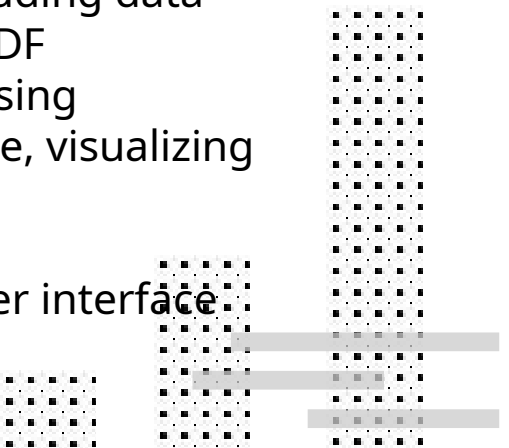
# Abstract

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 different emotions.

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.





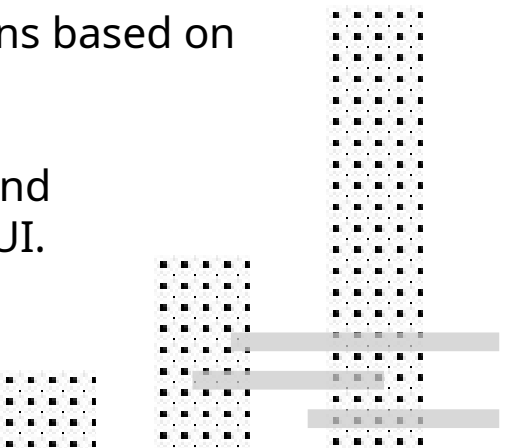
# Introduction

Text classification is a common task in natural language processing (NLP) that involves categorizing text into predefined classes or categories.

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, GUI.





# Text data and known labels

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

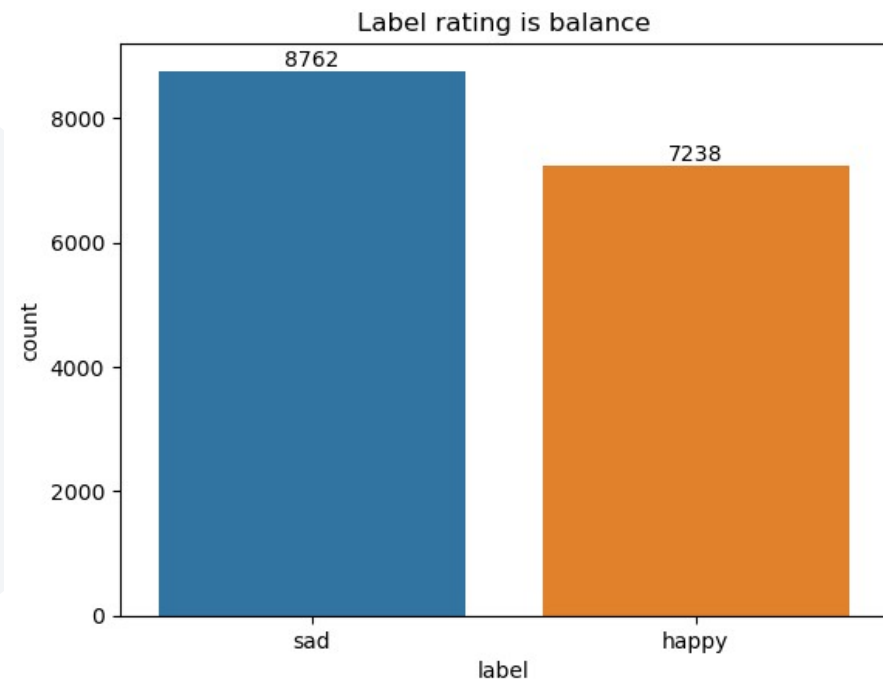
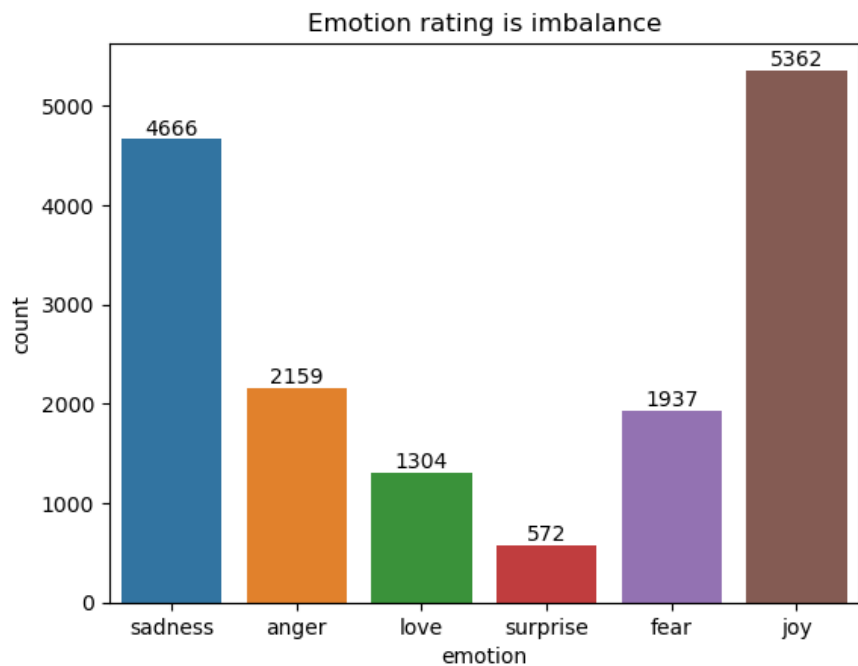
Displaying its information and shape and description.

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

Solve imbalance between classes is visualized using a countplot.

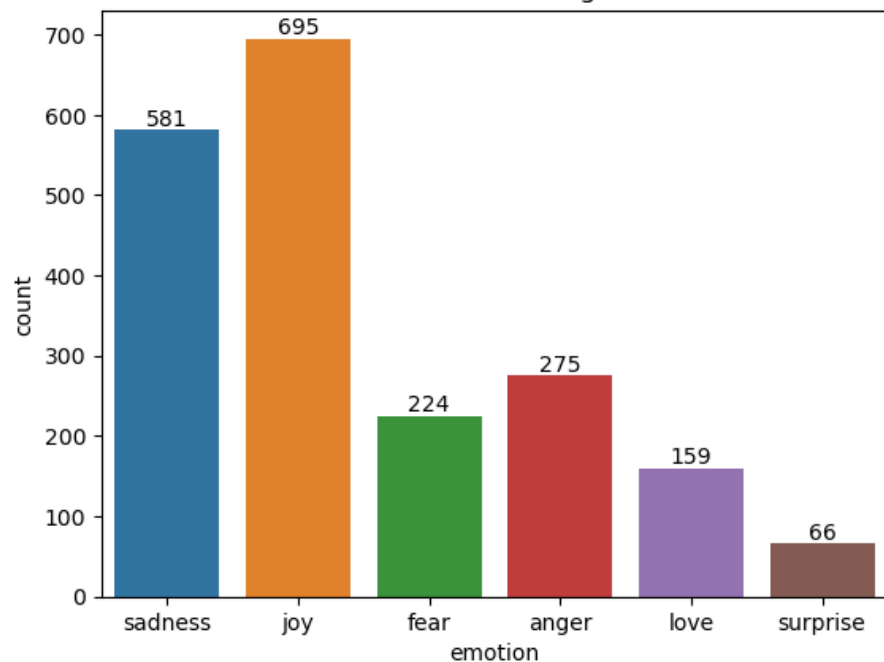


# Traning dataset

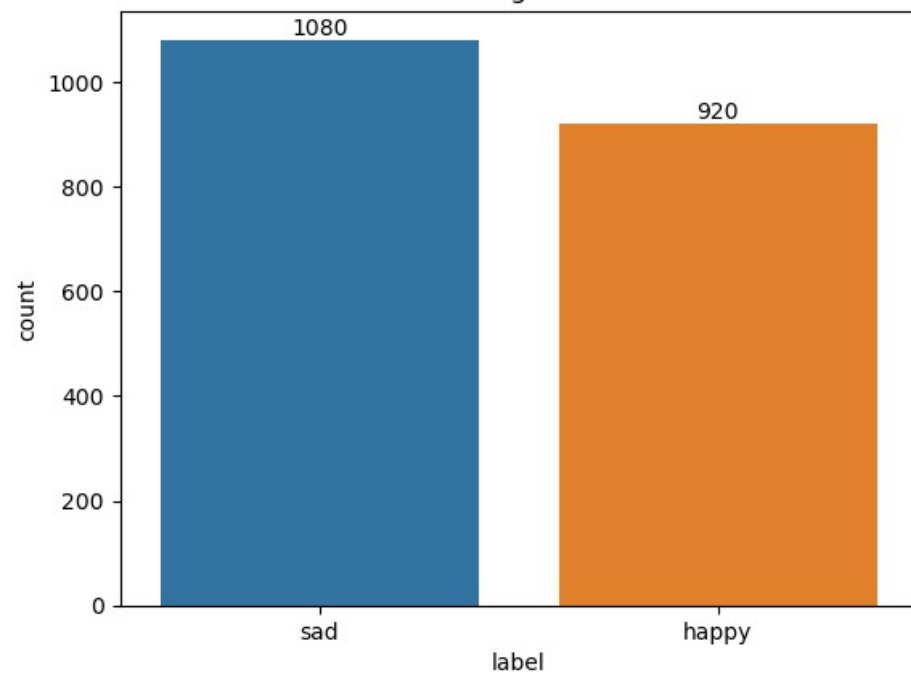


# Testing dataset

Emotion Rating



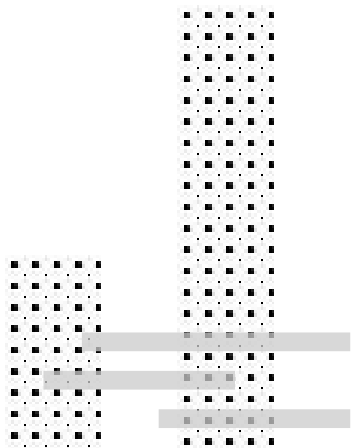
Label rating is balance





# Data preprocessing

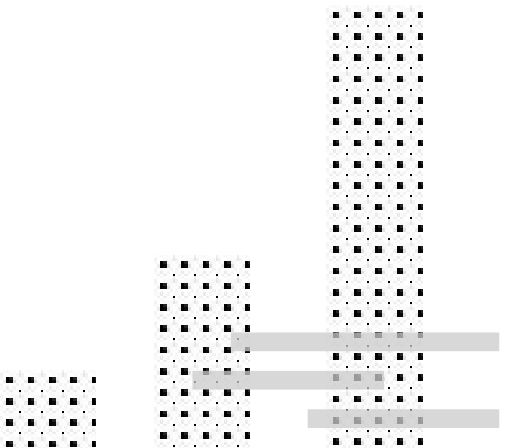
The project highlights the importance of data preprocessing in cleaning and preparing textual data for analysis:

- 
- + Noise removal:
    - stop words
  - + Tokenization :
    - word tokenize
  - + Word normalization:
    - PorterStemmer
    - WordNetLemmatizer.






# Build Pipeline

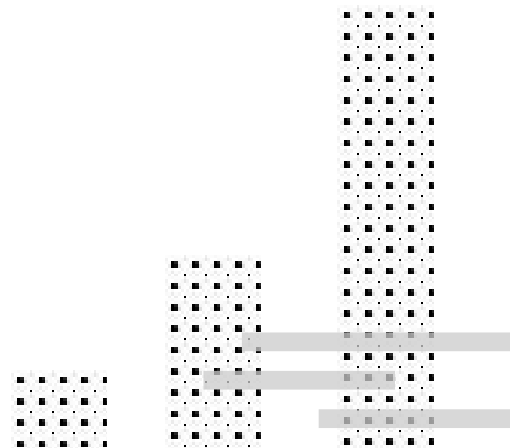
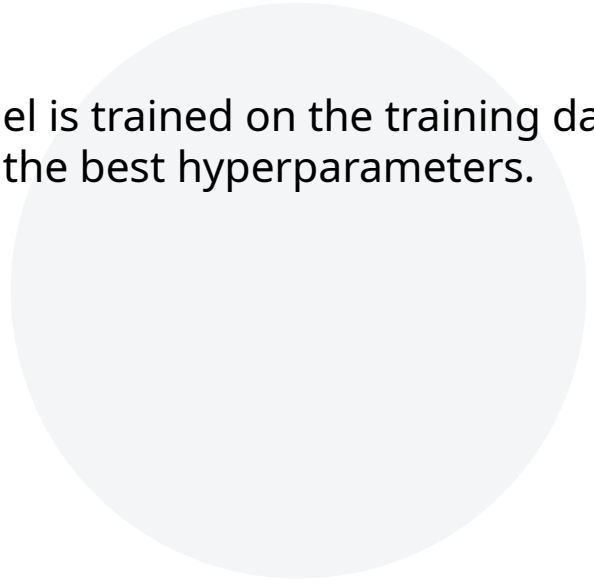
- The pipeline is built to simplify the training and evaluation process.
  - + The pipeline includes the
    - TfidfVectorizer for feature extraction
    - the Naive Bayes classifier for training.
  - RandomizedSearchCV is used to find the best hyperparameters for the pipeline
- 



# 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:**  
95%

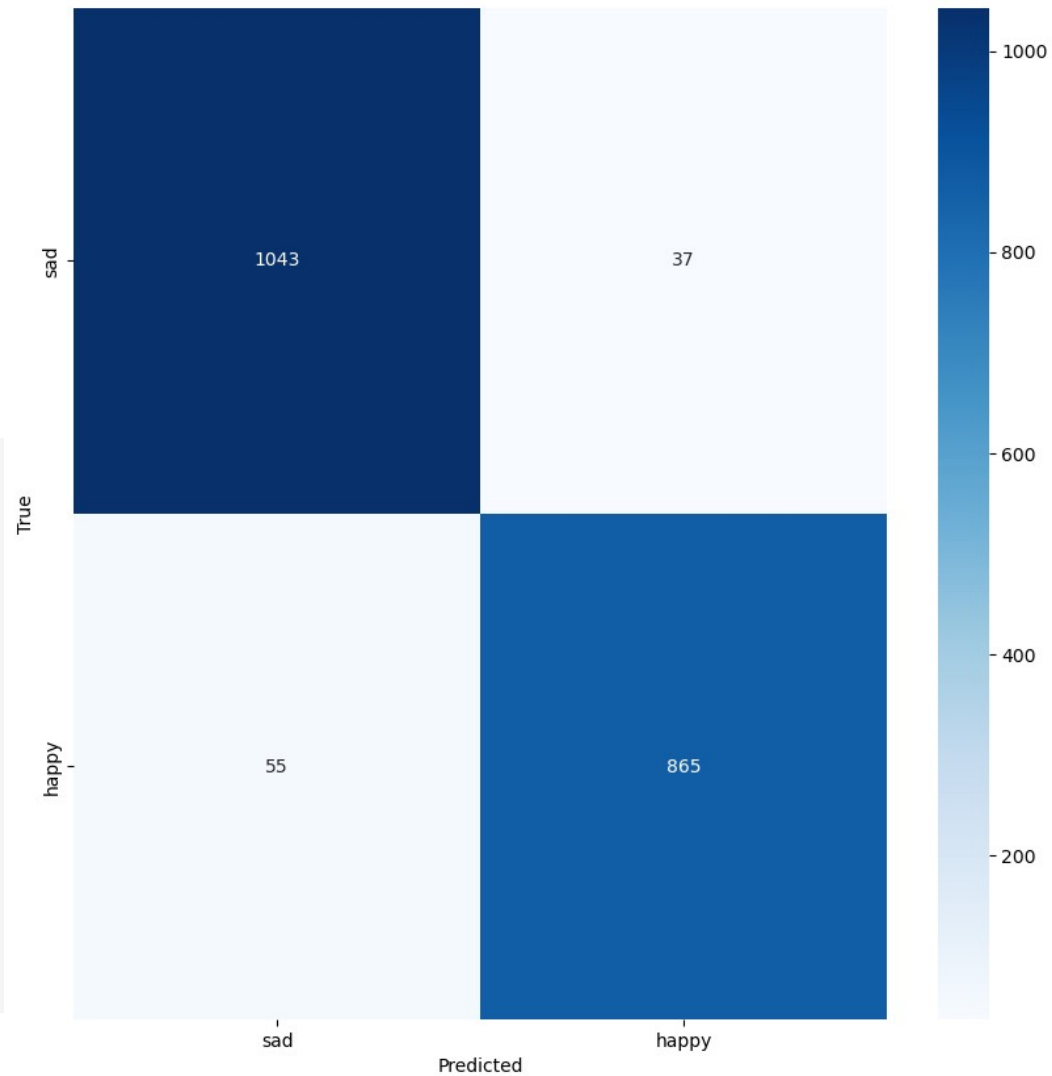
- **classification report:**

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

0	0.95	0.97	0.96	1080
1	0.96	0.94	0.95	920


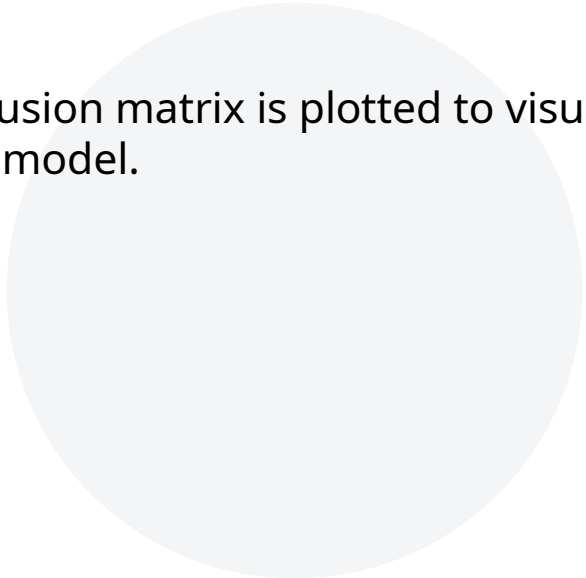
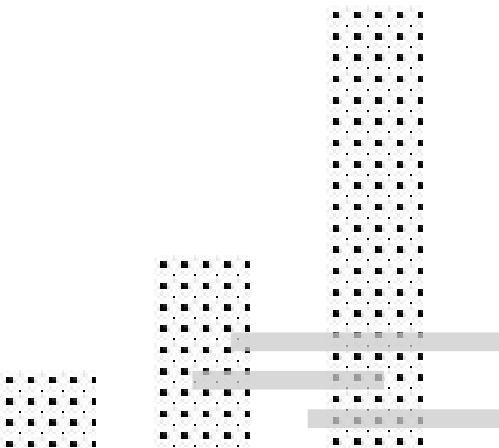
# Evaluation

- confusion matrix




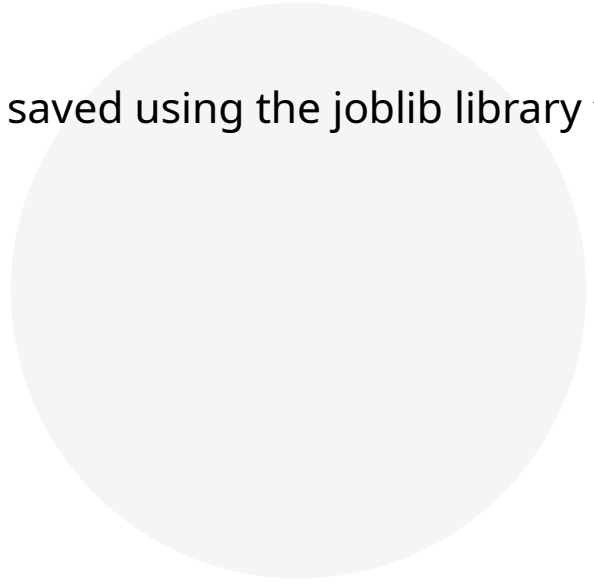
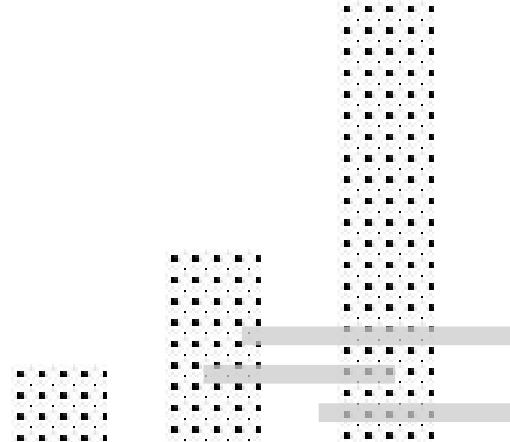


# Visualization

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


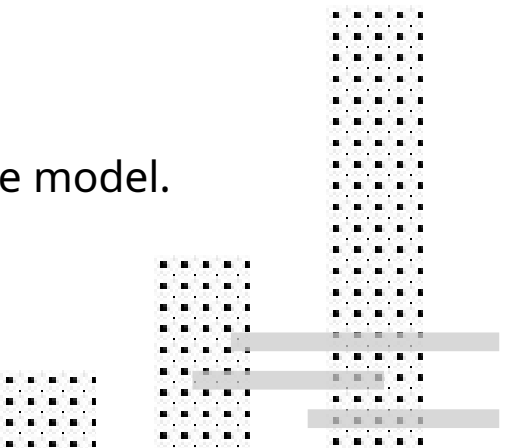


# Save the pipeline

- 
- 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 like my only role now would be to tear your sails with my pessimism and discontent

Predict

Text:

i feel like my only role now would be to tear your sails with my pessimism and discontent

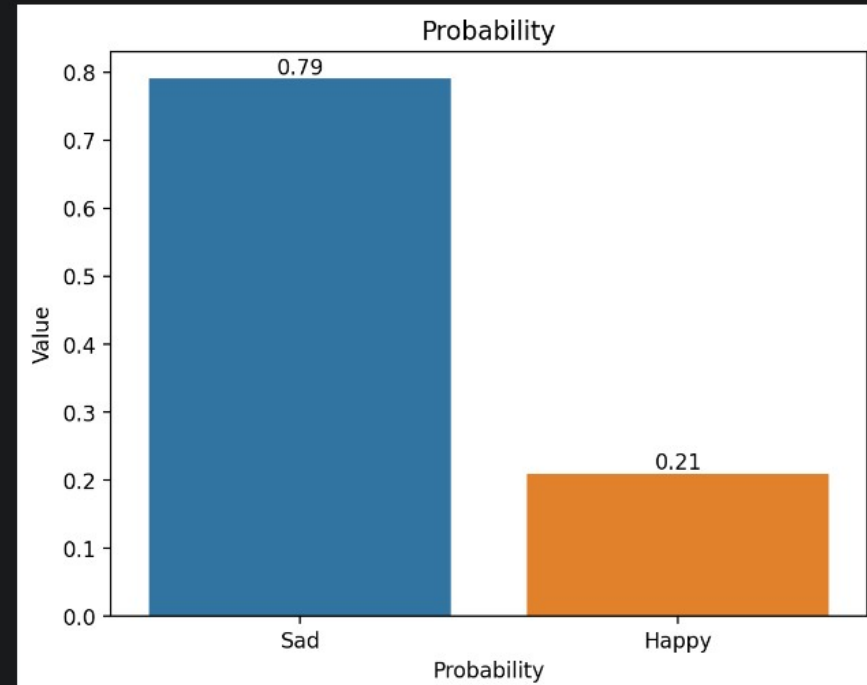
Prediction:

Sad: 😞

### Probability:

Probability of Sad: 0.79

Probability of Happy: 0.21





# 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](#)