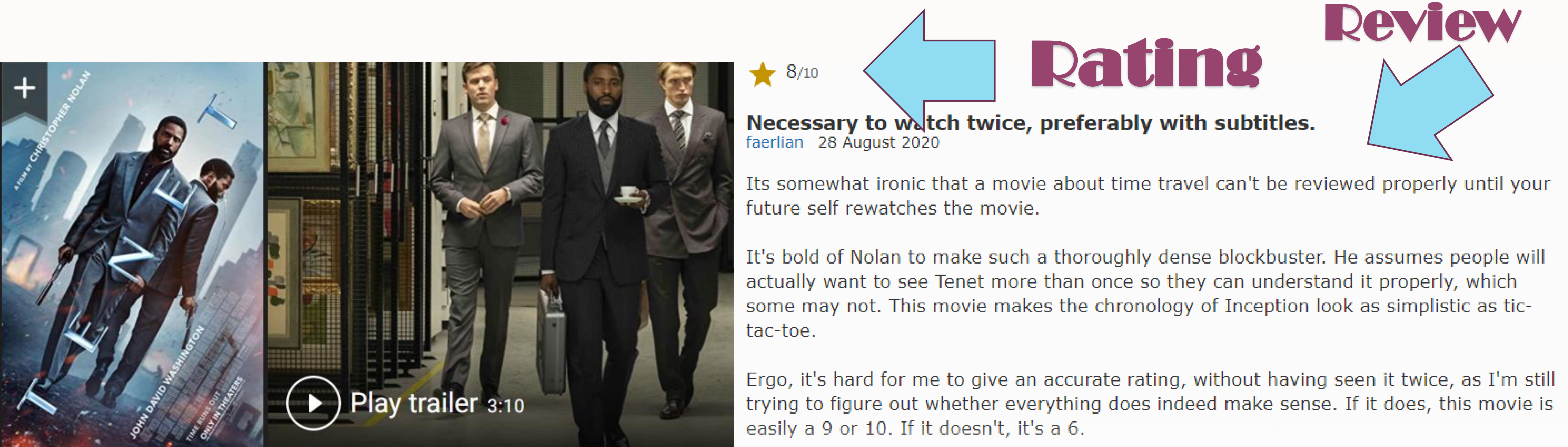
IMDB Movie Review Sentiment Classifier

Introduction

- ***Task***: Construct a sentiment classifier for movie reviews
- Once trained, the classifier can predict whether a movie review is positive or negative (binary classification)
- The classifier can then be deployed to see how well or badly a new movie is performing
- After a new movie is launched, there are various postings/comments in blogs, tweets, review sites etc. These comments are passed through the classifier check the sentiment towards the movie and see how well the movie is performing

Data Collection: IMDB Dataset

- IMDB dataset has 50K samples of reviews and labels of either positive or negative sentiments
- Dataset is balanced with 25K each of positive and negative sentiments
- Dataset was generated by the authors using the review rating (1 to 10), assigning 1 to 5 rating as negative and 6 to 10 as positive



8/10

Necessary to watch twice, preferably with subtitles.

faerlian 28 August 2020

Its somewhat ironic that a movie about time travel can't be reviewed properly until your future self rewatches the movie.

It's bold of Nolan to make such a thoroughly dense blockbuster. He assumes people will actually want to see Tenet more than once so they can understand it properly, which some may not. This movie makes the chronology of Inception look as simplistic as tic-tac-toe.

Ergo, it's hard for me to give an accurate rating, without having seen it twice, as I'm still trying to figure out whether everything does indeed make sense. If it does, this movie is easily a 9 or 10. If it doesn't, it's a 6.

Pre-Processing

- Change Lower Case
- Remove stop words
- Remove most common & non-meaningful words, abbreviations & html tags
- TF-IDF Vectorizer

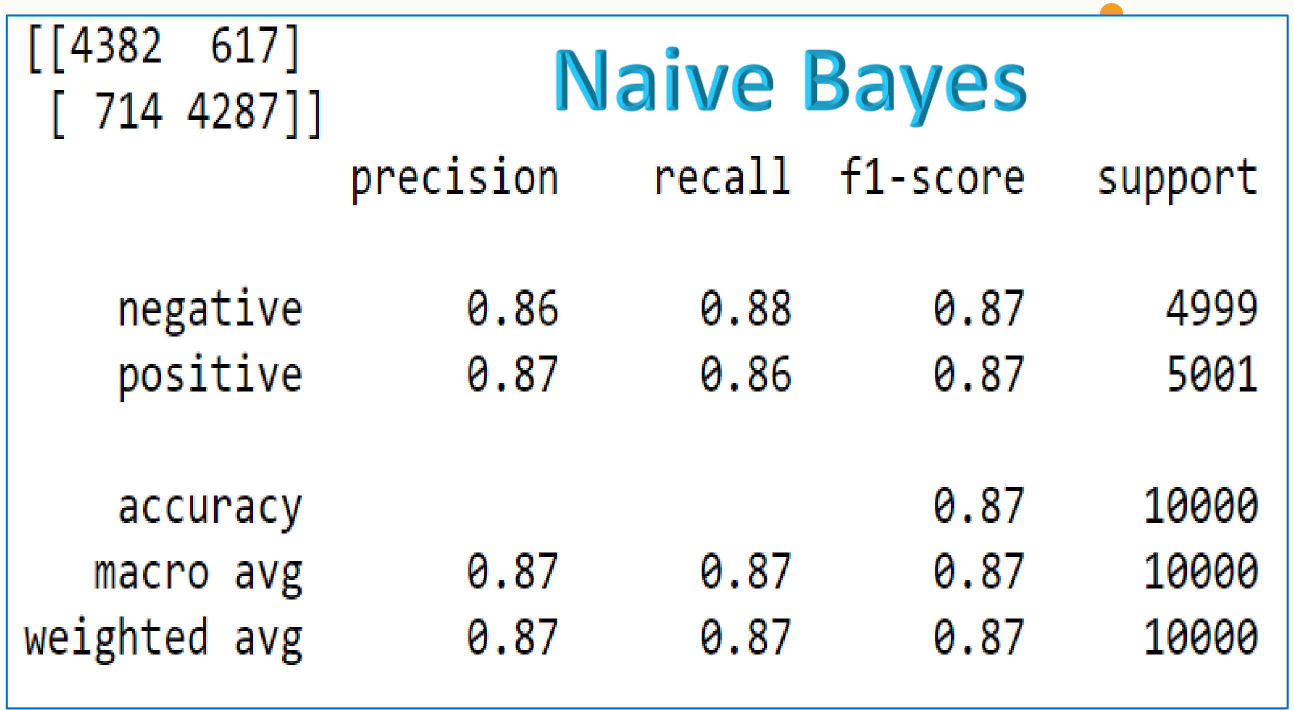
Algorithm Selection

Binary Classification Task:

- Logistic Regression
- Naive Bayes
- Support Vector Machine

Results

- All 3 algorithms perform well
- Accuracy, Precision & Recall all around 0.9
- Logistic Regression & SVM are on par and slightly better than Naïve Bayes

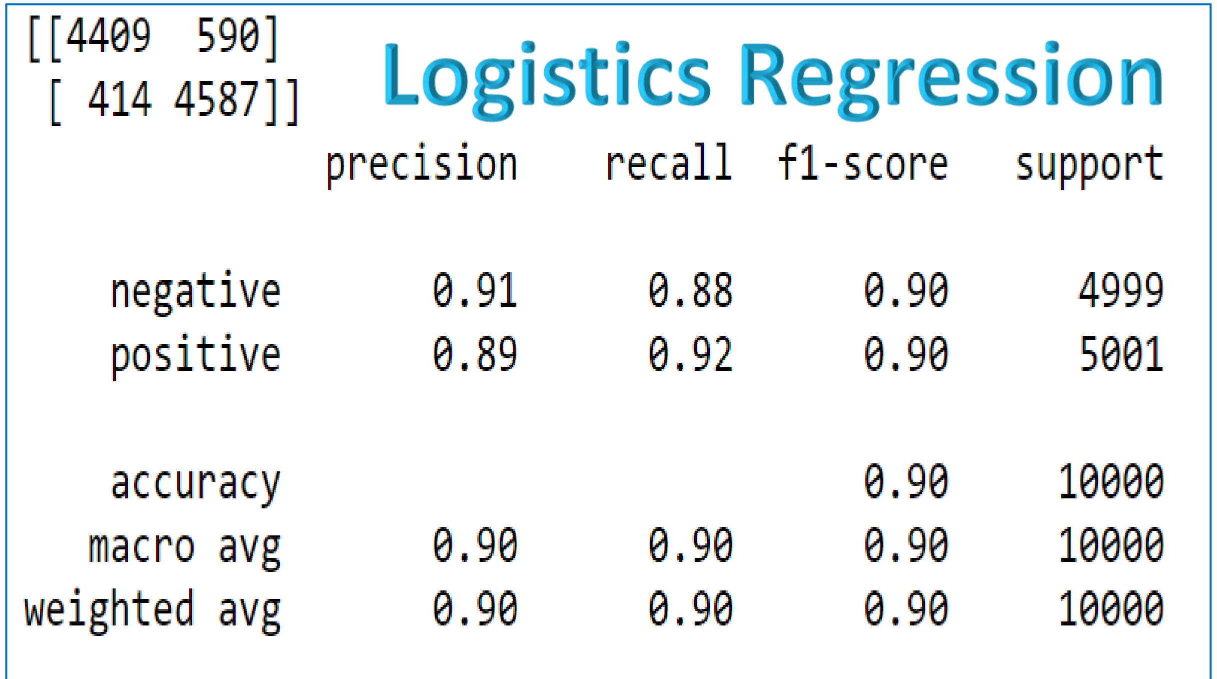


Naive Bayes

```
[[4382  617]
 [ 714 4287]]
              precision    recall  f1-score   support

    negative       0.86      0.88      0.87      4999
    positive       0.87      0.86      0.87      5001

    accuracy                           0.87     10000
   macro avg       0.87      0.87      0.87     10000
weighted avg       0.87      0.87      0.87     10000
```

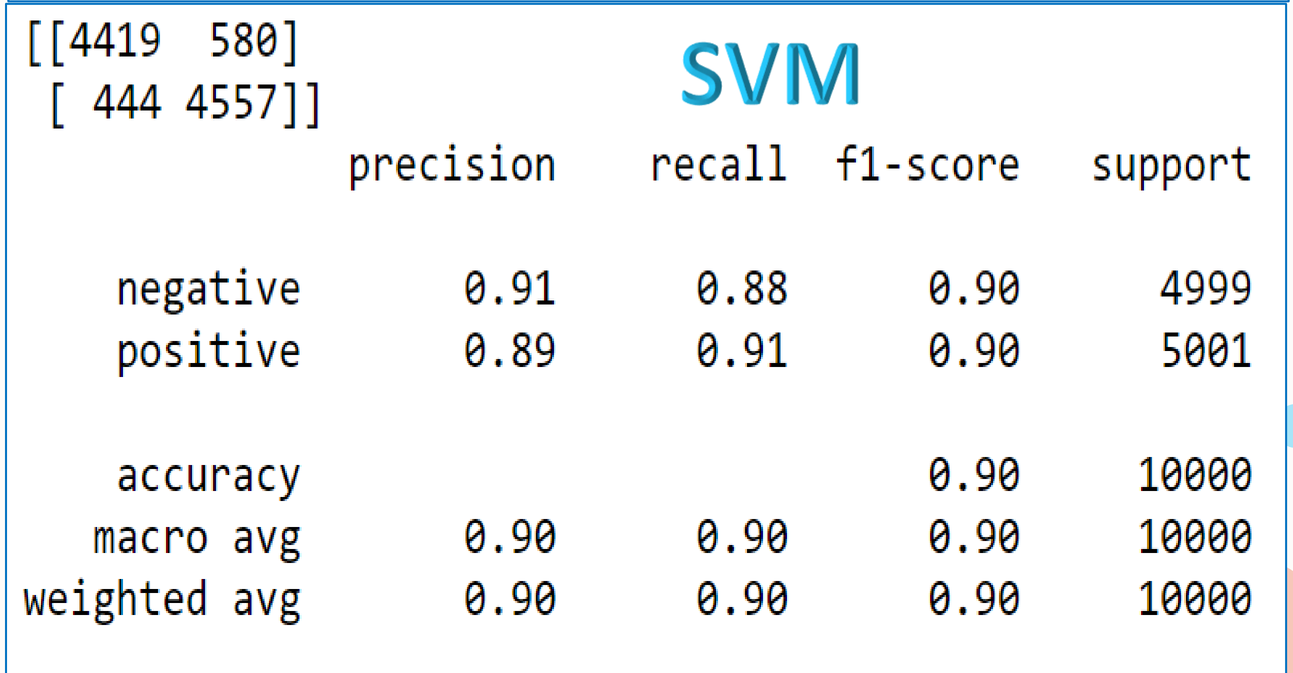


Logistics Regression

```
[[4409  590]
 [ 414 4587]]
              precision    recall  f1-score   support

    negative       0.91      0.88      0.90      4999
    positive       0.89      0.92      0.90      5001

    accuracy                           0.90     10000
   macro avg       0.90      0.90      0.90     10000
weighted avg       0.90      0.90      0.90     10000
```



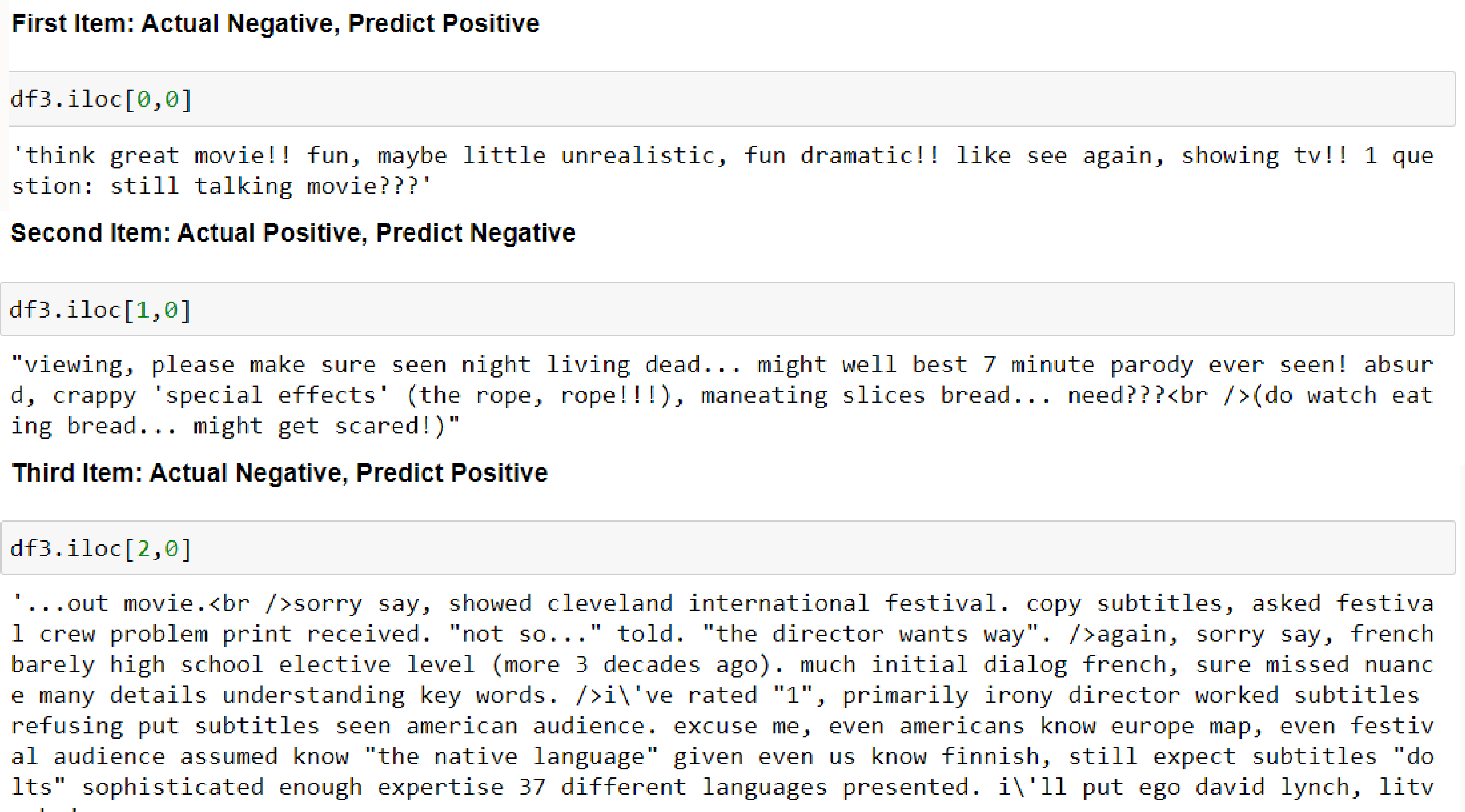
SVM

```
[[4419  580]
 [ 444 4557]]
              precision    recall  f1-score   support

    negative       0.91      0.88      0.90      4999
    positive       0.89      0.91      0.90      5001

    accuracy                           0.90     10000
   macro avg       0.90      0.90      0.90     10000
weighted avg       0.90      0.90      0.90     10000
```

Checking on wrong predictions



First Item: Actual Negative, Predict Positive

```
df3.iloc[0,0]
```

```
'think great movie!! fun, maybe little unrealistic, fun dramatic!! like see again, showing tv!! 1 que
stion: still talking movie???'
```

Second Item: Actual Positive, Predict Negative

```
df3.iloc[1,0]
```

```
"viewing, please make sure seen night living dead... might well best 7 minute parody ever seen! absur
d, crappy 'special effects' (the rope, rope!!!), maneating slices bread... need???<br />(do watch eat
ing bread... might get scared!)"
```

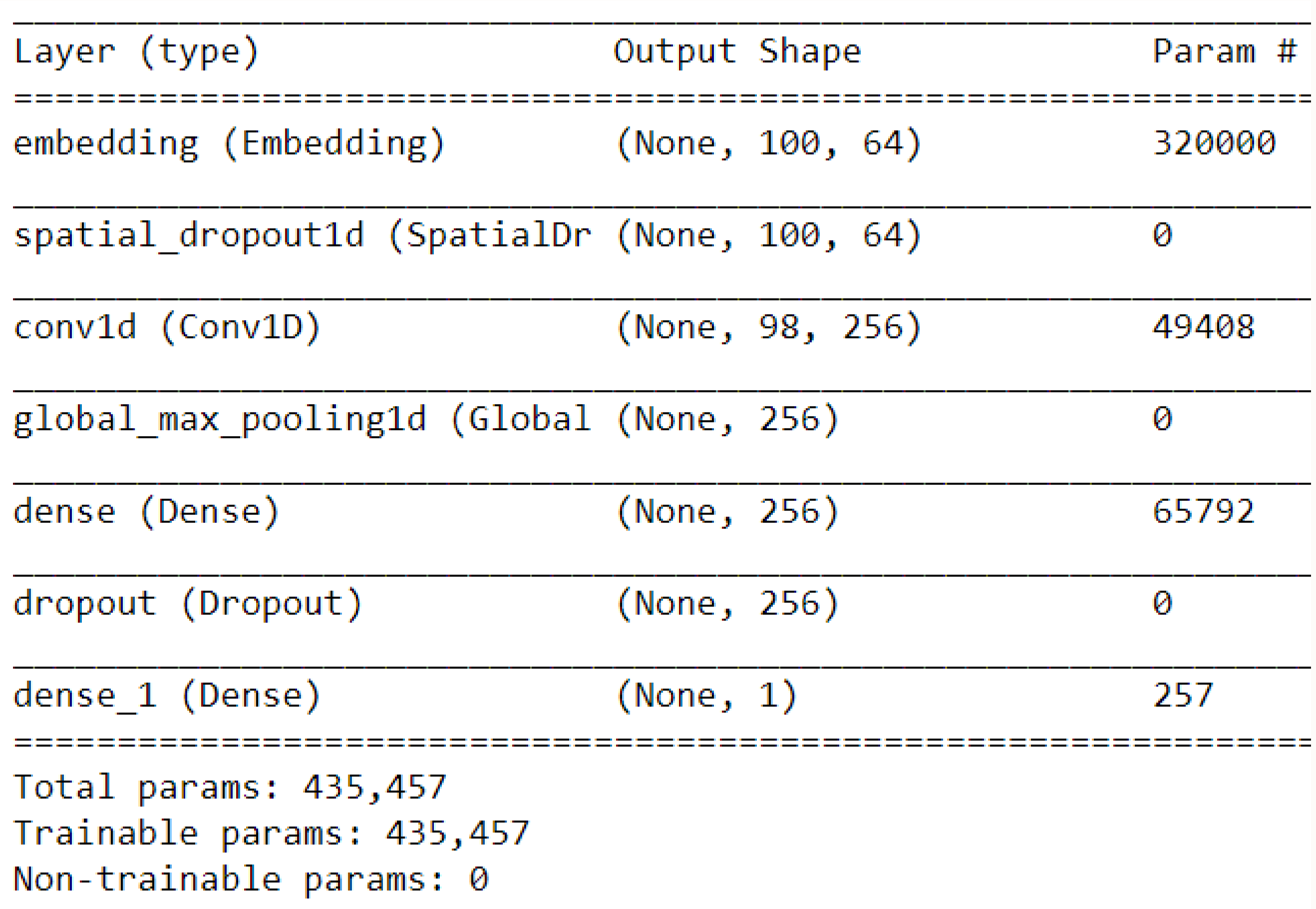
Third Item: Actual Negative, Predict Positive

```
df3.iloc[2,0]
```

```
'...out movie.<br />sorry say, showed cleveland international festival. copy subtitles, asked festiva
l crew problem print received. "not so..." told. "the director wants way". />again, sorry say, french
barely high school elective level (more 3 decades ago). much initial dialog french, sure missed nuanc
e many details understanding key words. />i\'ve rated "1", primarily irony director worked subtitles
refusing put subtitles seen american audience. excuse me, even americans know europe map, even festiv
al audience assumed know "the native language" given even us know finnish, still expect subtitles "do
lts" sophisticated enough expertise 37 different languages presented. i\'ll put ego david lynch, litv
```

CNN Architecture

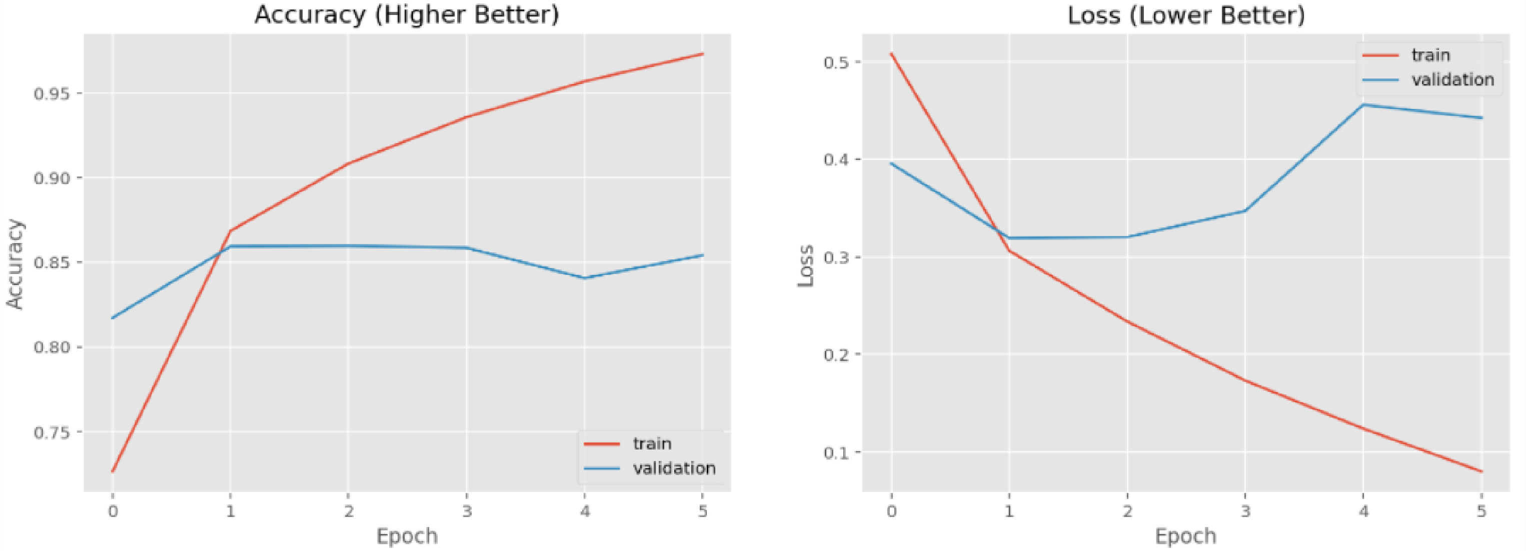
- Tensor Flow Keras has many built-in functions for pre-processing & text classification neural network design
- Word embedding (vectorization) is automatically done through embedding layer
- Pre-processing of keeping 5000 unique words, 100 words review length, pre-truncate or padding with zeros
- CNN 1D layer for text (2D for images)
- Final output neuron with sigmoid activation for binary classification
- Loss function: cross entropy & optimizer nadam

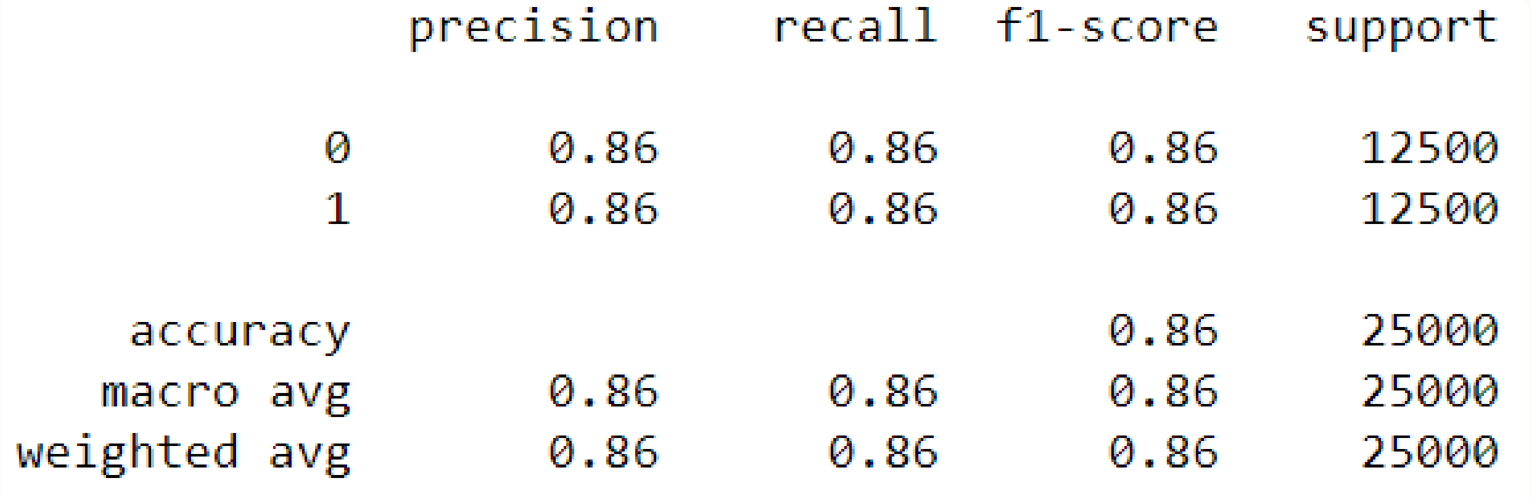


```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 64)           320000
_________________________________________________________________
spatial_dropout1d (SpatialDr (None, 100, 64)           0
_________________________________________________________________
conv1d (Conv1D)              (None, 98, 256)           49408
_________________________________________________________________
global_max_pooling1d (Global (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 256)               65792
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 257
=================================================================
Total params: 435,457
Trainable params: 435,457
Non-trainable params: 0
```

CNN Results

- Best results at accuracy, precision & recall at 0.86
- Reached at around 2 epochs
- After which overfitting on training set occurs without enhancing accuracy of validation set





```
              precision    recall  f1-score   support

           0       0.86      0.86      0.86     12500
           1       0.86      0.86      0.86     12500

    accuracy                           0.86     25000
   macro avg       0.86      0.86      0.86     25000
weighted avg       0.86      0.86      0.86     25000
```

ML vs Deep Learning Algorithms

- ML algorithms are quick to run & give good results for this IMDB dataset
- Deep learning algorithms using Tensorflow Keras are convenient with ready to use pre-processing & embedding code
- CNN architecture took longer to run and did not give improved results for this IMDB dataset
- Neural network design can be further tweaked, using other types of layers, changing dropout, more complex designs like LSTM etc
- Neural networks should work better for much larger dataset as it has many parameters for tuning