Twitter Sentiment Analysis

Submitted By:

Muhammad Talha 01-136221-055

Huzaib Khan 01-136221-036

Introduction

- Sentiment analysis detects emotions, opinions, and attitudes in text, classifying them as positive or negative.
- Industries use it to analyze customer opinions on products and services,
- especially through data from social media platforms like Twitter.
- As a popular microblogging site with millions of users,
- Twitter allows people to share emotions and thoughts through tweets.
- Using NLP techniques like tokenization and text cleaning,
- sentiment analysis helps track trends and understand public sentiment on specific topics effectively.

Dataset

- https://www.kaggle.com/datasets/kazanova/sentiment140
- Word limit of single tweet has 140 characters

Problem description/definition:

- To devise a sentimental analyzer for overcoming the challenges to identify the twitter tweets text sentiments (positive, negative) by implementing neural network using tensorflow.
- Ambiguity and Context Dependence: Words or phrases can have different meanings based on context (e.g., "This movie is sick" can mean either good or bad).
- Sarcasm and Irony: Detecting sarcasm is difficult because the literal meaning often contrasts with the intended sentiment.
- Domain-Specific Language: Sentiment words may vary by domain (e.g., "cold" might be negative in weather discussions but neutral in medical contexts).
- Spelling and Grammatical Errors: Social media data often contains typos, abbreviations, or slang that complicates analysis.

Solution

- Combined positive and negative tweets.
- We will convert the text into lowercase for further processing of tweet text.
- We will clean and remove stop words (of, a, in, etc.) from statements

because these words are not useful for supporting the labels of sentiment data.

- We will clean and remove punctuation marks as they are noise in the data and not meaningful.
- We will clean and remove repeating characters in the words.
- We will clean and remove emails.
- We will clean and remove URLs.
- We will clean and remove numbers in the data.
- We will apply tokenization (to separate the sentence into words).
- Applying Stemming lemmatization
- Seperate input features and labels

After preprocess

Removing stop words

Tokenization

Lemmatization

Stemming

```
[ ]  data['text'].head()
```

	text
799999	[love, healthuandpets, u, guys, r, best]
800000	[im, meting, one, besties, tonight, cant, wait...
800001	[darealsunisakim, thanks, twiter, ad, sunisa, ...
800002	[sick, realy, cheap, hurts, much, eat, real, f...
800003	[lovesbroklyn, efect, everyone]

dtype: object

Working

- Step 1 - The input to model is 500 words because these are the number features/words that we extracted above from text of tweets.
- Step 2- Embeddings provide the presentation of words and their relative meanings. Like in this, we are feeding the limit of maximum words, lenght of input words and the inputs of previous layer.
- Step 3- LSTM (long short term memory) save the words and predict the next words based on the previous words. LSTM is a sequance predictor of next coming words.
- Step 4- Dense layer reduce the outputs by getting inputs from Faltten layer. Dense layer use all the inputs of previous layer neurons and perform calculations and send 256 outputs
- Step 5- Activation function is node that is put at the end of all layers of neural network model or in between neural network layers. Activation function help to decide which neuron should be pass and which neuron should fire. So activation function of node defines the output of that node given an input or set of inputs.
- Step 6- Droupout layer drop some neurons from previous layers. why we apply this? We apply this to avoid the overfitting problems. In overfitting, model give good accuracy on training time but not good on testing time.

```
[ ] print('Test set\n  Accuracy: {:0.2f}'.format(accr1[1]))

Test set
  Accuracy: 0.74
```

Accuracy confusion matrix

- These are the evaluation measures to evaluate the performance of the model.

- Dark blue boxes are the correct predictions with the trained model and sky blue boxes shows the wrong predictions.

- 4776 tweets correctly predicted as negative sentiments. 1233 tweets predicted positive sentiments but that were actually negative sentiments.

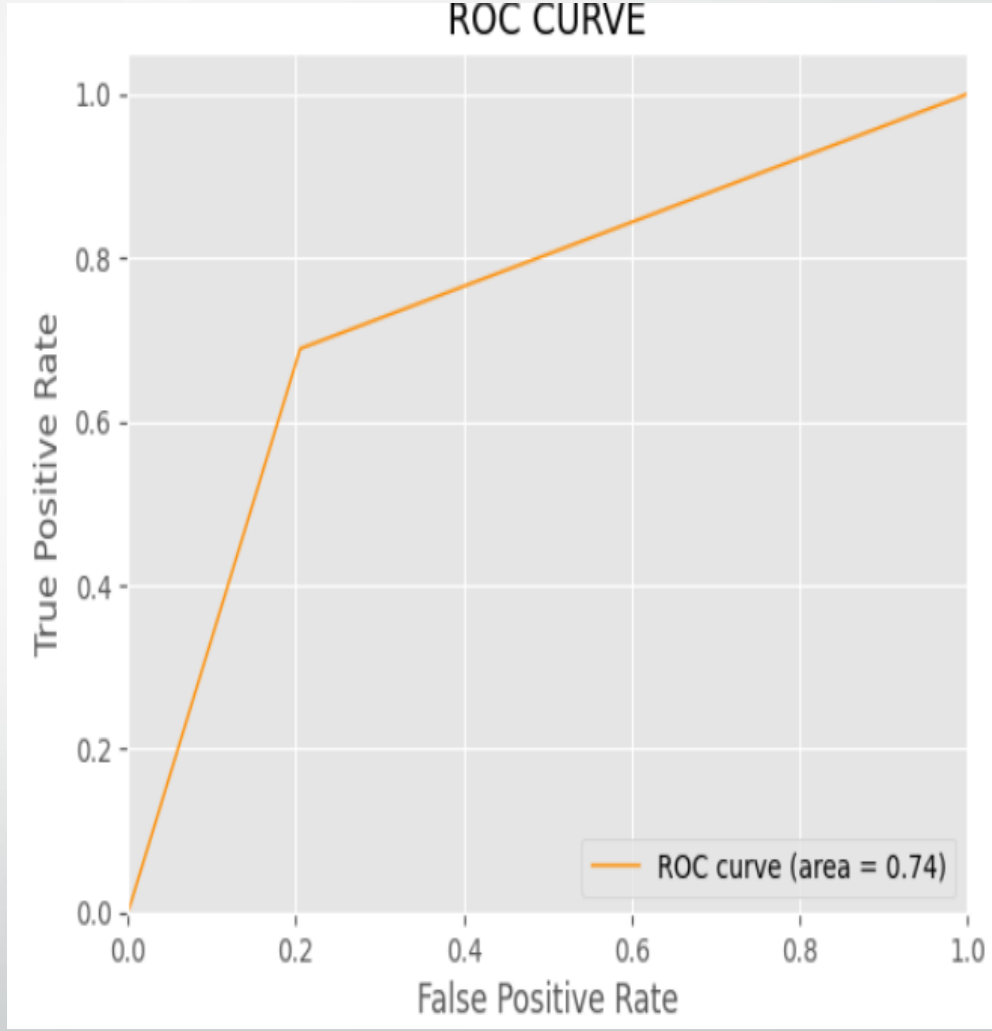
- 4127 tweets correctly predicted as postive sentiments. 1864 tweets predicted negative sentiments but that were actually positive sentiments.

ROC CURVE

- ROC curve show the performance of the model as well.

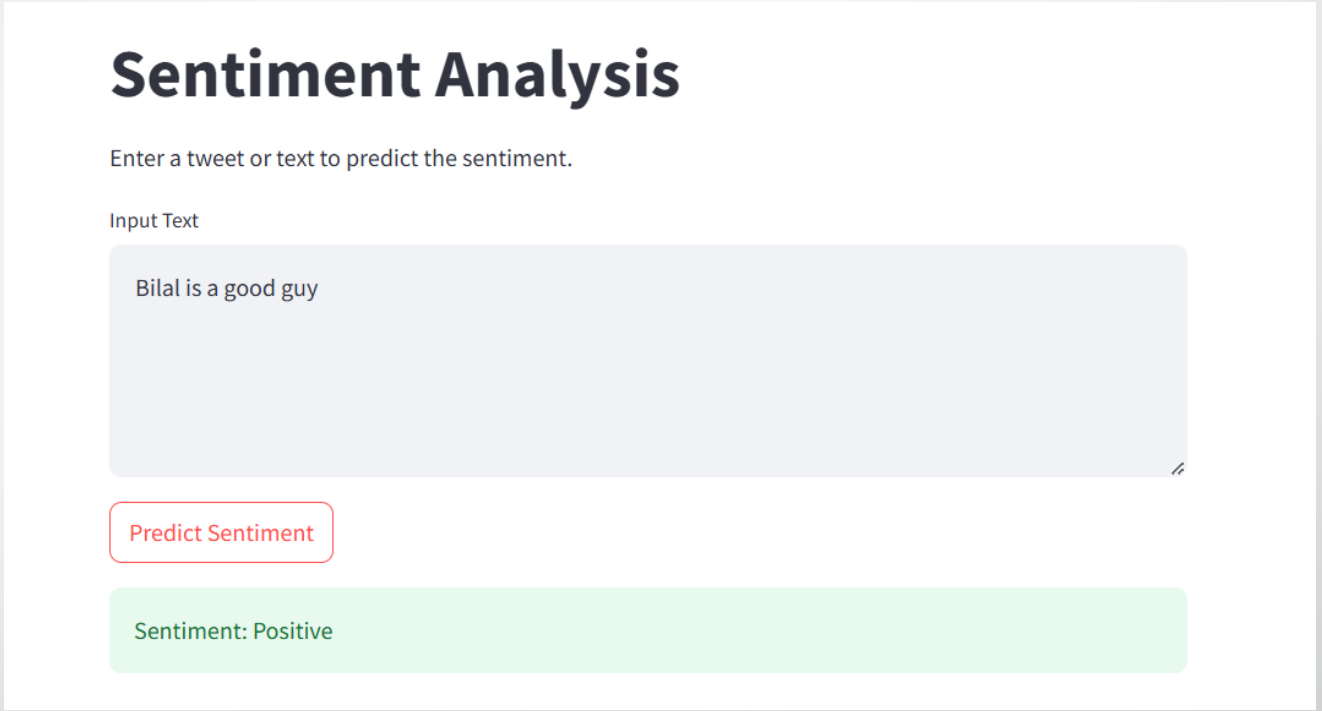
- We can see that the model started from the o percent predictions and then moved to true positive predictions that are correct

- ROC curve (receiver operating characteristic curve) show the performance of a classification model at all the classification thresholds. ROC plots two parameters, True Positive Rate (correct predictions/classifications) False Positive Rate (wrong predictions/classifications)



Final Output

Predicting sentiment anaylsis



Conclusion

- We used the Twitter sentiment analysis dataset and explored the data in different ways.
- We prepared the text data of tweets by removing unnecessary elements.
- We trained a model based on TensorFlow with all settings.
- We evaluated the model with different evaluation measures.
- We worked on the classification problem, specifically binary classification, which is a two-class classification.
- We can use it in tweets to analyze the sentiment of the tweet.