Install ollama and run local LLM(Llama 3.2), performance optimization and troubleshooting local methods

1. Introduction

This hands-on demonstrates how to install **Ollama** and run the **Llama 3.2 model** locally in a low disk space environment. Instead of downloading the full model (which requires large storage), I used **quantized variants (Q4)** and smaller models to optimize disk usage, memory, and performance.

2. Installation Steps

1. Install Ollama using:
2. curl -fsSL https://ollama.com/install.sh | sh
3. Verify the installation:
4. ollama --version

3. Running Llama 3.2 (Quantized Variant)

Since disk space is limited, I pulled the smaller **quantized model** instead of the full-size Llama 3.2:

ollama pull llama3.2:8b-instruct-q4

ollama run llama3.2:8b-instruct-q4

This reduced the storage requirement and allowed smooth execution.

4. Using Smaller Models (Alternative)

To save more space, I also tested lightweight models like **Gemma 2B** and **Mistral 7B Q4**, which require fewer resources:

ollama pull gemma:2b

ollama run gemma:2b

5. Performance Optimization

- Used **quantized models (Q4)** to reduce memory and disk usage.
- Adjusted **context length** to improve speed and response time.
- Monitored system resources using htop (CPU/RAM) and nvidia-smi (GPU).

6. Troubleshooting Methods

- **Error: Model not found** → Fixed by running ollama pull <model-name>.
- **Error: Out of memory** → Solved by switching to a smaller or quantized model.
- **Error: Dependency issues** → Reinstalled Ollama and ensured PATH was correctly set.

7. Results

- Successfully ran **Llama 3.2 quantized variant** with limited disk space.
- Tested alternative lightweight models like Gemma 2B.
- Achieved better performance with lower memory and storage requirements.

8. Conclusion

This hands-on proved that Ollama can run **large language models locally** even with limited disk space by using quantized or smaller variants. The setup is simple, resource-efficient, and can be optimized further for better speed and memory usage.