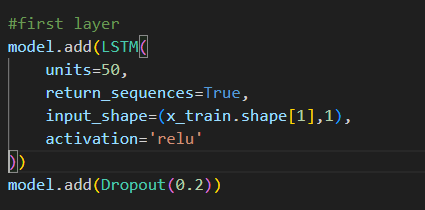
* Sequence of layers for building a Long Short-Term Memory (LSTM) neural network using the Keras library, which is commonly used for sequence prediction tasks, such as time series forecasting.



model.add(LSTM(...))🡺Adds an LSTM layer to the neural network.

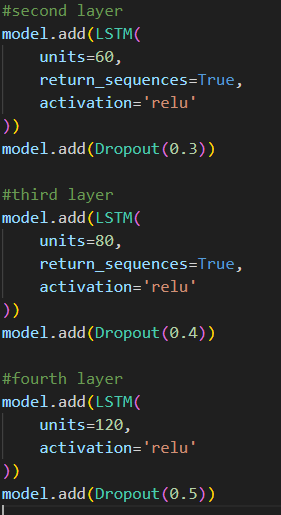
units=50🡺Specifies the number of LSTM units (neurons) in this layer.

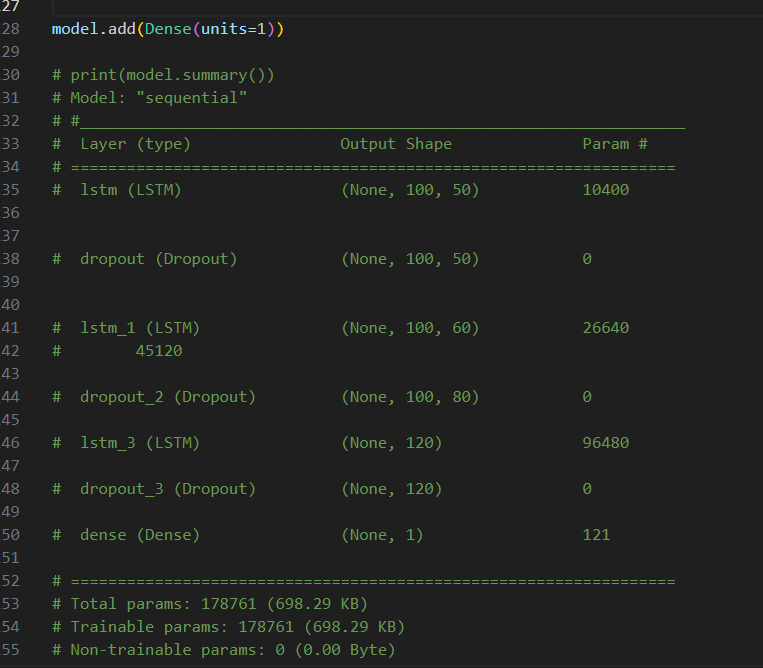
return\_sequences=True🡺 Indicates that the LSTM layer should return the full sequence of outputs, which is important when stacking multiple LSTM layers.

input\_shape=(x\_train.shape[1], 1)🡺Specifies the input shape for the first layer. x\_train.shape[1] is the number of time steps in each input sequence, and 1 is the number of features for each time step.

activation='relu'🡺Uses the Rectified Linear Unit (ReLU) activation function.

model.add(Dropout(0.2))🡺Adds a dropout layer after the first LSTM layer. Dropout is a regularization technique that helps prevent overfitting by randomly setting a fraction of input units to zero during training. Here, 0.2 is the fraction of units to drop.

Each layer introduces more LSTM units, and the dropout rate increases as you go deeper into the network. This is a common practice to prevent overfitting, especially in deep networks.



model.add(Dense(units=1))🡺 Adds a dense (fully connected) output layer with a single neuron. This layer is responsible for producing the final output of the network. The network is designed for a regression task since there is a single output neuron.

