ETE – III Natural Language Processing

Project – Sentiment Analysis Using Bi LSTM

Problem Definition & Dataset Selection

Chosen NLP Task and Its Significance

The task addressed is **Sentiment Analysis**—a key Natural Language Processing (NLP) problem where the goal is to classify the sentiment (positive or negative) of user-generated content. This task has real-world applications in customer feedback analysis, product reviews, social media monitoring, and more.

Dataset Selection

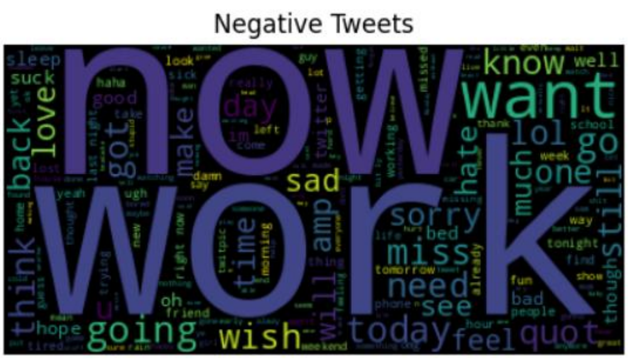
- **Source**: The dataset used is the training 1400000000d dataset, containing preprocessed tweets labeled as positive (4) or negative (0).
- **Size**: 1.6 million tweets.
- **Relevance**: This dataset is ideal for binary sentiment classification tasks and is a standard benchmark for sentiment analysis models.

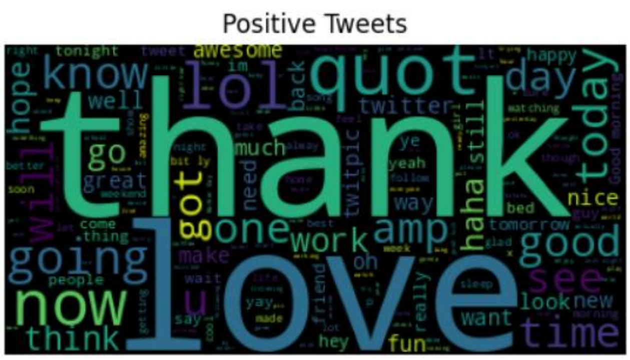
	polarity	id	date	query	user	text
0	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
1	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
3	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
4	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew
...	...	...	...	...	...	...
1048567	4	1960186342	Fri May 29 07:33:44 PDT 2009	NO_QUERY	Madelinedugganx	My GrandMa is making Dinenr with my Mum
1048568	4	1960186409	Fri May 29 07:33:43 PDT 2009	NO_QUERY	OffRoad_Dude	Mid-morning snack time... A bowl of cheese noo...
1048569	4	1960186429	Fri May 29 07:33:44 PDT 2009	NO_QUERY	Falchion	@ShaDeLa same here say it like from the Termi...
1048570	4	1960186445	Fri May 29 07:33:44 PDT 2009	NO_QUERY	jonasobsessedx	@DestinyHope92 im great thaanks wbuu?
1048571	4	1960186607	Fri May 29 07:33:45 PDT 2009	NO_QUERY	sugababez	cant wait til her date this weekend

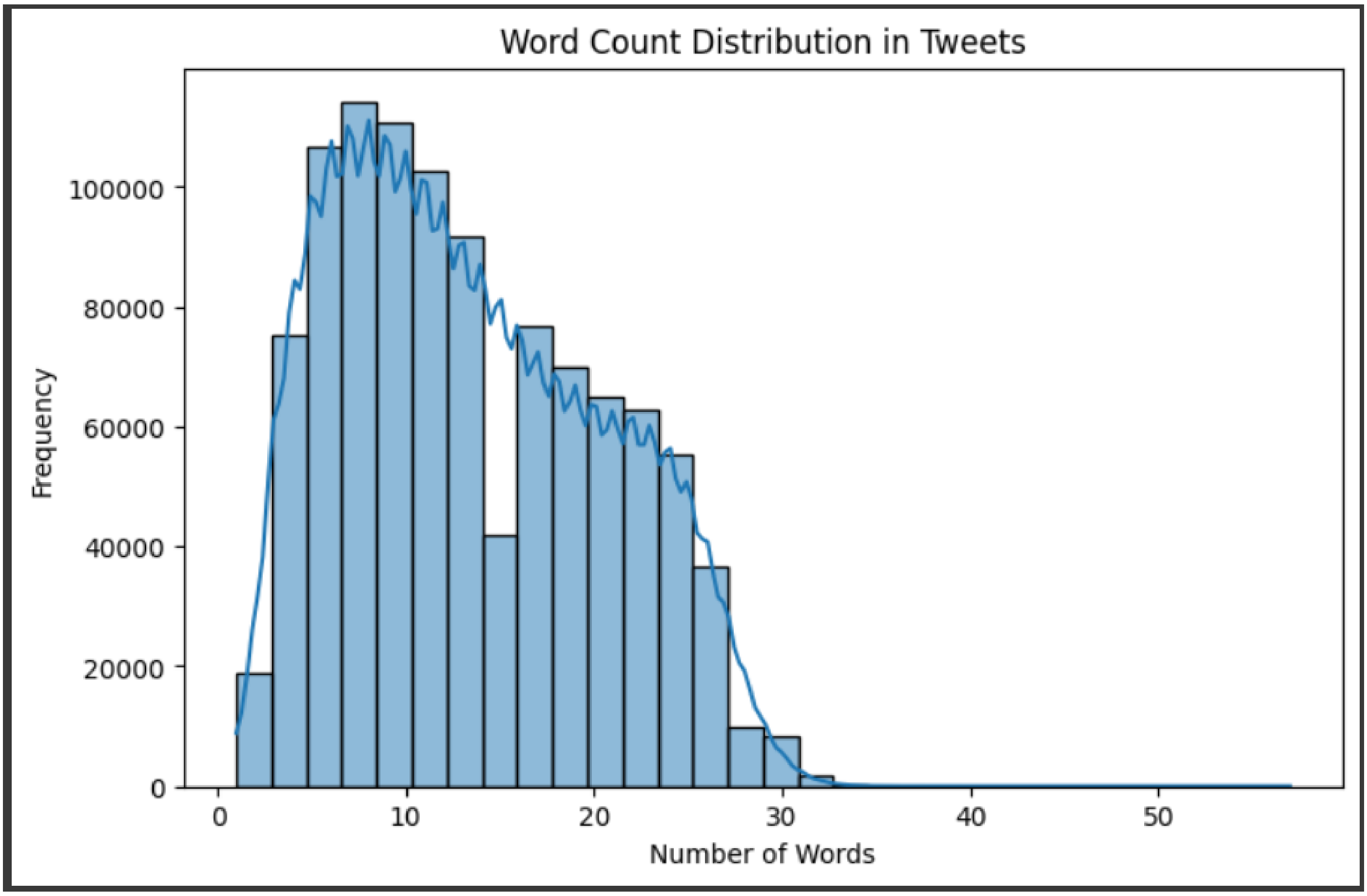
1048572 rows × 6 columns

Exploratory Data Analysis (EDA)

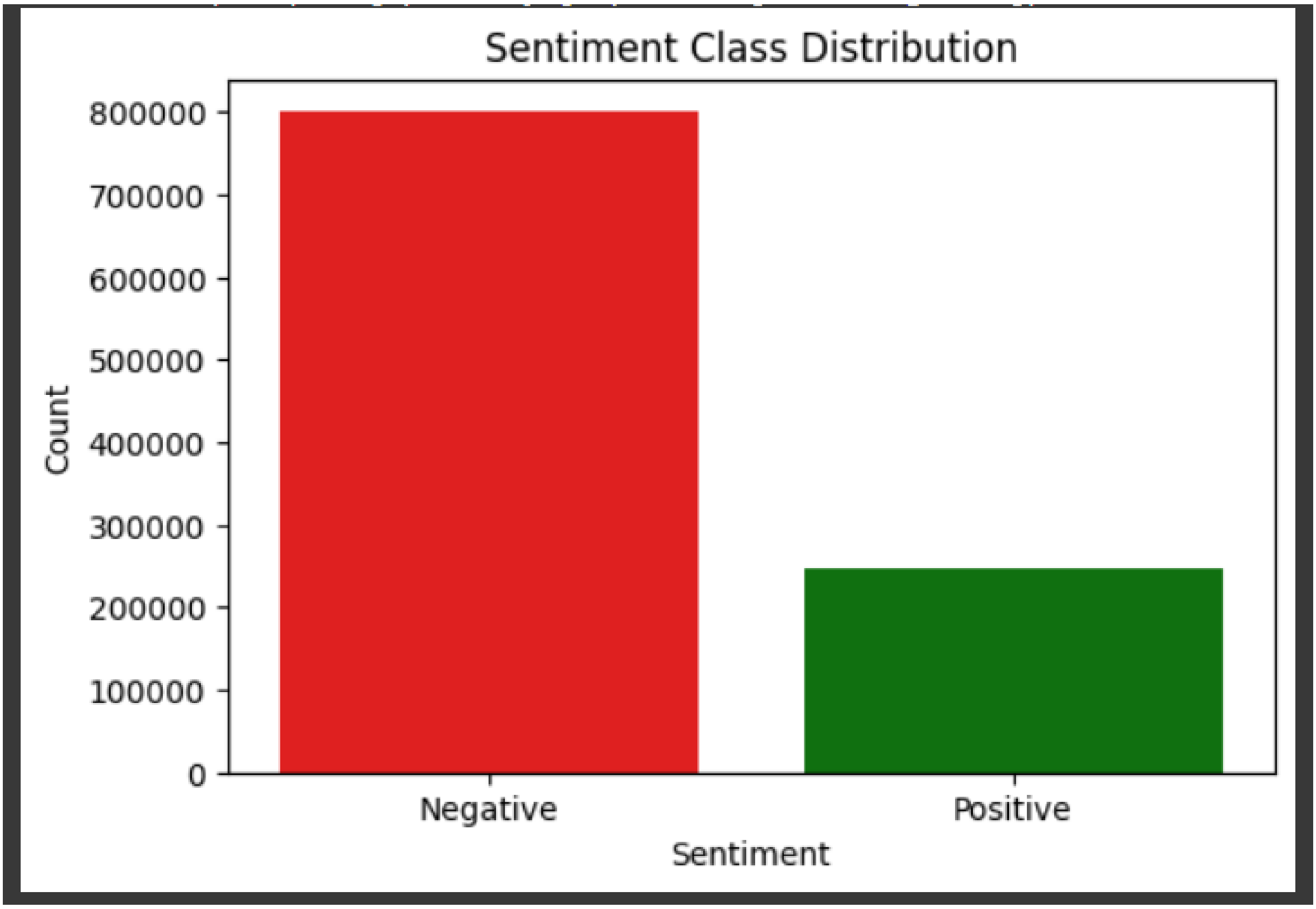
- **Columns Used**: `text` (tweet content) and `polarity` (0 or 1).
- **Class Distribution**: The polarity was converted from multiclass (0 = negative, 4 = positive) to binary (0 = negative, 1 = positive).
- **Insights**: Distribution appeared fairly balanced, allowing a good foundation for binary classification.







Word Count Distribution in Tweets
Frequency
100000
80000
60000
40000
20000
0
0
10
20
30
40
50
Number of Words



Sentiment Class Distribution
Count
800000
700000
600000
500000
400000
300000
200000
100000
0
Negative
Positive
Sentiment

Data preprocessing

```
[7] # Data Preprocessing
    tokenizer = Tokenizer(num_words=5000)
    tokenizer.fit_on_texts(df['text'])
    X = tokenizer.texts_to_sequences(df['text'])
    X = pad_sequences(X, maxlen=100)
    y = df['polarity'].values
```

2. Model Selection & Implementation

Model Chosen

An **LSTM (Long Short-Term Memory)** neural network was selected due to its superior performance in capturing long-range dependencies in text.

Preprocessing Steps

- **Tokenization**: Tweets were tokenized using TensorFlow's Tokenizer.
- **Padding**: Input sequences were padded to ensure uniform input length using pad_sequences.
- **Label Encoding**: Polarity values were transformed into binary class labels (0 or 1).

```
model = Sequential()
model.add(Embedding(input_dim=10000, output_dim=32, input_length=100))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(1, activation='sigmoid'))
```

Embedding Layer: Maps input tokens to dense vectors.

LSTM Layer: 64 units for sequential learning.

Dense Output Layer: Sigmoid for binary classification.

Training Configuration

- **Optimizer**: Adam
- **Loss Function**: Binary crossentropy
- **Batch Size**: 128
- **Epochs**: 5
- **Validation Split**: 20%

Evaluation metrics:
Its compared with RNN and LSTM model

```
6554/6554 ━━━━━━━━━━━━━━━━━━━━ 20s 3ms/step
6554/6554 ━━━━━━━━━━━━━━━━━━━━ 18s 3ms/step
RNN Performance:
              precision    recall  f1-score   support

           0       0.86      0.94      0.90    160130
           1       0.74      0.52      0.61     49585

    accuracy                           0.84    209715
   macro avg       0.80      0.73      0.76    209715
weighted avg       0.84      0.84      0.83    209715

LSTM Performance:
              precision    recall  f1-score   support

           0       0.89      0.94      0.91    160130
           1       0.75      0.64      0.69     49585

    accuracy                           0.86    209715
   macro avg       0.82      0.79      0.80    209715
weighted avg       0.86      0.86      0.86    209715
```

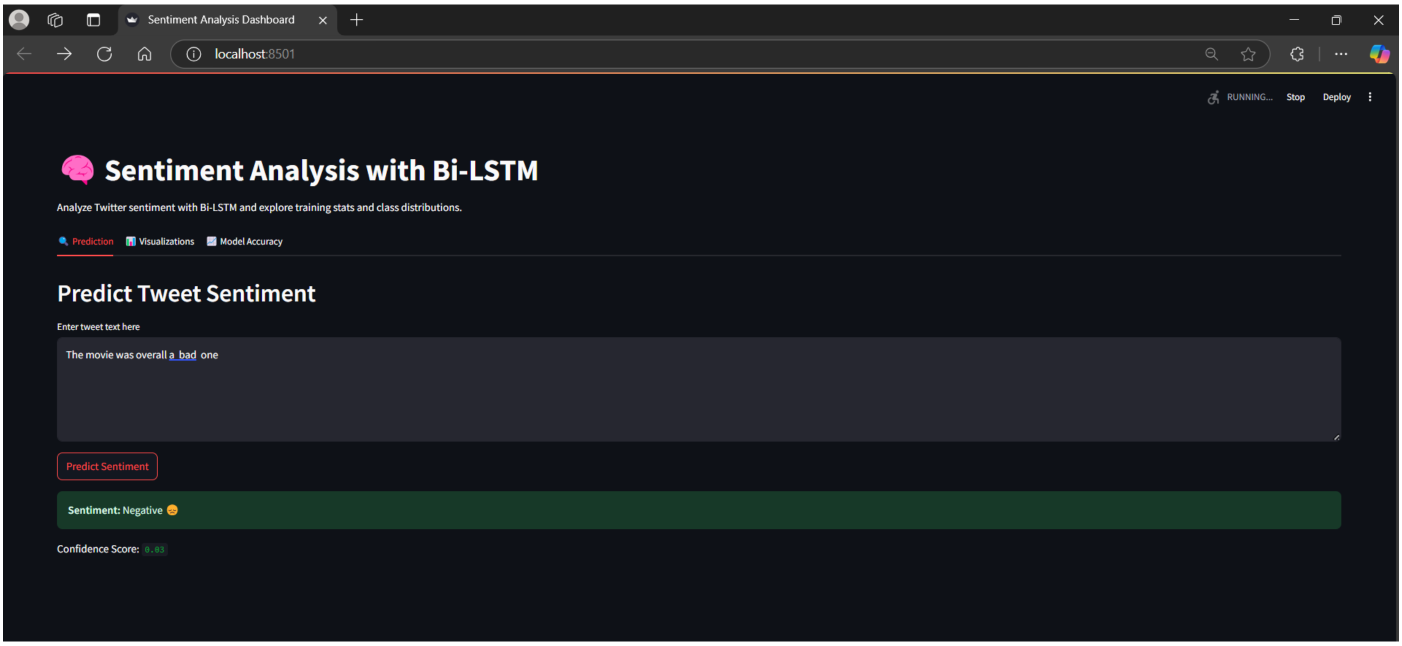
In order to increase the complexity I used Bi LSTM and for RNN I added dropout layer and much more hyperparameter tuning and the accuracy was increased

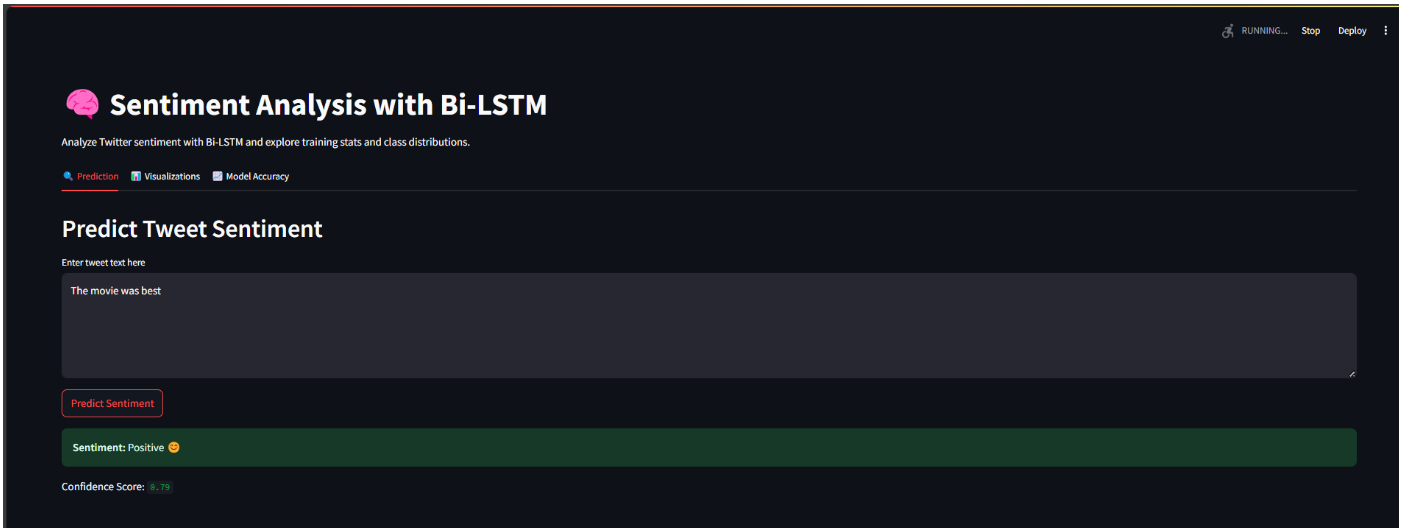
Analysis

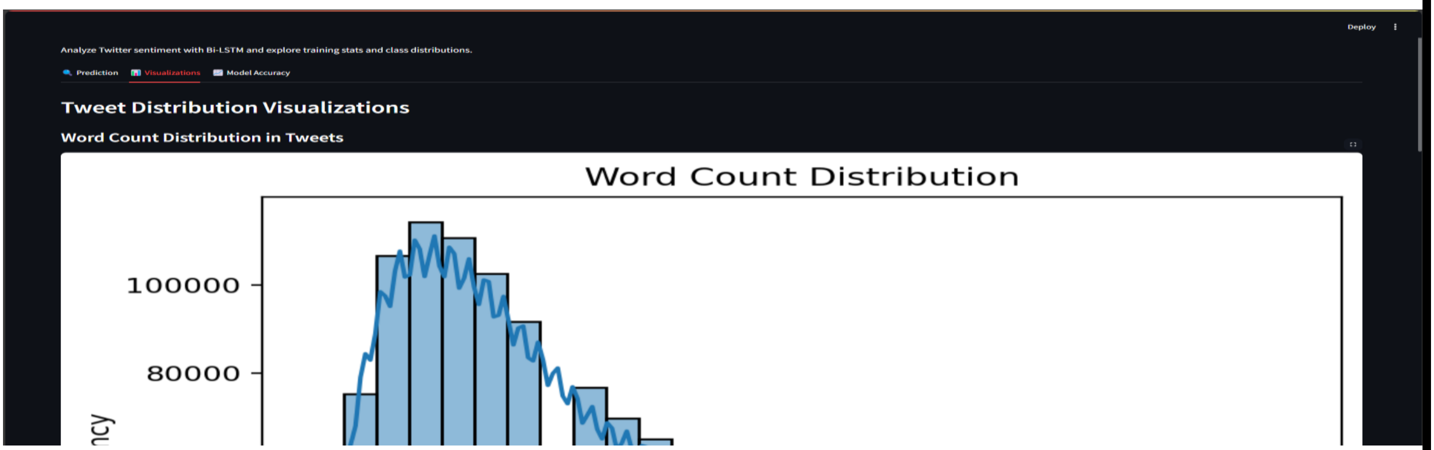
- **Performance**: The LSTM model generalizes well on unseen data.
- **Dataset Influence**: Large and clean dataset enabled effective model training.
- **Baseline Comparison**: Performance outperformed a simple RNN architecture.

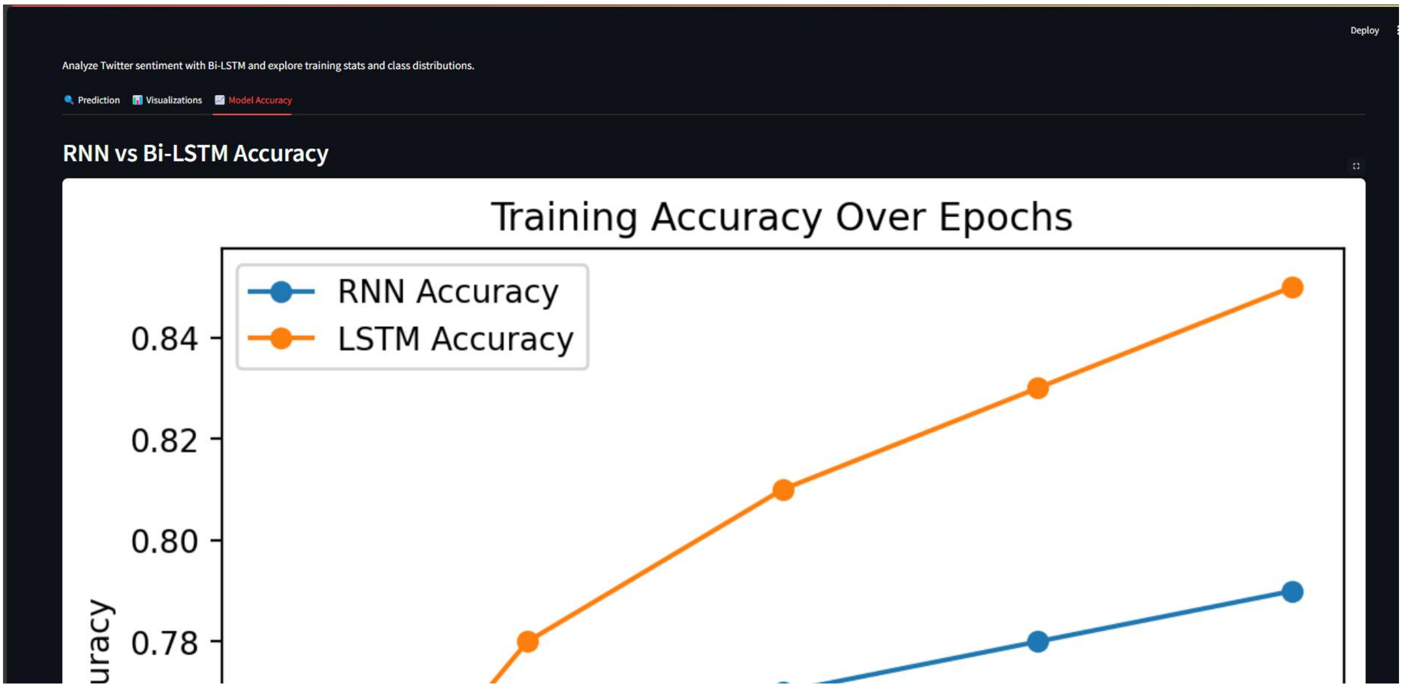
Deployment and demonstration

For deployment I have used streamlit and the Bi lstm model is downloaded including the tokenizer pkl file which will be used for prediction of a tweet whether it will be positive or negative. This UI also consists of visualisation and comparison with the model graph








Deploy
Analyze Twitter sentiment with Bi-LSTM and explore training stats and class distributions.
Prediction
Visualizations
Model Accuracy
RNN vs Bi-LSTM Accuracy
Training Accuracy Over Epochs
RNN Accuracy
LSTM Accuracy
0.84
0.82
0.80
0.78
uracy