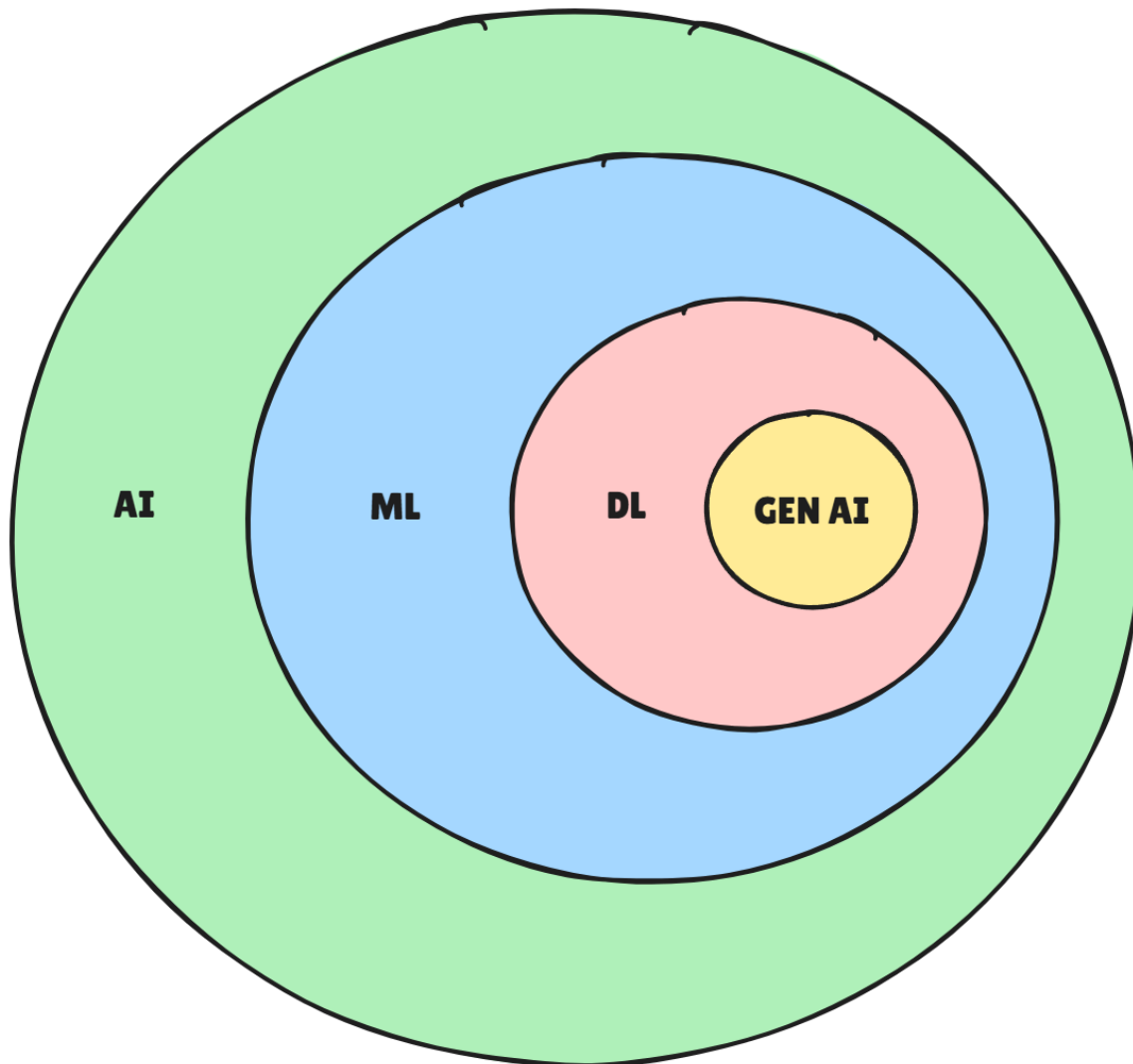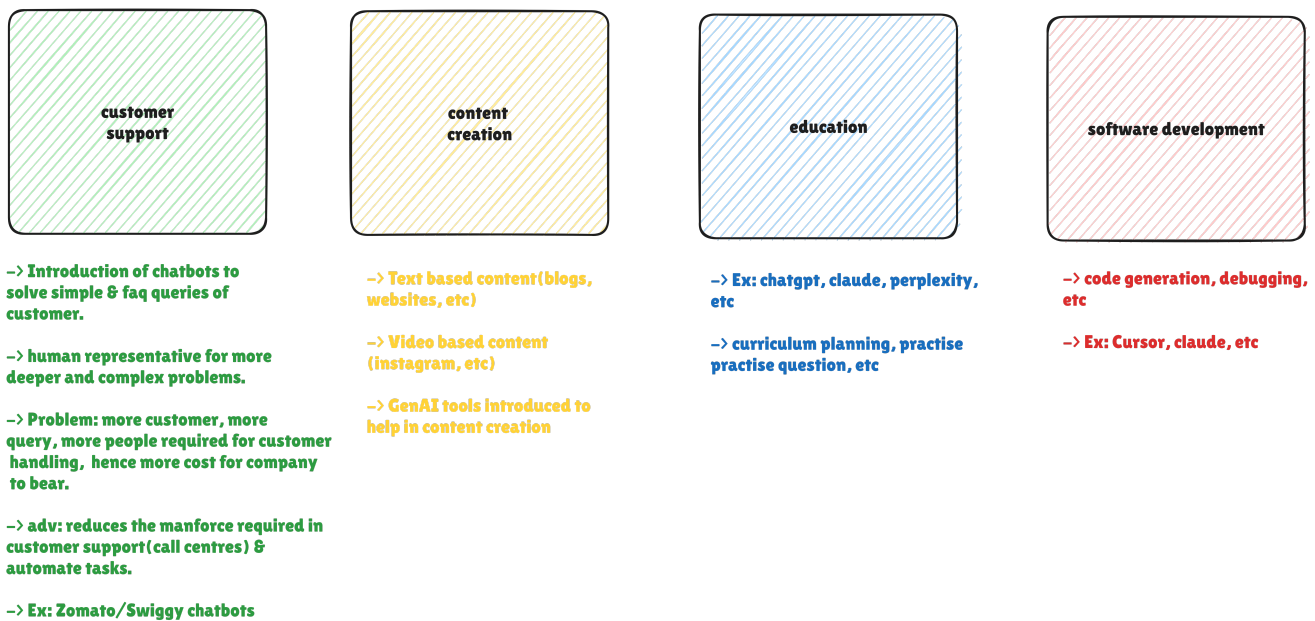# GenAI Roadmap for Beginners

## What is GenAI ?

Generative AI is a type of artificial intelligence that creates new content- such as text, images
, music or code-by learning patterns from existing data, mimicking human creativity.



## GenAI Impact Areas

| customer support | content creation | education | software development |
|---|---|---|---|

**customer support**

–> Introduction of chatbots to solve simple & faq queries of customer.

–> human representative for more deeper and complex problems.

–> Problem: more customer, more query, more people required for customer handling, hence more cost for company to bear.

–> adv: reduces the manforce required in customer support(call centres) & automate tasks.

–> Ex: Zomato/Swiggy chatbots

**content creation**

–> Text based content(blogs, websites, etc)

–> Video based content (instagram, etc)

–> GenAI tools introduced to help in content creation

**education**

–> Ex: chatgpt, claude, perplexity, etc

–> curriculum planning, practise practise question, etc

**software development**

–> code generation, debugging, etc

–> Ex: Cursor, claude, etc

# Is GenAI Successful ?

-> If the series of question asked answer is yes then it is successful.

1. Does it solve real world problems?
   Yes, in fields of customer support, education, etc as discussed earlier.
2. Is it useful on a daily basis ?
   Yes, Ex: We use chatgpt for assignments, etc.
3. Is it impacting the world economics?
   Yes, Ex: With the introduction of deepseek R1 into market, US based tech companies stocks plummeted 1 trillion dollar.
4. Is it creating new jobs?
   Yes, Ex: Job opening of post AI Engineer -> demand for this role is also increasing rapidly. In the next 5 years it might be as popular as SDE / Web Dev role.
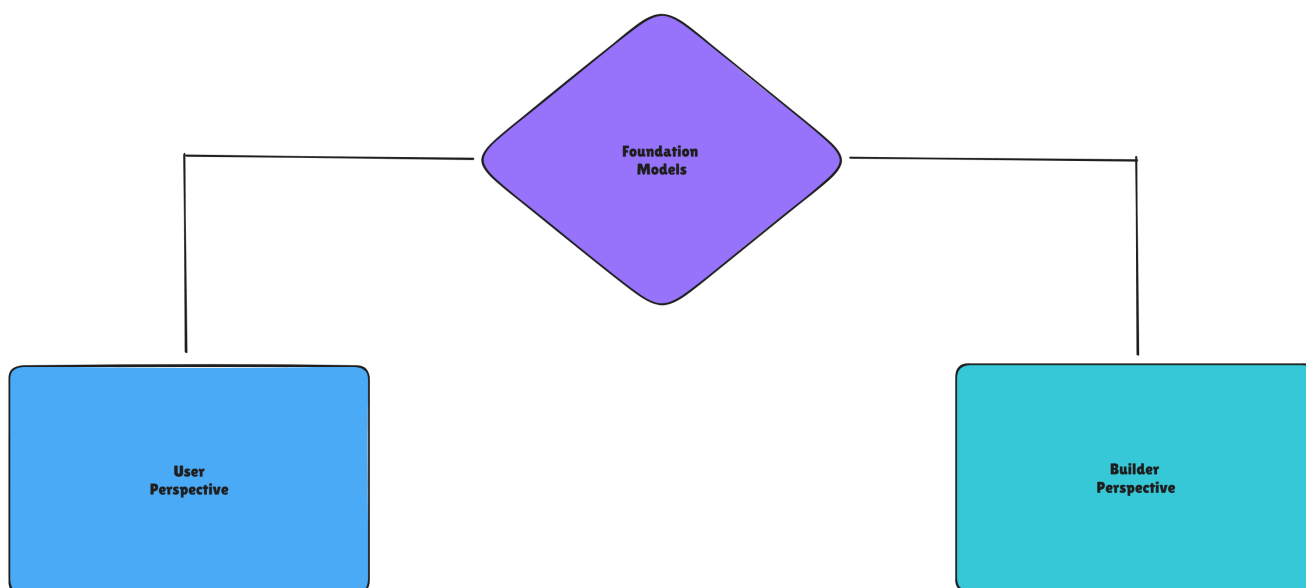5. Is it accessible?
   Yes, Ex: Chatbots like chatGPT etc can be accessed with simple english without needing to write any line of code.

# The Problem With GenAI

- Very fast progress => Daily new advancements => To be able to track it and learn it is challenging
- So much noise around GenAI =>There is so much FOMO merchants online ( social media influencers, etc who says if you don't leaarn GenAI you will be left behind, If you are not learning GenAI what are you doing, etc) this can be very demotivating.
- No single Source to learn (as it is a new technology).

# Mental Model

```
                    ┌─────────────┐
                    │  Foundation │
                    │    Models   │
                    └─────────────┘
           ┌──────────┴──────────────┐
  ┌─────────────┐            ┌─────────────┐
  │    User     │            │   Builder   │
  │ Perspective │            │ Perspective │
  └─────────────┘            └─────────────┘
```

# Foundation Models

- Foundation Models are at the centre of our Mental Model
- It is a type of large scale AI model which require huge amount of data and require a lot of computational power and very costly to train.
- One adv is that, unlike other ML models these are *specialised* models meaning till now whatever ML models we have built are task specific model, for ex: stock prediction model which will be able to predict future stock prices only, it cannot predict cricket scores. So these are specific models.
- But foundation models are *generalised* models, it can perform more than one task and it is like that because these models have 1. huge architecture and 2. are fed with huge amount of data.
- Ex of foundational model -> LLMs -> can perform text generation , sentiment analysis, summarization, Q/A, etc...
- All because during training process of LLM, AI model with many parameters are fed with huge internet size of data.
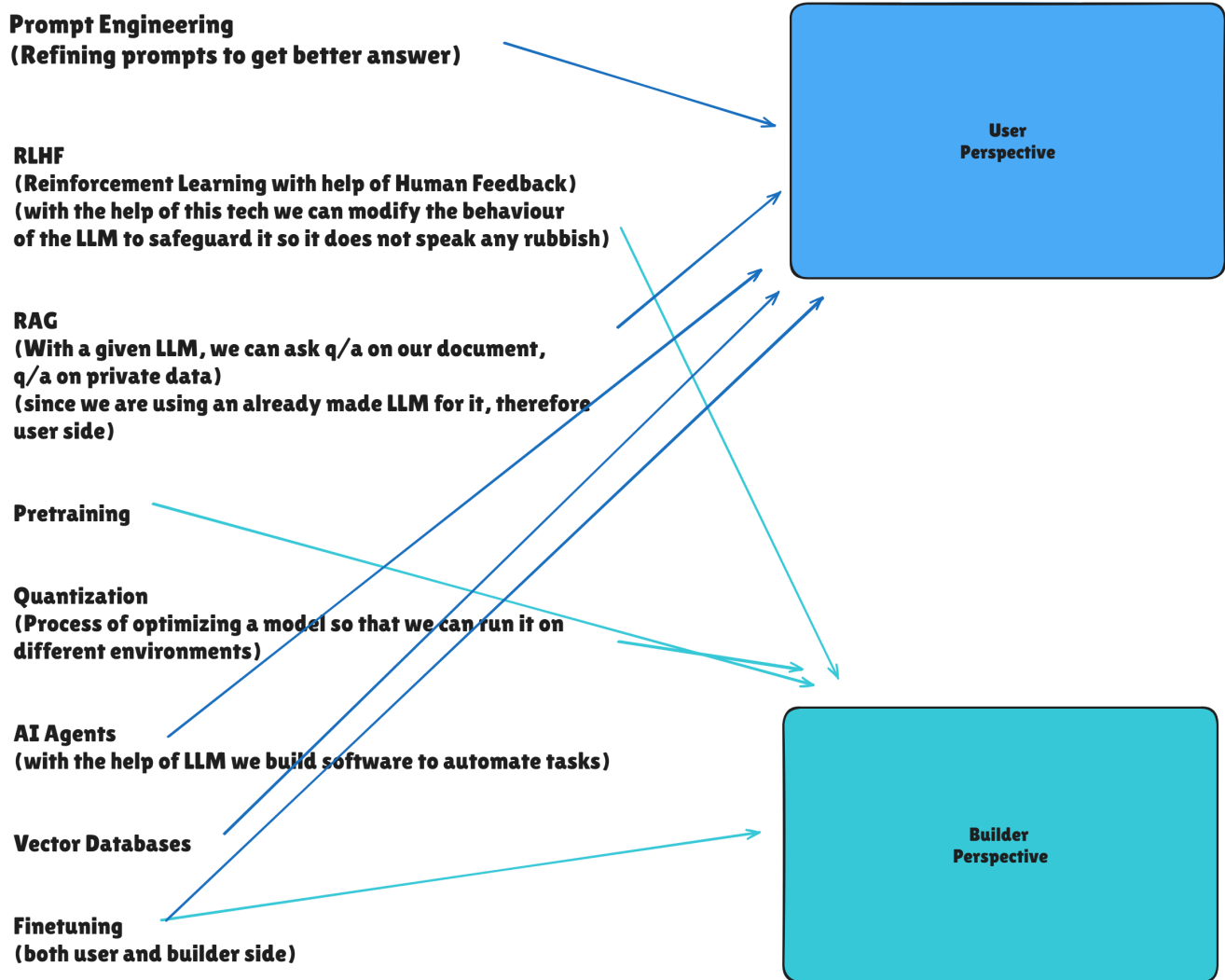- Foundation models also consist of LMMs (large multimodal models)

At the centre we have foundational model
In GenAI we can do two task

1. Build foundation models
2. Use foundation models

**User Perspective** -> Using already built LLMs for our tasks.
**Builder Perspective** -> Us, as a big company make foundation model and deploy it.

**Prompt Engineering**
**(Refining prompts to get better answer)**

**RLHF**
**(Reinforcement Learning with help of Human Feedback)**
**(with the help of this tech we can modify the behaviour**
**of the LLM to safeguard it so it does not speak any rubbish)**

**RAG**
**(With a given LLM, we can ask q/a on our document,**
**q/a on private data)**
**(since we are using an already made LLM for it, therefore**
**user side)**

**Pretraining**

**Quantization**
**(Process of optimizing a model so that we can run it on**
**different environments)**

**AI Agents**
**(with the help of LLM we build software to automate tasks)**

**Vector Databases**

**Finetuning**
**(both user and builder side)**

**User
Perspective**

**Builder
Perspective**

# Curriculum

# The Builder's Perspective

Prerequisite -> ML & DL fundamentals, DL framework (tensorflow/pytorch)

## Modules/Step

1. Transformer Architecture
2. Types of Transformers
   2.1 Encoder Only(BERT)
   2.2 Decoder Only(GPT)
   3.3 Encoder & Decoder based(T5)
3. Pretraining
   3.1 Training Objective
   3.2 Tokenization Strategies
   3.3 Training Strategies
   3.4 Handling challenges

4. Optimization
    4.1 Training Optimizations
    4.2 Model compression
    4.3 Optimizing Inference
5. Finetuning
    5.1 Task specific tuning
    5.2 Instruction tuning
    5.3 Continual Pretraining
    5.4 RLHF
    5.5 PEFT
6. Evaluation
7. Deployment

# The User's Perspective

1. Building Basic LLM Apps
    1.1 Open Source vs Closed Source LLMs
    1.2 Using LLM APIs
    1.3 LangChain
    1.4 HuggingFace
    1.5 Ollama
2. Prompt Engineering
3. RAG
4. Fine Tuning
5. Agents
6. LLMOps
7. Miscellaneous

# Do we need to learn both Sides ?

Builder Side
Data Scientist

AI
Engineer

User Side
Software Developers
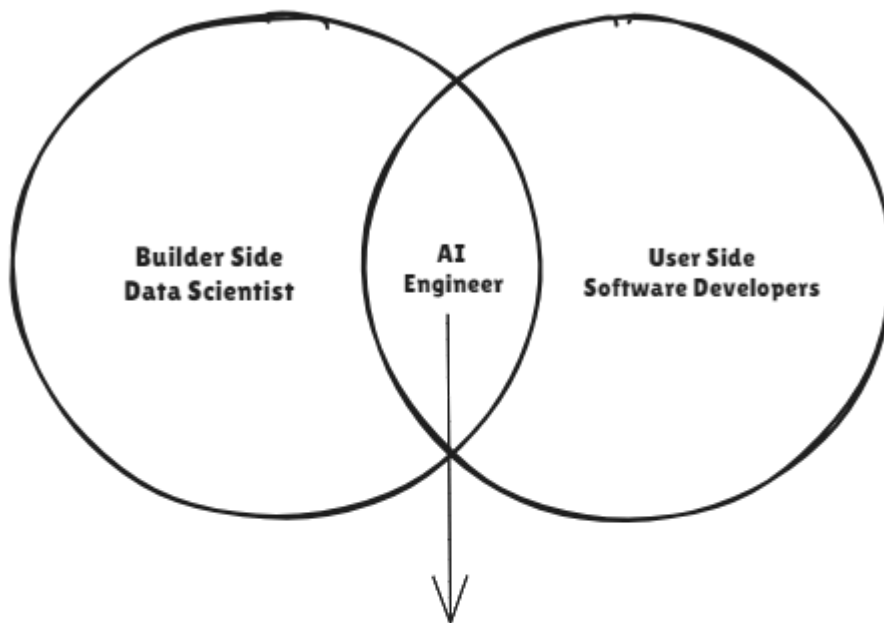
**Main role is to build LLM based app.**
**They are given a LLM (foundation model)**
**& on it they make an app but they also have deep**
**idea about  builder side too.**

**If we have a builder side knowledge then we can**
**operate better on user side.**