Nikolas Zelenikovski

48597359

CS 5393

Exam #2

Exploring Open-Source LLMs with Ollama

This project explores and evaluates the capabilities of open-source large language models (LLMs) using Ollama, a command-line tool that simplifies running LLMs locally. With increasing concerns about data privacy, internet dependency, and compute costs, local LLMs are becoming a practical alternative to commercial, cloud-based AI systems. Ollama bridges that gap by offering streamlined installation and runtime for various models directly on a user's machine.

The goal of this assignment was to gain hands-on experience with different LLMs, compare their performance on a range of tasks, and assess how prompt engineering impacts model behavior. I selected three models of varying size and complexity: TinyLlama (~1B parameters), Mistral-7B, and LLaMA2 (~7B). Each model was tested on tasks including question answering, summarization, code generation, and creative writing. I also conducted a focused experiment to see how adding structured reasoning to prompts affects model output.

This report summarizes the installation process, task results, and performance insights while reflecting on the practical applications of open-source LLMs in both personal and professional environments.

Model Name	**Size**	**Architecture**	**Notes**
TinyLlama	~1B	Lightweight	Very fast and low-resource usage
Mistral-7B	7B	Transformer	Mid-tier model with strong output
LLaMA2	~7B	Meta AI	High-quality outputs, resource-heavy

Each of the three selected models was evaluated on the following four tasks using consistent prompts:

- General Question Answering
- Text Summarization
- Python Code Generation
- Creative Writing

Task 1: Question Answering

Prompt Used:

> Who was the president of the United States in 1985? What were some of the major political events that year?

Outputs & Observations:

- TinyLlama:

  The President of the United States in 1985 was Ronald Reagan, who served from January 20, 1981, to January 20, 1989. The major political events of 1985 were:

1. *Iran-Contra affair: An ongoing scandal involving the sale of arms to Iran and the cover-up by the Reagan administration of clandestine transactions with Nicaragua's Daniel Ortega.*
2. *Reagan's budget: In 1985, President Ronald Reagan proposed a $700 billion national defense budget for the following year.*
3. *Space Shuttle Challenger disaster: On January 28, 1986, the Challenger exploded mid-air during a test flight, killing all seven crew members.*
4. *Fall of the Berlin Wall: On November 9, 1989, Germany formally reunified.*

Correctly identified Ronald Reagan, but included events that occurred after 1985 such as the Space Shuttle Challenger disaster (1986) and the Berlin Wall falling (1989). These hallucinations suggest weaker temporal reasoning.

- Mistral-7B:

The President of the United States in 1985 was Ronald Reagan. He assumed office on January 20, 1981 and served until January 20, 1989. In 1985, Reagan's second term was well underway. Some major political events that year include:

1. *Space Shuttle Challenger Disaster (planned in 1985, happened in 1986).*
2. *Live Aid Concert on July 13, 1985.*
3. *Geneva Summit between Reagan and Gorbachev in November 1985.*
4. *Iran-Contra Affair (began in 1985).*
5. *Ethics Law Amendment of the Ethics in Government Act.*

Gave a strong and accurate response. It correctly mentioned Reagan, the Live Aid concert, the Geneva Summit, and the early roots of the Iran-Contra Affair. While it did reference the 1986 Challenger disaster, it correctly noted that the decision to launch was made in 1985.

- <u>LLaMA2:</u>

The president of the United States in 1985 was Ronald Reagan. Some of the major political events that year included:

1. *The Iran-Contra Affair*
2. *The Immigration Reform and Control Act (1986)*
3. *The Fair Housing Act (1986)*
4. *The Gulf War (1990–91)*
5. *The Tax Reform Act of 1986*
6. *The Civil Rights Act of 1986*
7. *Supreme Court decisions on affirmative action*
8. *Contra War support in Nicaragua*

Returned an extremely long list of events. While Reagan was correctly identified, several items were incorrect — such as mentioning the Gulf War (which happened in 1990–91) and listing legislation from 1986. This shows a tendency to overgenerate and mix timelines.

<u>Task 2: Text Summarization</u>

Prompt Used:

> Summarize the following paragraph in 2–3 sentences:
> (*The paragraph discussed Reagan's Cold War policy, domestic economic concerns, drug use, and the HIV/AIDS epidemic.*)

Outputs & Observations:

- TinyLlama:

In 1985, Ronald Reagan was president of the United States, and his second term was marked by geopolitical tensions and economic struggles. The country was in the middle of a Cold War-era conflict with the Soviet Union, and Reagan's administration emphasized a strong anti-communist stance through policy initiatives like the Strategic Defense Initiative. Domestically, concerns about the economy, drug use, and HIV/AIDS were also on the radar of the administration. At the same time, international relations remained a priority, with military and diplomatic pushes for change in the Soviet Union through means like negotiations.

Provided a well-structured summary that exceeded the 2–3 sentence limit but captured all key ideas. Included details about both domestic and international issues.

- Mistral-7B:

In 1985, Ronald Reagan was serving his second term as US President amidst Cold War tensions. His administration pursued a strong anti-communist stance, including the Strategic Defense Initiative (SDI). Domestically, concerns about the economy, drug use, and the HIV/AIDS epidemic surfaced, while internationally, the Reagan Administration maintained pressure on the Soviet Union through diplomatic and military means.

Delivered a clear, concise 3-sentence summary that matched the tone and length constraints perfectly. It emphasized the Cold War, the Strategic Defense Initiative, and key domestic concerns.

- LLaMA2:

  Ronald Reagan was in his second term as president of the United States in 1985, during which he pursued a strong anti-communist agenda. Domestically, there were concerns about the economy, drug use, and the HIV/AIDS epidemic. Internationally, the Reagan administration continued to pressure the Soviet Union through diplomatic and military means.

Also summarized effectively, though its tone was slightly more robotic. Still, it accurately reflected the original content and stayed within the required sentence range.

All three models captured the core ideas, but TinyLlama exceeded the length limit and introduced repetition. Mistral-7B offered the most polished summary, matching the tone, length, and content expectations. LLaMA2 also performed well, though its tone was slightly robotic.

Task 3: Code Generation

Prompt Used:

Write a Python function that checks if a number is prime.

Outputs & Observations:

- TinyLlama:

    Produced a convoluted and incorrect function:

```
def is_prime(n):
    """
    Checks if n is prime or not based on the Euler's totient function.

    Args:
        n (int): The number to check if prime or not.

    Returns:
        bool: True if n is prime, False otherwise.
    """

    # Check for divisibility by 4
    if n % 4 == 0:
        return False

    # Loop through all positive integers up to sqrt(n)
    for I in range(5, int(sqrt(n)) + 1, 2):

        # Calculate the product of digits
        prod = 1
        for digit in str(n):
            prod *= int(digit)

        # Check if product is less than n or not divisible by I
        if prod < n or (i ** 2 <= prod and i != i):
            return False

    # If we reached here, then n is prime
    return True
```

- Mistral-7B

    Returned a correct and efficient implementation

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True
```

- LLaMA2:
  Gave a mostly correct function, but with incorrect example outputs in the explanation:

```
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True
```

Task 4: Creative Writing

Prompt Used:

> Write a short story (3–5 sentences) about a robot who wants to learn how to dance.

Outputs & Observations:

- TinyLlama:
  Returned an overly long and emotionally dramatic story well beyond 5 sentences:

  Sophia, the advanced robot made by ARTISTIA, longed to express herself through dance. As her responsibilities grew, she felt disconnected from her human coworkers. One day, she began dancing during a meeting, and her fluid movements shocked the team. They saw her transformation as something almost human. Inspired, they hoped more robots could learn to express themselves like Sophia someday.

- Mistral-7B:
  Delivered a balanced, creative response within the prompt limits:

In a bustling city, a robot named B-320 dreamed of dancing like the humans he observed. He enrolled in a dance studio and practiced every evening. When the music started, he twirled and tapped with joy. Dancing became his way to connect with the world beyond his programming.

- LLaMA2:
  Gave a concise and energetic story:

  Zeta, the metallic marvel, always admired the way humans danced. He downloaded tutorials, practiced nonstop, and soon became the life of the party. Now, at every factory celebration, Zeta is the first to hit the dance floor — and the last to leave.

3.5 Setup & Installation

Setting up the local environment was a key part of this project. I installed Ollama on a MacBook Pro with Apple Silicon using the Homebrew package manager. The initial setup was straightforward: a single terminal command (brew install ollama) installed the software, and I started the service using brew services start ollama. Once installed, running a model was as simple as using ollama run <model_name>.

However, performance varied significantly based on the model's size. TinyLlama downloaded quickly and ran with no issues. Mistral-7B and LLaMA2, on the other hand, required downloading multiple gigabytes of data, which took several minutes and caused my system to lag during initial execution. In one instance, Mistral-7B temporarily froze my computer, making the mouse unusable and causing background music to glitch. While these heavier models did eventually run, this experience highlighted how resource-intensive larger LLMs can be, especially when working with local hardware.

4. Focused Experiment: Prompt Engineering

To evaluate how different prompting styles affect model output, I tested each LLM on a basic arithmetic problem using two styles:

- Prompt A (Direct):
  Q: If there are 3 apples and you eat 2, how many are left?

- Prompt B (Chain-of-Thought):

Q: If there are 3 apples and you eat 2, how many are left?
A: Let's think step by step. First, we start with 3 apples. If we eat 2, then we subtract 2 from 3, which leaves us with 1 apple. So, the answer is 1.

Outputs & Observations:

Model	**Prompt A Output**	**Prompt B Output**
TinyLlama	Gave the correct answer, but reasoning was repetitive and sometimes illogical	Correct, but added confusing logic about “subtracting 1” and repeating steps
Mistral-7B	Gave a direct, clean answer	Repeated the CoT reasoning clearly and confirmed the logic
LLaMA2	Gave a correct and friendly response	Echoed the step-by-step breakdown in a natural tone

: Adding Chain-of-Thought (CoT) reasoning improved clarity and naturalness in the larger models. Mistral and LLaMA2 gave more thoughtful, human-like explanations in response to Prompt B. TinyLlama, however, did not benefit as much and occasionally introduced extra or confusing logic.

This demonstrates how prompt structure directly impacts output quality, especially for more advanced models. CoT prompts can significantly improve model reasoning — particularly in complex or multi-step tasks.

4.5 Model Comparison Summary

Comparing TinyLlama, Mistral-7B, and LLaMA2 side-by-side revealed distinct personalities and performance levels for each model. TinyLlama was impressively lightweight and quick to respond, but its answers lacked structure, accuracy, and depth. It’s best suited for basic tasks or environments where speed and efficiency outweigh output quality.

Mistral-7B consistently delivered the strongest results across tasks, balancing clarity, reasoning, and creativity. It handled code generation with ease and produced high-quality summaries and creative writing samples. However, its performance came at the cost of system strain.

LLaMA2 produced generally strong outputs, especially in creative writing and reasoning tasks, but it tended to overgenerate and occasionally hallucinate facts. It was more stable than

Mistral on my system, but still required significant resources. Overall, Mistral would be my model of choice for local use if system performance were not an issue.

5. Analysis of Strengths & Weaknesses

Across the four evaluation tasks, each model demonstrated unique strengths and limitations in terms of accuracy, reasoning ability, creativity, and performance on a local machine.

In terms of accuracy and logical reasoning, Mistral-7B consistently delivered the most reliable and coherent outputs. It performed particularly well in summarization and coding tasks, maintaining both structure and factual correctness. TinyLlama, although surprisingly fast and responsive, struggled with internal logic. It often reached the right conclusion but failed to explain its reasoning clearly. LLaMA2 generally produced strong outputs but occasionally included inaccuracies or hallucinated details, especially when generating long-form content.

For code generation, Mistral again stood out, producing clean and correct implementations for common problems like checking for prime numbers. LLaMA2 was close behind, though its included explanations sometimes contradicted the code or gave incorrect examples. TinyLlama's output was the weakest, generating unnecessarily complex logic that was not only incorrect but difficult to follow.

When it came to creative expression, all three models performed well but in different ways. TinyLlama returned a rich, emotional story that exceeded the length limit and lacked structure, but showed strong creative instincts. Mistral-7B offered the best balance, delivering an expressive and well-contained story that matched the prompt. LLaMA2 gave a short, upbeat response that felt natural and was perfectly suited to the requested length.

The experiment in prompt engineering highlighted a key difference in how these models handle structured reasoning. Chain-of-Thought (CoT) prompting significantly improved the clarity and natural tone of responses from both Mistral-7B and LLaMA2. In contrast, TinyLlama showed minimal improvement, and in some cases, its reasoning became more convoluted with the added guidance.

Finally, in terms of system performance, TinyLlama was by far the most efficient and responsive. It downloaded quickly, responded almost instantly, and never caused any noticeable system slowdown. In contrast, Mistral-7B and LLaMA2 were extremely resource-intensive. Both models took significantly longer to initialize, and in particular, Mistral-7B temporarily froze my system and caused music playback to glitch and the mouse to become unresponsive during the initial run. LLaMA2 also required substantial memory and time to load, though it was more stable than Mistral once operational. These experiences underscore the tradeoff between model complexity and usability, especially when running LLMs locally on consumer hardware.

6. Reflections & Real-World Implications

This project gave me valuable hands-on insight into the capabilities, limitations, and practical challenges of running open-source large language models locally. I expected the models to differ in output quality, but I didn't realize how significant the trade-offs would be between performance and resource usage. While smaller models like TinyLlama were lightning-fast and easy on my system, their responses lacked depth and often included flawed reasoning. On the other hand, larger models like Mistral-7B and LLaMA2 produced far better results in terms of accuracy, creativity, and logical coherence but at the cost of serious strain on my machine.

The prompt engineering experiment, in particular, stood out as an important takeaway. Even a basic Chain-of-Thought format made a clear difference in how the larger models reasoned through problems. This shows how much potential there is for improving LLM performance not just by scaling model size, but by learning how to ask better questions.

In a real-world context, these findings have major implications for developers working with constrained hardware or embedded systems. For quick tasks, simple tools, or use cases where responsiveness matters more than perfect accuracy, smaller models like TinyLlama are ideal. But for high-stakes applications involving complex reasoning or user interaction (such as tutoring systems, customer service agents, or content generation tools) larger models, paired with well-crafted prompts, are clearly the better option.

This project didn't just help me compare model outputs; it helped me understand how to work with them more strategically. It gave me a new level of respect for both the models and the prompts that drive them, and made it clear how important those decisions are when deploying LLMs in practice.