

Sentiment Analysis API

Sentiment Analysis is being performed in the given Tweets on Airline. There are two columns namely, "airline_sentiment" and "text". I have preprocessed the data by removing the stop words, removing urls and removing the twitter names. Then we encode the given column data["airline_sentiment"], and further tokenize the data["text"] to pass through our model using the Tokenizer

I have worked on two models, namely

- 1) Naive Bayes Classifier
- 2) LSTM Based Classifier

LSTM performed better than the Naive Bayes Classifier as NB belongs to a category of models called generative. This means that during training (the procedure where the algorithm learns to classify), NB tries to find out how the data was generated in the first place. It essentially tries to figure out the underlying distribution that produced the examples you input to the model.

On the other hand Bi-LSTM is a discriminative model. It tries to figure out what the differences are between the positive and negative examples, in order to perform the classification. The summary of the Bi-LSTM model is provided below

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding_9 (Embedding)	(None, 150, 100)	300000
bidirectional_9 (Bidirectional)	(None, 128)	84480
dense_18 (Dense)	(None, 24)	3096
dense_19 (Dense)	(None, 1)	25
Total params: 387,601		
Trainable params: 387,601		
Non-trainable params: 0		

I have hypertuned by trying out various activation functions in the Dense layers, various loss functions and optimizers, finally stuck with the following

```
model =keras.Sequential([
    keras.layers.Embedding(self.vocab_size, self.embedding_dim,
input_length=self.max_length),
    keras.layers.Bidirectional(keras.layers.LSTM(64)),
    keras.layers.Dense(24, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid')])

model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Finally I have achieved an accuracy around 90%