Recap - AI/ML Stack

– Agents, Flows, Model Evaluate –

Dev / ML Eng / Data Scienist — Amazon Bedrock → (Evaluate Model, Adapt Model)
— Amazon "Sagemaker AI"

→ "Transformer Architecture"

→ FM — unlabeded dataset (General purpose)
— KB (RAG?) + RAGAS, Amazon Bedrock | KB

→ Inference Parameters (fine-tuning)
- Temperature < low — very focused — Balanced (Mid)
  High — very creative
- Top P — % — Probability — 0.95 (cat dog, hat, sun)
  — lower
- Top K — # — Vocab < L, M, H
- max tokens # output tokens.

☆ setsequences → "}" "\ wor"

Day 2 - Agenda

① Prompt Eng (fine-tune Models)
  ↳ Concepts — Techniques.
  Basic/Adv — Zero-shot, few-shot, CoT, ToT
  Risks ☆ ⇒ Prompt Injection, other threats
  Sol: "Guardrails (Demo) → Model Eval, Agents, KB,

② Amazon Bedrock Components — Models — IPs
  +
③ Amazon Bedrock FMs > APIs functions
  = invoke Model

④ "Langchain" framework.
  ↳ Prompt Templates, extra features.

⑤ Architecture Patterns. → Demo — Agents
  — Guardrails.

Multi-modal Model = Text, images, videos, audio, web access (public) = general purpose.

FMs → Text generation, Text summarization, Code generation, Image generation (Delusion Architecture)
Information extration (KB, RAG, Chatbot, Agents)

~~Spl~~ spl models:
- Amazon < Nova, Titan
- Meta — Llama
- Anthropic — Claude
- AI21 — Jurassic

FMs — various stages of TRG to achieve the BEST Results.

① Pretraining — self-supervised learning (autogenerate labels)
  ↳ Provider
  RLHF (Reinforcement learning from Human feedback)
② fine-tuning (Supervised learning process) — (add specific, smaller datasets)
  - Instruction fine-tuning (specific instruction) → "Prompt tuning"
  - RLHF
③ Prompt Eng.

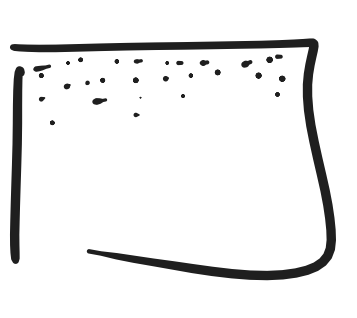
FMs < Text-to Text
  text to image

FM = Text to Text ⇒ LLM (NLP)
Natural Language Processing (Deep learning)

FM = Text to Image

Stability AI → Stable Diffusion (Model)

Diffusion Architecture
↳ Deep learning Architecture system.
↳ 2 step process.
  ↳ ① forward diffusion



  ② U-Net Model

LLMs → FMs

① Clear & Concise
BAD: Compute the sum ~~of~~ total of the seq of numerals: 4, 8, 12, 16.
Good: What is the sum of theses numbers - - - ?

② Include Context (if needed)
BAD: Summarize this article: [insert text]
Good: Provide a summary of this article to be used in a blog post: [——→]

Penalty parameters: (Text generation Model — Jurassic)
  ↓
  A121 labs
↳ Help control how repeative/varied the generated Text

① freq Penalty: looks at how often a token (word) has already been generated
  effect: Too frequently, the model is less likely to generate it again
  [very very very good,]

② Presence ⇒ check wheather a token is already present in the prompt
  effect: Reduces the chance of repeating words
  use case: Encourage new ideas rather than rephrasing the same thing.

③ Count Penalty = Applies a penatly based on # times a token appeared in the generated text so far.
  Effect: Similar to frequency, more fine-gained
  use case: Keeps the generated output diverse, avoid echoing the same token.