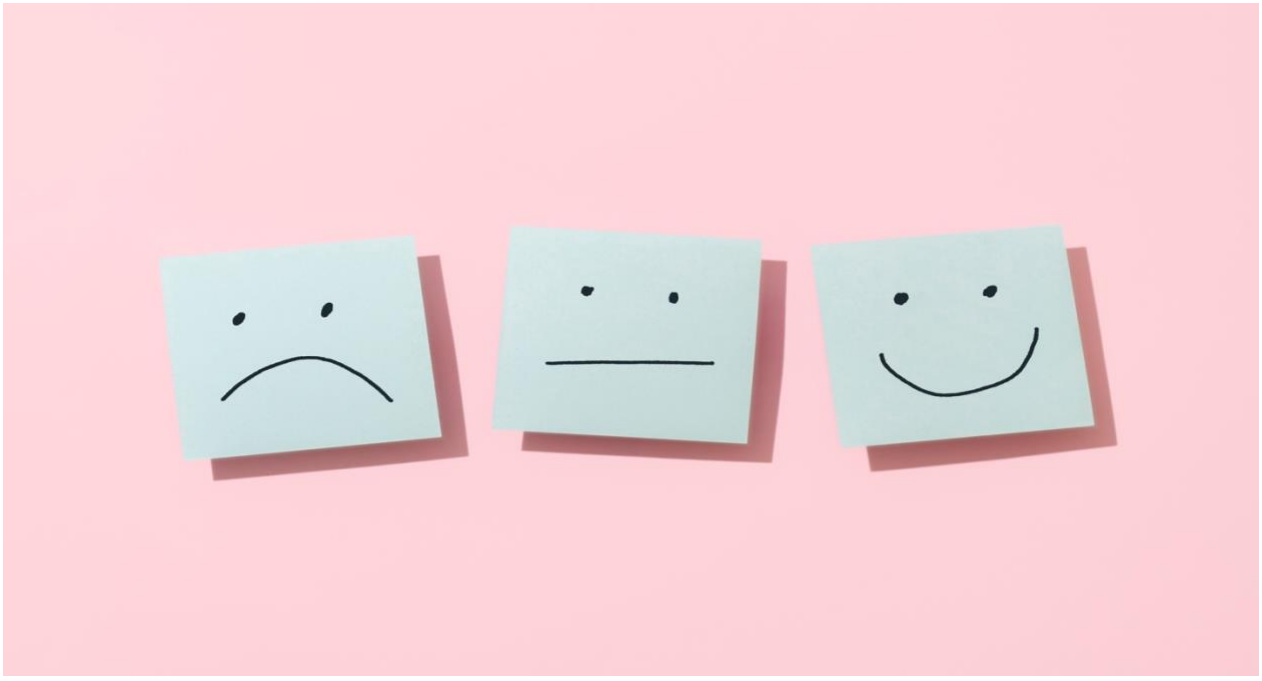
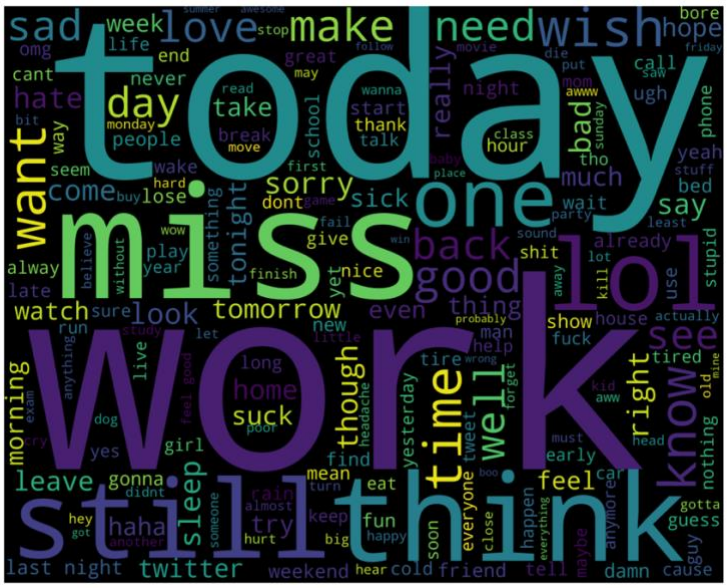
Twitter Sentiment Analysis: Classification Vs Deep Learning Approach



Project MVP by: Himani Kaushik

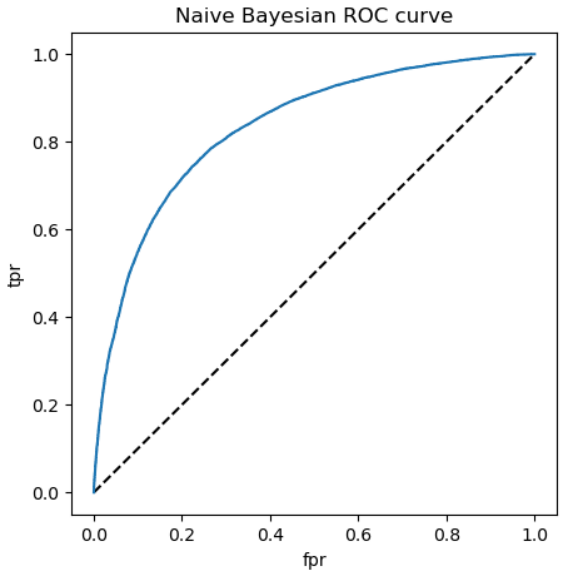
The purpose of the model is to analyze tweets and predict whether it expresses positive or negative sentiment. The model aims to implement and evaluate Naive Bayes model for classification and LSTM model for deep learning, and then select the better performing model.
The data was obtained from the kaggle website - https://www.kaggle.com/datasets/kazanova/sentiment140 . The dataset has 1.6 million tweets extracted using the twitter api. Since the dataset was too huge to work on a regular machine, I trimmed the dataset to 1/4th of its original size. The data balance was maintained while trimming the data.
The data was cleaned and preprocessed using NLTK. I used TweetTokenizer to tokenize the data and split the text based on various criterions well suited for tweets. I used WordNetLemmatizer to lemmatize the text. WordCloud was used to show frequencies of positive and negative words.



Naive Bayesian model was used to build the initial classification model. Following are the results of Naive Bayesian model:

Accuracy on train data: 0.8107305555555555

Accuracy on test data: 0.755575



LSTM Model:

GloVe embeddings were used to convert words to representational vectors. Following model was built:

Model: "sequential"

_____

Layer (type) Output Shape Param #

==========

embedding (Embedding) (None, 25, 50) 20000050

_____

bidirectional (Bidirectional (None, 25, 256) 183296

_____

bidirectional_1 (Bidirection (None, 256) 394240

_____

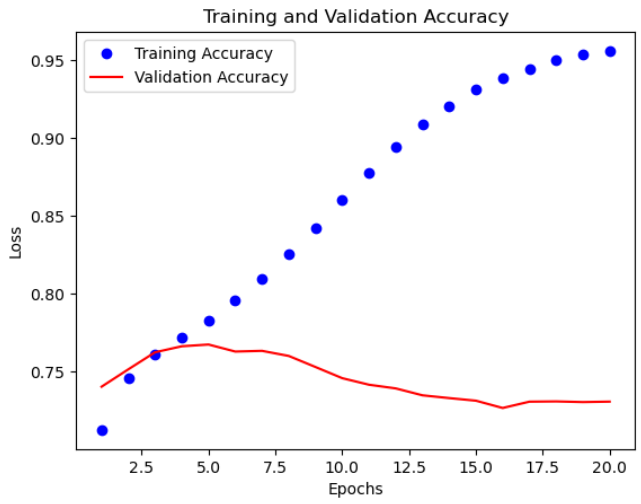
dense (Dense) (None, 1) 257

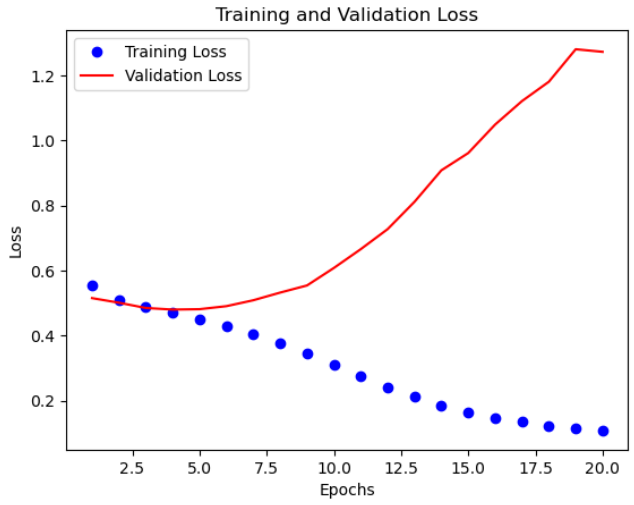
==========

Total params: 20,577,843

Trainable params: 577,793

Non-trainable params: 20,000,050





The model is currently over-fitting. I will continue working to improve the model by using regularization techniques.