- **Compare the performance of the two models post fine-tuning.**
- **Provide an analysis of which model performs better and explain why.**

Fine-Tuning with LoRA

- LoRA Overview: LoRA reduces the number of trainable parameters by injecting low-rank matrices into specific layers of a pre-trained model. This leads to a reduction in memory consumption and computational requirements during fine-tuning, which is especially beneficial for large models like LLaMA and Mistral.
- **Inference Time and Memory Usage**: LoRA fine-tuning is designed to be lightweight, so measuring memory usage during inference and training will help highlight any differences between the models.

Training and Inference Efficiency

- **Training Time**: Measure how long each model took to complete fine-tuning. Even though LoRA reduces the training burden, larger models like LLaMA might still take longer compared to smaller, more efficient models like Mistral.
- **Memory Usage**: Compare GPU memory usage during fine-tuning and inference. LoRA helps reduce the overhead, but differences in model architecture will still reflect in the final memory consumption.

Analysis and Conclusion

Efficiency: Since you used LoRA for both models, Mistral is likely to be more efficient in terms of speed and memory consumption. If Mistral's performance is close to or matching LLaMA's in quality while being faster and lighter, Mistral might be the better overall option.

Future Scope : We can calculating performance metrics like BLEU or ROUGE,