Sentiment Analysis of IMDB Movie Review Using CNN And LSTM

Srijon Mukhopadhyay
Zhuoya Li

Introduction

Sentiment Analysis has been a very favored task especially in business sectors.

- It provides insights into customer trends
- It may be used for adaptive customer service and training chatbots

Deep learning models are very useful for making use of sentiment analysis tools. Both supervised and unsupervised approaches have been used in the past to implement sentiment analysis.

In this project we'll compare two popular models based on their ability to learn from labeled data and perform classification tasks.

Outline

- Load IMDB movie reviews and analyze the dataset
- Data preprocessing
- Build the deep learning network
    - Create the model
    - Tune hyperparameters
    - Train the model
- Evaluate the model
- Reference

Importing and Analyzing the Dataset

Loading the data

- Training set
  - 20000 rows
- Testing set
  - 5000 rows

- **id -** Unique ID of each review
- **sentiment -** Sentiment of the review
  - 1 for positive reviews
  - 0 for negative reviews
- **review -** Text of the review

Data Preprocessing

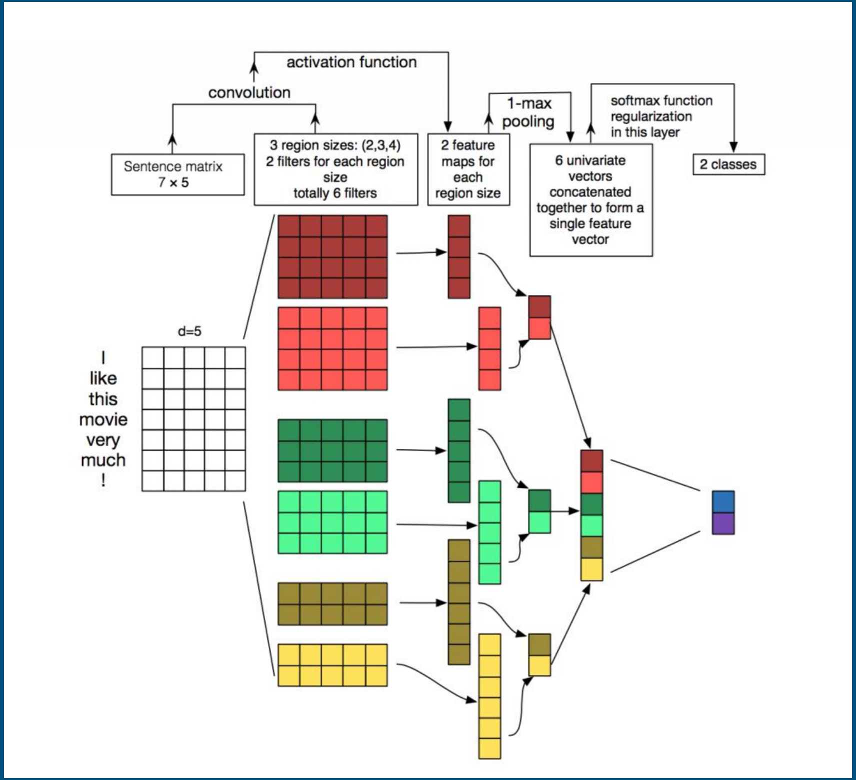
- Text preprocessing steps
    - Take a text string as a parameter
    - Perform preprocessing on the string to remove special characters from the string
        - Remove html tags
        - Remove punctuations and numbers
        - Remove any single characters
        - Remove multiple spaces
- Preprocess our reviews and store them in a new list
- Divide the dataset
    - Training set 80%
    - Testing set 20%

Preparing the embedding layer

- Find the vocabulary size and then perform padding on both train and test set
- Adding 1 because of reserved 0 index
- Load the GloVe word embeddings
- Create a dictionary that will contain words as keys and their corresponding embedding list as values
- Create an embedding matrix where each row number will correspond to the index of the word in the corpus

Deep Learning Network

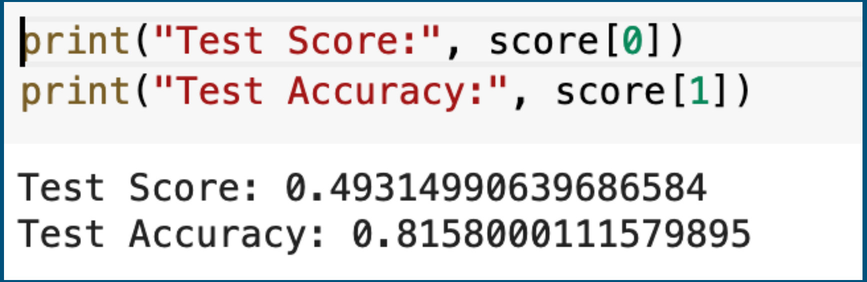
CNN (Convolutional Neural Network)


activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

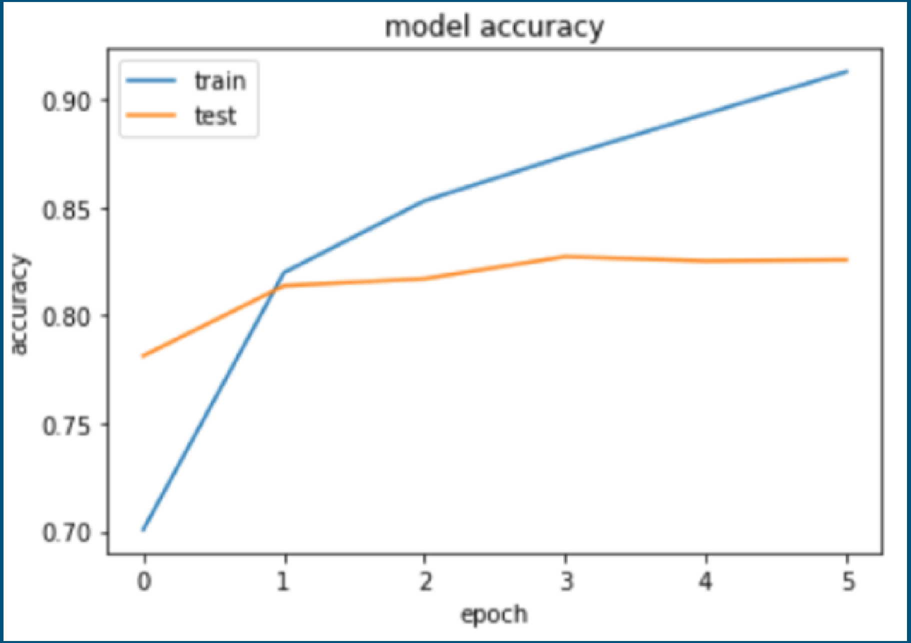
- A CNN is able to use filters that act as word embeddings
    - This allows ngram analysis
- 1-max pooling is performed over each map
- A feature vector can thus be generated from the maps
- The final layer (in our case, a sigmoid layer) then receives this feature vector as input and uses it to classify the input text

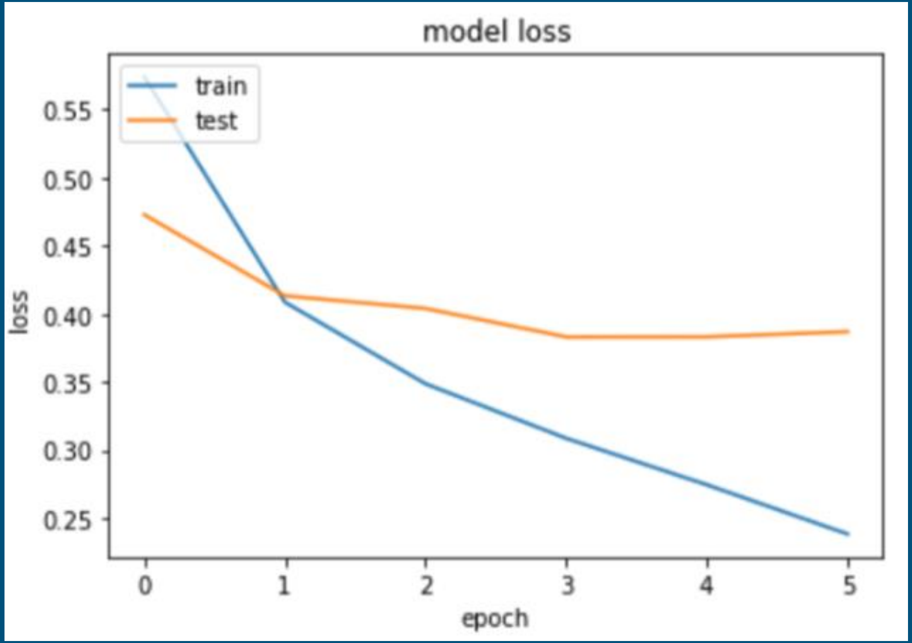
Text Classification with CNN

- Create a simple CNN with 1 convolutional layer and 1 pooling layer
- Train our model and evaluate it on the training set
  - Batch_size = 128
  - Epochs = 6
  - Verbose = 1
  - Validation_split = 0.33

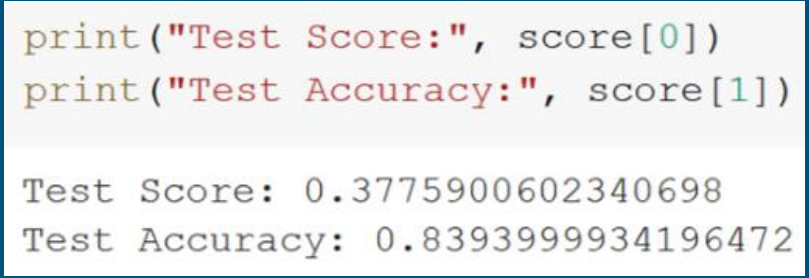


```
print("Test Score:", score[0])
print("Test Accuracy:", score[1])

Test Score: 0.49314990639686584
Test Accuracy: 0.8158000111579895
```


model accuracy
train
test
accuracy
0.90
0.85
0.80
0.75
0.70
epoch
0
1
2
3
4
5

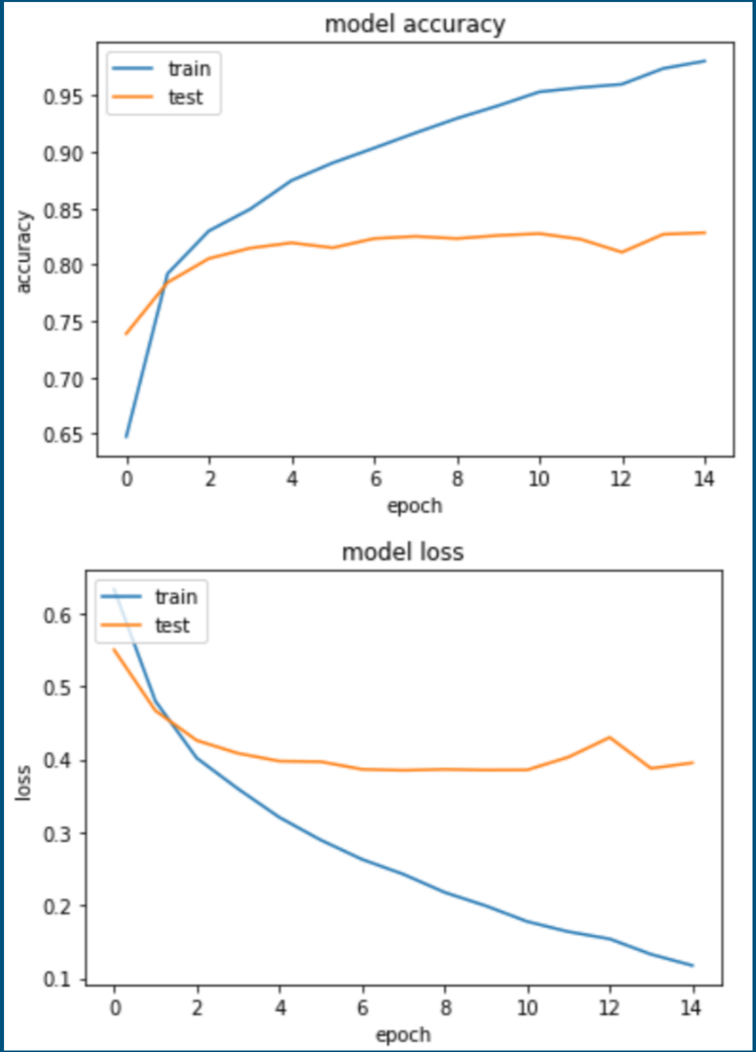

model loss
train
test
loss
0.55
0.50
0.45
0.40
0.35
0.30
0.25
epoch
0
1
2
3
4
5

- Epochs = 15
- Batch_size = 32
- Validation_split = 0.33

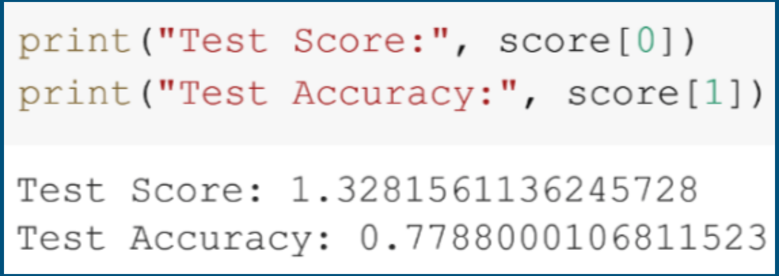


```
print("Test Score:", score[0])
print("Test Accuracy:", score[1])
```

```
Test Score: 0.3775900602340698
Test Accuracy: 0.8393999934196472
```

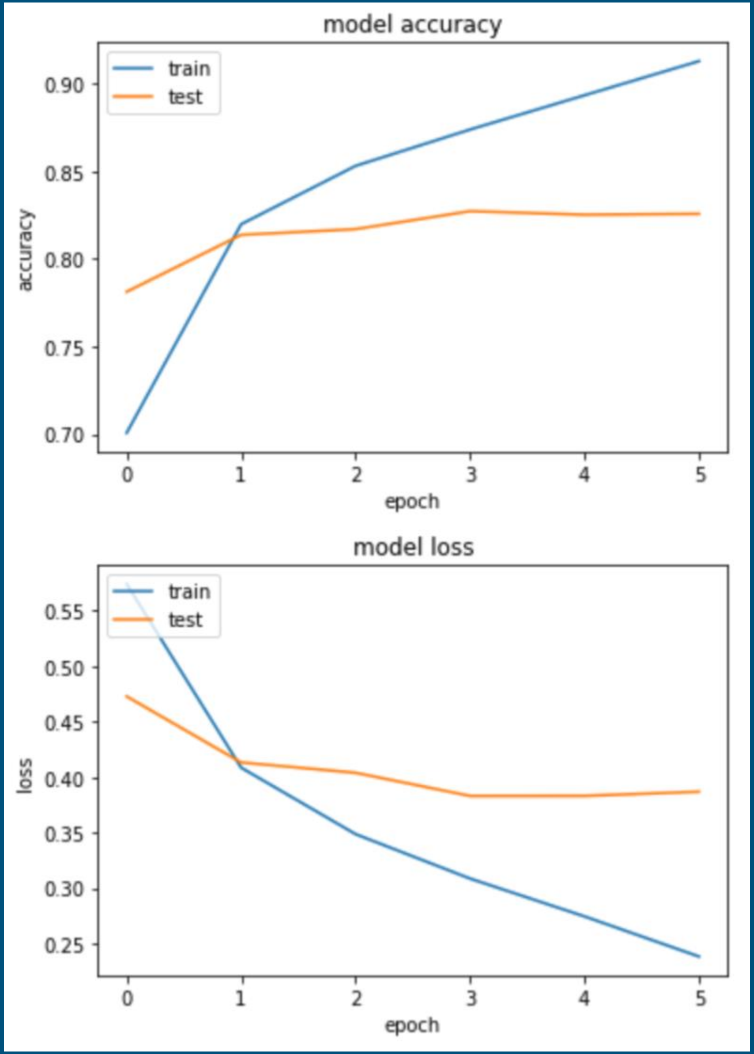


- Learning_rate = 0.01
- Batch_size = 32

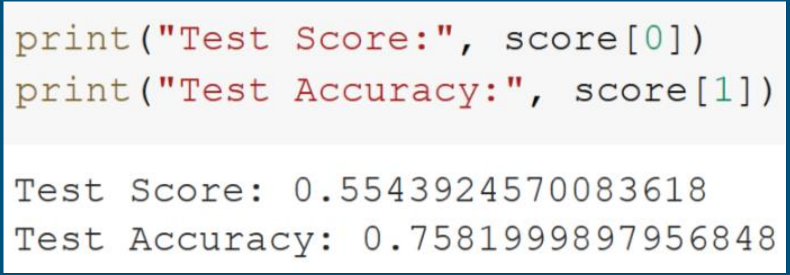


```
print("Test Score:", score[0])
print("Test Accuracy:", score[1])
```

```
Test Score: 1.3281561136245728
Test Accuracy: 0.7788000106811523
```

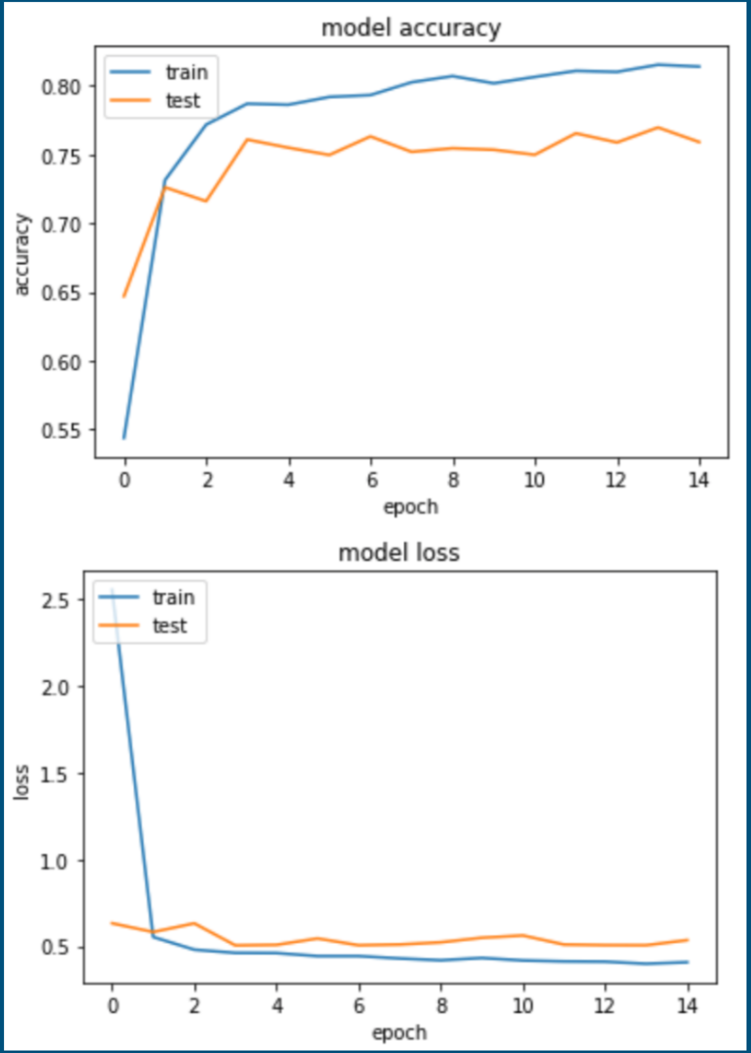


- Learning_rate = 0.1
- Batch_size = 256

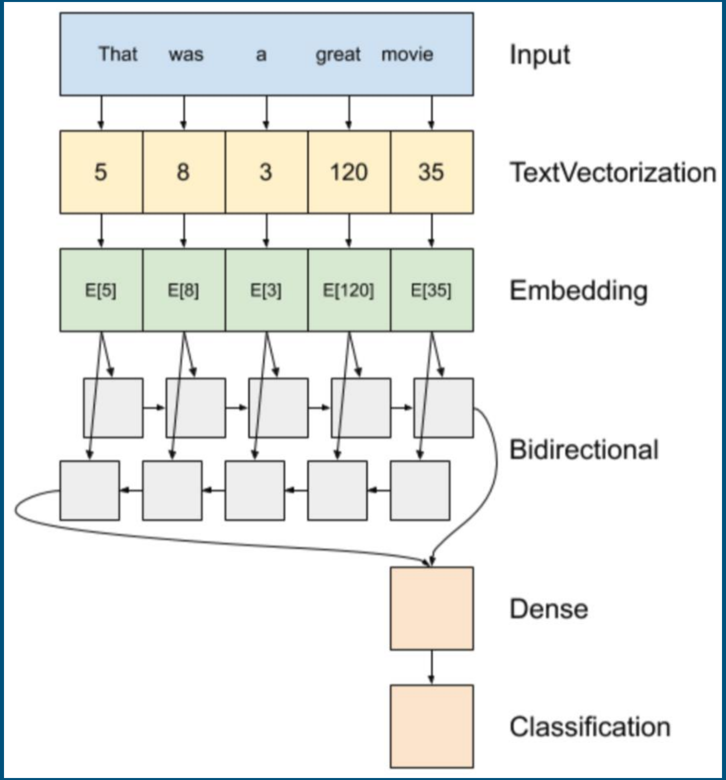


```
print("Test Score:", score[0])
print("Test Accuracy:", score[1])
```

```
Test Score: 0.5543924570083618
Test Accuracy: 0.7581999897956848
```



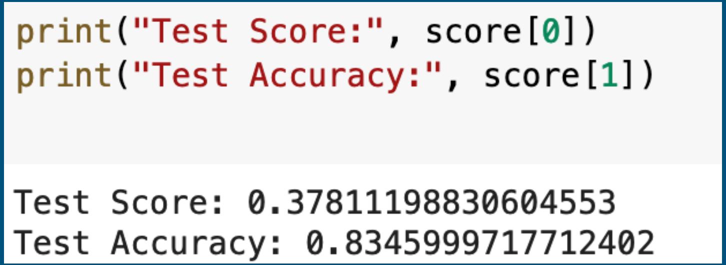
LSTM (Long Short-Term Memory)



- LSTM is a Sequential model
- It is able to save past information about inputs
- An embedding layer provides spatial representations of text
- Bidirectional LSTMS are able to access past inputs while also processing future inputs in reverse direction
  - This is especially useful for modeling text sequences which are usually complex

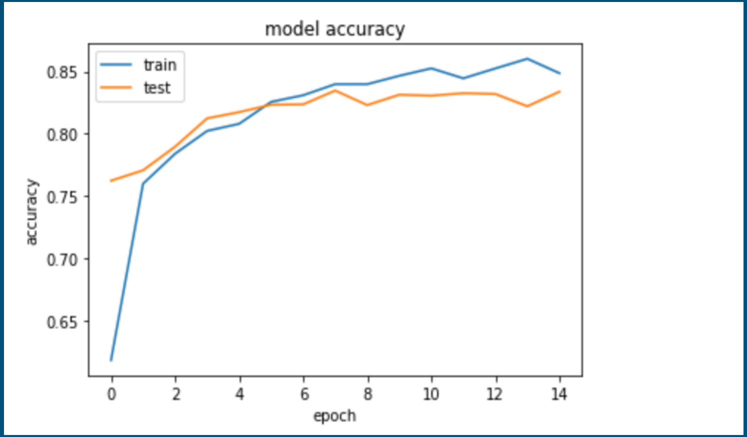
Text Classification with LSTM

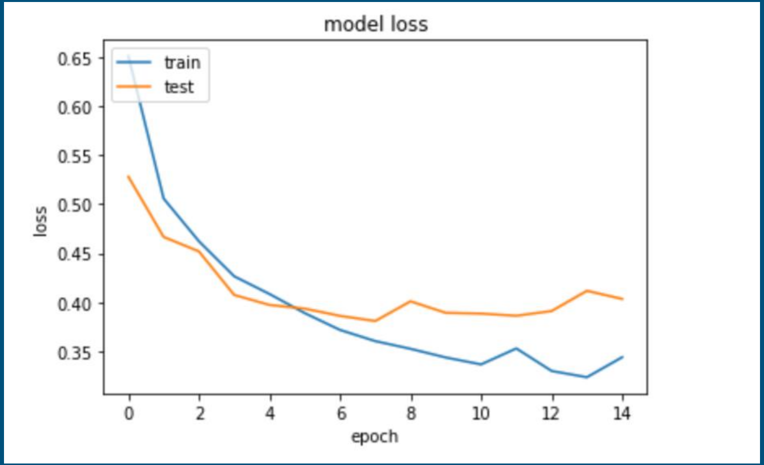
- Create an LSTM model with a Bidirectional LSTM layer fitted with word embeddings
  - Batch size = 128
  - Learning Rate = 0.01
  - Validation Split = 0.33
  - Dropout = 0.3



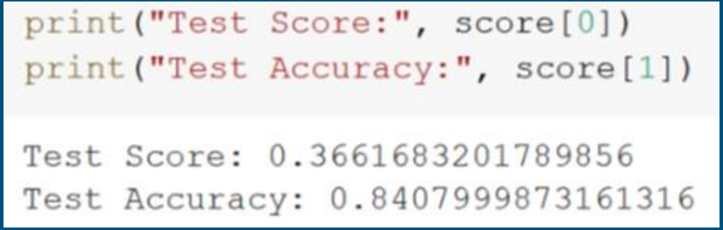
```
print("Test Score:", score[0])
print("Test Accuracy:", score[1])
```

```
Test Score: 0.37811198830604553
Test Accuracy: 0.8345999717712402
```

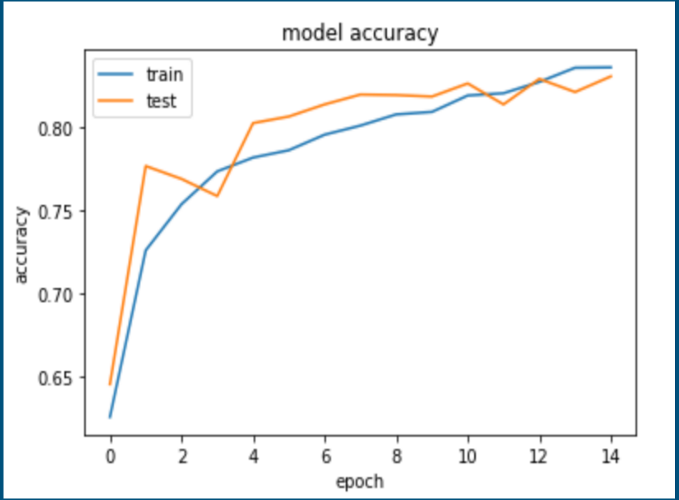

model accuracy
train
test
accuracy
epoch
0.65
0.70
0.75
0.80
0.85
0
2
4
6
8
10
12
14

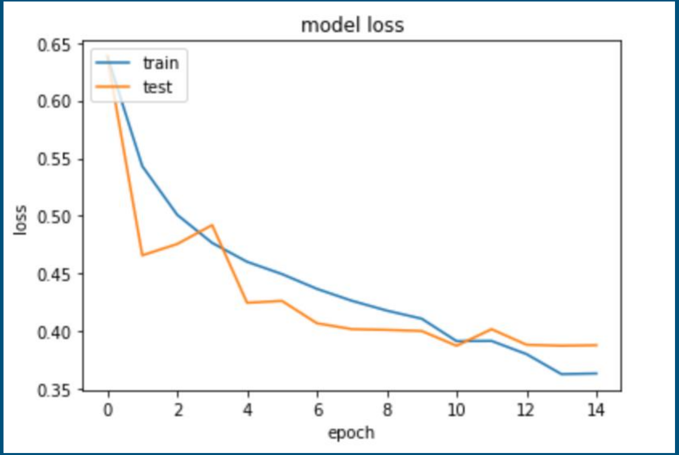

model loss
train
test
loss
epoch
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0
2
4
6
8
10
12
14

- Batch Size: 32
- Learning Rate:0.001

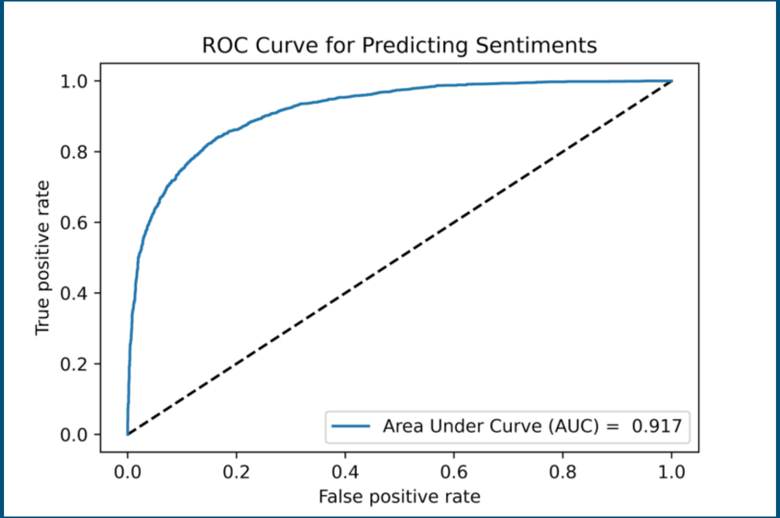


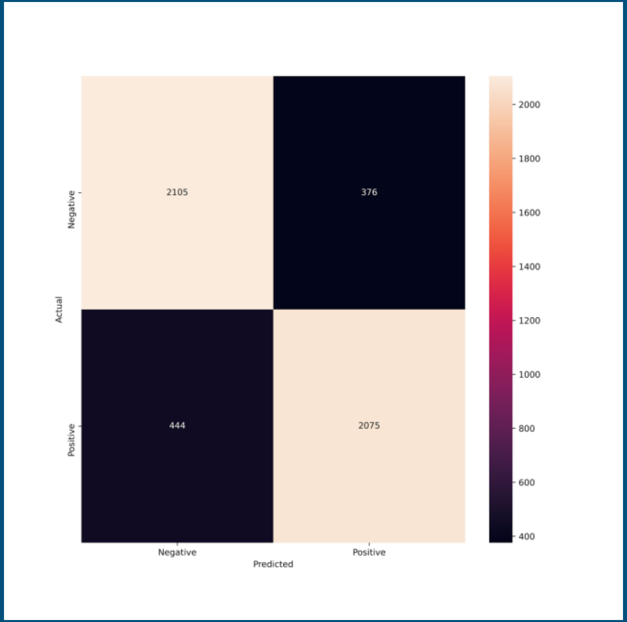
```
print("Test Score:", score[0])
print("Test Accuracy:", score[1])

Test Score: 0.3661683201789856
Test Accuracy: 0.8407999873161316
```





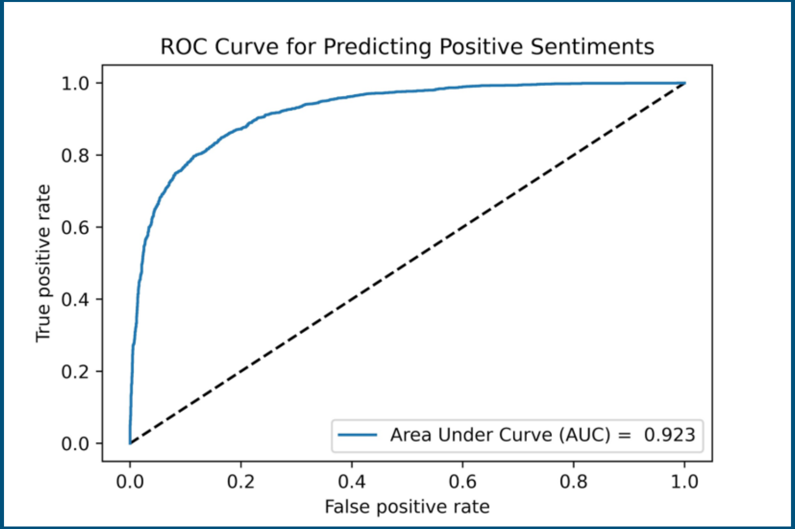
Evaluation - CNN

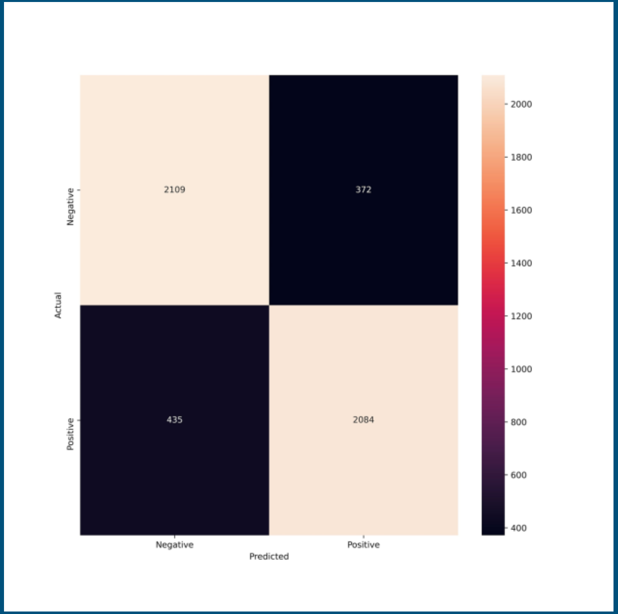




Sentiment	Precision	Recall	F-1 Score
Negative	0.83	0.85	0.84
Positive	0.85	0.82	0.84

Evaluation - LSTM





Labels	Precision	Recall	F1 Score
Negative	0.83	0.85	0.84
Positive	0.85	0.83	0.84

Summary and Conclusions

- Though the F1 scores of both models are the same, the LSTM model edges out the CNN model
  - slightly lesser number of FN and FP errors
  - greater AUC- ROC value
- Both CNN and LSTM models perform better on test set with lower batch sizes and learning rates
- Model can still be improved by making changes
  - Further hyperparameter tuning like decaying learning rate
  - Increasing samples in training set to allow train longer

Reference

http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

https://www.tensorflow.org/tutorials/text/text_classification_rnn

https://machinelearningmastery.com/develop-word-embedding-model-predicting-movie-review-sentiment/

https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e

Thank you!