Model: "sequential": This indicates that the model is built sequentially, where each layer has exactly one input and one output.

Embedding Layer (Embedding):

Output Shape: (None, 100, 128)

Parameters: 640,000

This layer converts word indices into dense vectors of fixed size. Here, the vocabulary size is large (hence the 640,000 parameters), and each word is mapped to a 128-dimensional vector.

LSTM Layer:

Output Shape: (None, 64)

Parameters: 49,408

The LSTM layer has 64 units, meaning it produces a 64-dimensional output for each time step in the input sequence.

Dense Layer (Dense):

Output Shape: (None, 5)

Parameters: 325

This is a fully connected layer that takes the output from the LSTM and transforms it into a vector of size 5. This suggests that the model is designed to classify inputs into 5 different categories.

Total Parameters:

Total trainable parameters: 689,733

There are no non-trainable parameters, which means all parameters in the model are adjustable during training.

None: This dimension is typically used to represent the batch size, which can vary. It means that the model can process any number of input samples at once.

(None, 100, 128):

Embedding Layer: The output shape indicates that for each input sequence, there are 100 time steps (or words), and each word is represented by a 128-dimensional vector. The None allows for a variable batch size.

(None, 64):

LSTM Layer: The output shape here indicates that for each input sequence processed by the LSTM, the output is a 64-dimensional vector for each sample in the batch. Again, None represents the variable batch size.

(None, 5):

Dense Layer: The output shape indicates that for each input to this layer, the output is a 5-dimensional vector. This is typical for a classification task with 5 classes. The None represents the variable batch size.

In summary, these shapes are used to describe how data flows through the model, with None allowing flexibility in how many samples are processed at once.

Metrics Explained

Precision: The ratio of correctly predicted positive observations to the total predicted positives. Precision = TP / (TP + FP), where TP is true positives and FP is false positives.

Recall (Sensitivity): The ratio of correctly predicted positive observations to all actual positives. Recall = TP / (TP + FN), where FN is false negatives.

F1-Score: The weighted average of precision and recall. F1 Score = 2 \* (Precision \* Recall) / (Precision + Recall). It balances precision and recall.

Support: The number of actual occurrences of the class in the specified dataset.

Interpretation

The model performs exceptionally well on the "Not Hate" category with high precision, recall, and F1-score, indicating it is very good at identifying non-hate content. However, it struggles more with categories like "Racism" and "Sexism," where both precision and recall are lower, suggesting difficulty in correctly identifying these instances.

The overall accuracy of the model is high at approximately 94.66%, but the macro average indicates there is a disparity in performance across different classes, highlighting areas for potential improvement in detecting certain types of hate speech.

This report provides a comprehensive view of how well the LSTM model classifies different types of content in the dataset, allowing for targeted improvements in weaker areas.

**GRU**

**Interpretation**

**The model performs exceptionally well in identifying "Not Hate" content with high precision, recall, and F1-score.**

**The performance on "Racism" is notably lower, indicating difficulty in accurately predicting this class.**

**"Homophobia," "Sexism," and "Xenophobia" have moderate performance, with room for improvement in precision and recall.**

**The overall accuracy and weighted averages suggest that the model is generally effective but may struggle with minority classes like "Racism."**