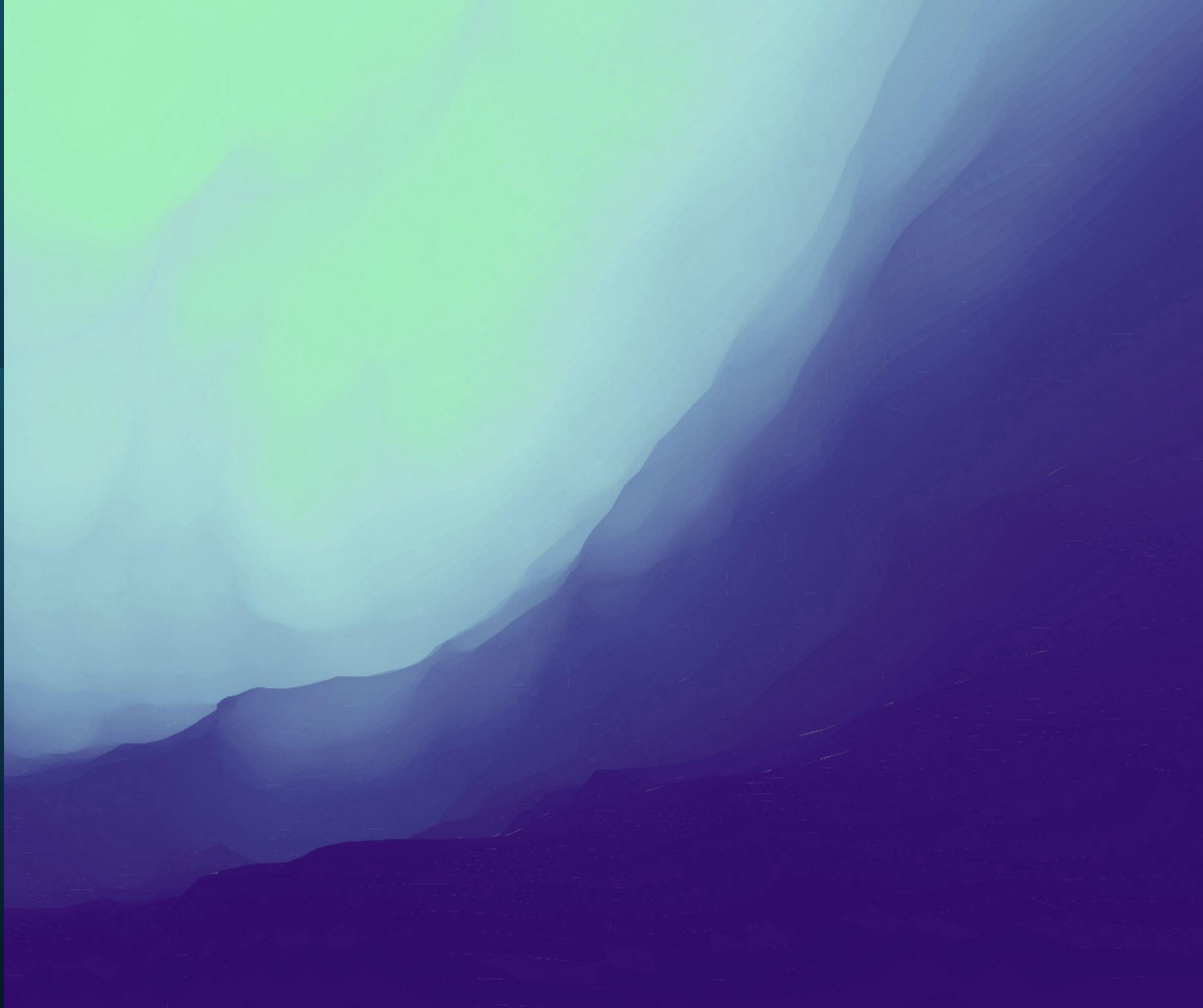
Andrew N

Fake News Detection



Dataset

- Dataset can be found here: https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification
- Collection of various news articles
- Labeled either fake (0) or real (1)
- Only the first 5000 samples are used.
- 52.56% are true
- 47.44% are fake

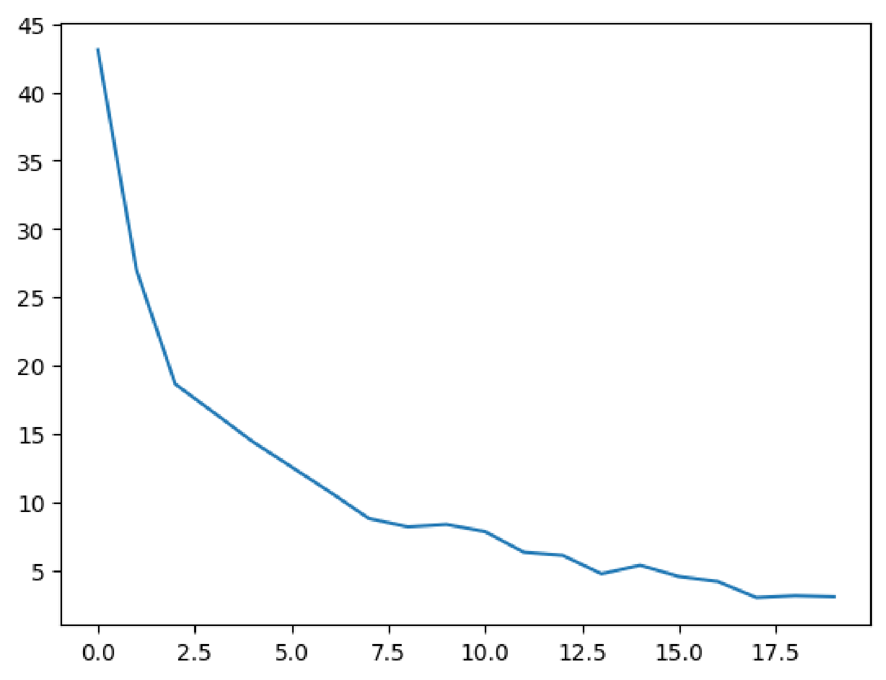
Preprocessing

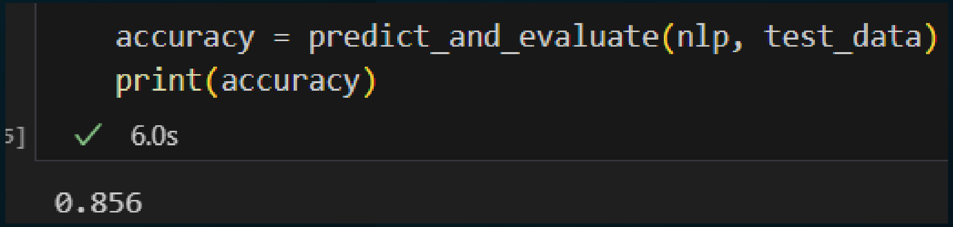
- Remove stop words
- Remove numbers
- Remove extra spaces
- Remove punctuation
- Leave named entities untouched
- Lemmatize the words

Spacy Model

- Standard set up from class
- 0.5 drop out
- Stochastic Gradient Descent

Spacy
Run 1:
Epochs: 20
Batch Size: 16

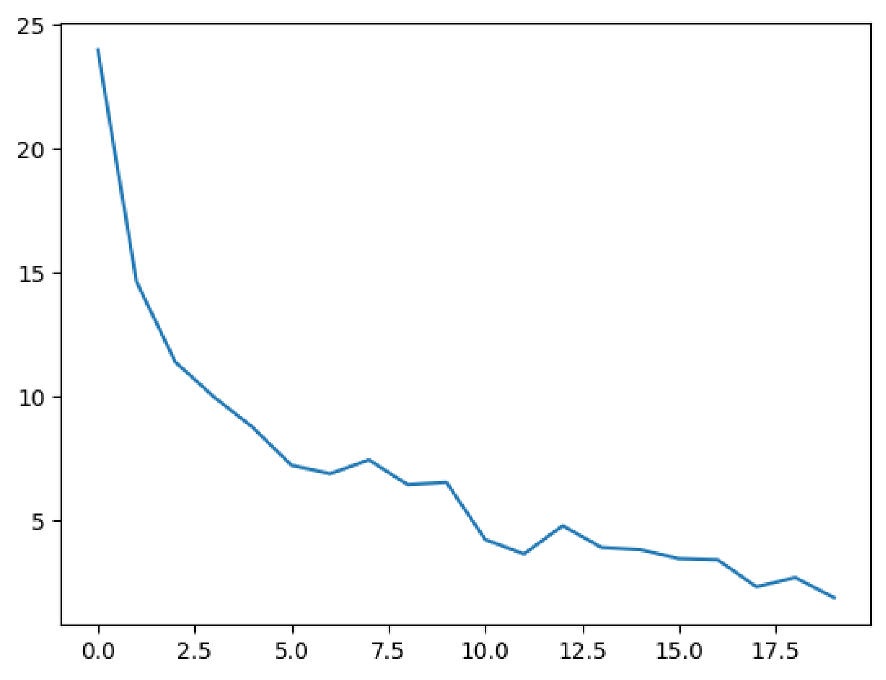


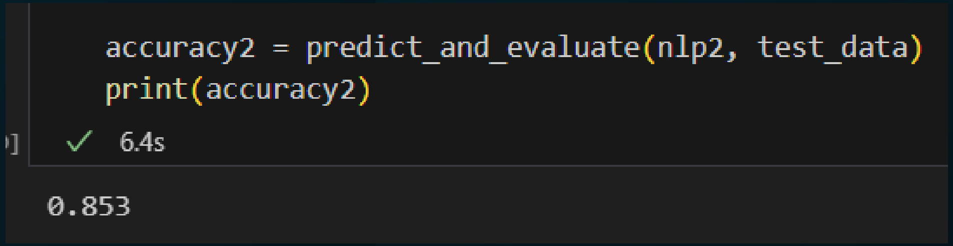


```
accuracy = predict_and_evaluate(nlp, test_data)
print(accuracy)
```

```
0.856
```

Spacy
Run 2:
Epochs: 20
Batch Size: 32



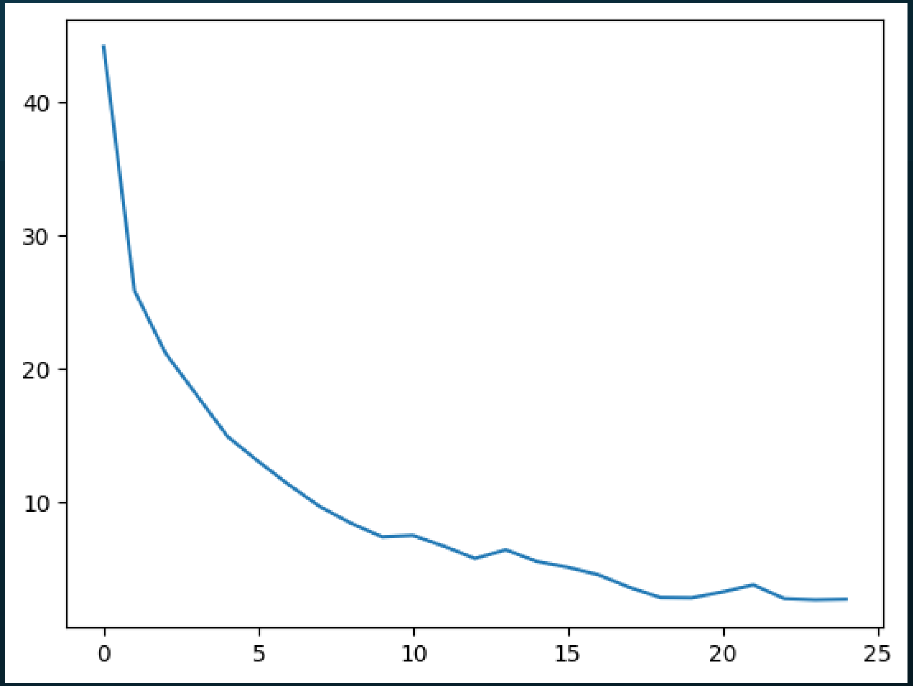


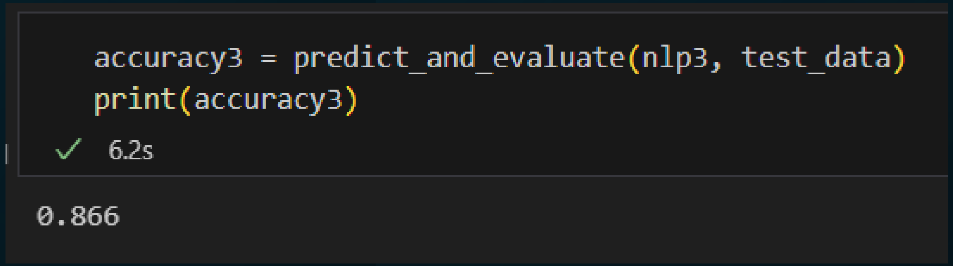
```
accuracy2 = predict_and_evaluate(nlp2, test_data)
print(accuracy2)
```

✓ 6.4s

```
0.853
```

Spacy
Run 3:
Epochs: 25
Batch Size: 16



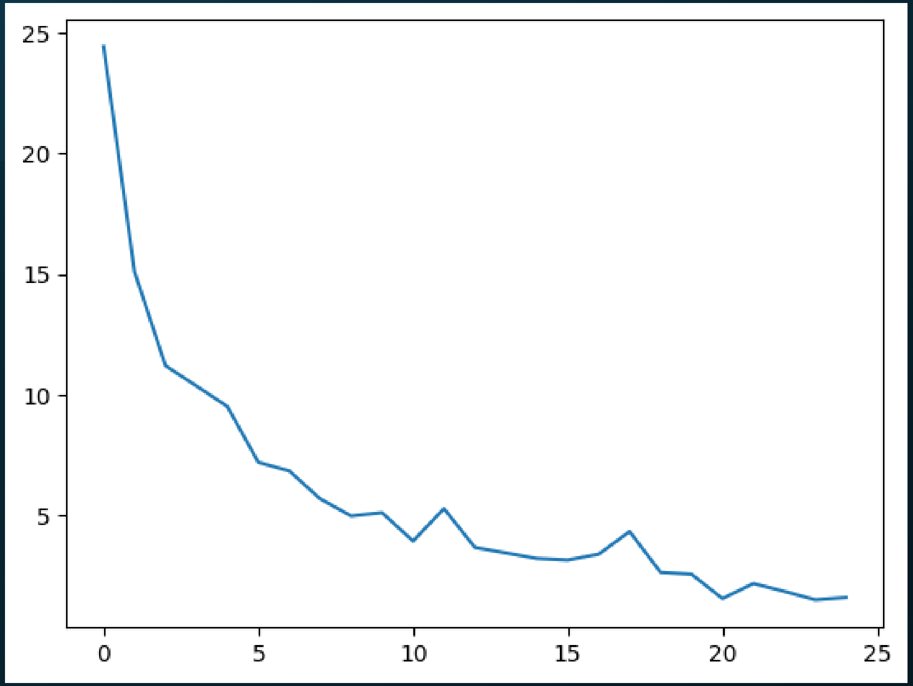


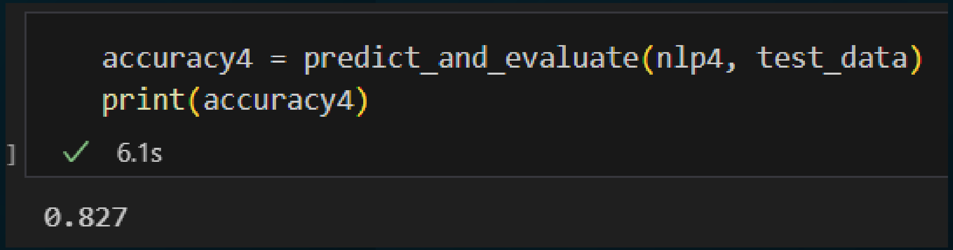
```
accuracy3 = predict_and_evaluate(nlp3, test_data)
print(accuracy3)
```

✓ 6.2s

```
0.866
```

Spacy
Run 4:
Epochs: 25
Batch Size: 32





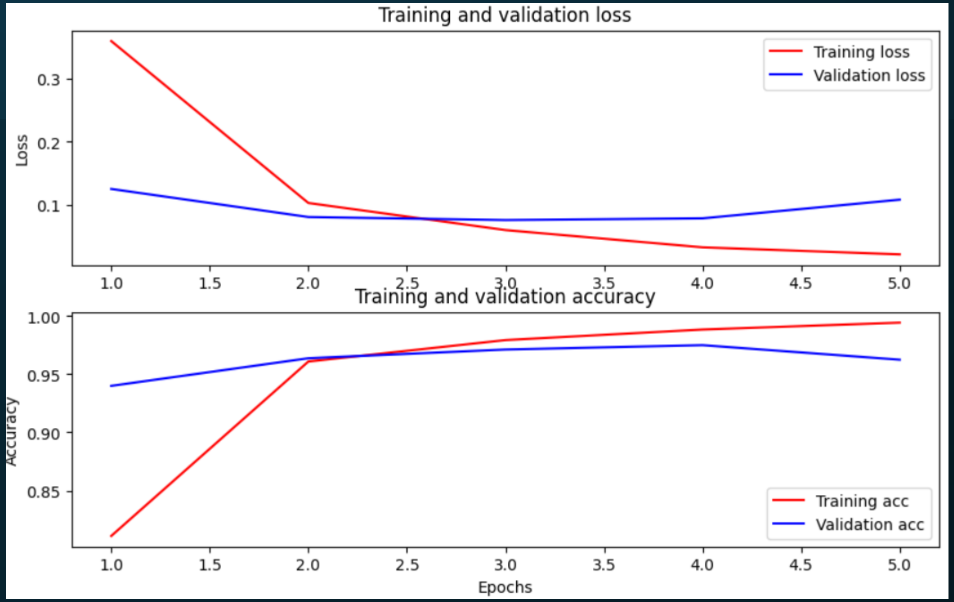
```
accuracy4 = predict_and_evaluate(nlp4, test_data)
print(accuracy4)
```

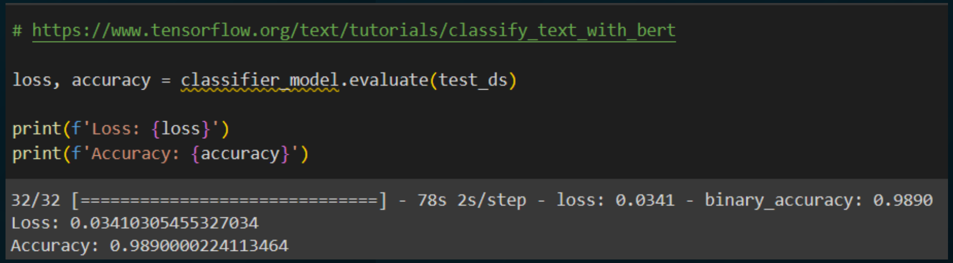
```
0.827
```

BERT

- Found tutorial here: https://www.tensorflow.org/text/tutorials/classify_text_with_bert
- Followed step by step
- Made modifications to use my dataset

BERT
Run 1:
Epochs: 5
Learning Rate: 3e-5

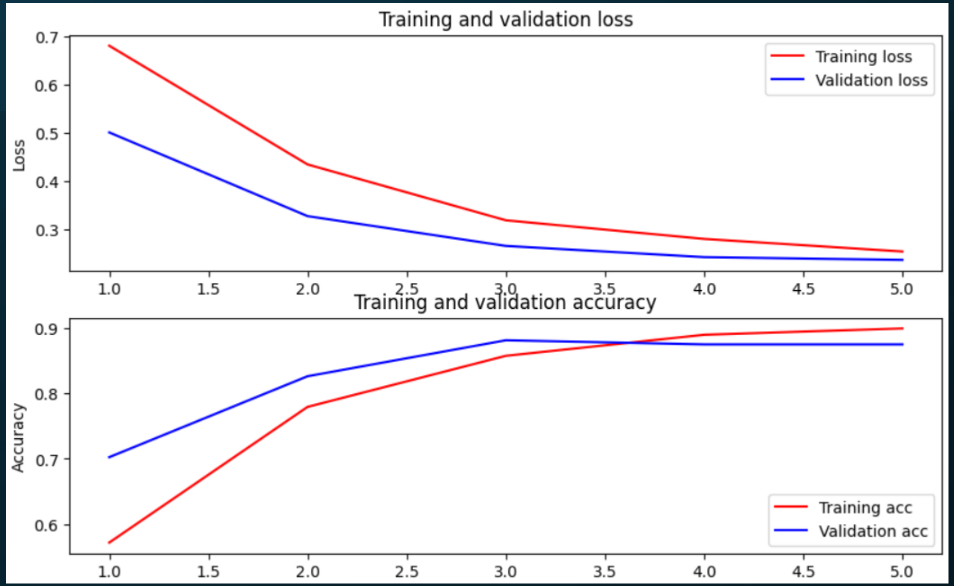


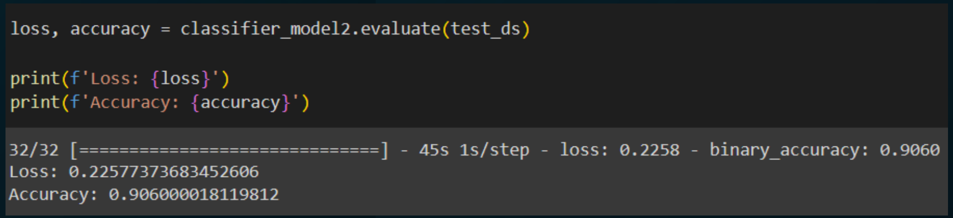


```
# https://www.tensorflow.org/text/tutorials/classify_text_with_bert

loss, accuracy = classifier_model.evaluate(test_ds)

print(f'Loss: {loss}')
print(f'Accuracy: {accuracy}')
```

```
32/32 [==============================] - 78s 2s/step - loss: 0.0341 - binary_accuracy: 0.9890
Loss: 0.03410305455327034
Accuracy: 0.9890000224113464
```

BERT
Run 2:
Epochs: 5
Learning Rate: 3e-6

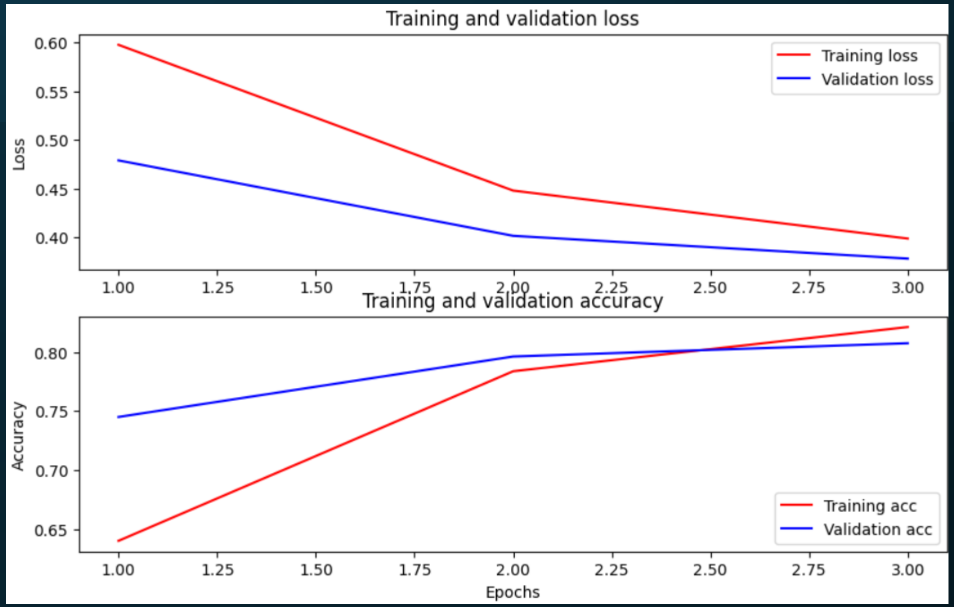


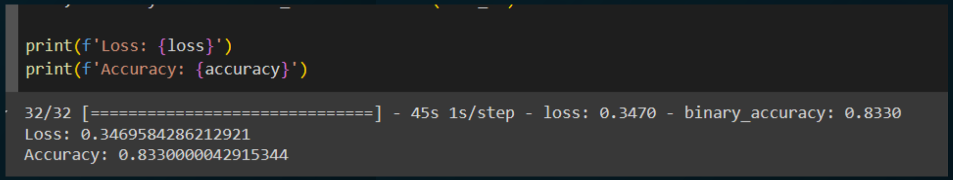


```
loss, accuracy = classifier_model2.evaluate(test_ds)

print(f'Loss: {loss}')
print(f'Accuracy: {accuracy}')
```

```
32/32 [==============================] - 45s 1s/step - loss: 0.2258 - binary_accuracy: 0.9060
Loss: 0.22577373683452606
Accuracy: 0.906000018119812
```

BERT
Run 3:
Epochs: 3
Learning Rate: 3e-6

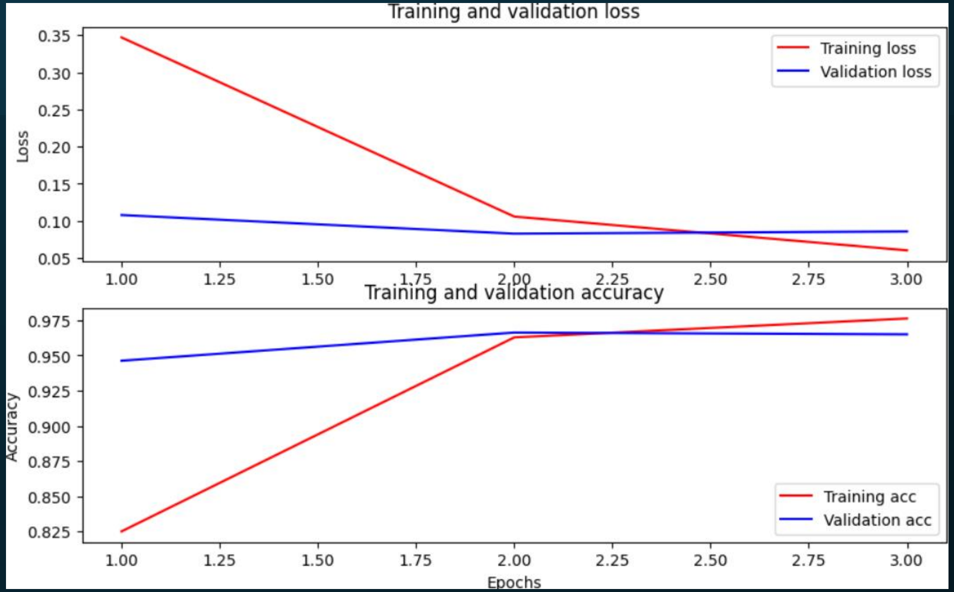


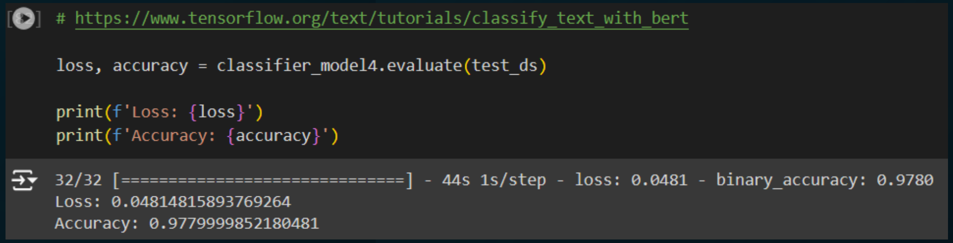


```
print(f'Loss: {loss}')
print(f'Accuracy: {accuracy}')
```

```
32/32 [==============================] - 45s 1s/step - loss: 0.3470 - binary_accuracy: 0.8330
Loss: 0.3469584286212921
Accuracy: 0.8330000042915344
```

BERT
Run 4:
Epochs: 3
Learning Rate: 3e-5





```
# https://www.tensorflow.org/text/tutorials/classify_text_with_bert

loss, accuracy = classifier_model4.evaluate(test_ds)

print(f'Loss: {loss}')
print(f'Accuracy: {accuracy}')
```

```
32/32 [==============================] - 44s 1s/step - loss: 0.0481 - binary_accuracy: 0.9780
Loss: 0.04814815893769264
Accuracy: 0.9779999852180481
```

Spacy vs BERT

- Spacy best test performance 86.6% Accuracy with 25 epochs, and batch size 16.
- BERT best test performance 98.9% Accuracy with 5 epochs, and learning rate 3e-5
- BERT took 1 hour to train 5 epochs
- SPACY took around 30 minutes to train 25 epochs
- BERT set up is much more complex
- SPACY is fairly simple

Conclusion

- Performance of the spacy model is decent, and the computation is very fast.
- Performance of the BERT model is superior, but the computation time is a lot longer.
- If you want to get decent results, fast then Spacy is a good choice.
- If you want to get extremely accurate results, then BERT is a good choice.