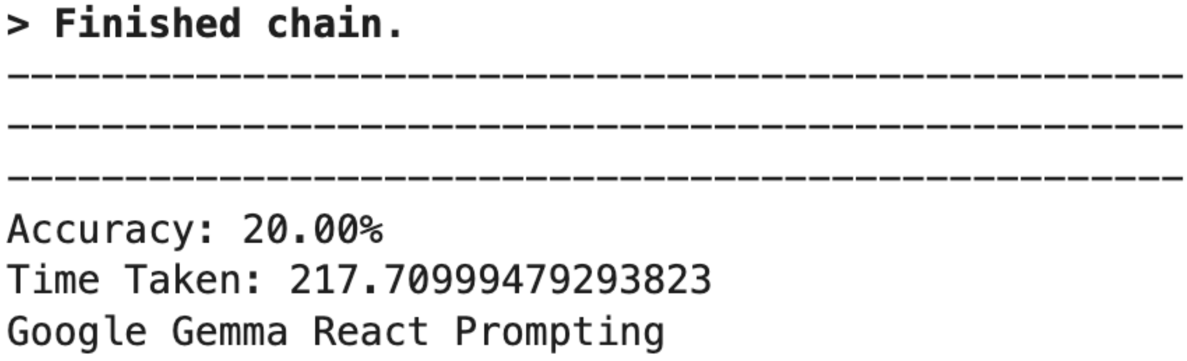
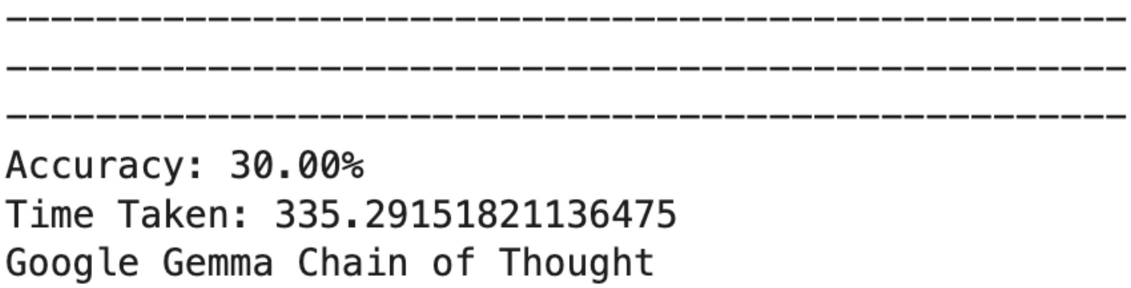
LLM Assignment 2
Aayush Ranjan (2021003)

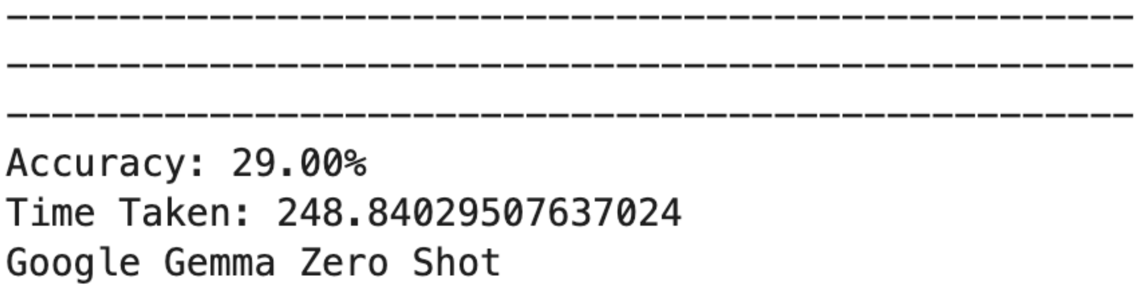
Results:



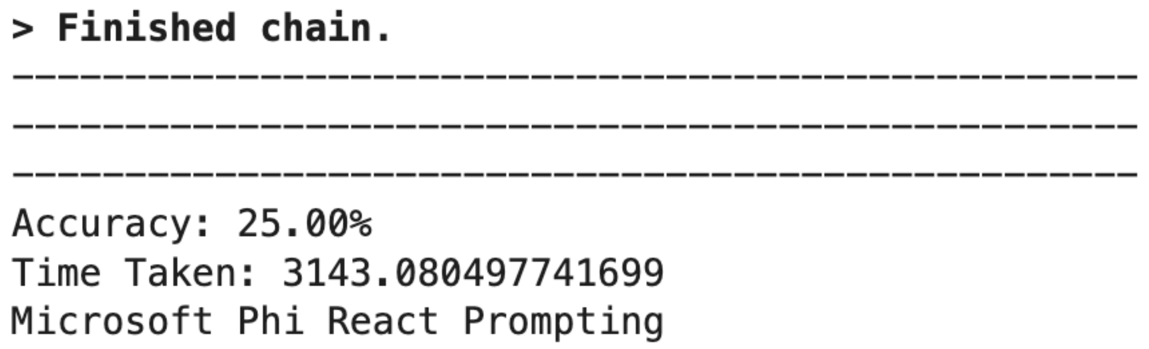
```
> Finished chain.
--------------------------------------------------
--------------------------------------------------
--------------------------------------------------
Accuracy: 20.00%
Time Taken: 217.70999479293823
Google Gemma React Prompting
```



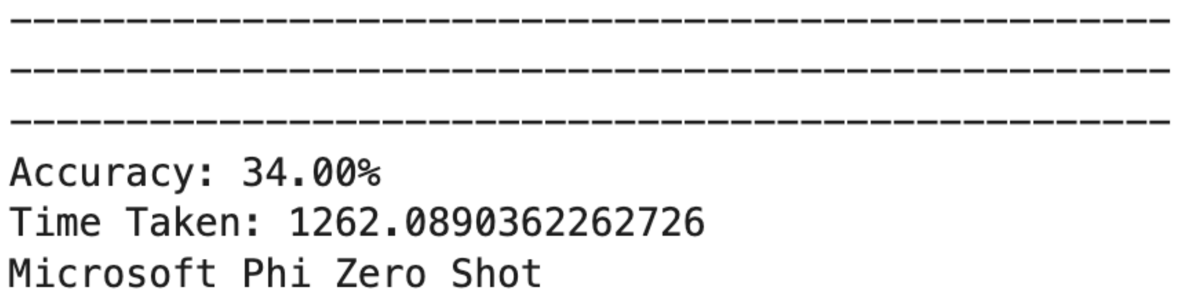
```
--------------------------------------------------
--------------------------------------------------
--------------------------------------------------
Accuracy: 30.00%
Time Taken: 335.29151821136475
Google Gemma Chain of Thought
```



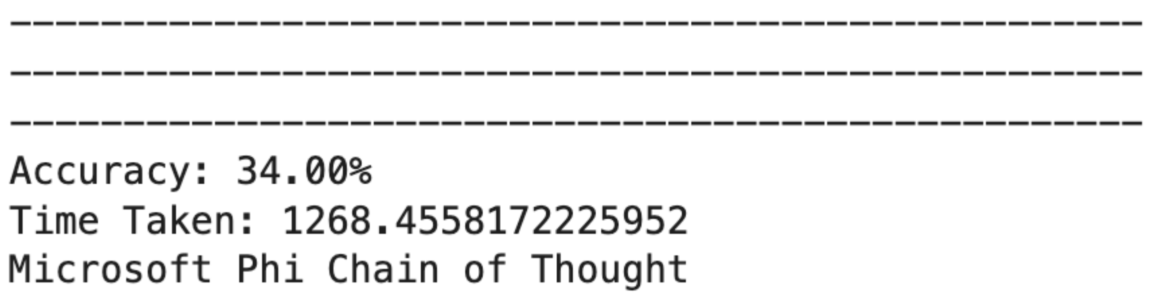
```
--------------------------------------------------
--------------------------------------------------
--------------------------------------------------
Accuracy: 29.00%
Time Taken: 248.84029507637024
Google Gemma Zero Shot
```



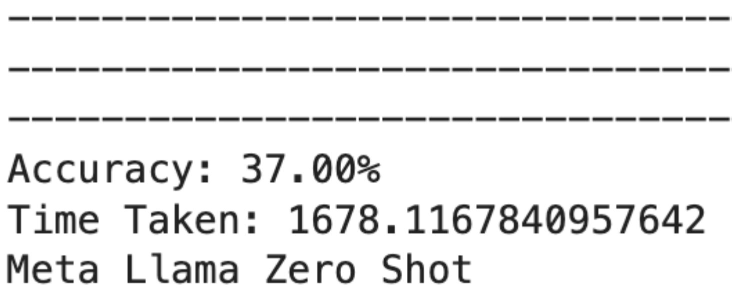
```
> Finished chain.
--------------------------------------------------
--------------------------------------------------
--------------------------------------------------
Accuracy: 25.00%
Time Taken: 3143.080497741699
Microsoft Phi React Prompting
```



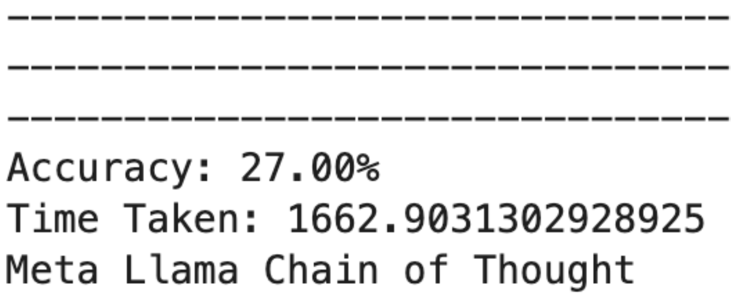
```
--------------------------------------------------
--------------------------------------------------
--------------------------------------------------
Accuracy: 34.00%
Time Taken: 1262.0890362262726
Microsoft Phi Zero Shot
```



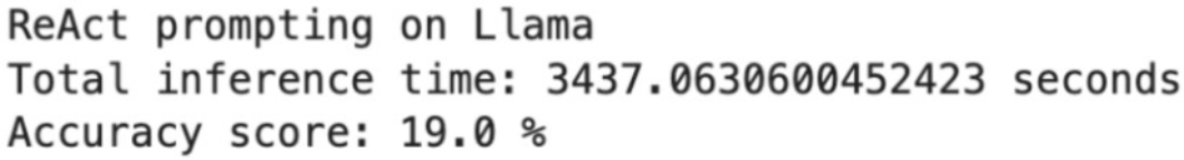
```
--------------------------------------------------
--------------------------------------------------
--------------------------------------------------
Accuracy: 34.00%
Time Taken: 1268.4558172225952
Microsoft Phi Chain of Thought
```



```
--------------------------------------------------
--------------------------------------------------
--------------------------------------------------
Accuracy: 37.00%
Time Taken: 1678.1167840957642
Meta Llama Zero Shot
```



```
--------------------------------------------------
--------------------------------------------------
--------------------------------------------------
Accuracy: 27.00%
Time Taken: 1662.9031302928925
Meta Llama Chain of Thought
```



```
ReAct prompting on Llama
Total inference time: 3437.0630600452423 seconds
Accuracy score: 19.0 %
```

Analysis of the Accuracies

1. Zero Shot Prompting:

- **Meta Llama (37%)** performed the best in this mode, indicating its ability to handle mathematical questions without additional guidance or reasoning prompts.
- **Microsoft Phi (34%)** is close, showing that it also has a strong general understanding and inference capability.
- **Google Gemma (29%)** lags behind, suggesting that it struggles with direct, unguided mathematical question-answering.

This indicates **Meta Llama's superior generalization ability** in mathematical problems compared to the others, likely due to its larger parameter size and training focus on general knowledge tasks.

2. Chain of Thought Prompting:

- **Microsoft Phi (34%)** leads in this mode, excelling at structured, step-by-step reasoning. This suggests Phi's architecture may better leverage logical processes or sequence completion when explicitly guided.
- **Google Gemma (30%)** performs similarly, benefiting from reasoning prompts, though slightly less efficiently.
- **Meta Llama (27%)**, interestingly, underperforms here compared to Zero Shot. This suggests that Meta Llama might not benefit as much from reasoning-based instructions or is more prone to deviating from accurate step-by-step reasoning under guided prompting.

Here, **Microsoft Phi stands out**, showing strength in tasks where logical steps are necessary for correct answers.

3. React Prompting:

- **Microsoft Phi (25%)** again outperforms the others, although its performance drops compared to Chain of Thought. This suggests that Phi can handle action-reasoning prompts but struggles with answer extraction and actions in React prompting, likely due to the nature of its architecture.
- **Google Gemma (20%)** continues to show weaker results, likely because it is less suited for prompt structures that require dynamic engagement with the prompt.
- **Meta Llama (19%)** is the lowest, indicating the model's limitations when handling React prompting scenarios, especially in mathematical question-answering. Since it wasn't designed for such tasks, it encounters issues extracting answers properly.

Analysis of the Inference Times

Google Gemma has the fastest inference times across all prompting types due to its smaller size (2B), making it ideal for real-time applications where speed is prioritized over accuracy.

Microsoft Phi performs well in Zero Shot and Chain of Thought tasks but faces a significant slowdown in React Prompting, likely due to its architecture struggling with dynamic, multi-step tasks.

Meta Llama, being the largest model, is the slowest overall. While it excels in reasoning accuracy, its high computational cost and slow React Prompting time make it less suitable for time-sensitive or interactive applications.

Model Size and Efficiency

Gemma 2B IT: This is the smallest model in this group (2 billion parameters), designed for efficient deployment on low-resource hardware like CPUs and edge devices. This makes it ideal for applications where computing resources or power are constrained.

Phi 3.5 Mini Instruct: A mid-sized model, balancing between size and performance. It offers a good compromise for applications needing more capacity than small models but still needing speed and low latency

Meta Llama 3.1 8B: With 8 billion parameters, this model is much larger, which means it generally requires more compute resources. However, it delivers stronger performance on complex tasks like reasoning, coding, and understanding deeper semantics. It excels in research and industrial applications where accuracy is more important than speed

Inference Speed

Gemma 2B IT: Being the smallest, this model offers the fastest inference times. It's suited for real-time applications where response speed is critical, such as chatbots or edge devices.

Phi 3.5 Mini Instruct: Slightly larger, it offers moderately fast inference times but adds more flexibility in task performance compared to Gemma 2B.

Meta Llama 3.1 8B: The largest model here, Meta Llama's inference speed is slower due to its size but it handles more complex and multi-step reasoning tasks more effectively. This makes it less suitable for applications requiring instant responses.

Output Quality

Gemma 2B IT: Strong performance in instruction-following and reasoning for its size. It performs well on text-based tasks, particularly in safety and responsibility-focused applications, but can struggle with tasks requiring high complexity.

Phi 3.5 Mini Instruct: Offers balanced output quality across multiple domains. It is good for general-purpose use and instruction-following but does not outperform larger models like Meta Llama 3.1 in more challenging tasks.

Meta Llama 3.1 8B: Delivers superior quality in complex text generation, multilingual capabilities, and logical reasoning. This model is especially strong for tasks like question answering, creative writing, and coding.

Performance

Gemma 2B IT: This model excels in mathematical reasoning and coding tasks, especially given its relatively small size. However, its instruction-following and general language understanding performance are lower compared to larger models like Meta Llama 3.1 . Its design is optimized for lightweight applications, making it efficient but not the best for deep reasoning tasks.

Phi-3.5 Mini Instruct: Phi-3.5 Mini strikes a balance between performance and efficiency. It outperforms Meta Llama 3.1 on certain multilingual and instruction-following benchmarks, showing versatility in diverse applications . Additionally, its small size allows it to be deployed in low-resource environments without significant performance trade-offs .

Meta Llama 3.1 8B Instruct: As the largest model, Meta Llama 3.1 provides superior performance in complex reasoning and deeper language understanding, especially in tasks requiring extensive context and knowledge. It surpasses both Gemma 2B and Phi-3.5 Mini in high-resource benchmarks but requires more computational resources .

References:

1) https://arxiv.org/pdf/2403.08295
2) https://arxiv.org/pdf/2404.14219
3) https://arxiv.org/pdf/2302.13971

Github Repository Link: https://github.com/aayush-2021003/LLM-Assgn-2