Week 5 Assignment: Transformers Sentiment Classification

Candidate: Sneha Santha Prabakar

Coding platform: Jupyter Notebook

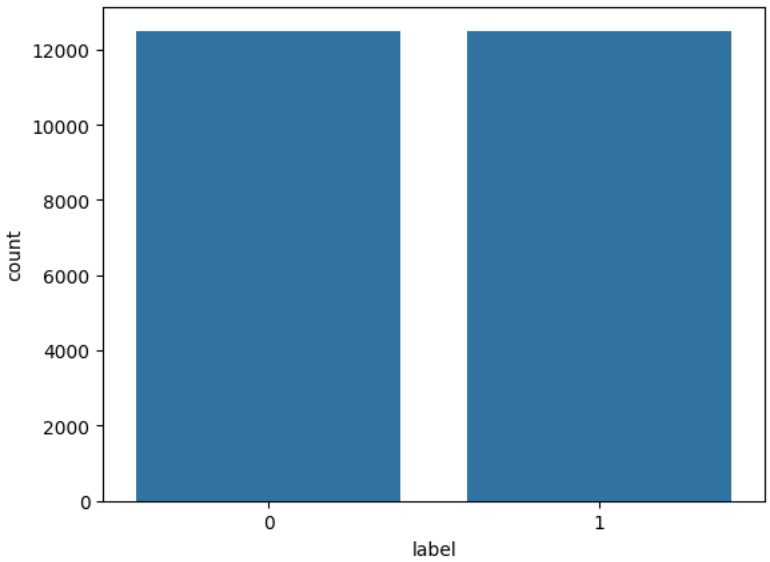
Please Note:

The python notebook included in the submission contains all the charts and steps mentioned in the grading criteria. This PDF document only provides a summary for each section executed in the python notebook. Hence, please refer to the python notebook for the execution steps.

Part 1: Data Pre-Processing

- We are using the Hugging Face dataset, IMDB, for movie reviews.
- There are 25,000 records for training and 25,000 for testing.
- Each review is labelled as 0 (negative) or 1 (positive).

EDA 1: Class distribution



We can see that the dataset is quite balanced.

- Missing values: We remove records with value = NA
- Outliers: We filter out reviews that are longer than 1000 words, to remove extreme outliers.
  - Reviews with less than or equal to 1000 words are kept
- Tokenization:
  - We create a vocabulary of the most frequent 15,000 words.
  - Unknown words will be replaced by OOV ("out-of-vocabulary").
- Padding:
  - Each review becomes a sequence of integers (each word → number).

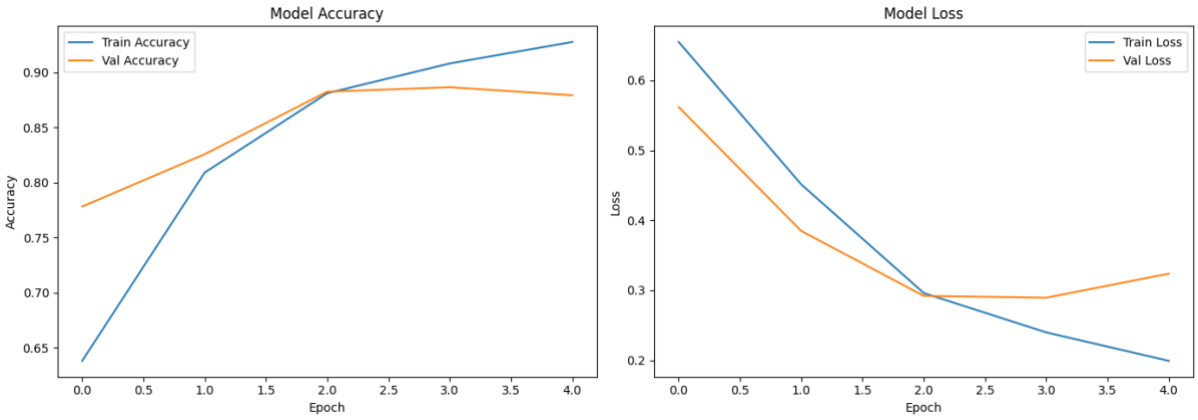
    - We then pad all sequences to the same length (256 words), so they can be processed in batches.
- Data splitting: We will split the training data with stratification to create a training and validation split (80/20), to help us monitor generalization during training.

Part 2: Basic Model

Layer (type)	Output Shape	Param #
input_layer_18 (InputLayer)	(None, 256)	0
embedding_10 (Embedding)	(None, 256, 64)	960,000
transformer_encoder_8 (TransformerEncoder)	(None, 256, 64)	37,664
global_average_pooling1d_8 (GlobalAveragePooling1D)	(None, 64)	0
dropout_36 (Dropout)	(None, 64)	0
dense_28 (Dense)	(None, 1)	65

Total params: 997,729 (3.81 MB)
Trainable params: 997,729 (3.81 MB)
Non-trainable params: 0 (0.00 B)

- We will build a basic regularized transformer model with:
    - Embedding layer
    - Custom TransformerEncoder block with multi-head attention and FFN
    - GlobalAveragePooling1D
    - Output Dense(1, activation='sigmoid')
- We will also use dropout and L2 regularization to prevent overfitting.
- We will compile the model with 'binary_crossentropy' loss and adam optimizer (with low learning rate), and train it for 5 epochs with a batch size of 64.



Accuracy plot:

- Epoch 0–1: Rapid improvement in both training and validation accuracy. Model is learning useful patterns from the data.
- Epoch 2: Training and validation accuracy both reach ~88%, indicating the model is generalizing well.
- Epochs 3–4: Training accuracy continues to increase (above 91%). Validation accuracy plateaus and then slightly declines (~88% → ~87.5%).

This shows that early training phases show excellent convergence.
The best generalisation is achieved with epoch = 2 where both training and validation accuracy intersect, around 88%.

Around Epoch 3+, a generalization gap emerges, as validation accuracy no longer improves and may be slightly degraded, and the model starts to overfit.

Loss plot:

- Training loss steadily decreases, which is a sign of effective learning.
- Validation loss:
    - Decreases initially (Epochs 0–2).
    - Stalls at Epoch 2.
    - Increases slightly after Epoch 3.

Final comments
Epoch 2 appears to be the optimal stopping point — both accuracy and loss are at their best.
From Epoch 3 onward: Training loss keeps dropping, but validation loss rises — a classic sign of overfitting.

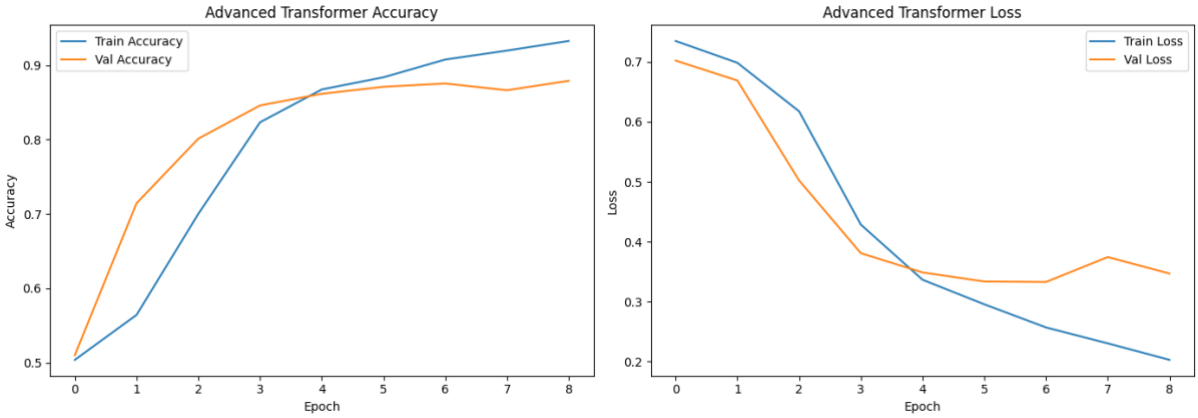
Based on the **classification report**, we can see that the basic model has strong generalization, especially in identifying positive sentiment (Recall = 0.92).

It's slightly more cautious when predicting negative sentiment, prioritizing precision over recall.

With AUC-ROC = 0.9528, the basic model distinguishes classes very well — better than accuracy alone shows.

Both classes are above 85% F1-score, which is solid.

Part 3: Advanced Model



Accuracy plot:

- Epochs 0–4:
    - Steady rise in both training and validation accuracy.
    - Model learns rapidly and generalizes well up to this point.
- Epochs 5–6:
    - Training accuracy continues increasing (~91% → ~93%).
    - Validation accuracy plateaus around 88%–89%.
- Epoch 7:
    - Minor drop in validation accuracy while training accuracy still increases.

We can see that there is strong generalization in early stages, and after epoch 5, there is minor overfitting as the validation accuracy no longer improves even as training accuracy increases.

Loss plot:

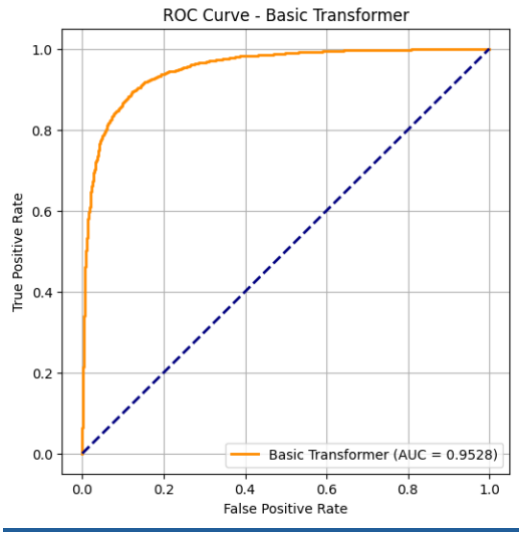
- Epochs 0–4:
  - Both training and validation loss drop rapidly — great learning phase.
- Epochs 5–6:
  - Training loss keeps decreasing.
  - Validation loss begins to flatten, then slightly increases at Epoch 7.
- Epoch 8:
  - Small recovery in validation loss.

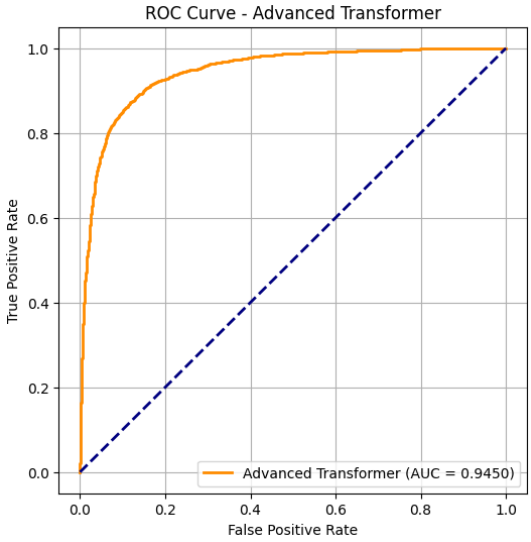
Final comments

The best performing epoch is likely Epoch 5 or 6.
The advanced model generalizes better than the basic model (higher and more stable validation accuracy/loss).
This validates the improvements from:

- Positional encoding
- Multi-head attention
- Proper regularization

Part 4: Model Comparison





Basic Transformer ROC
AUC-ROC = 0.9528
Curve rises steeply toward the top-left.
High true positive rate with low false positives.
Strong overall separation between classes.
Slightly sharper curve than the advanced model, meaning it's more confident in its predictions.

Advanced Transformer ROC
AUC-ROC = 0.9450
Still an excellent curve — well above random baseline.
Smooth and strong classification boundary.
Slightly lower AUC than the basic model — possibly due to increased regularization and slightly more conservative decision boundary.

Comparison
Both models are excellent at binary classification on IMDB reviews.
The basic transformer performs slightly better on AUC but risks overfitting. The advanced transformer, while having a slightly lower AUC, is more stable and consistent across epochs.
For deployment, the advanced model is the safer choice.