What is a Quantized Model?

In the context of large language models (LLMs), **quantization** is a model optimization technique used to reduce the memory and compute requirements by lowering the precision of the numerical values (weights and activations) used in the model. This is done **without retraining the model**, and usually with **minimal loss in performance**.

🎯 Why Quantize?

Original models are typically stored in **32-bit floating point (FP32)** format, which offers high precision but demands significant RAM and compute power. Quantization reduces this to smaller formats like **8-bit (INT8)** or even **4-bit**, making models:

- ✅ **Smaller in size**
- ✅ **Faster to run**
- ✅ **Able to run on hardware with limited RAM or VRAM**

Precision Format	Bits per Weight	RAM Usage	Speed	Accuracy Impact
FP32	32	Very High	Slow	None
INT8 / Q8_0	8	Medium-High	Moderate	Very Low
Q5_1	5	Moderate	Fast	Low
Q4_K_M	4 (optimized)	Low (~30 GB)	Very Fast	Minimal

Why I Chose Q4_K_M

Given my hardware setup:

PROF

- **30 GB of system RAM**
- **8 GB of GPU VRAM**

I needed a model format that balances **performance**, **compatibility**, and **accuracy**. After testing different quantized versions, I selected:

🧠 `Q4_K_M` — "4-bit Quantization with K-means + Multipliers"

This is an advanced 4-bit quantization scheme that:

- Maintains better accuracy than older 4-bit methods (`q4_0`, `q4_1`)
- Uses **multipliers** and **grouped quantization** to preserve precision
- Fits comfortably in **<32 GB RAM**, making it ideal for my system

> With Q4_K_M, I can run large models (up to 14B parameters) on my local machine, without needing a dedicated high-end GPU.

Summary

Quantization allows developers and researchers to **run large-scale AI models on consumer-grade hardware**. It is a powerful technique for local, offline AI inference, especially when paired with optimized formats like `Q4_K_M`.

By choosing a quantized model, I've enabled my project to:

- Function efficiently within hardware constraints
- Deliver real-time inference
- Maintain model performance with minimal trade-offs