By

*Kesini Kandasamy*

Assignment 5.1: Transformer-based Sentiment Classification

Contents

[Scenario and Instructions 2](#_Toc207556981)

[Assignment & Solutions 3](#_Toc207556982)

[Task 1: Utilise Libraries/Dataset 3](#_Toc207556983)

[Task 2: Data Processing and Exploration 3](#_Toc207556984)

[Task 3: Construct a basic Transformer model 5](#_Toc207556985)

[Task 4: Train the basic model 6](#_Toc207556986)

[Task 5: Display model architecture and training progress 6](#_Toc207556987)

[Task 6: Construct an advanced transformer model 7](#_Toc207556988)

[Task 7: Train the advanced model 7](#_Toc207556989)

[Task 8: Display model architecture and training progress 8](#_Toc207556990)

[Task 9: Compare training and validation results for each model: 9](#_Toc207556991)

|  |  |
| --- | --- |
| Assignment 5.1  Transformer-based Sentiment Classification |  |

# Scenario and Instructions

#### **Assignment Instructions:**

Develop and evaluate basic and advanced Transformer models for binary text classification to understand the application of attention mechanisms and positional encoding in NLP. This assignment focuses on building hands-on experience with modern Transformer architectures applied to customer sentiment analysis using the IMDB dataset.

#### **Dataset:**

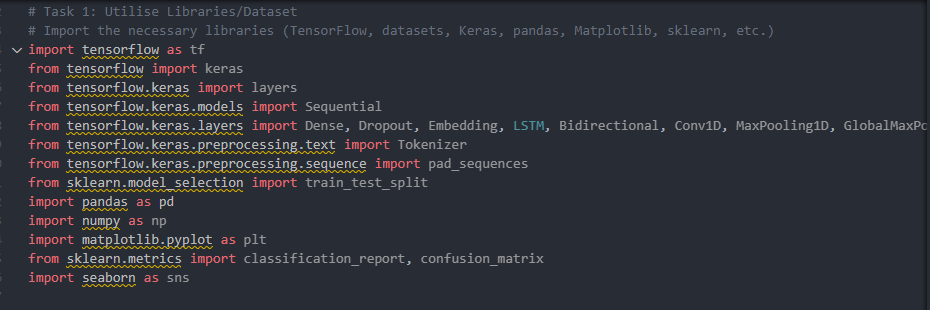
IMDB movie reviews

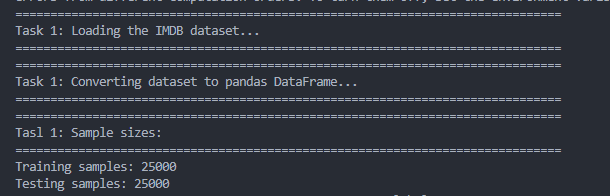
* 50,000 total reviews (25K train, 25K test)
* Binary classification:
  + 0 → Negative review
  + 1 → Positive review

# Assignment & Solutions

## Task 1: Utilise Libraries/Dataset

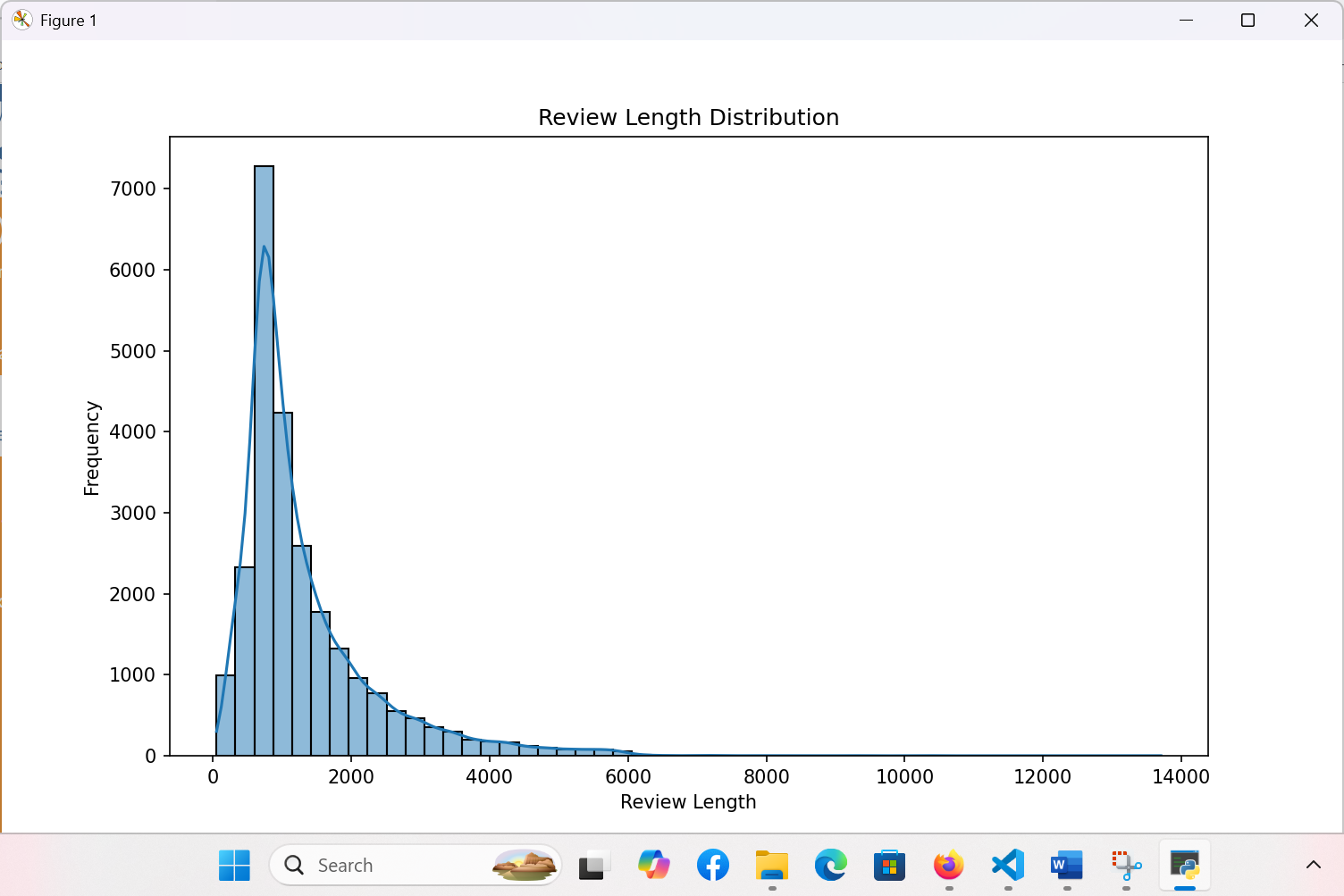
* Import the necessary libraries (TensorFlow, datasets, Keras, pandas, Matplotlib, sklearn, etc.)
* Load the IMDB dataset using load\_dataset("imdb")
* Convert data to pandas DataFrames and print sample sizes



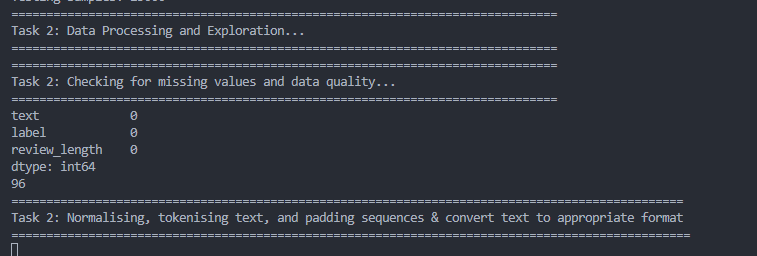


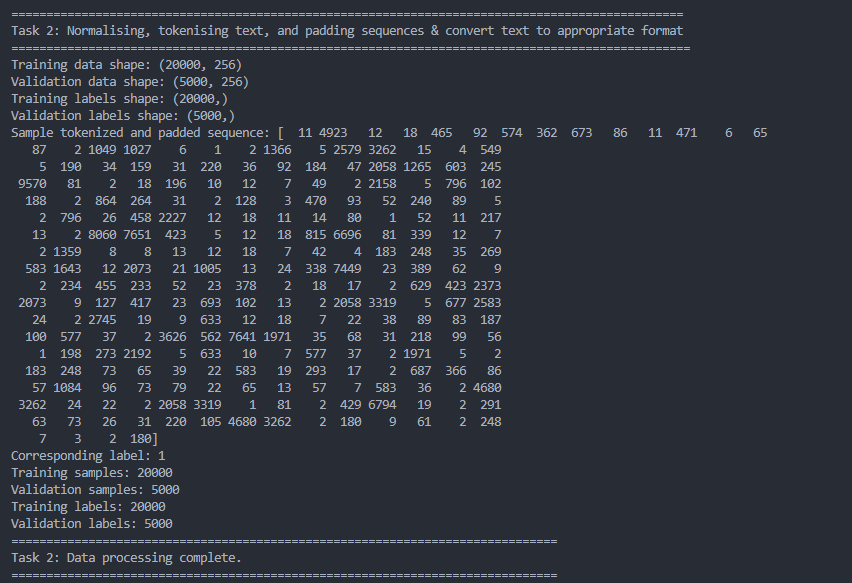
## Task 2: Data Processing and Exploration

* Generate two EDA visualisations (e.g., review length distribution, class distribution)
* Check for missing values and data quality
* Normalise and tokenise text, pad sequences to a consistent length
* Convert text into appropriate format using Keras Tokenizer (e.g., maxlen=256)
* Create training and testing splits



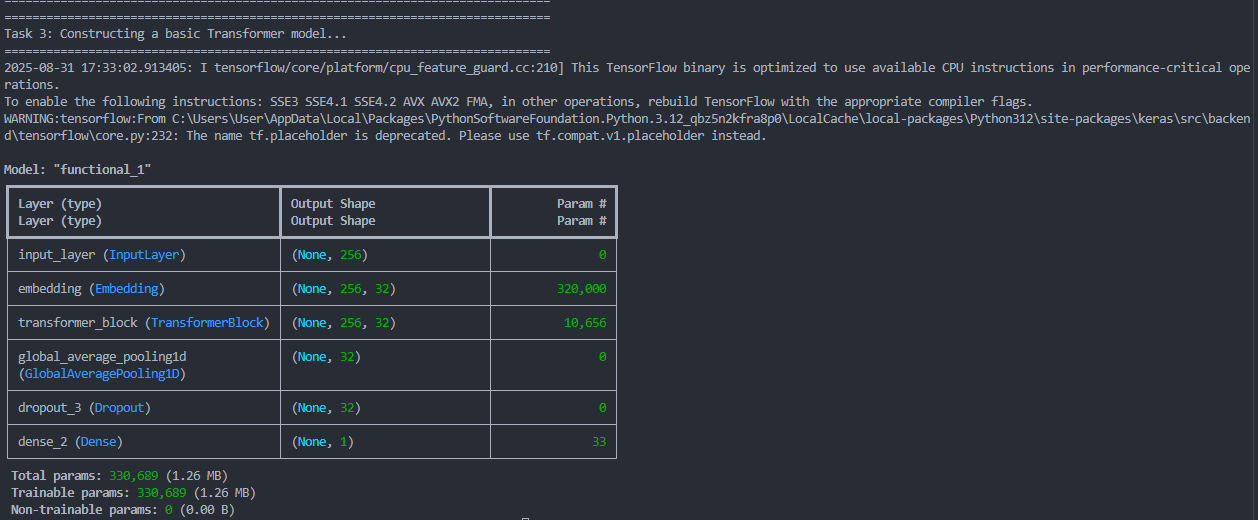






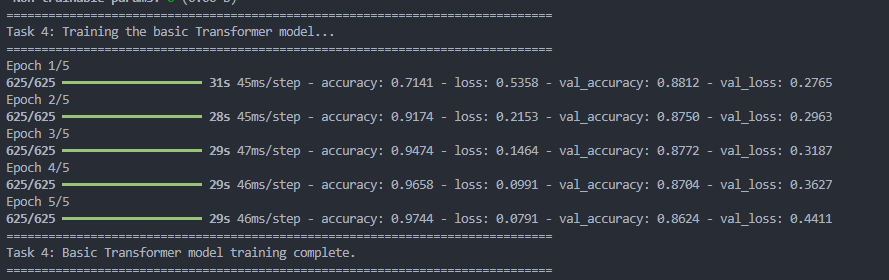
## Task 3: Construct a basic Transformer model

* Construct a basic transformer model with appropriate layers (Embedding, TransformerEncoder, GlobalAveragePooling1D, Dense)



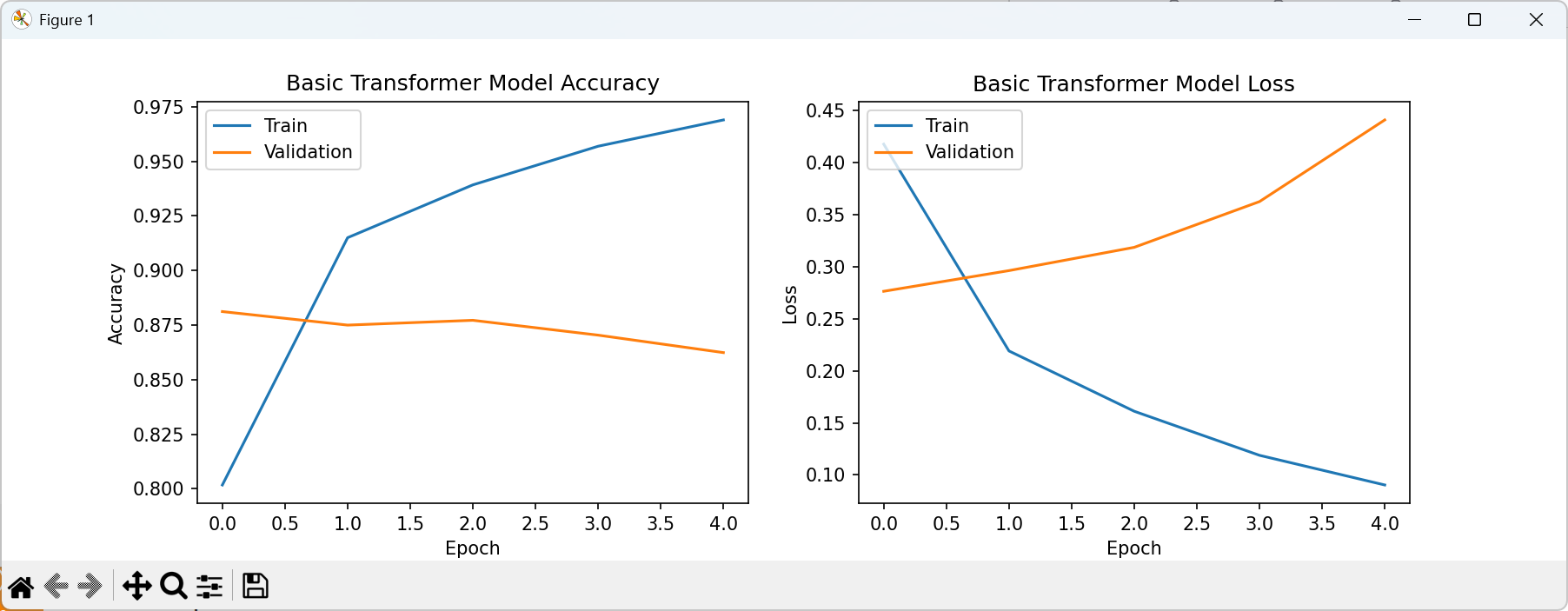
## Task 4: Train the basic model

* Train the model on the IMDB training data for 5 epochs using binary\_crossentropy



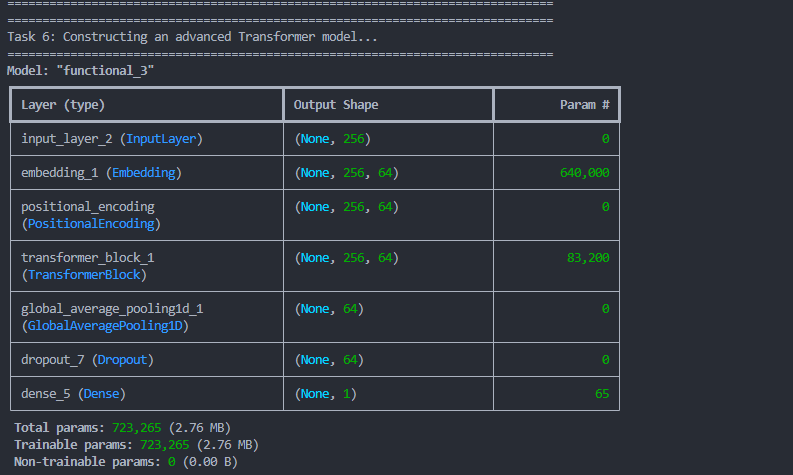
## Task 5: Display model architecture and training progress

* Display model architecture and plot training performance (accuracy, loss)



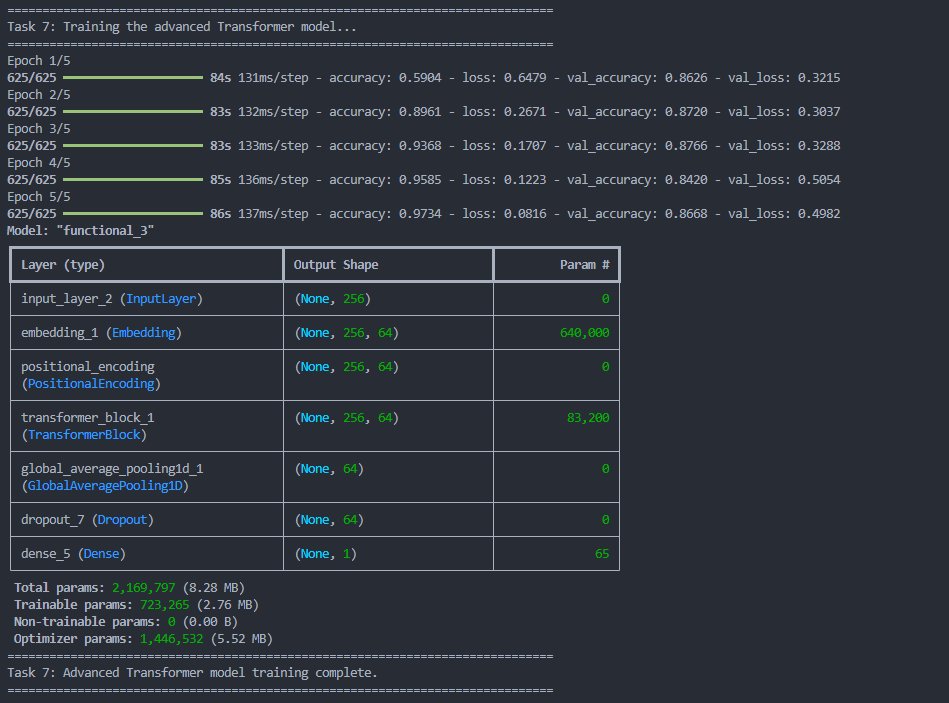
## Task 6: Construct an advanced transformer model

* Implement a custom positional encoding layer
* Use a Transformer block with multi-head attention (num\_heads ≥ 4) and Dropout



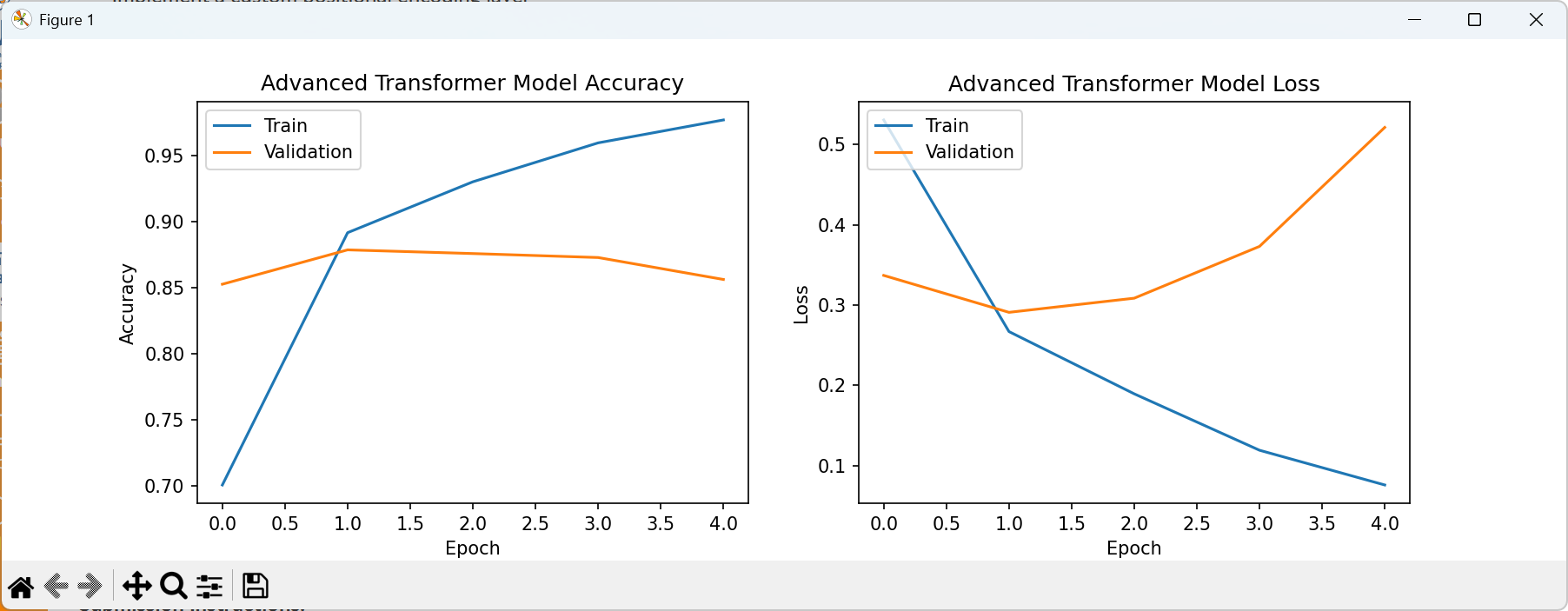
## Task 7: Train the advanced model

* Train the enhanced model and display the training curves and model summary



## Task 8: Display model architecture and training progress

* Display model architecture and plot training performance (accuracy, loss)



## Task 9: Compare training and validation results for each model:

* Evaluate both models using Accuracy, Precision, Recall, F1-score and AUC-ROC

