LAB #3

In this lab, you will test different prompt types and experiment with local language models to determine what works best for your preprocessed data. The goal is to explore how different approaches impact the quality and relevance of the responses.

01

LM Studio:

- Download a model from LM Studio
- Load it (keep the context size in mind)
- Start the server

02

Example script:

This script shows how you can use the LM Studio API.
Simply copy and paste the code into your IDE, adjust it as needed, and run it.

```
import requests

LM_STUDIO_API_URLimport requests


LM_STUDIO_API_URL = "http://localhost:1234/v1/chat/completions"


prompt = "YOUR_PROMT"


payload = {
   "model": "YOUR_MODEL",
   "messages": [
       {"role": "user", "content": prompt}
   ],
   "temperature": 0.7,
   "max_tokens": 200
}


response = requests.post(LM_STUDIO_API_URL, json=payload)


if response.status_code == 200:
   data = response.json()
   reply = data['choices'][0]['message']['content']
   print("LM Studio reply:\n", reply)
else:
   print("Request failed with status code:", response.status_code)
   print(response.text)
```

03

Test all six prompt types with your data:

- Zero-shot
- One-shot
- Few-shot
- Chain-of-thought
- Negative prompting
- Hybrid prompting

Your goal is to figure out which prompt type and instructions give the best results for our data.