Twitter Sentiment Prediction

(Group-1)

Hamsalakshmi Ramachandran	017423666
Himani Shah	017411407
Soumya Challuru Sreenivas	017518618
Sugandha Chauhan	017506190

Sentiment analysis is the process of automatically classifying text into categories such as positive, negative, or neutral. This tool is highly beneficial for companies as it helps them understand customer opinions on their products and services. By leveraging sentiment analysis, businesses can make data-driven decisions, identify potential public relations issues early, and manage their brand reputation effectively, ensuring a positive public image.

Twitter (now referred to as "X") as a prominent social media platform where users share their thoughts, opinions, and updates through short messages known as tweets. Twitter has evolved into a vital platform for interaction and community building on a global scale, making it a rich source of data for sentiment analysis.

Problem Understanding and Formulation

The Challenge

The platform faces issues with misuse, especially in the spread of hateful content and misinformation.

The Goal

- Build a robust classifier model using Natural Language Processing (NLP) for Twitter sentiment analysis.
- Classify tweets as positive, neutral, or negative to help identify and flag harmful or hateful content.

BRIEF WORKFLOW

1. Dataset Acquisition
2. Exploratory Data Analysis (EDA)
3. Data Cleaning and Preprocessing
4. Model Selection and Training
5. Model Evaluations
6. Web Interface with Gradio

DATA EXAMINATION & CLEANING

- **Columns:** textID - unique ID for each piece of text, text - the text of the tweet, sentiment - the general sentiment of the tweet, selected_text - the text that is selected for the tweet's sentiment

- ~ 27,000 data points.

```
In [10]: # Loading the dataset
         df = pd.read_csv('Tweets.csv')
         #Let's check the samples of data
         df.head()
```

Out[10]:

	textID	text	selected_text	sentiment
0	cb774db0d1	I`d have responded, if I were going	I`d have responded, if I were going	neutral
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative
2	088c60f138	my boss is bullying me...	bullying me	negative
3	9642c003ef	what interview! leave me alone	leave me alone	negative
4	358bd9e861	Sons of ****, why couldn`t they put them on t...	Sons of ****,	negative

- Handling Missing Values and checking for duplicates

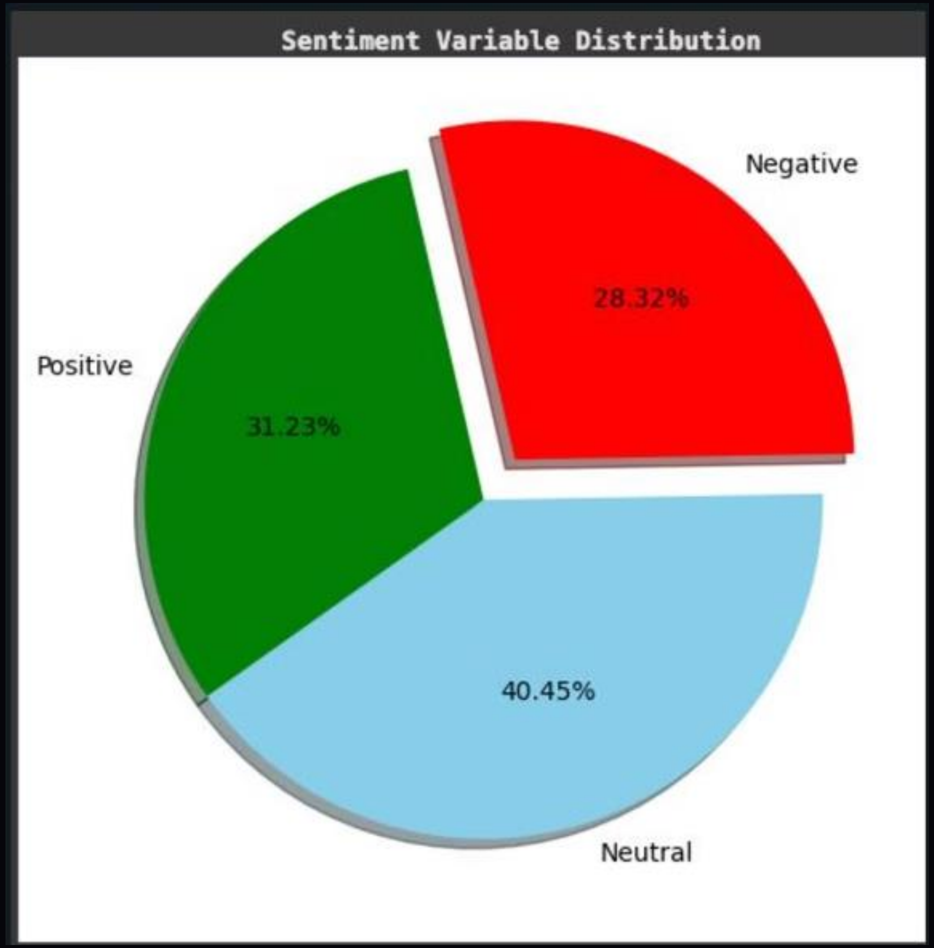
```
In [14]: #Let's check Null values
         df.isnull().sum()

Out[14]: text         1
         sentiment    0
         dtype: int64
```

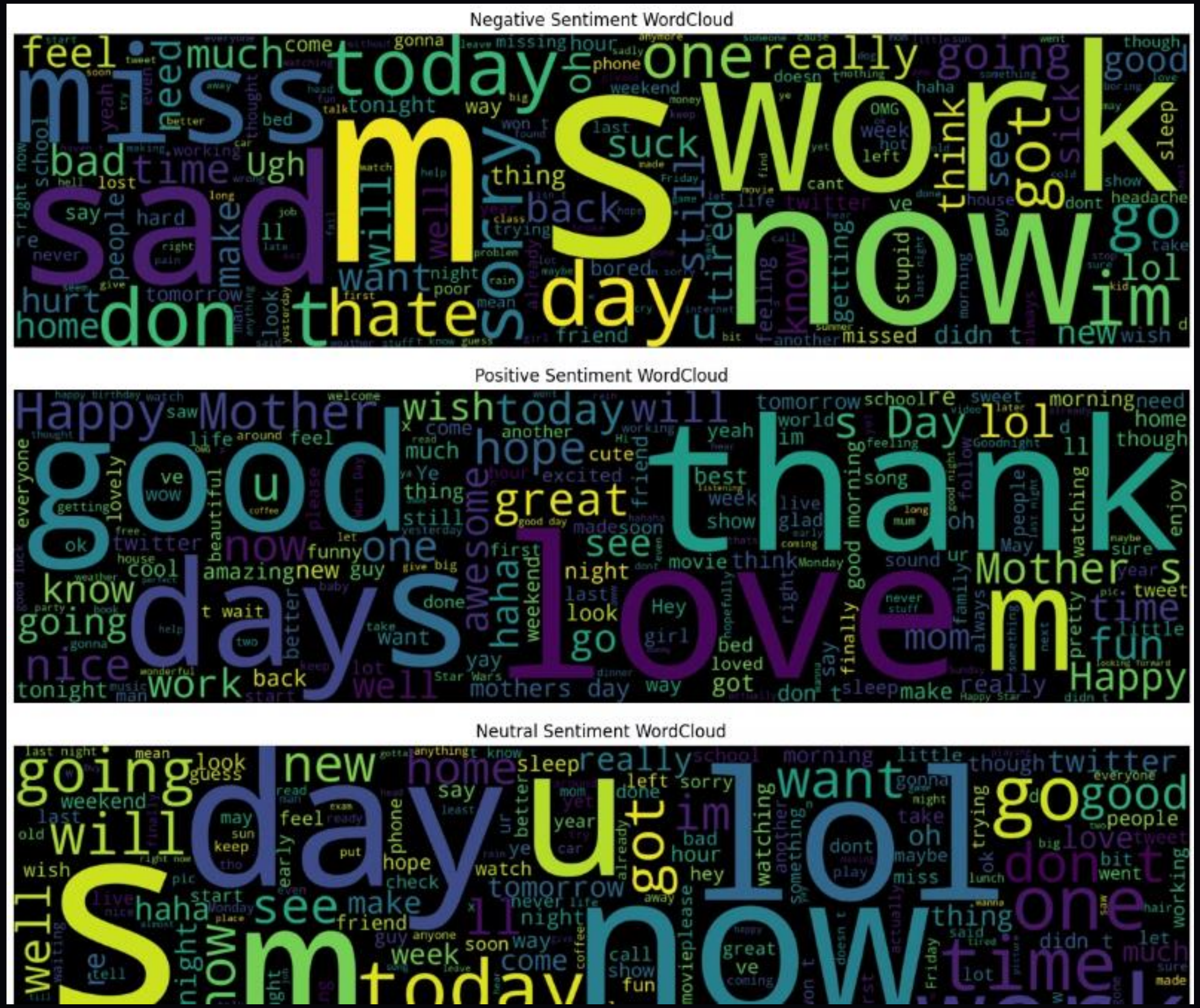
The dataset has one null row. So, we are dropping it

```
In [15]: #Dropping the null values
         df.dropna(inplace=True)
         original_df = df.copy()

In [16]: #checking Duplicates
         df.duplicated().sum()

Out[16]: 0
```

No duplicates found

EXPLORATORY DATA ANALYSIS (EDA) INSIGHTS





Data Preparation and Preprocessing

Replaced Backticks with apostrophes

Handled contractions: eg couldn't -> could not

Removed HTML tags, URLs, digits and special characters

Converted text to lower case

Tokenized the tweet text

Removed stopwords

Performed Lemmatization

```python
# Define a function to clean and preprocess the text
def preprocess_text(text):

    # Replace Backticks with apostrophes
    text = text.replace('`', "'")

    # Replacing contractions like "don't" with "do not" for better sentiment context
    text = fix(text)

    # Remove HTML tags and URLs
    text = re.sub(r'<.*?>|http\S+', '', text)

    # Remove digits and punctuations
    text = re.sub(r'[^a-zA-Z\s]', '', text)  # Keep only alphabets and spaces

    # Convert text to lower case
    text = text.lower()

    # Tokenize the text
    tokens = word_tokenize(text)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    stop_words = stop_words - set(essential_stopwords)   # Remove essential stopwords from the standard list
    tokens = [word for word in tokens if word not in stop_words]

    # Perform Lemmatization
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(word) for word in tokens]

    # Join the tokens back into a single string
    cleaned_text = ' '.join(tokens)
    return cleaned_text

# Apply preprocessing function to text column
df['cleaned_text'] = df['text'].apply(preprocess_text)
```

```python
df.head(7)
```

	text	sentiment	cleaned_text
0	I`d have responded, if I were going	neutral	would responded going
1	Sooo SAD I will miss you here in San Diego!!!	negative	sooo sad miss san diego
2	my boss is bullying me...	negative	bos bullying
3	what interview! leave me alone	negative	interview leave alone
4	Sons of ****, why couldn`t they put them on t...	negative	son could not put release already bought
5	http://www.dothebouncy.com/smf - some shameles...	neutral	shameless plugging best ranger forum earth
6	2am feedings for the baby are fun when he is a...	positive	feeding baby fun smile coo

Modeling and Evaluation

Decision Tree Classifier

```
Best Parameters: {'dt__criterion': 'gini', 'dt__max_depth': 20, 'dt
Accuracy: 0.5816957787481805

Classification Report:
              precision    recall  f1-score   support

    negative       0.76      0.23      0.35      1572
     neutral       0.51      0.89      0.65      2236
    positive       0.78      0.50      0.61      1688

    accuracy                           0.58      5496
   macro avg       0.68      0.54      0.54      5496
weighted avg       0.66      0.58      0.55      5496
```

- Decision Tree was used for its interpretability and ability to model non-linear relationships.

Overall performance was not up to the mark.

Random Forest Classifier

```
Best Parameters: {'rf__max_depth': 30, 'rf__min_samples_lea
Accuracy: 0.6273653566229985

Classification Report:
              precision    recall  f1-score   support

    negative       0.80      0.33      0.47      1572
     neutral       0.54      0.87      0.67      2236
    positive       0.79      0.58      0.67      1688

    accuracy                           0.63      5496
   macro avg       0.71      0.59      0.60      5496
weighted avg       0.69      0.63      0.61      5496
```

- Random Forest is used reduces errors, making it effective for some aspects of tweet sentiment prediction.
- It provides high precision for certain classes, such as positive sentiment.

Low accuracy and struggles to identify neutral tweets.

Naive Bayes Classifier

```
Best Parameters: {'nb__alpha': 1.0, 'nb__fit_prior': True, 'prepr
Accuracy: 0.6277292576419214

Classification Report:
              precision    recall  f1-score   support

    negative       0.72      0.44      0.54      1572
     neutral       0.55      0.80      0.65      2236
    positive       0.76      0.57      0.65      1688

    accuracy                           0.63      5496
   macro avg       0.68      0.60      0.62      5496
weighted avg       0.66      0.63      0.62      5496
```

- Naive Bayes is computaitonally efficient and performs well on text data. It achieved reasonable performance compared to random forest.

Compared to Random Forest this model was able to achieve better recall for all the classes and improved accur

We still wanted to achieve better accuracy and balanced precision and recall.

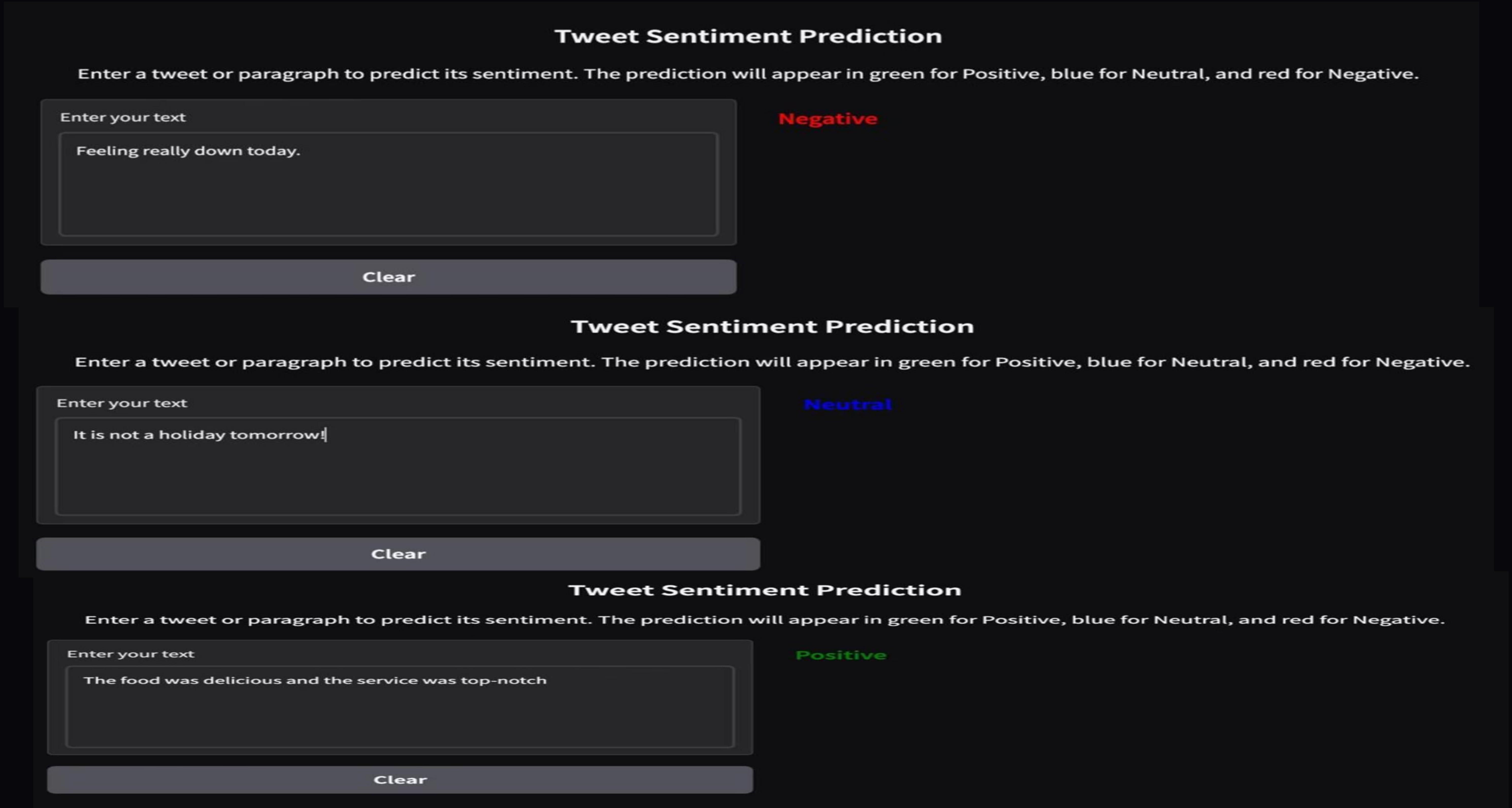
RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) is highly suitable for sentiment analysis. It is a transformer-based model that excels in understanding contextual relationships in text. RoBERTa's pretraining on a large corpus and its ability to fine-tune for specific tasks ensures state-of-the-art performance in classifying sentiments in text like tweets. Its robustness to diverse linguistic patterns makes it ideal for social media sentiment analysis.

```
Some weights of RobertaForSequenceClassification were not initialized from the model checkpoint at roberta-base and
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
Epoch 1, Loss: 0.6581155532073593
Epoch 2, Loss: 0.5344152894342831
Epoch 3, Loss: 0.47582745651837866
Epoch 4, Loss: 0.4197073759486712
Epoch 5, Loss: 0.36235465985903137
              precision    recall  f1-score   support

    negative       0.77      0.78      0.78      1572
     neutral       0.76      0.73      0.74      2236
    positive       0.81      0.83      0.82      1688

    accuracy                           0.78      5496
   macro avg       0.78      0.78      0.78      5496
weighted avg       0.78      0.78      0.78      5496
```

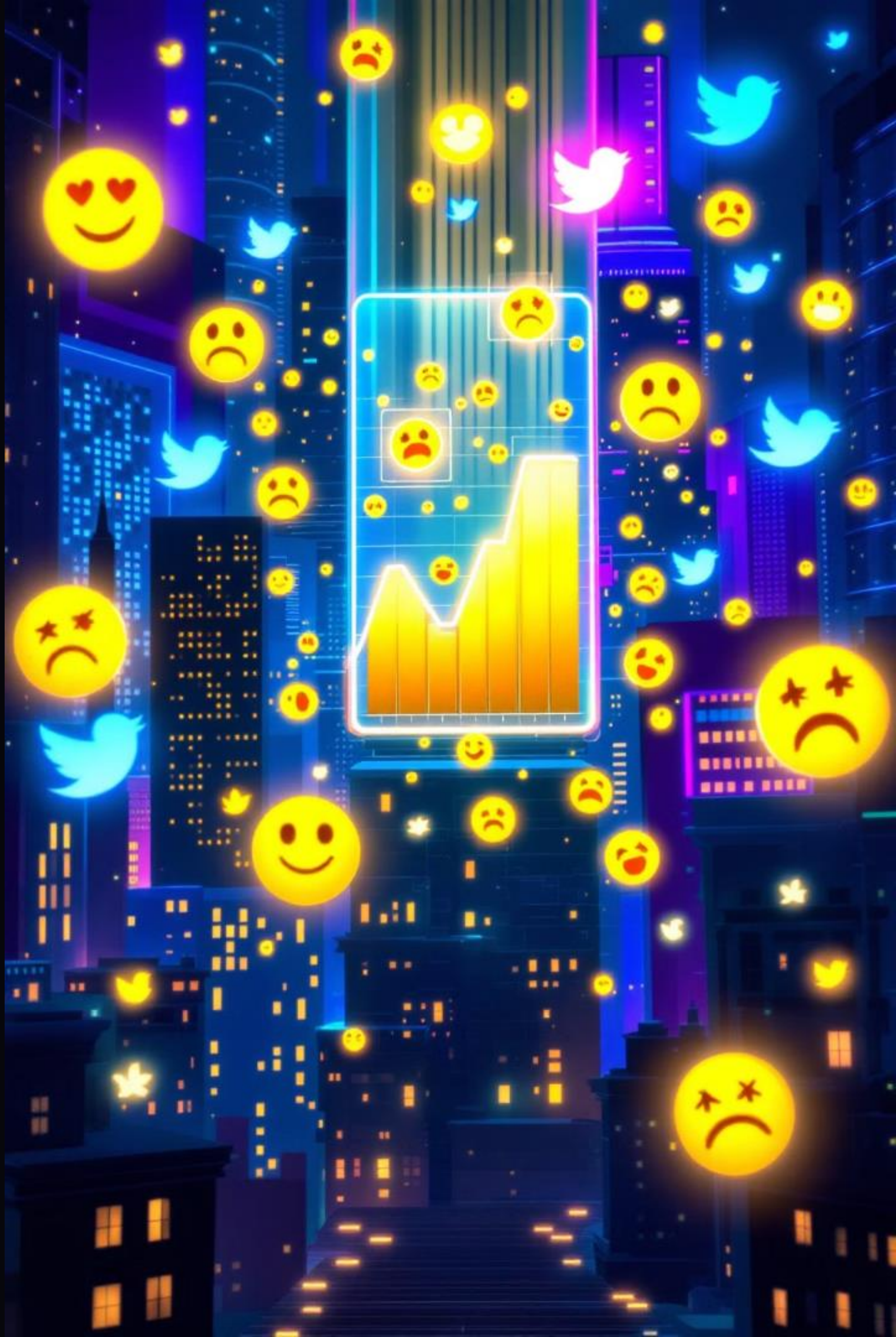
The RoBERTa model achieves excellent performance with an overall accuracy of **78%** and balanced precision, recall, and F1-scores (0.78) across all sentiment classes. It performs particularly well for the **positive class** (F1-score: 0.82) and maintains consistent performance for all sentiments.

Deployment and Monitoring (Gradio)



Conclusions

Twitter sentiment analysis offers invaluable insights into public opinion. This project highlights the importance of meticulous data preparation, choosing the right model, and ongoing monitoring for optimal results. Machine learning empowers us to understand and interpret the complex world of social media.



Questions?