Approach Used

1. **Data Preprocessing**
    - Tokenization of text data using Tokenizer from tensorflow.keras.preprocessing.text.
    - Padding sequences to a fixed max_length of 200 for uniform input size.
    - Splitting data into training and validation sets using train_test_split from sklearn.model_selection.
2. **Model Architecture**
    - **Embedding Layer:** Converts words into vector representations (embedding_dim=100 and vocab_size=10000).
    - **LSTM Layers:** Two LSTM layers with 128 and 64 units, where the first LSTM returns sequences for deeper learning.
    - **Dropout Layers:** Applied after LSTM and Dense layers (0.3 rate) to reduce overfitting.
    - **Dense Layers:** Fully connected layers with 32 neurons (ReLU activation) and an output layer with a sigmoid activation for binary classification.
3. **Training Setup**
    - The model was trained for **10 epochs** with validation.
    - Binary cross-entropy loss and Adam optimizer were used for learning.
    - The accuracy and loss were monitored for training and validation.

Challenges Faced

- **Training Time:** The model took **~500 seconds per epoch**, making training time-consuming.
- **Overfitting:** From **epoch 5 onwards**, training accuracy increased significantly while validation accuracy fluctuated slightly. This indicates a potential **overfitting issue**.
- **Sudden Accuracy Drop (Epoch 8):** A drop in validation accuracy at epoch 8 (0.9913) was observed due to possible model instability.

Model Performance & Improvements

1. **Final Accuracy:**
    - **Training Accuracy: 99.87%**
    - **Validation Accuracy: 99.89%**
    - **Final Validation Loss: 0.009**
2. **Performance Observations:**
    - The model generalizes well with a **high validation accuracy**.
    - The loss is very low, indicating an effective learning process.
    - Overfitting is minimal but can still be addressed for robustness.
3. **Possible Improvements:**
    - **Early Stopping:** Use early stopping to prevent unnecessary epochs that may cause overfitting.