Introduction to Machine Learning Series

Rola Dali
ML Architect
February 2025

>whoami

- Machine Learning Architect @ RapidScale
- Academic: PhD in NeuroScience & Bioinformatics, 2017
- AWS enthusiast:
  - AWS Community Builder
  - AWS Montreal Meetup co-lead
  - AWS Ambassador (Golden Jacket All Star)

Series Content

ML101: Introduction to Machine Learning

ML102: Machine Learning under the hood

ML103: Introduction to GenerativeAI

ML104: Architecting GenAI systems

Thank you

ML101: Introduction to Machine Learning

Rola Dali
ML Architect
February 2025

Agenda

01 Machine Learning

- ML Definition
- ML vs Computer Science
- ML vs Statistics
- ML patterns & anti-pattens
- ML Lifecycle

02 ML on AWS

03 AI vs HI

What is Machine Learning?

Definition

Dictionary

Definitions from Oxford Languages · Learn more

machine learning

noun

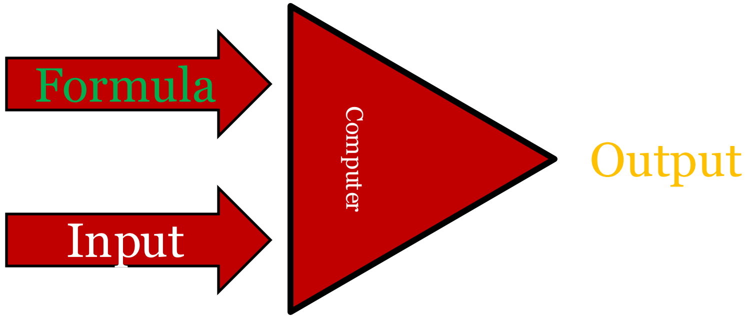
the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyze and draw inferences from patterns in data.

=> Computer systems + powerful mathematics to learn patterns in data without being explicitly taught

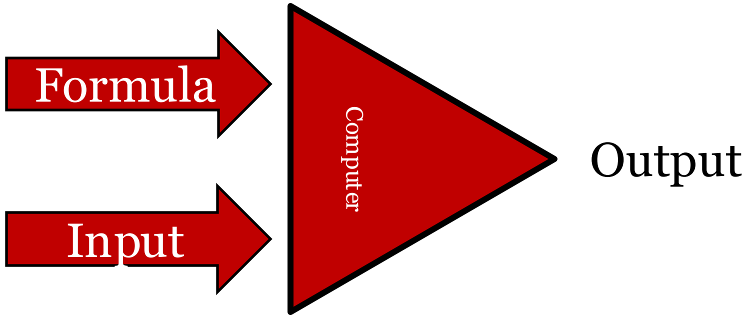
=> Mathematical algorithms powered by computer systems and learn from data

What is Machine Learning?
ML vs Computer Science

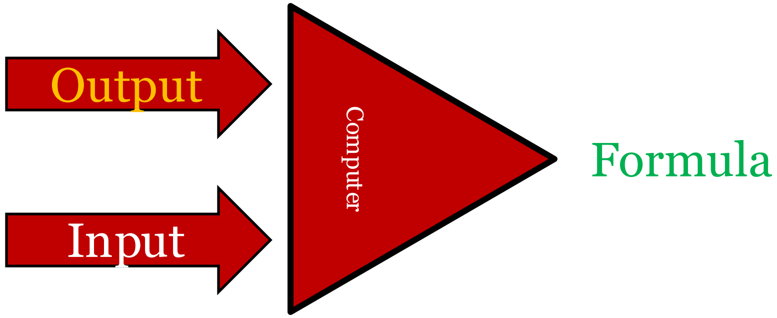
Traditional Software during <u>development</u>



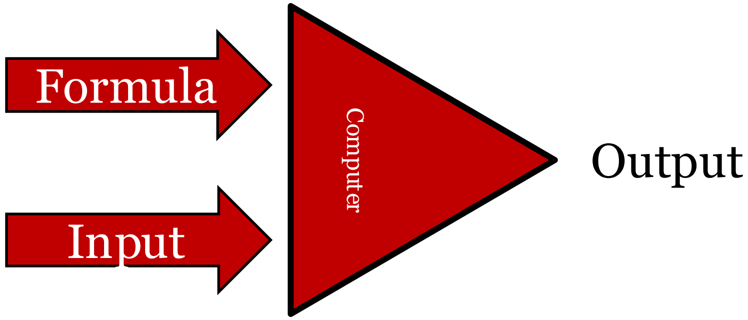
Traditional Software during <u>deployment</u>



Machine Learning during <u>development</u>/Training



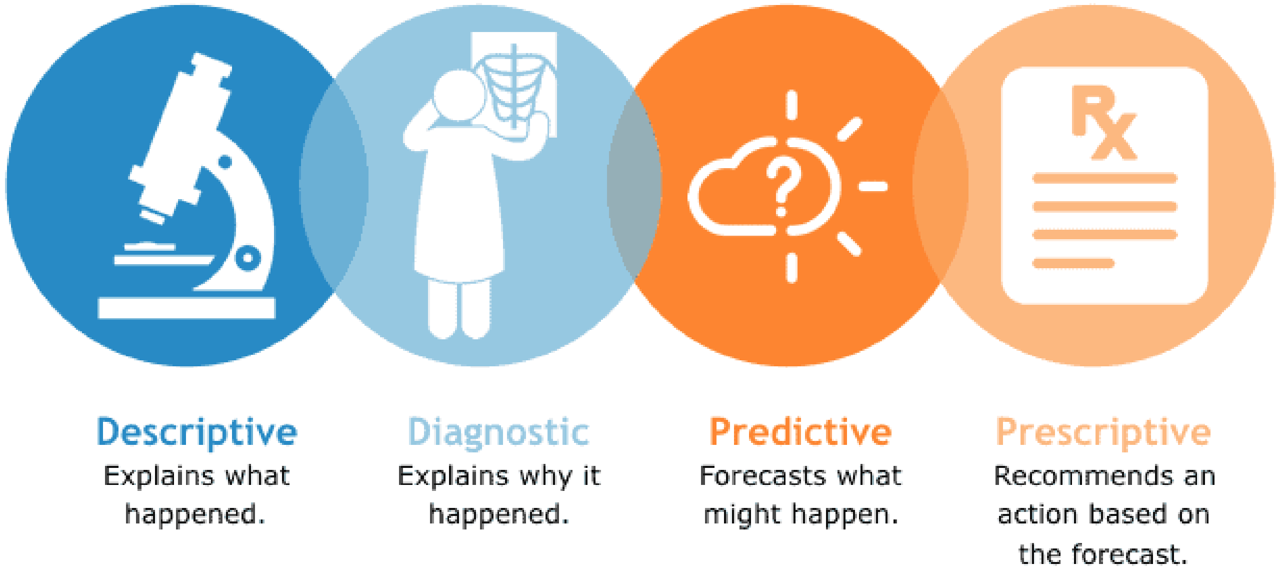
Machine Learning during <u>deployment</u>/Inference



Formula ~ pattern ~ algorithm ~ recipe ~ instructions ~ model

What is Machine Learning?

ML vs statistics: types of analytics



Hindsight **Insight** **Foresight**

Statistics **Machine Learning** **+ Business knowledge**

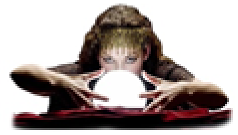
Statistics & ML:

- both are part of the data toolset
- both aim to define a mathematical representation of a real-world system (model)
- Statistics is concerned with understanding population descriptors
- ML is more concerned with predicting unseen data

ML Super Power:

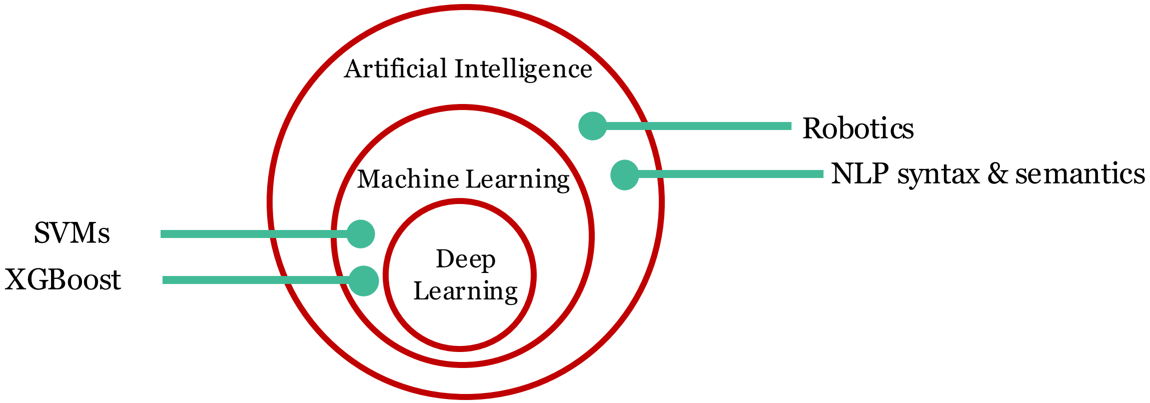
"Prediction": Using information you have to fill in for information you do not have

Importance: reducing uncertainty & being more prepared



What is Machine Learning?
ML terminology

- Artificial Intelligence (AI): Techniques that enable computers to mimic human behavior
- Machine Learning (ML): AI techniques that allow computers to learn without explicit programming
- Deep Learning: A subset of ML which uses multi-layer neural networks inspired by the human brain
- Generative AI: A type of AI that allows computers to generate new content



What is Machine Learning?
The rise of ML

- “AI godfathers”: 2018 Turing Award: deep neural networks:

Geoffrey Hinton (PhD, 1977), Yann LeCun (PhD, 1987) and Yoshua Bengio (PhD, 1991)

- ML has been around for a long time (1943). Why did it become popular more recently?
  - Increase in number of sensors/devices: We have loads of data
  - Increase in computer speed and memory: We can process the data
  - Better ML algorithms and software for easy deployment: Barrier for entry is being lowered
  - Increasing demand for customized solutions and data-driven systems: realizing the potential power in data & ML

Why has AI gained mass attention?

ML Foundational Pillars:

Machine Learning

- Data
- Mathematical Algorithms
- Compute systems

- A convergence of algorithmic advances, data proliferation, and tremendous increases in computing power and storage has propelled AI from hype to reality.

Reference: McKinsey. Executive Guide to AI

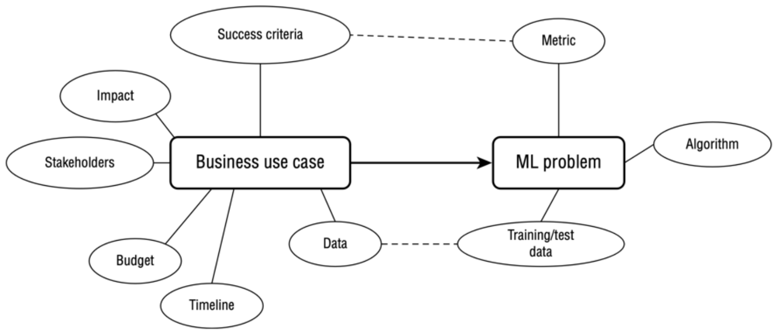
ML fit & anti-patterns

- When is ML a great fit?
  - Scale: Machines scale better than humans
  - Change: Patterns that change regularly can be better automated with MLOps than manually
  - Complexity: Patterns that are too complex to tease out can be better captured by ML

- ML anti-patterns:
  - If there is not enough data or it is not good quality (if you are training your own model)
  - If simpler solutions do the trick
  - If it is not cost-effective

- A word of caution:
  - Technical objective vs business goals

FIGURE 1.1 Business case to ML problem

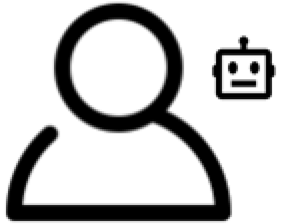


Official Google Cloud Certified Professional ML Engineer

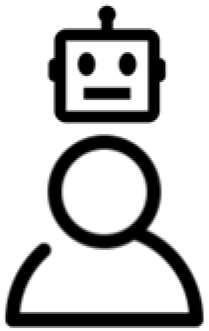
ML automation Stages: Human-Machine Interactions

Possible integrations of ML/automation in a working environment:

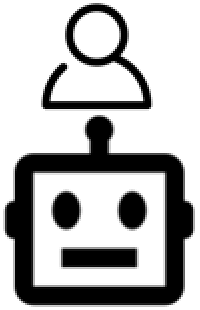
Human only systems
- human running the full process



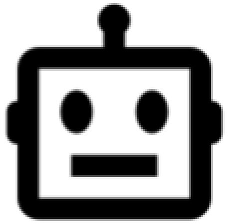
Shadow mode
- ML shadows human but its output not used



AI assistance
- Human is primary driver
- Human can tap into an ML system for help



Partial automation
- ML does the work
- Human gating for validation & approval

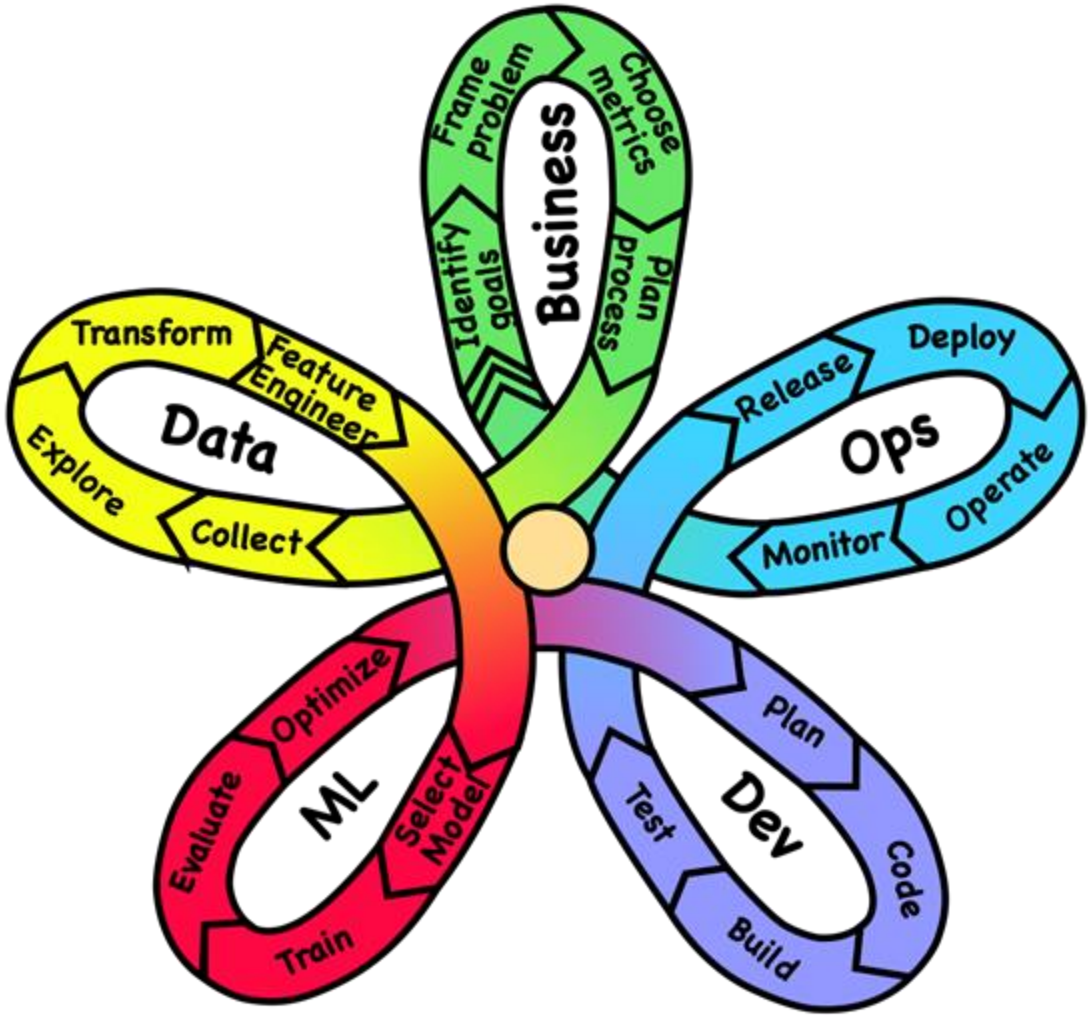


Machine only systems
- Full automation
- ML running the full process

- ML is an iterative & experimental process towards maturity: start early
- “If you wait for the technology to prove its worth to the rest of the industry before you jump in, you might end up years or decades behind the competition”

Machine Learning Lifecycle
The ML process

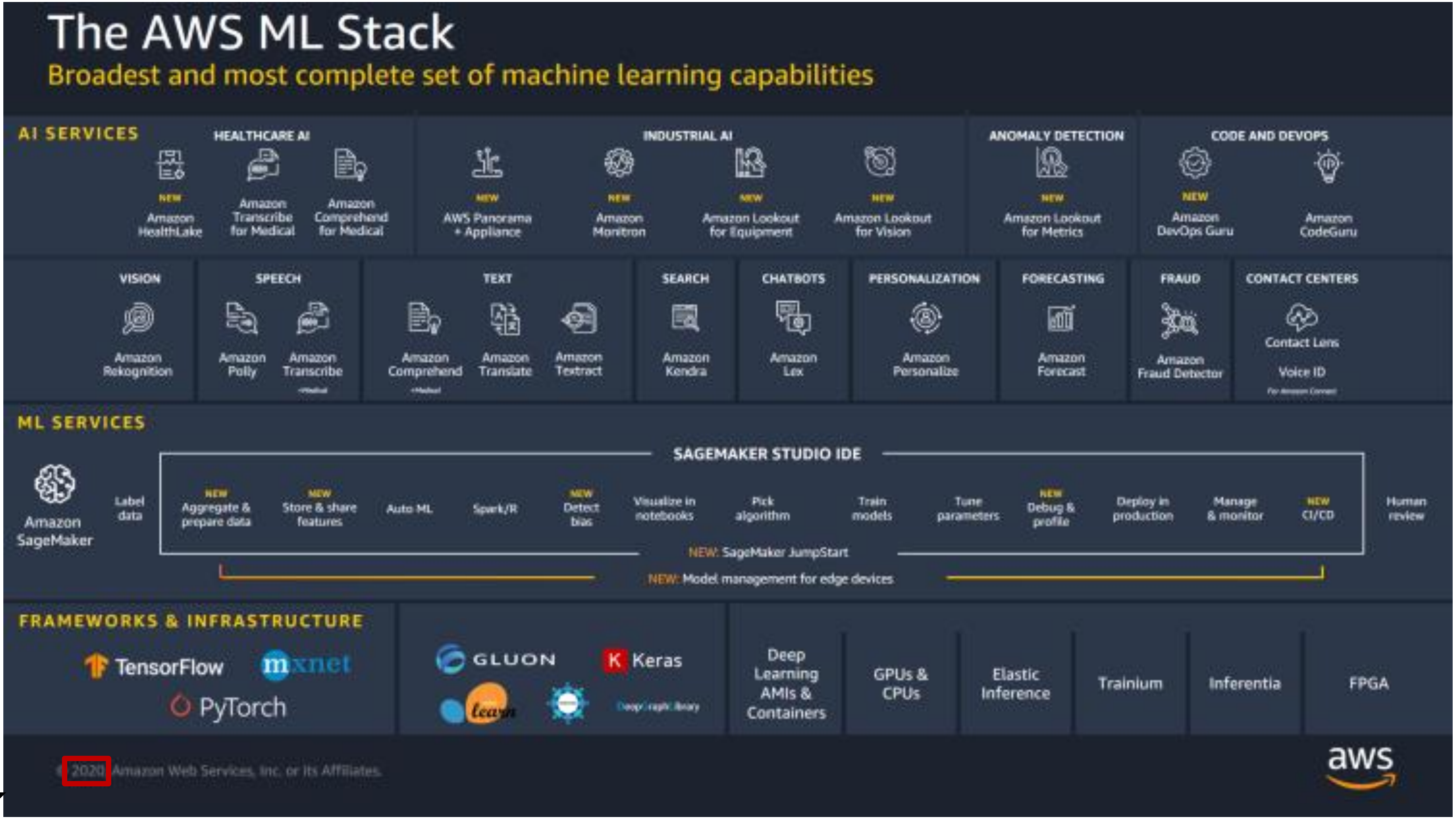
1. Define & Frame:
   1. Understand objective & frame problem as ML task
2. Data Preparation
   1. Collect data
   2. Data Engineering
   3. Feature Engineering
   4. Data labeling
3. Model Development
   1. Model Selection
   2. Model Training
   3. Model Evaluation
4. Deployment & Serving Model
5. Maintenance & Monitoring



ML on AWS: Overview

EASE OF USE

CONTROL



Personas

Anyone

Data Scientist

ML engineer

Generative AI Stack

APPLICATIONS THAT LEVERAGE LLMs AND OTHER FMs

- Amazon Q
- Amazon Q in Amazon QuickSight
- Amazon Q in Amazon Connect
- Amazon CodeWhisperer

TOOLS TO BUILD WITH LLMs AND OTHER FMs

Amazon Bedrock

Guardrails | Agents | Customization Capabilities

INFRASTRUCTURE FOR FM TRAINING AND INFERENCE

- GPUs
- Trainium
- Inferentia
- SageMaker
- UltraClusters
- EFA
- EC2 Capacity Blocks
- Nitro
- Neuron

Brains & Bots: Human Brain VS Artificial Intelligence

	Brains	Bots	Conclusion	Winner
Predictive Machines	Predicts events	Predicts events	Both work predictively & adjust	
Base Counts	100T synapses	~2T parameters in SoTA	Brains ~50X interconnected Better at integrating data wholistically	
Training Time	Evolving for 300K+ your age in fine tuning	~100 yrs old as a field product of Brain ingenuity	Bots have had much less time BUT benefit from brain ingenuity	
Speed	Neurotransmitters in liquid: ~ 200 Hz	Electrons in transistors CPU clock rate > 10GHz	Bots ~50X faster than brains	
Input Modalities	Tethered to biology 5 Senses	Unlimited Input	Bots have unlimited input streams	

- Machine Learning systems have HUGE potential given their speed and augmentation capability
- They will be the workhorse of the future

=> LEARN ML

ML 101 resources

Andrew Ng on Coursera

Towardsdatascience

Kaggle

KDNuggets

DLRL 2015 lectures: DLRL2015
DLRL 2016 lectures: DLRL2016
DLRL 2017 lectures: DLRL2017
DLRL 2018 lectures: DLRL2018
DLRL 2019 lectures: DLRL2019

Learn ML on AWS

DeepLearning.ai

ML 101 resources

AI For Everyone

Machine Learning Specialization

Supervised Machine Learning

Deep Learning Specialization

Unsupervised Learning, Recommenders, Reinforcement Learning

Structuring Machine Learning Projects

Machine Learning Engineering for Production (MLOps)

Thank you

ML102: Machine Learning under the hood

Rola Dali
ML Architect
February 2025

Types of Machine Learning

Types of ML:

- Supervised Learning: model learns from data with labels
- Unsupervised Learning: model learns from data without labels
- Reinforcement learning: agent solves multi-step problem by maximizing reward
- Generative AI: models that generate new content

Predictive/Classical Machine Leaning

- AI is a set of tools for general purpose tasks. It is essentially input to output mapping.
- Last decade was about optimizing and understanding supervised learning. This decade is more GenAI.
- If a model is small, increasing amount of data will eventually plateau performance (bias). With larger models, the more data, the better the performance.

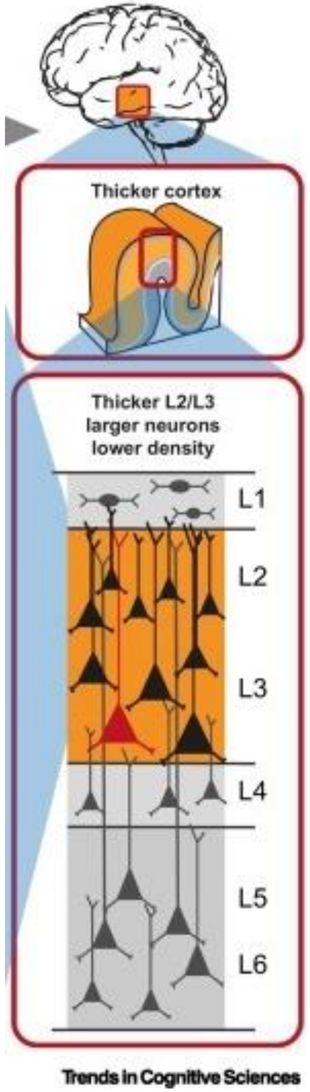
What is a 'Model'?

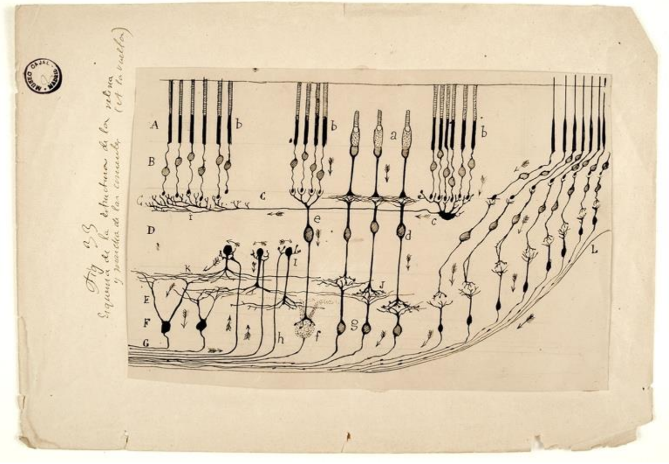
- A Model is a Mathematical representation of a real-world system
- Model ~ Algorithm ~ Formula ~ Pattern
- Machine learning models are just BIG statistical calculators
- Model Components:
  - Y = Class to be predicted = Model output
  - X = Input data = features
  - Parameters = Weights = tunable numbers in the model that encode the learning
  - Model Architecture/class = Formula = Equation
- The model class defines the mathematical formula used:
  - Linear Regression: $y = ax + b$
  - Logistic Regression: $\text{sigmoid}(y = ax + b)$
  - As models become too big, the exact mathematical equation becomes too complex. We visualize their architecture with abstract diagrams

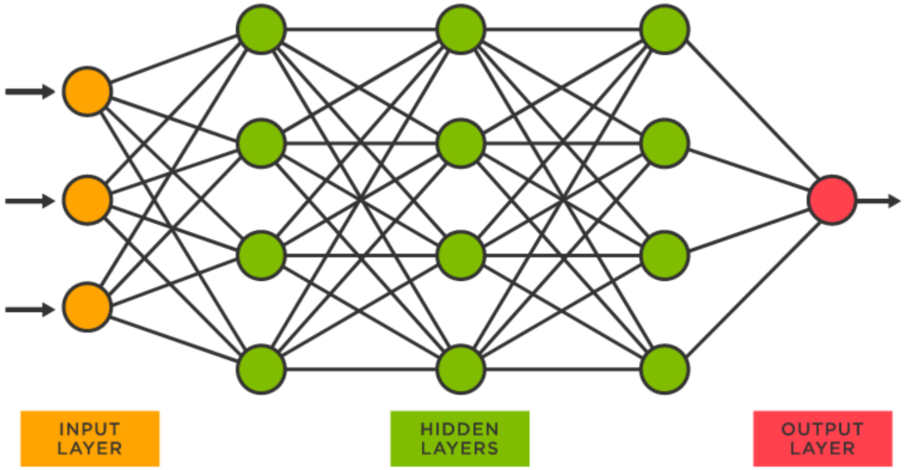
Types of Algorithms

- Linear Regression
- Logistics Regression
- Decision Trees
- Random Forest
- Naive Bayes
- SVM
- Neural Networks
- Nearest Neighbor
- Clustering
- Dimensionality reduction: PCA, autoencoders, TSNe

Deep Learning inspired by the human brain







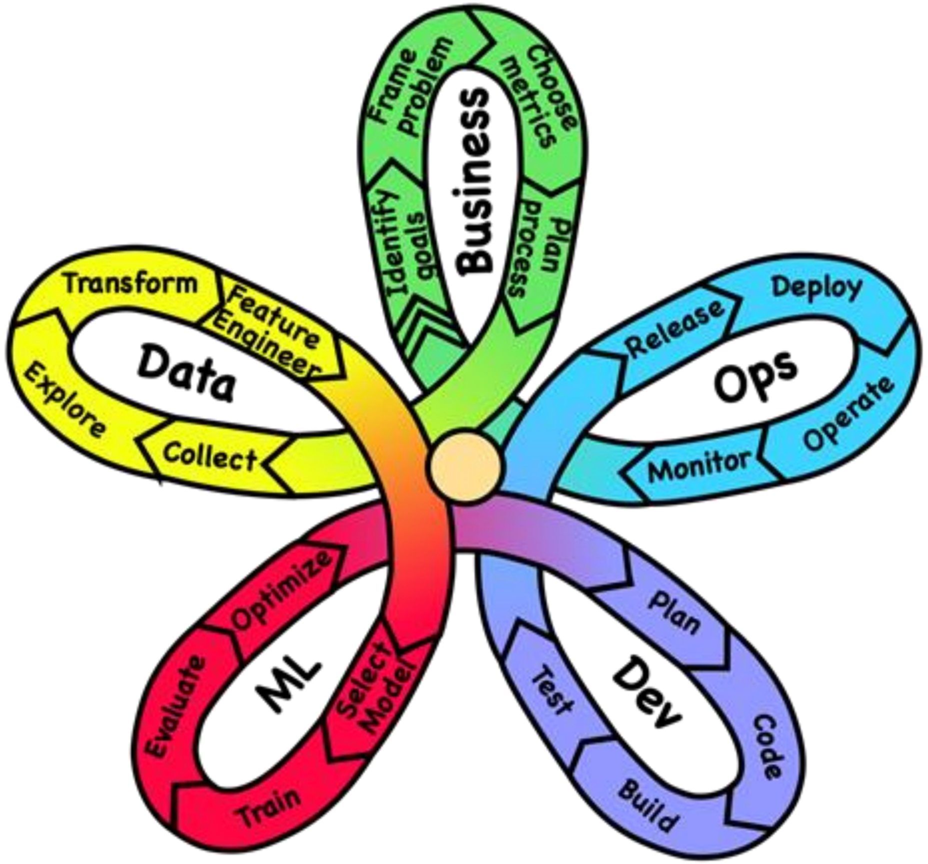
Trends in Cognitive Sciences, ISSN: 1364-6613, Vol: 26, Issue: 11, Page: 909-922

Parameters vs HyperParameters

- A model parameter is a tunable variable that is internal to the model & whose value can be learned from data through training.
- A model hyperparameter is a variable that is external to the model & whose value cannot be learned from data. Hyperparameters instead control the learning process
  - Examples: learning rate, number of epochs, regularization parameter, …
- When you download a pre-trained model, you are downloading the parameters/weights.
  - Models are sized by the number of parameters
  - Hyperparameters are good to know as they might explain model behavior for troubleshooting but are not relevant to deploying a model.

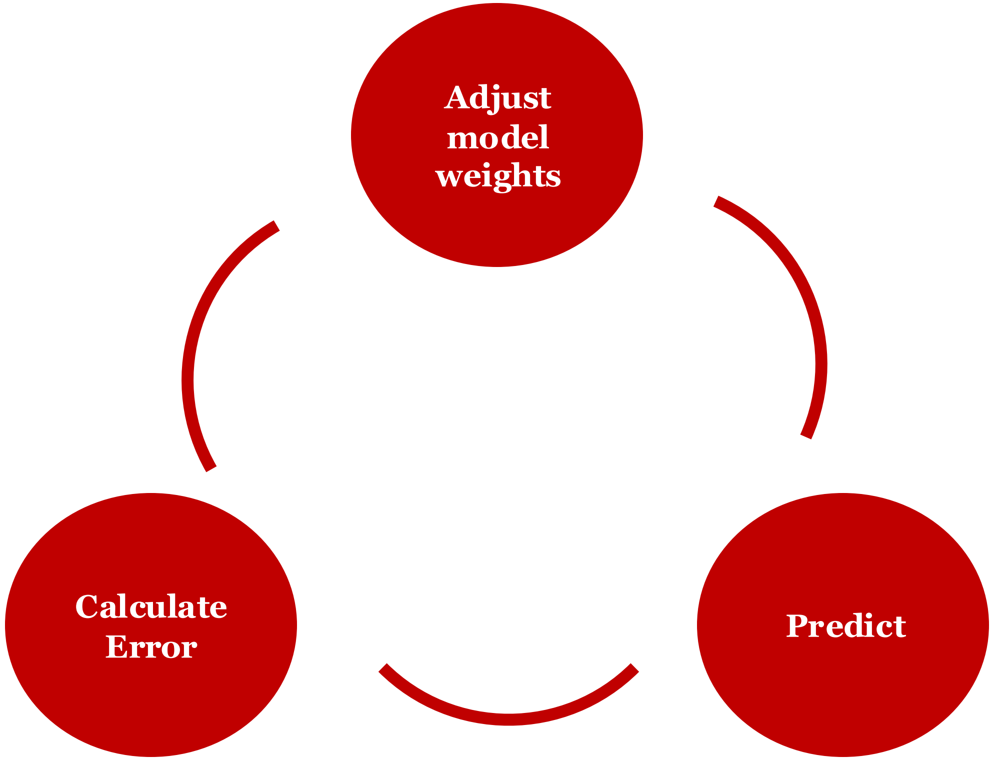
Solving a Supervised Learning Problem

1. Define & Frame:
    1. Understand objective & frame problem as ML task
2. Data Preparation
    1. Collect data
    2. Data Engineering
    3. Feature Engineering
    4. Data labeling
3. Model Development
    1. Model Selection
    2. Model Training
    3. Model Evaluation
4. Deployment & Serving Model
5. Maintenance & Monitoring



Training a Supervised Model

1. Initialize weights/parameters at random
2. Make predictions with that model
3. Calculate the error between model predictions & real answers
    - Optimize Objective function
4. Edit weights/parameters to reduce error
5. Repeat steps 2-4 until error no longer goes down



Demo

- Software vs Machine Learning System

Demo

- Watching a Model Train

Demo

- Tensorflow Playground

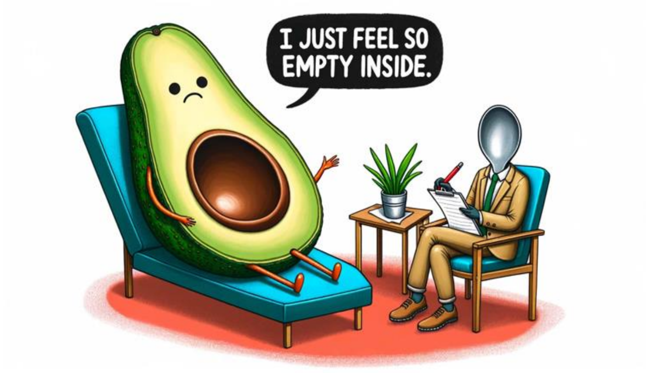
Thank you

ML103: Introduction to GenAI

Rola Dali
ML Architect
February 2025

What is GenAI?

- GenerativeAI is subset of Machine Learning that generates new content
- Emerged into public eye in Nov 2022 when OpenAI launched ChatGPT for free



What is a Large Language Model?

- An LLM is, as the name suggests, a Language Model Model
  - LLMs are a sophisticated type of Neural Networks
  - LLMs are characterized by their large number of parameters, often in the billions
  - LLMs understand a probability distribution of words in a sequence
  - Primary LLM goal is to predict the next word based on previous words
  - They capture word syntax (the arrangement of words) and semantics (the meaning of words)
  - LLMs have proven useful for: translations, natural language generation, part-of-speech tagging, parsing, information retrieval, …
  - LLMs use self-supervised learning for pre-training, removing the need for explicit labeling

Example Tasks LLMs can accomplish

LLMs can Read:

- Proofreading
- Summarizing large texts
- Analyzing text content
- Classification of text
- …

LLMs can write:

- Stylistic polishing
- Answering questions
- First draft generation
- Translating text
- Code writing
- Creating templates
- …

Because LLMs can read & write, they can converse:

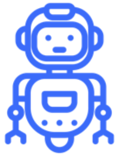
- Chatbots
- Natural Language Interfaces
- …

- **Common use cases of LLMs:**
  - Essay writing
  - Summarization
  - Translation
  - Information retrieval
  - conversational assistanta
  - image generators
  - automated coding tools

Possible LLM Issues

- Hallucination
- Knowledge cut offs
- Context windows: input and output lengths are limited
- Bias and Toxicity
- Still not great with structured/tabular data
- Still not great with number processing

ML Glossary Summary



Artificial Intelligence (AI): Techniques that enable computers to mimic human behavior

Machine Learning (ML): AI techniques that allow computers to learn without explicit programming = mimics “learning”

Generative AI: A type of AI that allows computers to generate new content

LLMs: Large Language Models: umbrella term for models specialized in language

Transformers: Algorithm/neural network that revolutionized GenAI and underlies LLMs

Prompt: Input to the model

Token: a word or a part of a word. Currency of LLMs. Unit of measuring input/output size

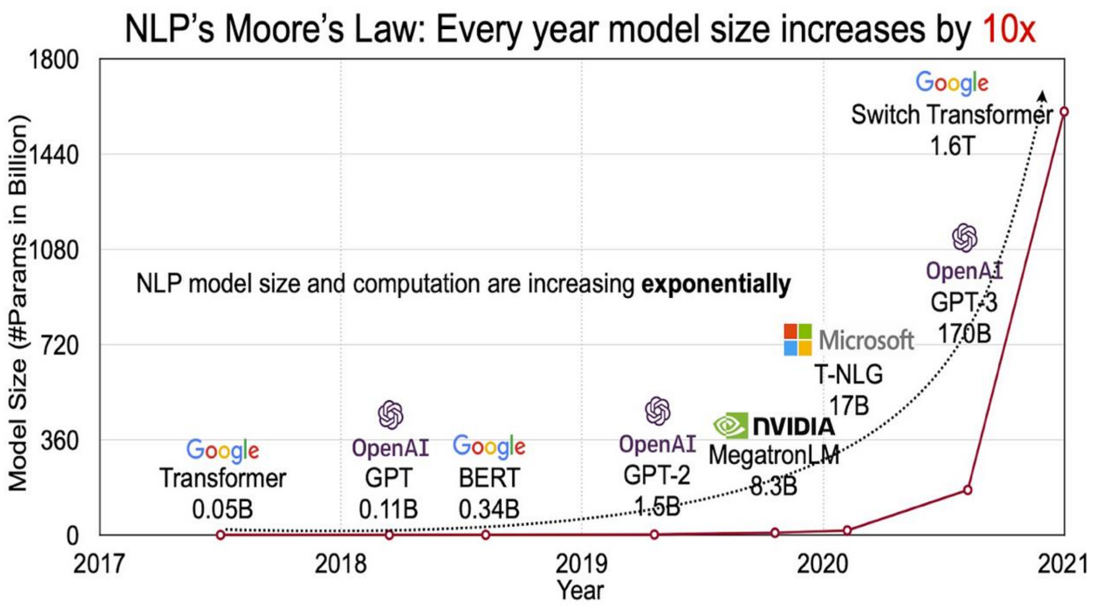
Embedding: numerical representation of non-numeric entities => projection into mathematical space

RAG: Retrieval Augmented Generation: using an external knowledge base to augment the system

Foundational model: ML model trained on vast datasets so it can be applied across a wide range of use cases

GenAI Size Ballparks

- ML models are often sized by “number of parameters” = model weights
- Size ranges from 1 param (y = ax) to ~2T param (GPT 4)
- Predictive ML ~ million params
- GenAI ~ billion-trillion params



Sizing Ballpark:

1 parameter @ 32 bit float = 4 bytes
1 billion parameters ~ 4 GB of RAM JUST FOR PARAMS

BUT you need ~ 20X more space (optimizer, gradients, activation, …) to train

1 billion param model ~ 80 GB of RAM (limit of the Nvidia A100 GPU)
⇒ Imagine the requirements for a 1.8T param model?!

⇒ These models have put constraints on compute/data and made parallelization and optimizations (hardware & software) a must
⇒ The sheer size and compute demands limit training to organizations with significant resources => “Foundational Models”

Reference: LLMs & generative AI project lifecycle: Coursera

Training GenAI resource consumption estimates

Example of current SOTA Llama3 400B

- Params: 405B
- Data: 15.6T tokens
- Token/param ratio: 40 tokens/param => train compute optimal
- FLOPS: 6NP = 6*dataset* model size = 6 * 15.6 *10^12 * 4.5 *10^9 = 3.8 e^25
- Compute: 16K H100 Nvidia GPUs with average throughput of 400 TFLOPS
- Time: ~ 70 days (paper says ~ 30M GPU hours)
- Cost: rented computer + salary = ~ $65-85M
- Carbon emitted: ~ 440 tCO2eq ~ 2000 return tickets to JFK-LHR
- Notes:
  - Next Model? 10X more FLOPs
  - Complexity grows Quadratically with the length of the sequence

Predictive ML vs GenAI:

	Predictive ML	GenAI
Algo Size (params)	< Millions	Billions-Trillions
Data Demands	+	+++
Training Compute	Laptop/reasonable machines	Super computers. Parallelization is critical
Training	Often customized with data	Pre-trained by big providers
Use cases	Specific tasks	General tasks
Cost	$	$$$
Interactions	Custom	API calls
Difficulty	Data, ML algorithms, MLOps	Model selection, prompt engineering, Evaluation
AWS tools	Amazon SageMaker	AWS Bedrock

Choosing GenAI vs predictive ML

- Main differences between GenAI and predictive ML
  - Model training
  - Model Size & required resources
  - Open Ended output

- Use GenAI Foundational models for:
  - General tasks
  - Quick turnaround
  - Do not have expertise
  - Do not have data

- Use Predictive ML for:
  - Specific tasks
  - Tasks that require great accuracy
  - Long term projects
  - Large load projects
  - High Compliance projects

Training data

- Collecting data is hard work
- Heavily guarded by companies
  1. Download all of the internet
     - use web crawlers to scrape pages
     - currently there are 250 billion pages online => 1 PB of data
  2. Text extraction from HTML
  3. Filter undesired content
  4. Deduplicate
  5. Heuristic filtering: remove low quality document
  6. Model based filtering
  7. Mix Data: Classify data categories
  8. Reweight domains using scaling laws to get high downstream performance

Open-Source academic datasets:

- C4 (150B tokens | 800 GB): collection of English-language text sourced from the public Common Crawl web scrape
- The Pile (280B tokens | 825 GiB): open source language modelling data set that consists of 22 smaller, high-quality datasets combined together
- Dolma (3T tokens): an open dataset of 3 trillion tokens from a diverse mix of web content, academic publications, code, books, and encyclopedic materials.
- FineWeb (15T tokens)

- The more params the model has, the more data it needs to see
- Models are training on a dataset on the order of 15T tokens
  - "20K years to read worth of data": Yann Lecun

Reference: Yann Dubois. Stanford lecture

How to Choose an LLM?

Input Output Modalities:

- Models have predefined inputs and outputs.

Performance on required task:

- LLMs are many different capabilities. Look at evaluation metrics to evaluate performance on different tasks: reasoning, coding, summarizing, …

You will need to read the model card to know which models will work for you

Model Size:

- 1B parameters: good with pattern matching & basic knowledge of the world
- 10B parameters: greater world knowledge. Can follow basic instructions
- 100B+ params: Have rich world knowledge & complex reasoning

Closed sources models/SaaS options:

- Easy to use in an application
- More powerful models
- Some risk of vendor lock in

Open source models:

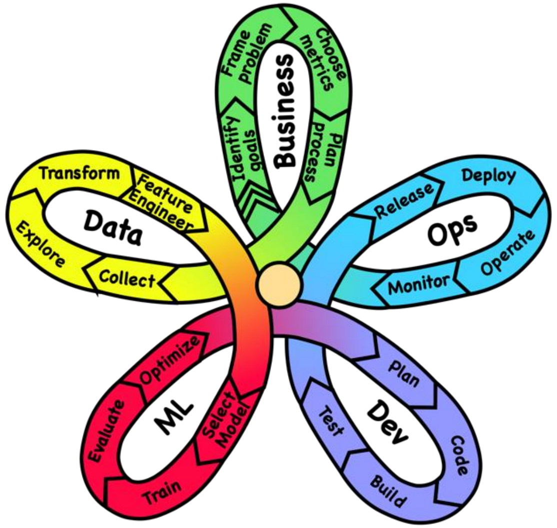
- Full control over model
- Can run your on own device
- Full control over data privacy/access

Predictive ML vs GenAI
LifeCycle

GenAI project LifeCycle. Source: Ping. The ML Solutions Architect Handbook 2024

Predictive ML life Cycle:

- Frame Business Problem
- Source & Prepare Data
- Choose Model Class
- Train Model
- Test Model
- Deploy Model
- Maintain & Monitor



- GenAI life Cycle:
  - Scope project
  - Build system
    - Choose Foundational Model
    - Prompt Engineering
    - Model adaptation & customization
  - Evaluate Performance
  - Deploy Application
  - Monitor Performance

Whereas in Predictive ML, much of the work is about customizing the model to excel at a specific task, GenAI is more about extracting what you want from a general purpose model

=> Asking for a needle in a hay stack

Prompt Engineering
Anatomy of a prompt

- By using Foundational models, the task shifts from data/model to prompting in order to ”extract” what we need from the model

- Prompt: the input to the model and can vary in structure & content
- prompt engineering: editing the input text to drive the desired output from the model

Prompt Engineering Best Practices:

- Give clear/specific instructions
- Structure prompts
- Include examples
- Add contextual information
- Use system instructions
- Instruct the model to explain its reasoning (Chain of thought)
- Break down complex tasks
- Prompt iteration strategies

Prompt

- Query: what is the task?
- Instructions: steps to perform
- Objective: mission/goal to achieve
- Persona: role/view
- Constraints: restrictions to respect
- Examples: demo of output
- Context: relevant information
- Tone: style to use

Enhancing LLM Performance

- Weight preserving techniques
  - Model does not change
  - Focusing on model interactions
  - Examples: prompt engineering, RAG, …
- Weight altering techniques:
  - Model itself changes
  - Training model on your own data
    - Pre-training
    - fine tuning

Retrieval Augmented Generation (RAG)

- A method created by the FAIR team at Meta to enhance the accuracy of LLMs
- Improves LLMs by adding an information retrieval step before generating an answer

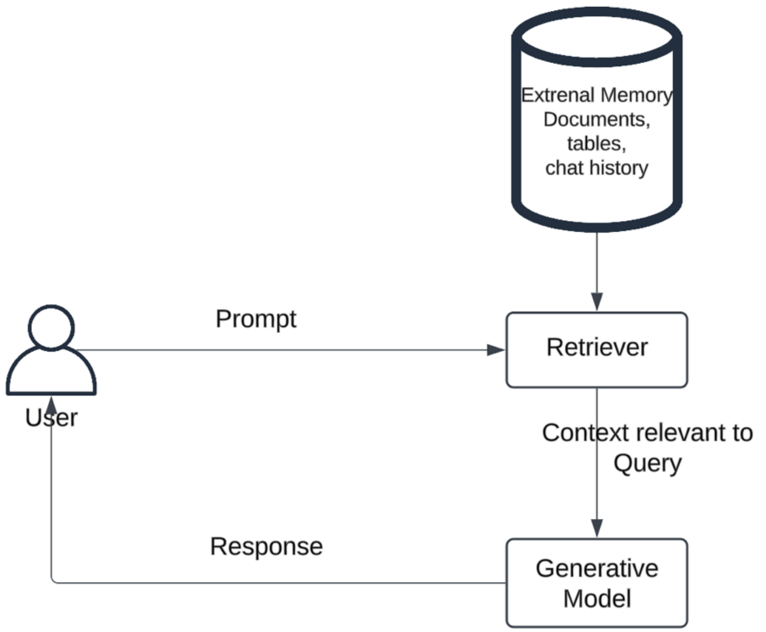
Benefits of RAG:

- Cite sources
- Up to date information
- Domain specific knowledge
- Reduces hallucinations
- Improves LLM performance without training

RAG tradeoffs:

- Increased latency
- Increased cost

A basic RAG architecture. Source: Huyen. AI Engineering 2025

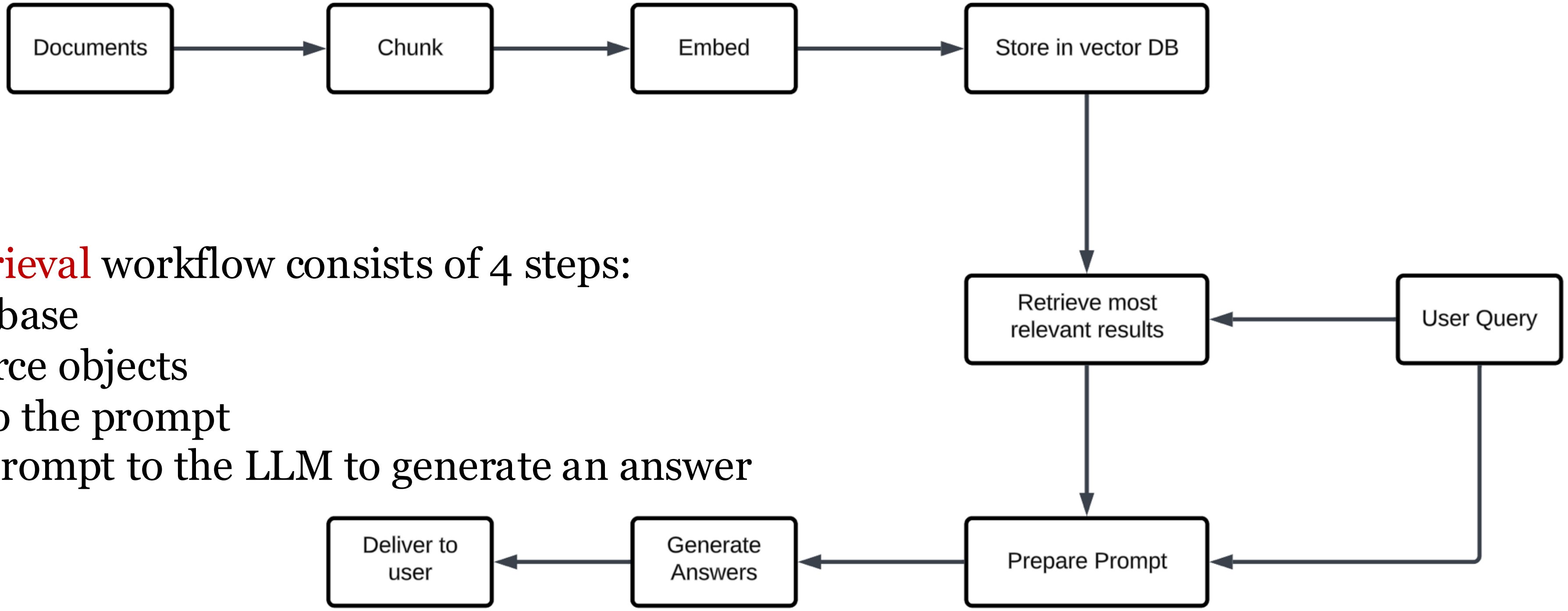


Retrieval Augmented Generation (RAG)

To set up a RAG knowledge base:

- Chunk
- Embed/vectorize
- Store

Typical LLM pipeline. Source: Bouchard. Building LLMs for Production



At its simplest the RAG Retrieval workflow consists of 4 steps:

1. Query a vector Database
2. Obtain relevant source objects
3. Stuff the objects into the prompt
4. Send the modified prompt to the LLM to generate an answer

GenAI under the hood

Concepts in LLM processing

- Tokenization
  - the initial phase of interacting with LLMs. It involved breaking the input text into smaller pieces known as tokens
  - Tokenization is model dependent. The models are released as a pair of pre-trained tokenizer and model weights
- Embeddings
  - Involves transposing words into mathematical space
- The Transformer
  - The model at the heart of the LLM

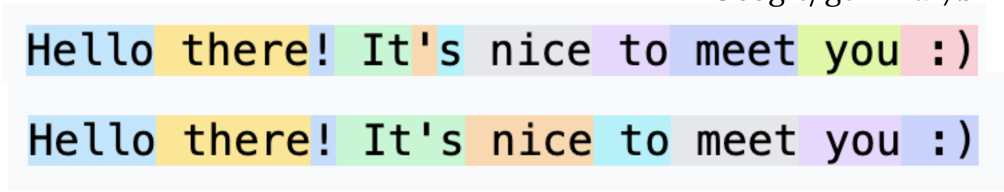
Tokenization

Tokenization is the process of breaking down text into smaller units called tokens

Tokenization Benefits:

- Cost Efficiency: Transformer performance is quadratic in terms of input token
- Reduces the space complexity of word

Google/gemma-7b



GPT-4o

Tokenization Side effects:

- LLMs can't spell words
- LLMs cannot do simple string processing tasks like reversing a string
- LLMs are worse at non-English languages
- LLMs are bad at simple arithmetic
- LLMs can have trouble coding in Python

Tokenizer visualization: https://tiktokenizer.vercel.app/

Word Embeddings/Vectorization

- Machine learning models are just large statistical calculators. They work with numbers, not words.
  - We need to convert words into numbers
  - Vector embeddings: is the projection of text, images… into mathematical space
  - ==> numerical representation of non-numeric entities like words
  - examples: one hot encoding, TF-IDF, Word2Vec, …

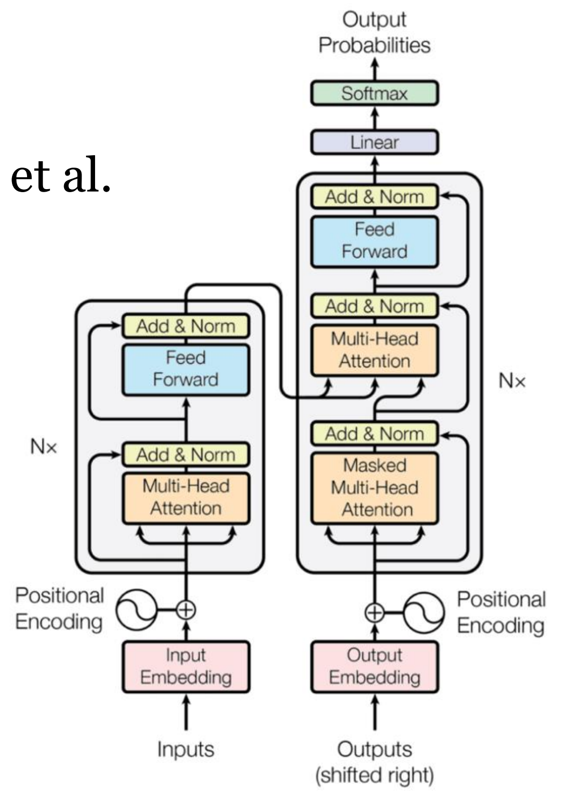
Historically:

- Bag of words: 1954: simple approach to document classification by counting word occurrence
- TF-IDF: 1972: word count based on rarity or frequency
- Word2Vec: 2013: word embeddings
  - high dimensional vectors encapsulating semantic associations
  - This was a substantial advancement in capturing textual semantics
  - => Can do math with words: (king - man) + woman = queen => transformed NLP and modeling on text

Word2vec paper: https://arxiv.org/pdf/1301.3781

The Transformer

- The heart of many LLMs
- A Neural Network
- Introduced in the paper “Attention Is All You Need” - 2017. Vaswani, Ashish et al.
  - by researchers from Google and University of Toronto
- Initially designed for sequence-to-sequence tasks like translation
- Copy/Paste the architecture was used across fields



The encoder-decoder structure of the Transformer architecture
Taken from “Attention Is All You Need“

Differentiators of Transformers:

- Parallelizable
- Positional encoding: don't need to retain sequence
- Attention heads

Transformer Explained: https://poloclub.github.io/transformer-explainer/

References

- Yann Dubois. Aug 27, 2024. https://www.youtube.com/watch?v=9vM4p9NN0Ts
- Andrej Karpathy. January 10, 2023. https://www.youtube.com/watch?v=XfpMkf4rD6E
- Andrej Karpathy. Jan 17, 2023. https://www.youtube.com/watch?v=kCc8FmEb1nY
- Andrej Karpathy . nanoGPT: https://github.com/karpathy/nanoGPT
- Andrej Karpathy . building GPT tokenizer: https://www.youtube.com/watch?v=zduSFxRajkE
- Andrew Ng. Coursera
- Andrew Ng. DeepLearning.AI

Thank you

ML104: Architecting GenAI systems

Rola Dali
ML Architect
February 2025

Tech Serves the Business

Is ML/AI the right tool for the job?

- Keep in mind that ML, AI, analytics, tech, … are all tools to solve problems & serve the business
- You always want to choose the right tool to solve the use case
- Any repeating task in a system is worth automating:
  - If the task is deterministic: it can be automated with DevOps/rule based systems
  - If the tasks involves cognition/reasoning or patterns we do not understand, it can be automated with ML

Predictive ML vs GenAI
Fine or Coarse tools…

	Metric	Predictive ML	GenAI
Custom	Cost	$	$$$$$
SaaS	availability	++	+++++

Main differences to keep in mind:
- Model Training
- Size
- Open Ended results

Cost:
- Training cost
- Inference cost
- Total Cost of ownership

Predictive ML vs GenAI

Choosing the right tool for the Job

	Custom Predictive ML	GenAI Foundational Model
Model Size	+	+++++
Training Investment	+++++	Done by model owner
Data availability	Required	Not Required
Expertise	Required	Not Required
Initial Investment	+++++	+
Inference Cost	+	+++
Model Control	+++++	-
Deterministic or Not	More Deterministic	Less Deterministic
Project length	Makes sense for long term investments	
Workload Size	Makes sense for large workloads	
High Compliance	+	-

Predictive ML vs GenAI SaaS cost

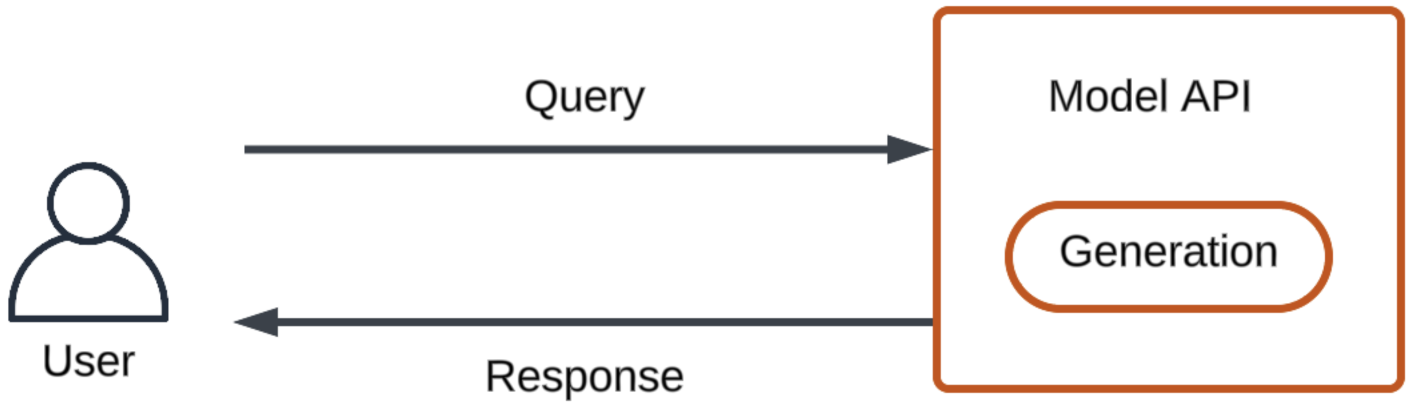
- Document processing: Textract (SaaS ML) vs LLM (SaaS GenAI)
  - 1M pages per month processing: ~676 words per page

Service	Price	Factor
Amazon Textract	1,500 USD	1X
Amazon Bedrock: Nova Pro	2,689 USD	~2X
Amazon Bedrock: Claude sonnet	11,600 USD	~8X

Example GenAI architecture

- What I will present is a series of CONCEPTUAL diagrams that show an abstract form of the main building blocks of a system and how they are connected.
- Many different tools can fit into that block
- The main point of this exercise is to get a high-level view of a system. It is important to understand what function each component serves, the pros and cons of having it and later on, what technical implementation options are available for each component (out of scope for this talk).

GenAI Architecture
Ground 0: A model Call



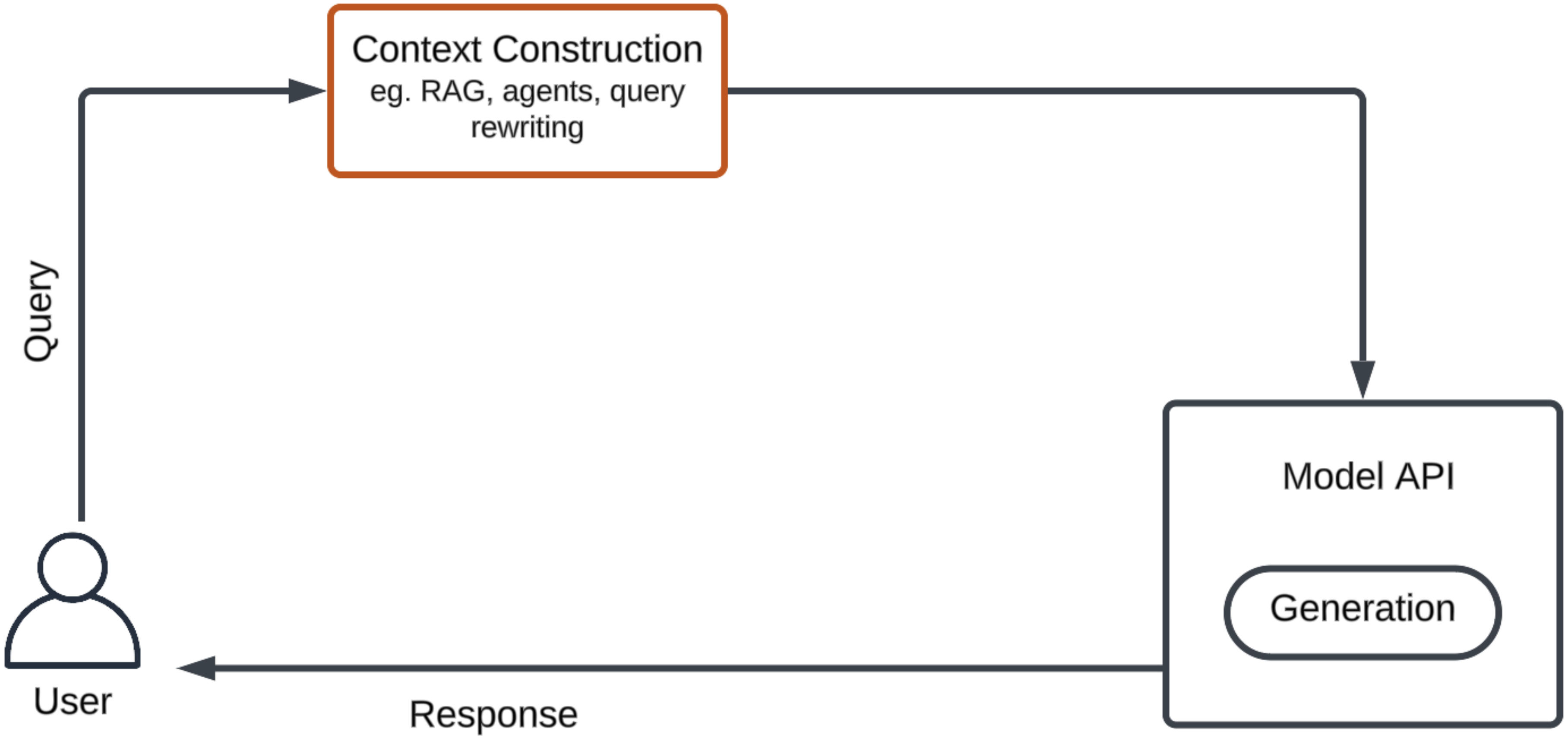
Anatomy of a prompt

- By using Foundational models, the task shifts from data/model to prompting in order to ”extract” what we need from the model
- Prompt: the input to the model and can vary in structure & content
- prompt engineering: editing the input text to drive the desired output from the model

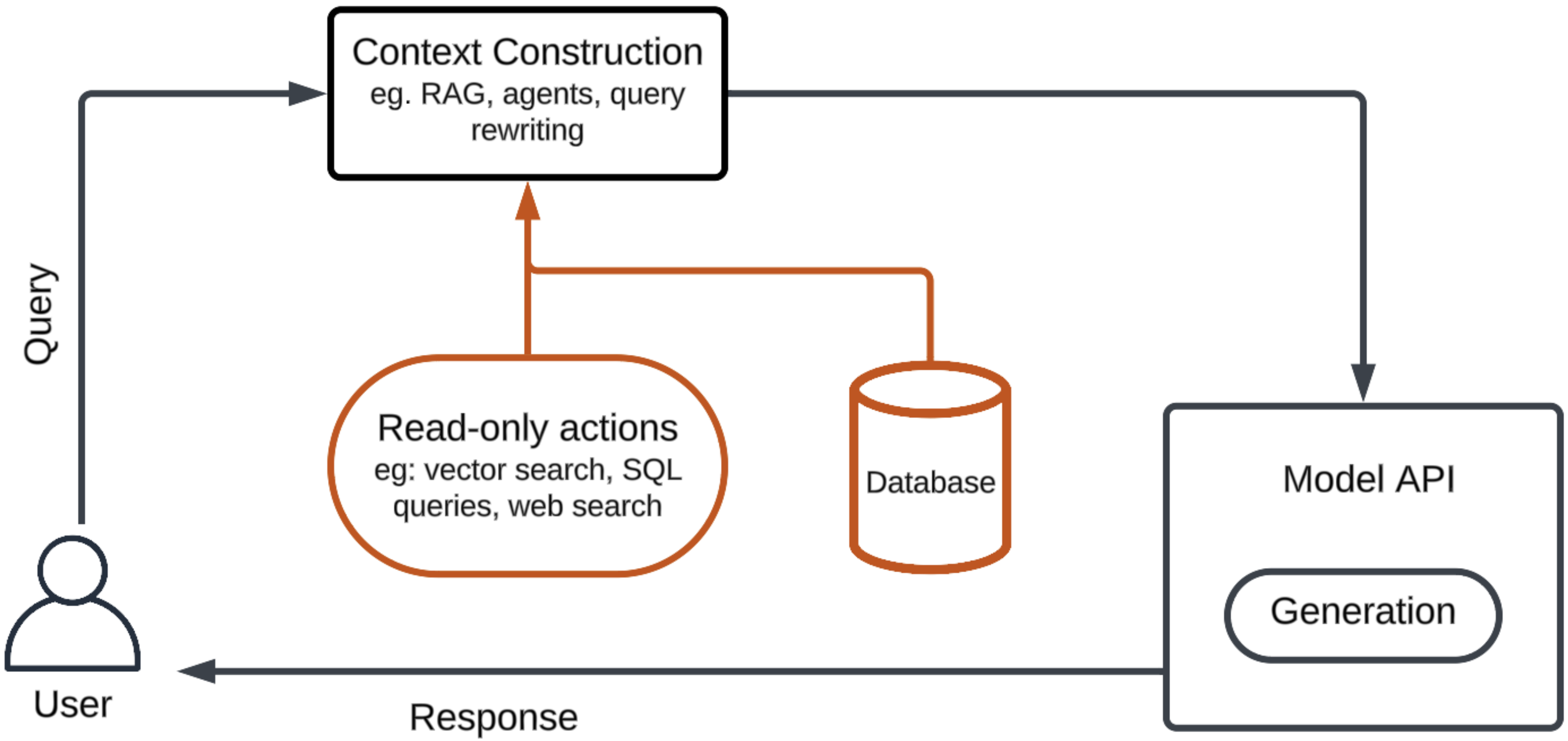
Prompt

- Query: what is the task?
- Instructions: steps to perform
- Objective: mission/goal to achieve
- Persona: role/view
- Constraints: restrictions to respect
- Examples: demo of output
- Context: relevant information
- Tone: style to use

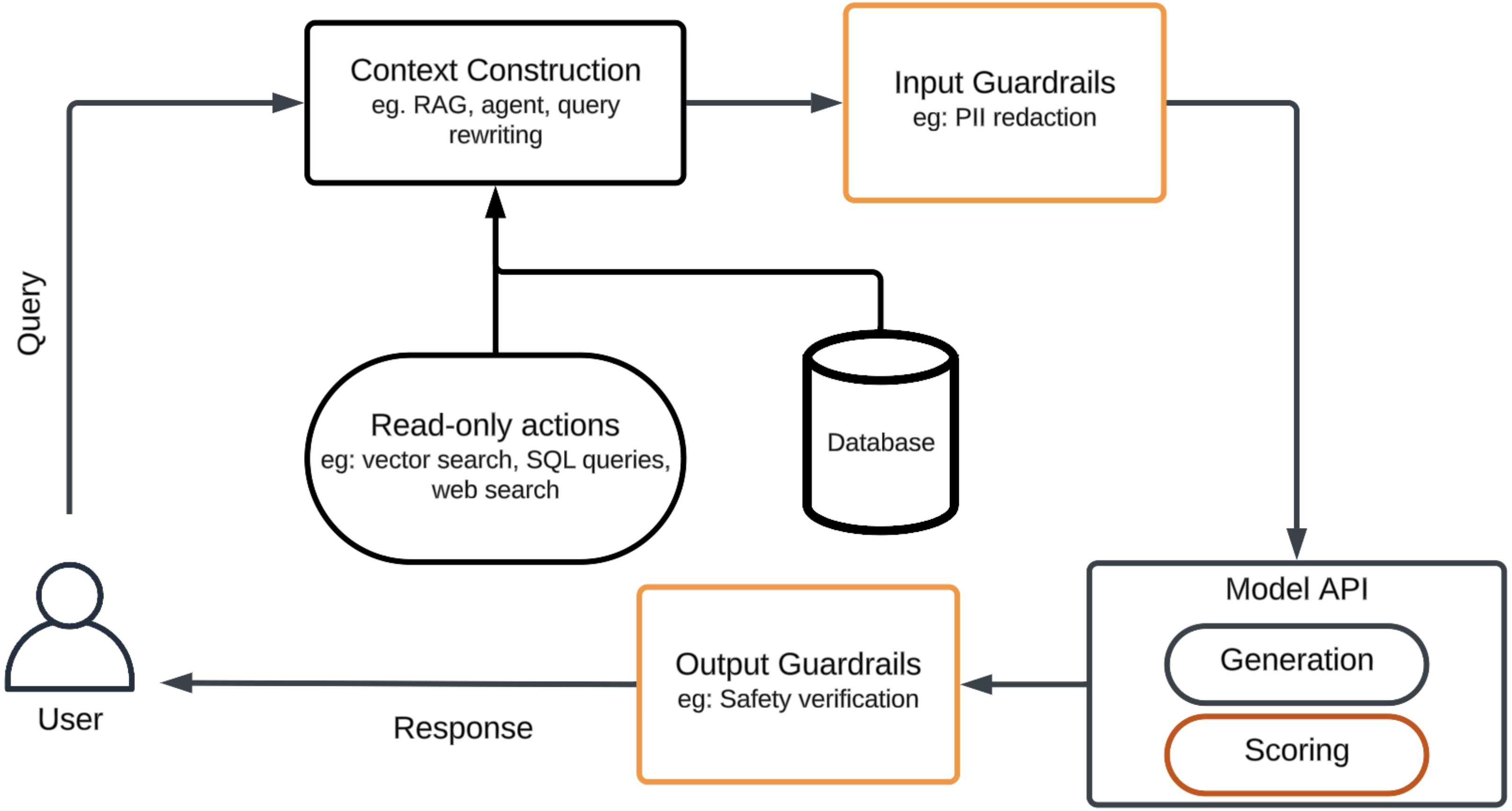
GenAI Architecture
Enhance Context/prompt/model input



GenAI Architecture
Add external knowledge base

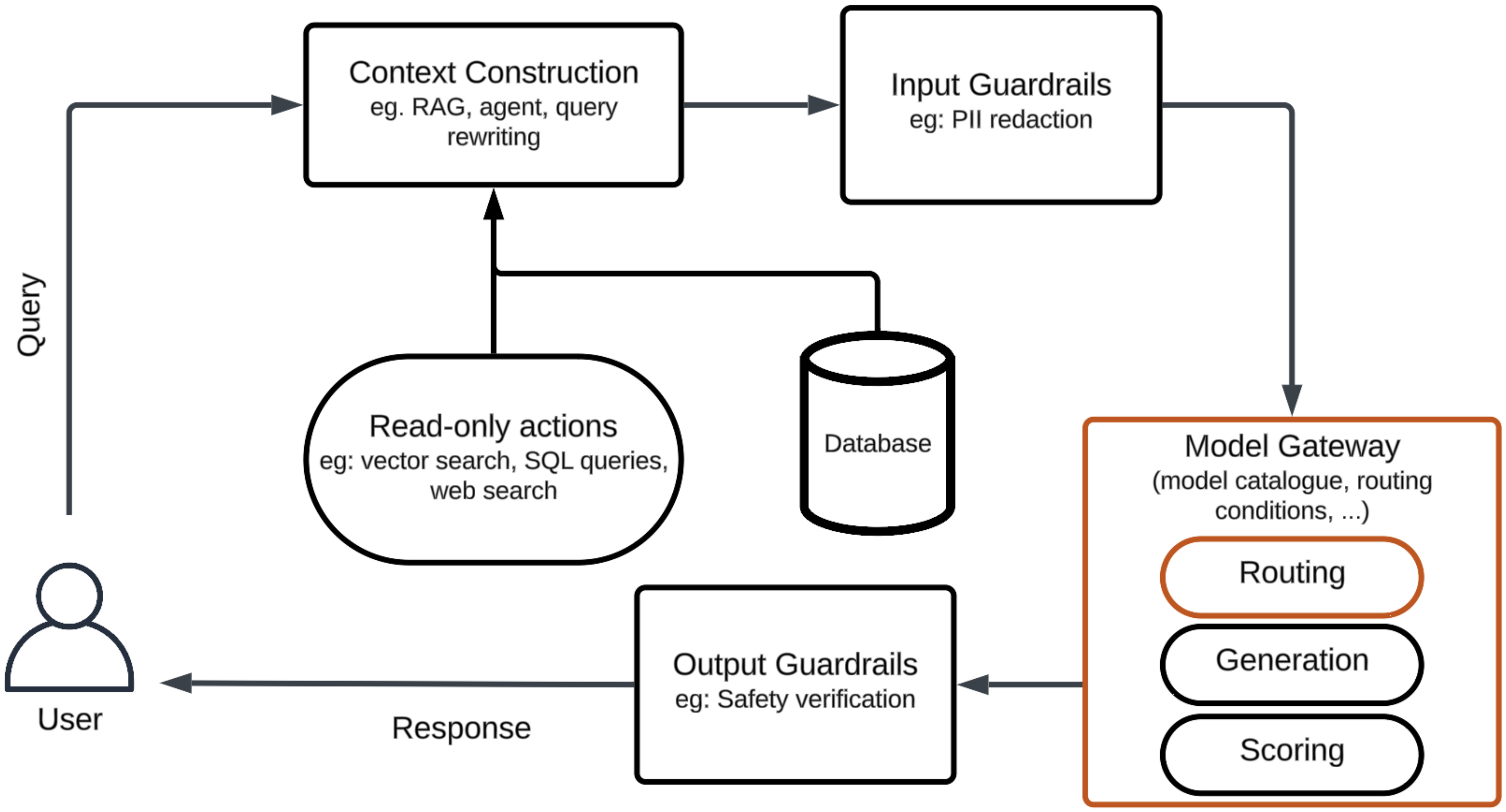


GenAI Architecture
Put Guardrails: input & output controls



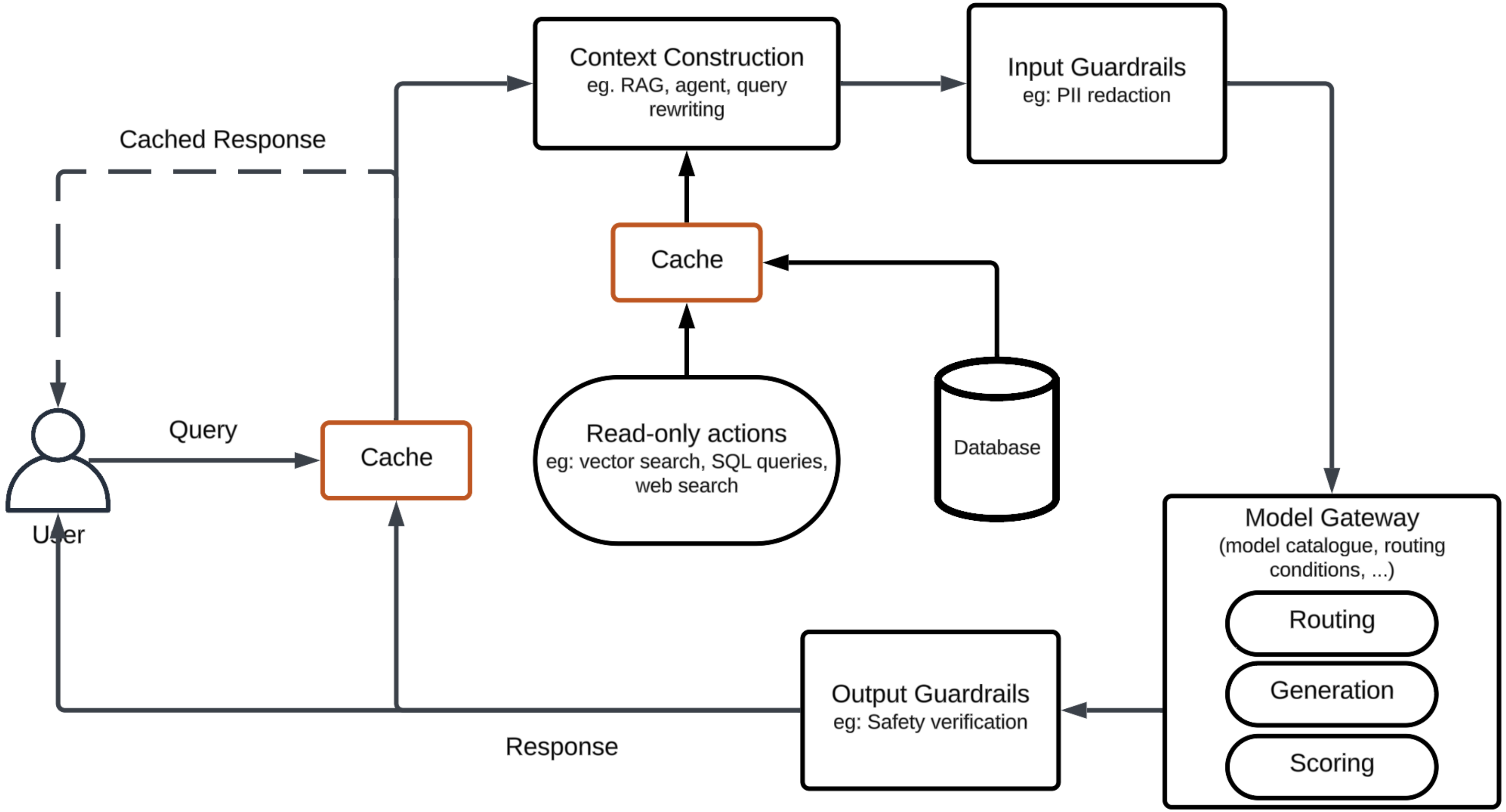
GenAI Architecture

Abstract Model interactions: Add model Router & Gateway

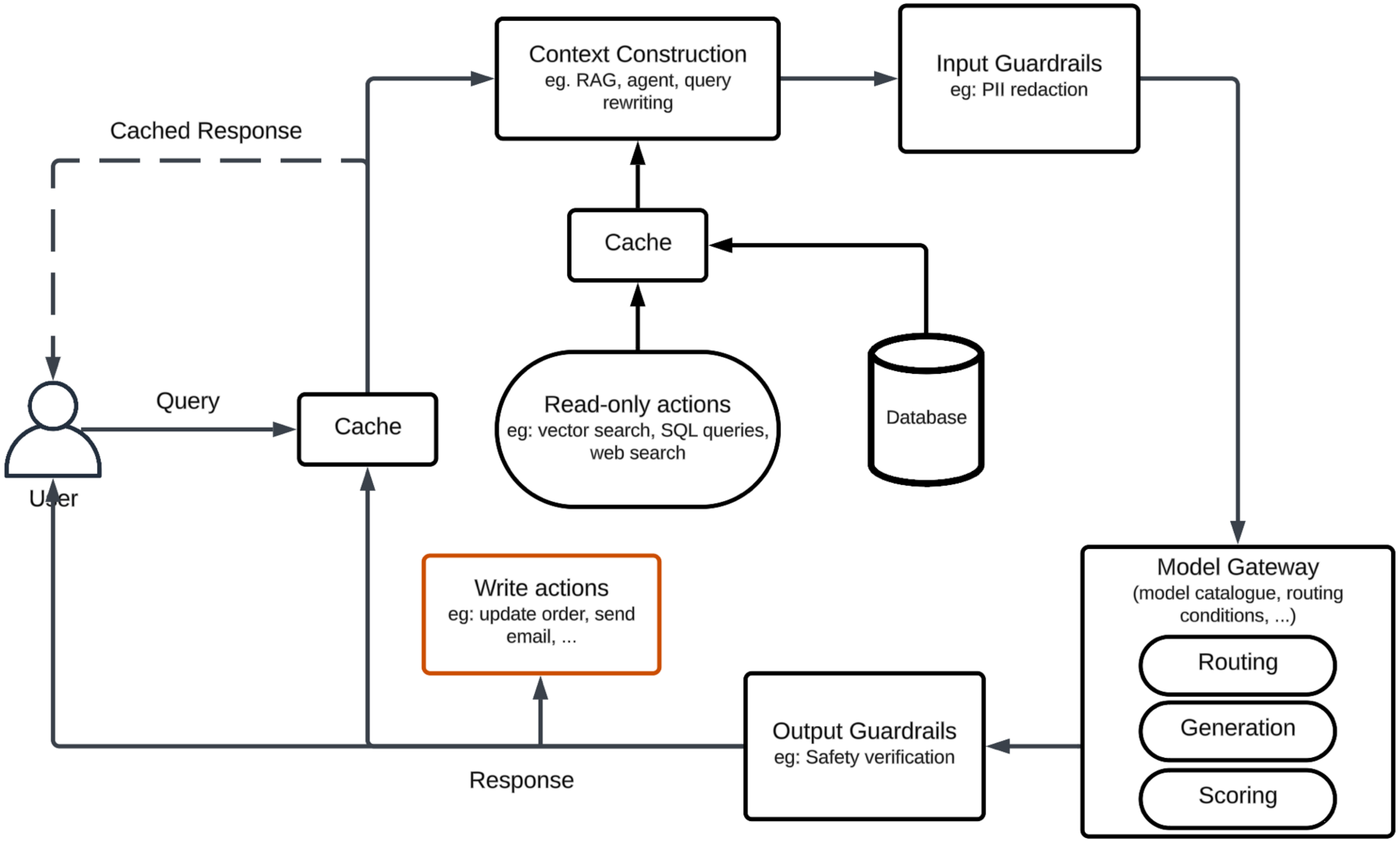


GenAI Architecture

Reduce latency with caches



GenAI Architecture
Add functionality with agents



GenAI Architecture
Other functionalities

- Add authentication & authorization
- Add state and session management
- Add monitoring & observability
- Add pipeline orchestration
- Add Human Feedback
- … … …

Thank you