# Machine Learning Engineer in the Generative AI Era - 3 Course Series

**Series 1: Data Engineering**, includes but not limited to data collection, data extraction, data filtering, data visualization, data analysis, synthetic data generation, data annotation platform, human data, prompt engineering, model evaluation, lightweight model finetuning, RAG, database, agent, etc.
**Series 2: Model Serving**, includes but not limited to …
**Series 3: Agent**, Includes but not limited to …

## Series 1 Data Engineering

All lectures should be at least 90 minutes, ideally 2 hours long.

Projects should have 2 levels of difficulty, **medium** and **challenging**. For the medium level, it should take students 2-4 hours to complete the homework. For the challenging level, it should take students 4-10 hours to complete the homework.

For all the projects, we need to make sure they are connected under one use case. For now, we want to build a research agent.

### Week 1: Introduction to large language models (LLM) and prompt engineering

- Lecture
    - What is generative AI and agentic AI, overview of the AI landscape
    - What is LLM, and what can they do (list use cases in both to-enterprise and to-consumer)
    - How LLM is trained (very high-level) -> leads to scaling law
    - How do we interact with LLM - prompt engineering
    - System prompt and user prompt
    - Few shot prompt engineering, chain-of-thought prompt engineering
    - Prompt engineering best practices (like the CO-STAR framework)
- Project 1 (medium): doesn’t require GPU resources
    - Task 1: Interact with ChatGPT, build basic prompt for your personalized research agent
    - Task 2: Rephrase your prompt using CO-STAR framework

  - Task 3: Learn about structured output, e.g., output in JSON format and contain multiple key
  - Task 4: Use XML format to rewrite your prompt, better to use nested XML format for complicated logic inside prompt, see how far can prompt engineering + ChatGPT go.
  - Task 5: List out defects you find, e.g., model doesn't follow your instruction, model make common sense mistakes, model doesn't know domain knowledge, etc.

## Week 2: LLM overview

- Lecture
  - Transformer and attention
  - Next token prediction and hallucination
  - LLM pretraining
  - Supervised finetuning (SFT)
  - Alignment (DPO and PPO)
  - Data needs for each phase, cost and challenges for each phase
  - Test-time scaling (O1, O3)
- Project 2 (medium): requires light GPU resources
  - Task 1: Familiar with Huggingface, download model like llama3.3 and perform inference
  - Task 2: Familiar with vLLM serving, use default code to serve a model and call the model for inference, e.g, Llama3 70B.
  - Task 3: Use your designed prompt from project 1, see how your local served model performs compared with ChatGPT

## Week 3: Pretrain data collection and extraction

- Lecture
  - More details about pretraining
  - Data needs for pretraining, diversity and quantity.
  - Data scraping, common crawl, html processing
  - OCR, get text from images
  - ASR, get text from audios
  - Pretraining data processing techniques, e.g., heuristic filtering, data deduplication, n-gram repetition, PII information removal. Can just follow llama3 or recent papers to go through all techniques.
- Project 3 (Challenging): requires light GPU resources for OCR and ASR
  - Task 1: Scrape a website, like arxiv for a specific subject

    - Task 2: Perform OCR on the downloaded papers
    - Task 3: Perform some basic filtering techniques like data deduplication.

## Week 4: RAG

- Lecture
    - RAG overview
    - Text embedding and distance metrics
    - Tokenization, filtering, chunking
    - Vector database
    - RAG and LLM integration, langchain
    - Evaluation and metrics
- Project 4 (Challenging): requires light GPU resources
    - Task 1: Perform text embedding and build an index
    - Task 2: Build a RAG system and integrate into your LLM, and see if RAG improves your experience.

## Week 5: Supervised finetuning (SFT) - I

- Lecture
    - What is SFT
    - How to get SFT data
    - ChatML format
    - Full finetune and LoRA finetuning
    - Deepspeed and TRL package
- Project 5 (Challenging): requires GPU resources for actual model tuning
    - Task 1: Find some related dataset on Huggingface and familiar with SFT data samples and ChatML format
    - Task 2: Perform LoRA finetuning to quickly experience SFT
    - Task 3: Perform full finetuning and see how easy it is to overfit, how to adjust hyperparameters, and compare with original model to see if SFT helps

## Week 6: Supervised finetuning (SFT) - II

- Lecture
    - How to make sure diversity of SFT data, both in terms of human and model
    - Synthetic data generation
    - How to make sure quality of SFT data
    - Use LLM as judge and rejection sampling
    - LLMsys arena and the Elo rating system
- Project 6 (Challenging): requires GPU resources for actual model tuning

    - Task 1: Given the model’s drawbacks and our scraped data from project 3, synthesis SFT data in ChatML format to help model tuning
    - Task 2: Perform ablation study on the quantity and diversity of the data you generated, data mixture ratio with the HF datasets you downloaded and fine the best ratio to tune the SFT model. The goal is to make sure the SFT tuned model is better than the original model (regardless of pretrained or instruction tuned llama)

## Week 7: Alignment

- Lecture
    - What is RLHF and preference data
    - Reward model and RL
    - PPO and DPO
    - Iterative DPO as in llama3 paper
    - Recent methods like GRPO
- Project 7 (Challenging): requires GPU resources for actual model tuning
    - Task 1: Get some huggingface datasets for alignment
    - Task 2: Build a data annotation platform in gradio, to self label some important data (or private data you really want)
    - Tasl 3: Use default TRL or other packages to perform DPO. Alignment is hard, so don’t expect this project will actually enhance the model's performance, just prepare the student with alignment techniques.

## Week 8: Hallucination, jailbreak and ethical considerations

- Lecture
    - What is hallucination
    - How to overcome hallucination
    - Talk about common jailbreak methods
    - Talk about ethical considerations, like bias mitigation
- Project 8 (Medium): doesn’t require GPU resources
    - Task 1: Try to jailbreak your hosted model or commercial models
    - Task 2: Try to get your model to hallucinate and consider ways to prevent it
    - Task 3: Look into safety related datasets on HF and Github, have basic understanding of how to safety align a model
    - Task 3: If any of the previous projects is not done perfectly, ask the students to spend more time on it. This week is kind of a break for them.

# Week 9: Voice agent

- Lecture
    - What is voice agent, talks about GPT4o-realtime
    - Chained method (ASR + LLM + TTS) and end2end models
    - How does ASR works (related to project 3) and popular models
    - How does TTS works and popular models
    - Talks about common data processing pipeline in audio, e.g, Emilia
- Project 9 (Challenging): requires light GPU resources for replicating Emilia pipeline
    - Use GPT4o-realtime or audio SDK to build a voice agent, experience the difference between the text-based and audio-based chatbot.
    - Or we can try to build sth similar to notebookLM.

# Week 10:  Agent

- Lecture
    - What is agent and what are the things agent can do but LLM cannot
    - Function call
    - MCP protocol
- Project 10 (Challenging):
    - Task 1: Build an agentic workflow that can search recent papers (maybe use perplexity API), perform OCR (use gemini API), generate a survey report (using deep research), and finally generate a podcast for you to listen to.