# Sentiment Analysis on an IMDB Movie Review Data Set

## 1 Introduction

This paper presents a sentiment analysis of the IMDB movie review dataset using a beginner-friendly approach. The analysis incorporates GloVe embeddings, tokenization, and performance evaluation of XGBoost, Convolutional Neural Network, and Recurrent Neural Network models.

## 2 Data Set Analysis

The data set consists of 50000 reviews and two columns, where one is a text review and the other a sentiment labeled positive or negative. There are no null values in the data set.

| | 0 |
|---|---|
| review | 0 |
| sentiment | 0 |

dtype: int64

Figure 1:

The data set s evenly distributed, with a total of 25000 positive and 25000 negative reviews.

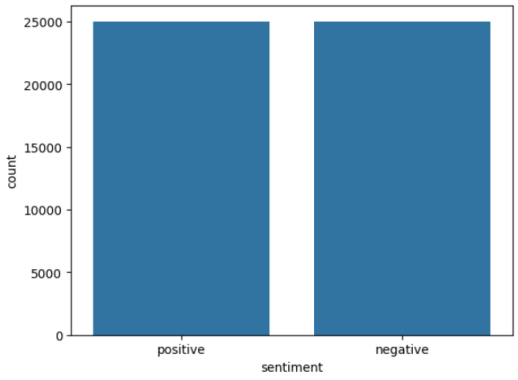



Figure 2:

### *2.1 Example Reviews*

The data set was inspected before set cleaning. The reviews include HTML tags, and the format of the reviews was briefly explored. Reviews often contain both positive and negative words, such as stunning and regret:

> What an absolutely stunning movie, if you have 2.5 hrs to kill, watch it, you won't regret it, it's too much fun! Rajnikanth carries the movie on his shoulders and although there isn't anything more other than him, I still liked it. The music by A.R.Rehman takes time to grow on you but after you heard it a few times, you really start liking it.

Several of the reviews have low ambiguity, and the reviewers show strong emotions in their review, as is the case with this reviewer.

> Phil the Alien is one of those quirky films where the humour is based around the oddness of everything rather than actual

> punchlines.<br /><br />At first it was very odd and pretty funny but as the movie progressed I didn't find the jokes or oddness funny anymore.<br /><br />Its a low budget film (thats never a problem in itself), there were some pretty interesting characters, but eventually I just lost interest.<br /><br />I imagine this film would appeal to a stoner who is currently partaking.<br /><br />For something similar but better try "Brother from another planet"

Many of the reviews are long and contain some ambiguity, where the reviewer argues for both the positive and negative qualities of the film. This reviewer found some of the characters interesting, but they eventually lost their interest in the movie.

> I remember this film,it was the first film i had watched at the cinema the picture was dark in places i was very nervous it was back in 74/75 my Dad took me my brother & sister to Newbury cinema in Newbury Berkshire England. I recall the tigers and the lots of snow in the film also the appearance of Grizzly Adams actor Dan Haggery i think one of the tigers gets shot and dies. If anyone knows where to find this on DVD etc please let me know. The cinema now has been turned in a fitness club which is a very big shame as the nearest cinema now is 20 miles away, would love to hear from others who have seen this film or any other like it.

Some of the reviews are anecdotal, where the movie has brought up old memories in the reviewer. The reviewer recalls their first cinema visit and describes the feeling of going to the cinema for the first time. In this review, we can also see that often, people confuse summaries with reviews. Summaries mostly describe the plot and do not give insight into the reviewer's experience of what is reviewed.

> My first exposure to the Templarios & not a good one. I was excited to find this title among the offerings from Anchor Bay Video, which has brought us other cult classics such as "Spider Baby". [...]

As we can see, reviews vary in quality, length and ambiguity. Other noteworthy aspects of the reviews are the use of negations. Instead of using the antonyms of words, many reviewers use "not" for negation. The data set reflects the real-life complexity of natural languages, which is highly contextual and ambiguous.

### *2.2 Cleaning of the Data Set*

The data set was prepared by removing HTML tags, punctuations, numbers, single characters, and multiple spaces.

As this analysis uses Glove, stop words were removed as Glove does not do forward-checking to contextualize words.

## 3 Training, Validation, and Testing Data Sets

In the original data frame, the sentiment is expressed as "positive" and "negative". To be able to work with machine learning models, negative labels were mapped to 0 and positive to 1.

The number 42 is used as a seed for randomization for splitting the data and models through this whole analysis. The data set was split into three sets:

- 70% training data
- 15% validation data

- 15% testing data

The test set comprises 35000 examples, and the validation and test sets are 7,500 examples each.

# 4 Formatting of the sets

In this paper, three machine learning models are explored. The XGBoost model needs a TF-IDF vector representation of the data set, and the two neural networks need tokenization and embedding.

## *4.1 TF-IDF*

TfidfVectorizer was used to convert the collection of reviews to a matrix of TF-IDF features. TF-IDF stands for term frequency-inverse document frequency. After this step, XGBoost has training, validation, and testing data sets, which include reviews in a suitable format for this model.

## *4.2 Embedding preparation*

To prepare the CNN and RNN models, training, validation, and testing sets must be tokenized and embedded.

Tokenizer was used to convert text into tokens, creating a word index that maps words to numbers and assigning lower numbers to frequently appearing words.

After this, pre-trained Glove embeddings were used to map each word to a pre-trained vector. In Glove, similar words have similar vectors, allowing the model to recognize semantic word relationships.

# 5 Models

In this section, the training and performance of the XGBoost, CNN, and RNN models are explored.

## *5.1 XGBoost*

XGBoost is an optimized distributed gradient boosting library that implements machine learning algorithms under the Gradient Boosting framework. It is a supervised learning boosting algorithm that uses gradient descent.

Boosting is an ensemble method that combines multiple individual decision trees to form a strong learner. Each weak tree is trained sequentially to correct the errors of the previous models. As iterations increase, weak learners are transformed into strong learners. XGBoost is more robust than decision trees, which are often prone to overfitting.

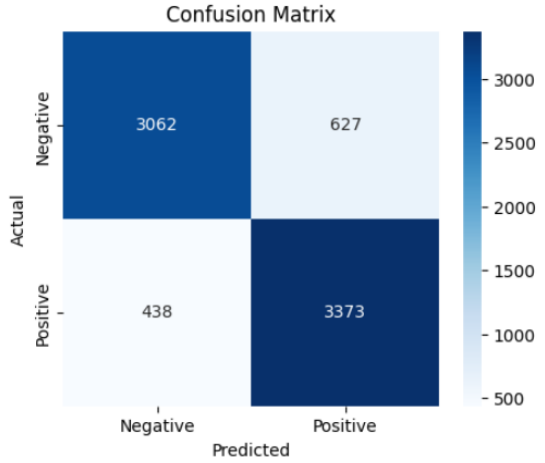
The following parameters were used in this analysis:

- n_estimators=200
- max_depth=6
- learning_rate=0.1
- colsample_bytree=0.8
- subsample=0.8

The parameter choices are in a moderate range to handle over-fitting while not compromising the training time. Each tree

randomly selects 80% of the features, and 80% of the training data is randomly selected for each tree.

The XGBoost accuracy on the validation set was 0.85, thus scoring high accuracy on the first run. The model predicted the labels with a 0.85 accuracy on the test data.




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.874857 | 0.830035 | 0.851857 | 3689.000 |
| 1 | 0.843250 | 0.885070 | 0.863654 | 3811.000 |

The model predicted true positives and true negatives well. Both precision and recall were high for both the positive and negative classes.

XGBoost provided a simple approach to sentiment analysis.

## *5.2 CNN*

Convolutional Neural Networks (CNNs) are a specific form of feed-forward neural networks. Originally invented for image classification, they have also gained popularity in textual tasks. CNNs are composed of multiple layers of artificial neurons.

The model uses pre-trained frozen word embeddings of Glove with 100-dimensional vectors. Words are mapped to the vector word representations to capture the semantic meaning.

The convolutional layer has 64 layers and a kernel size of 5, with a ReLU activation.

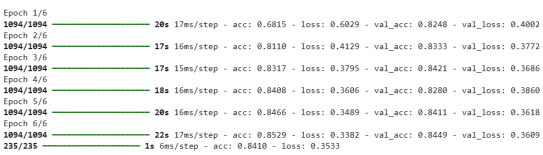
Global Max Pooling reduces the dimensionality and shortens training time. Dropout is used for regularization.

The output layer is fully connected and uses a sigmoid activation function, as the goal is to predict the probability of classes 0 and 1.

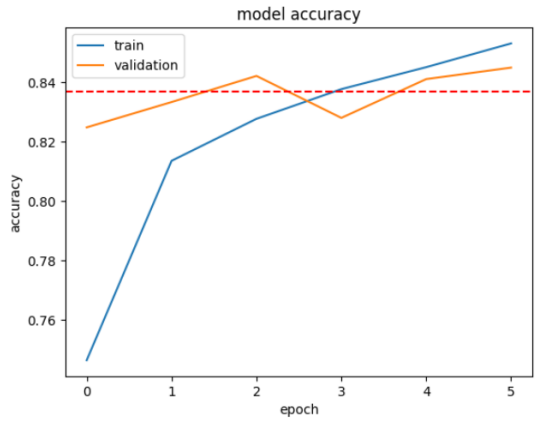
At compilation, Adam is used as an optimizer as it is designed to improve training speed and converge quickly. The loss function is binary cross entropy as this is a binary classification task.

The model was fitted on the training set created for the neural network tasks and validated on the validation set for neural networks, with a batch size of 32 in 6 epochs.

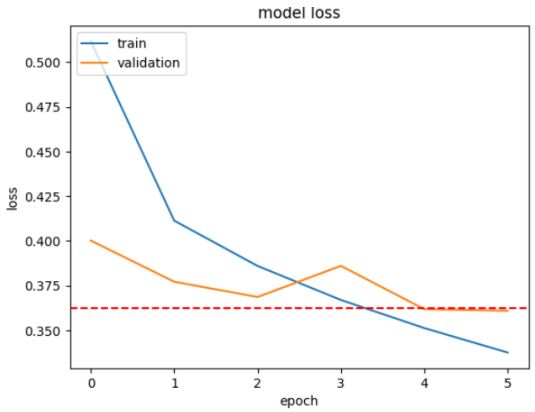
The model was then evaluated on the test set created for the neural network tasks.



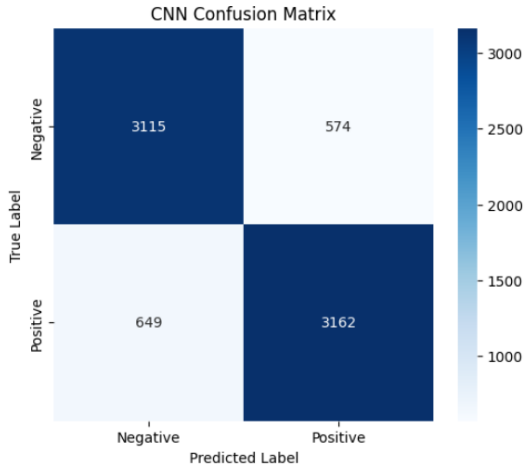

The test set had an accuracy of 0.84 and a loss of 0.35.




The training data quickly rose to an accuracy of over 0.82 in the fourth batch, after which it plateaued. The accuracy of the validation set was consistent in all epochs. The dotted line represents the test set accuracy, which aligns with the validation set accuracy.




By the 4th epoch, training and validation loss were down to under 0.375, aligning with the test loss of 0.37. The training and validation loss decrease each epoch, which indicates effective learning. The CNN model generalizes well.




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.827577 | 0.844402 | 0.835905 | 3689.000000 |
| 1 | 0.846360 | 0.829703 | 0.837949 | 3811.000000 |

The model had a slightly higher precision on positive labels than negatives.

## *5.3 RNN*

Recurrent Neural Networks (RNNs) are useful NLP tasks as the order and context of the data points are important. RNNs outputs depend on the prior elements within the sequence.

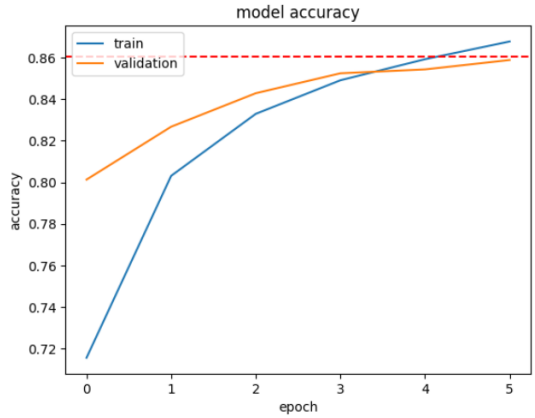
A sequential model and a Long-Short-Term Memory (LSTM) with 128 units are used.

The output layer outputs a single output and uses a sigmoid function for binary classification.

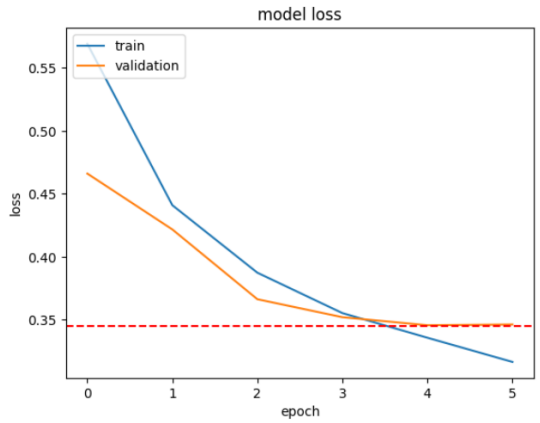
RMSprop is used as an optimizer with a binary cross-entropy loss function while training. The model's accuracy is tracked during training.

```
Epoch 1/6
547/547 ━━━━━━━━━━━━━━━━━━━━ 112s 201ms/step - acc: 0.6528 - loss: 0.6255 - val_acc: 0.8013 - val_loss: 0.4661
Epoch 2/6
547/547 ━━━━━━━━━━━━━━━━━━━━ 108s 198ms/step - acc: 0.7943 - loss: 0.4559 - val_acc: 0.8268 - val_loss: 0.4217
Epoch 3/6
547/547 ━━━━━━━━━━━━━━━━━━━━ 142s 197ms/step - acc: 0.8301 - loss: 0.3931 - val_acc: 0.8429 - val_loss: 0.3664
Epoch 4/6
547/547 ━━━━━━━━━━━━━━━━━━━━ 142s 197ms/step - acc: 0.8448 - loss: 0.3640 - val_acc: 0.8525 - val_loss: 0.3521
Epoch 5/6
547/547 ━━━━━━━━━━━━━━━━━━━━ 143s 200ms/step - acc: 0.8552 - loss: 0.3422 - val_acc: 0.8544 - val_loss: 0.3457
Epoch 6/6
547/547 ━━━━━━━━━━━━━━━━━━━━ 139s 196ms/step - acc: 0.8664 - loss: 0.3175 - val_acc: 0.8589 - val_loss: 0.3463
235/235 ━━━━━━━━━━━━━━━━━━━━ 12s 51ms/step - acc: 0.8639 - loss: 0.3387
```

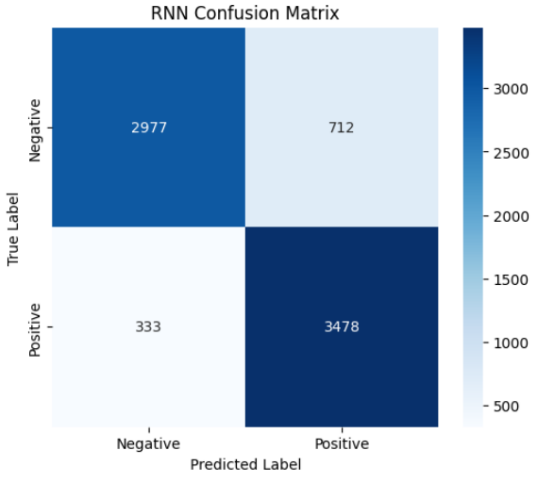
The test accuracy is 0.86 and loss 0.34.




The RNN model's training and validation accuracy steadily increases up to the 5th epoch, when they align with the test accuracy represented by the dotted line.




The training and validation loss drop steadily during the training up to the fifth epoch, when the validation loss plateaus. The model generalizes and learns well.




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.899396 | 0.806994 | 0.850693 | 3689.000000 |
| 1 | 0.830072 | 0.912621 | 0.869391 | 3811.000000 |

The precision of the negative labels was 0.90 and the positive ones 0.83.

## 6 Conclusions

Each model had a decent accuracy of around 0.85 for both the validation and test sets:

- XGBoost: 0.858
- CNN: 0.841
- RNN: 0.864

The accuracy is relatively high, considering the ambiguous nature of the reviews and that many of them displayed features of plot summaries.

RNN performed best in terms of accuracy, but the differences are insignificant. XGBoost and CNN performed similar results with much shorter runtimes.

During this analysis Randomized Grid Search was used to identify the best

parameters. Due to computational constraints and extended runtime, this was not included in this notebook. For even better results, Grid Search could have been used to explore even more parameter combinations. Fine-tuning the parameters was the most computationally demanding aspect of this exercise.

The most effective way to improve model performance would be to explore BERT embeddings instead of GloVe embeddings. Unlike GloVe, BERT processes text bidirectionally, allowing it to better capture contextual meaning, ambiguity, and negations in reviews.

There are numerous ways to further explore this dataset and experiment with different models. However, this project served as a valuable opportunity to approach natural language tasks from a beginner-friendly perspective.

# References


Lecture materials

Text feature extraction, Scikit Learn
https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

XGBoost documentation
https://xgboost.readthedocs.io/en/stable/

What is XGBoost, IBM
https://www.ibm.com/think/topics/xgboost

What is a recurrent neural network?
https://www.ibm.com/think/topics/recurrent-neural-networks