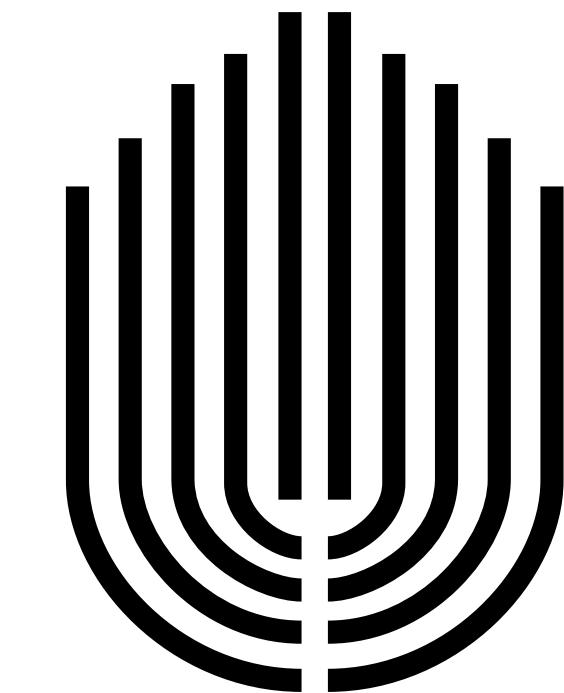


GEN AI FOUNDATION

multisoft
systems



edquest

By Dinesh Chopra

AGENDA

- Introduction to Generative AI
- Generative AI Evolution
- Foundation Models (Text Generation, Image Generation)
- Generative AI Applications
- Fine-Tuning and Customization
- Generative AI Ethics
- Hands-on Project

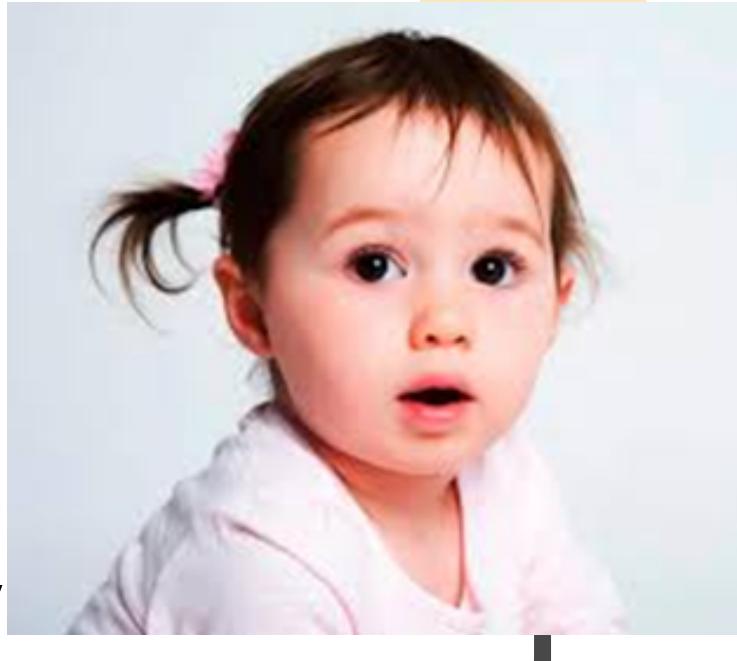
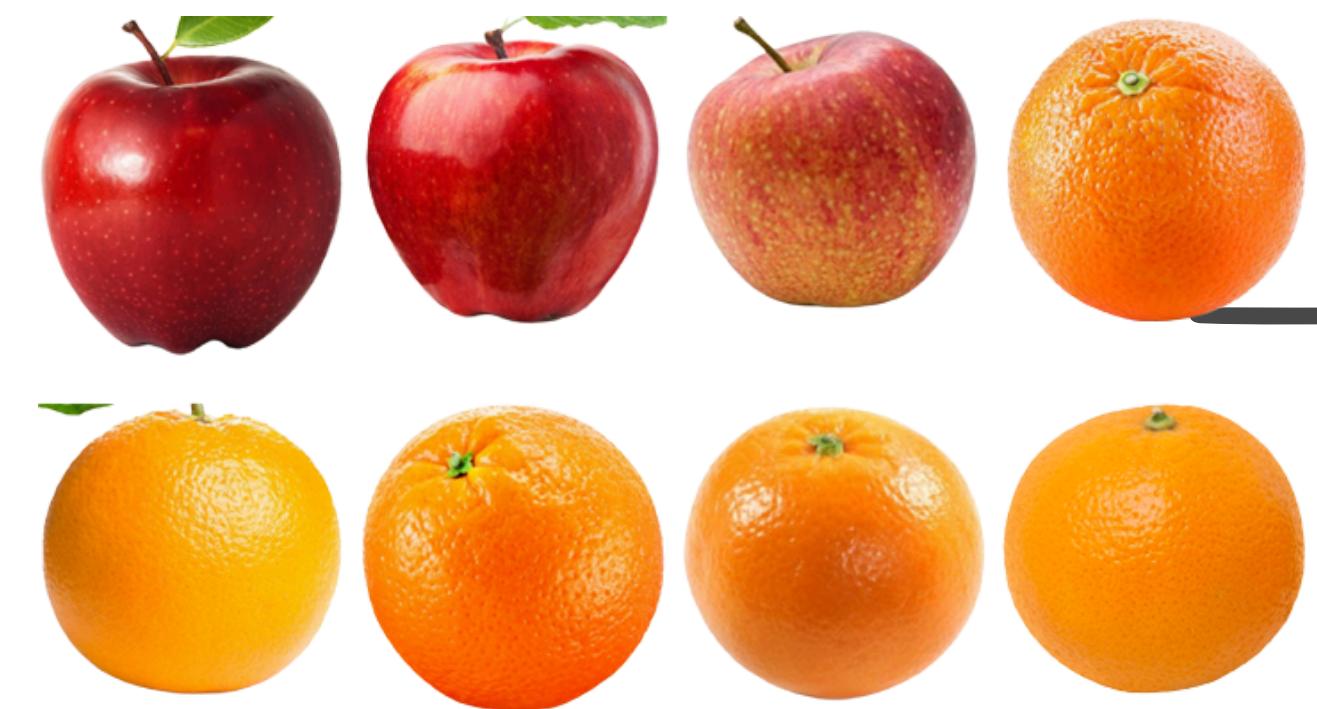
Introduction to Generative AI & Its Evolution

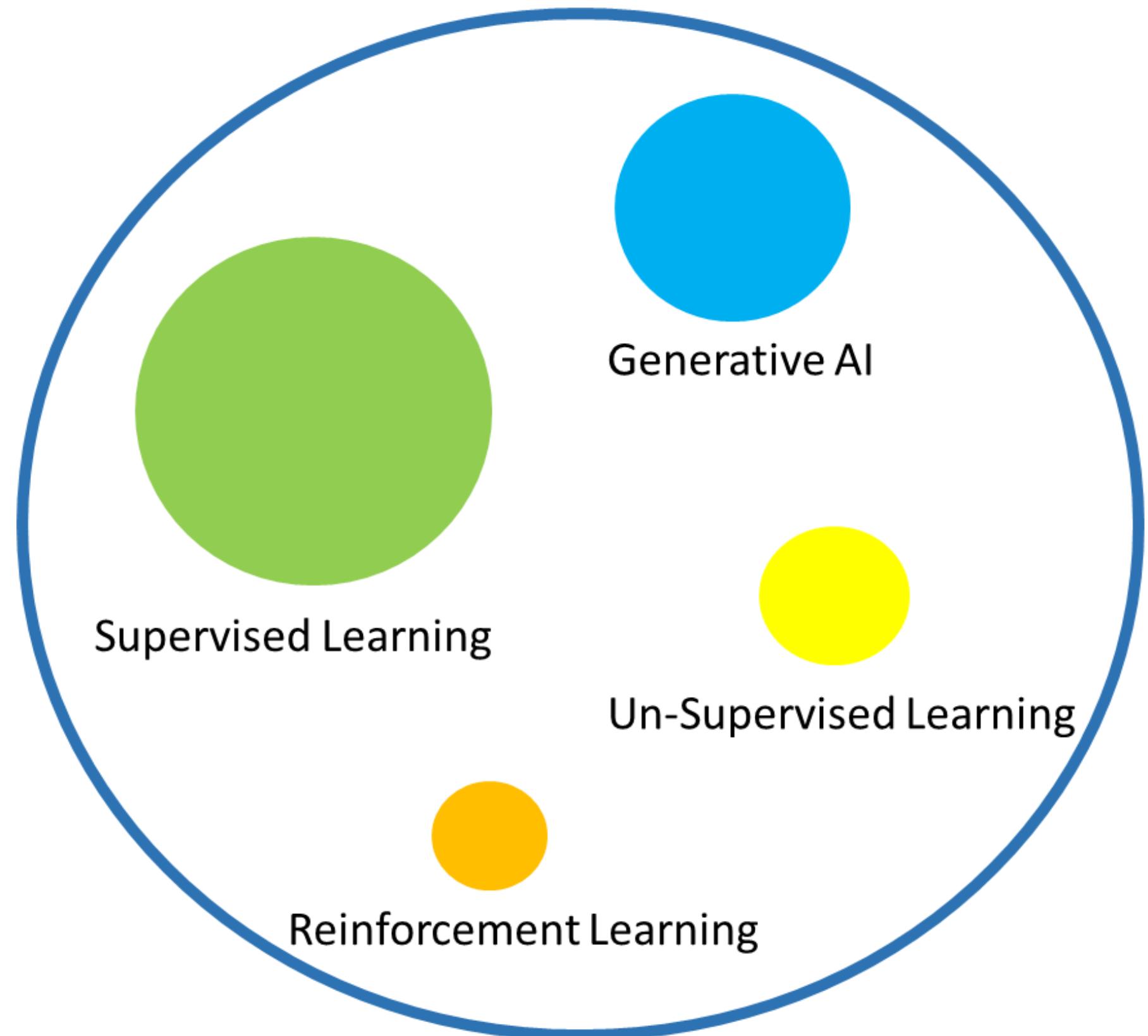


AI/ML ?

Machine Learning (ML) is a subset of Artificial Intelligence that involves the development of algorithms enabling computers to learn from data and improve their performance on a specific task without being explicitly programmed.

- It focuses on the creation of models that can recognize patterns,
- Models can make predictions,
- Models can adapt to new information





HOW TO CONFUSE MACHINE LEARNING



AI, ML & Generative AI

- AI: Simulating human intelligence
- ML: Learning from data to predict
- Generative AI: Creating new content
 - Text
 - Image
 - Video

AI, ML & Generative AI

Modi Taps Bhashini AI Platform to Bridge the Language Divide

Modi used it to enable people from south India to understand his speech

Suraksha P

Bengaluru: Prime Minister Narendra Modi used Bhashini, the Government of India's national public digital platform for languages that develops services and products for citizens by leveraging AI, at two recent public events to enable travellers and students from southern India to understand his speech.

Bhashini counts among its various products and technologies, real time speech to speech machine translation, and automatic speech recognition, which Prime Minister Narendra Modi used on December 19 to interact with students in various parts of India via

video conferencing. During the interaction with the participants of the grand finale of Smart India Hackathon, 2023, students from Tamil Nadu spoke to the PM from Royal Global University, Guwahati, Assam. As the PM greeted Reshma, a student from Nadar Saraswathi College of Engineering & Technology, from Theni in

Tamil Nadu, she and her five teammates could hear a Tamil translation of the PM's message through Bhashini.

"You've reached Guwahati after trav-

elling several kilometers from southern India. You're all ambassador Ek Bharat Shreshtha Bharat (a Government of India initiative that aims to improve the present cultural connection across various states of India). Please tell me about your team and the problem area you are working on," the PM said, which the students heard in Tamil after a couple of seconds. Reshma could immediately talk about her solution: AI-based generative design of hydro power plants.

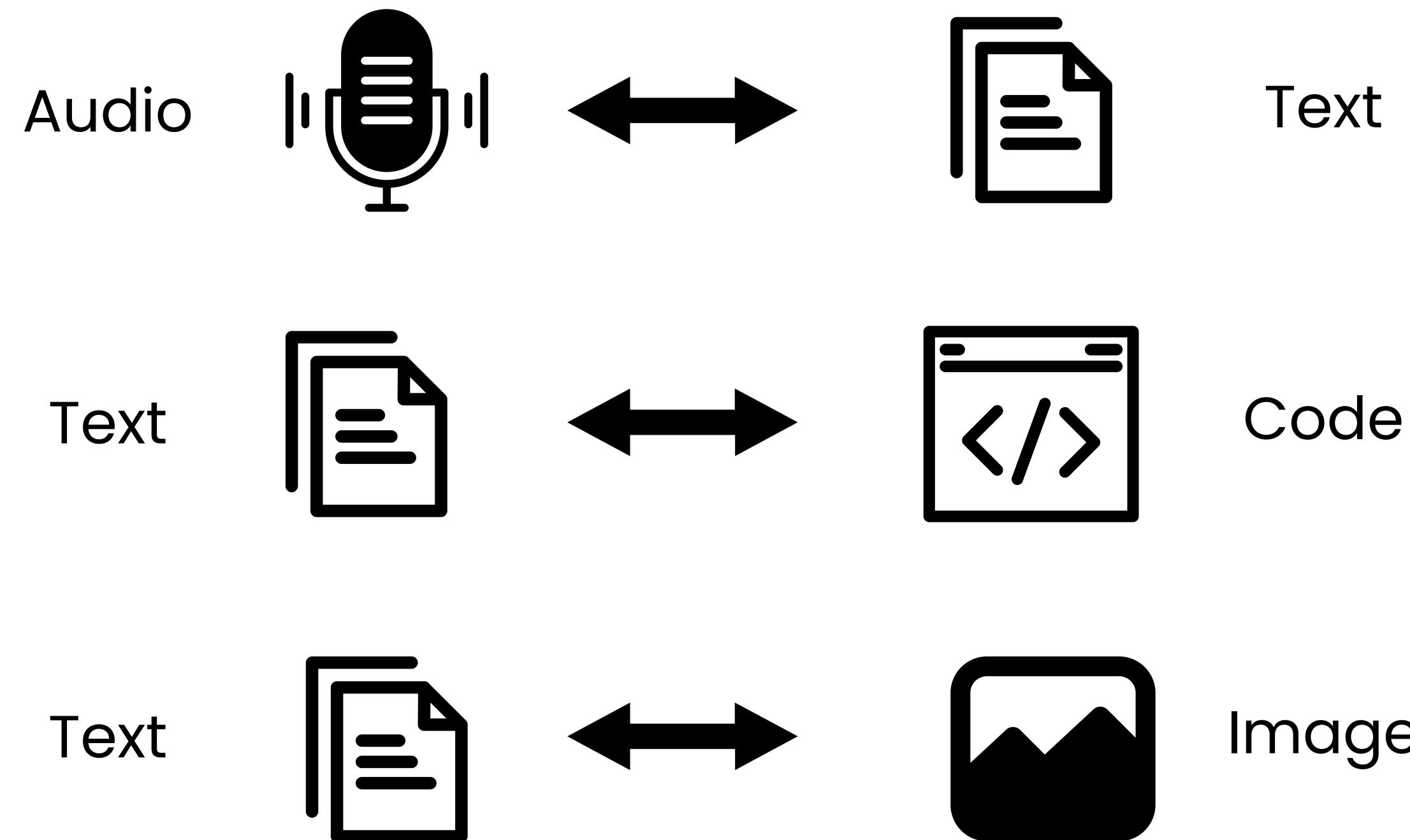
This isn't the first time that Bhashini has been deployed. Recently, in Ranasi, PM Modi urged travel from Tamil Nadu to plug in their headphones to listen to a real time Tamil translation of his speech at Kashi Tamil Sangamam, 2023.

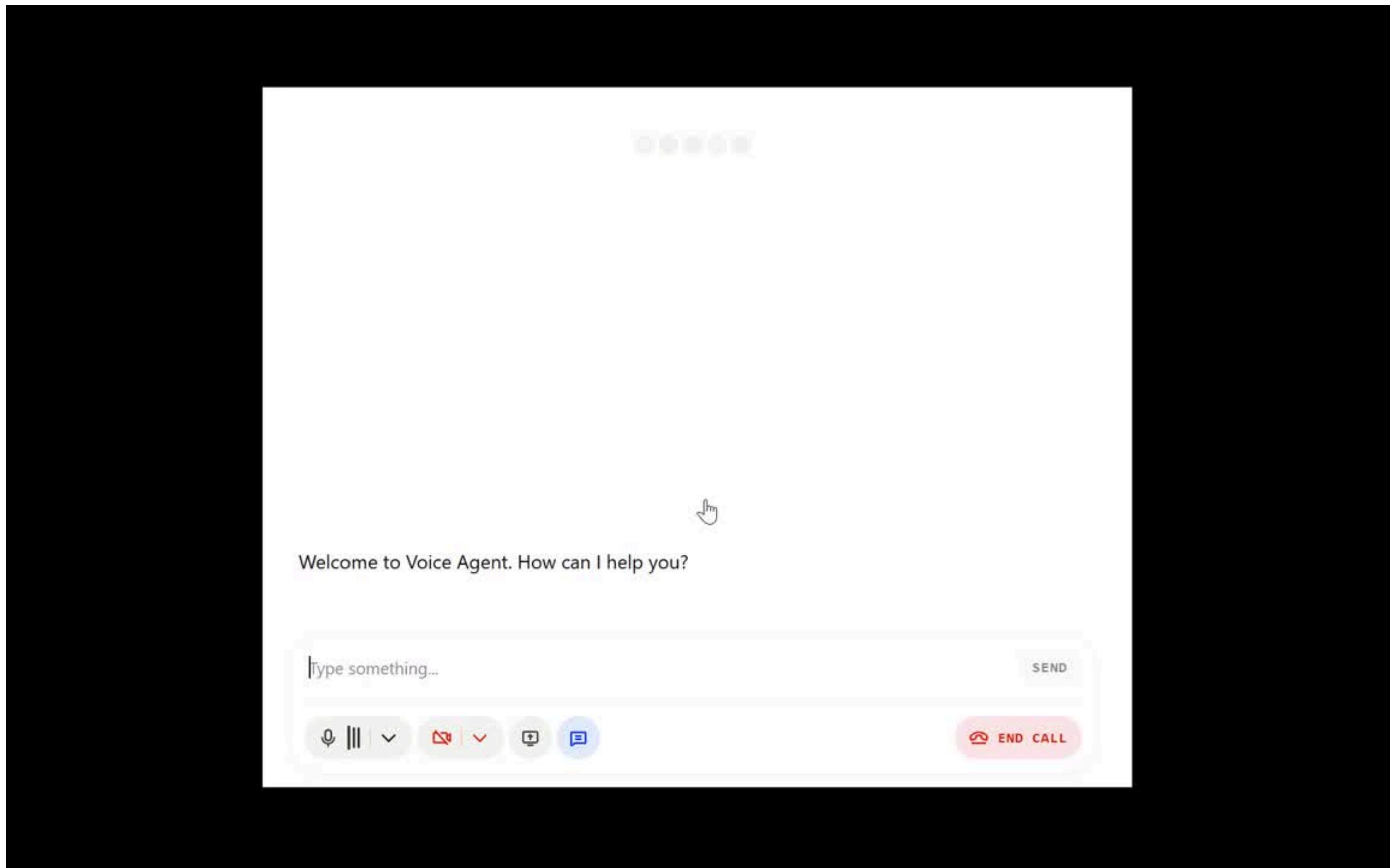


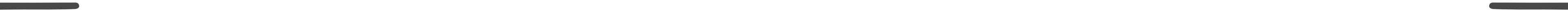
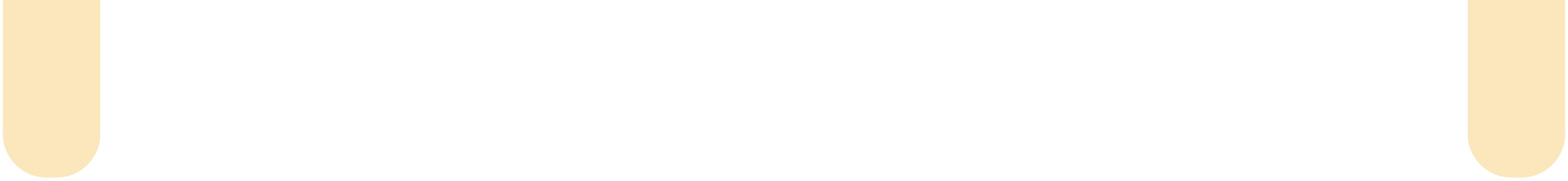
FILE PHOTO

suraksha.p@timesgroup

Generative AI







Software is changing.
(again)

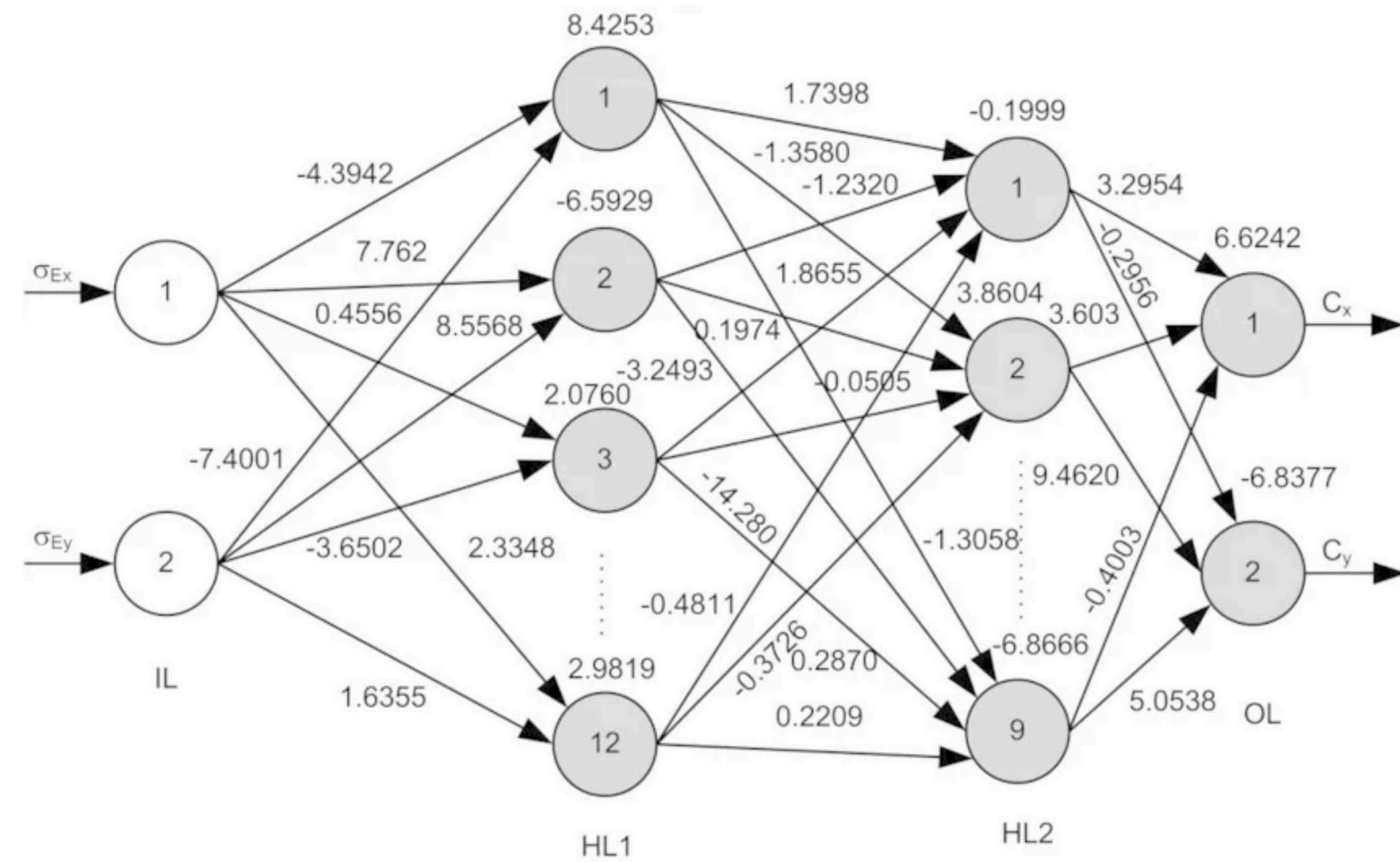
Software 1.0

Code

```
    if$(a[d].word == b) {
        return c;
    }
}
function dynamicSort(a) {
    var b = 1;
    if(b == -1, a = a.substr(1)); a = return(function(c, d) {
        if(c[a] > d[a] ? 1 : 0)*b; sli}; } function occurrence(b) {
        if(0 >= b.length) {
            return 0;
        }
        b += ("");
        if(c == fc = ? 1 : b.length;) {
            for((c = fc = ? 1 : b.length; i < c; i++)) {
                if(a[i].word == b) {
                    a[i].occurrence++;
                }
            }
        }
        return b;
    });
}
```

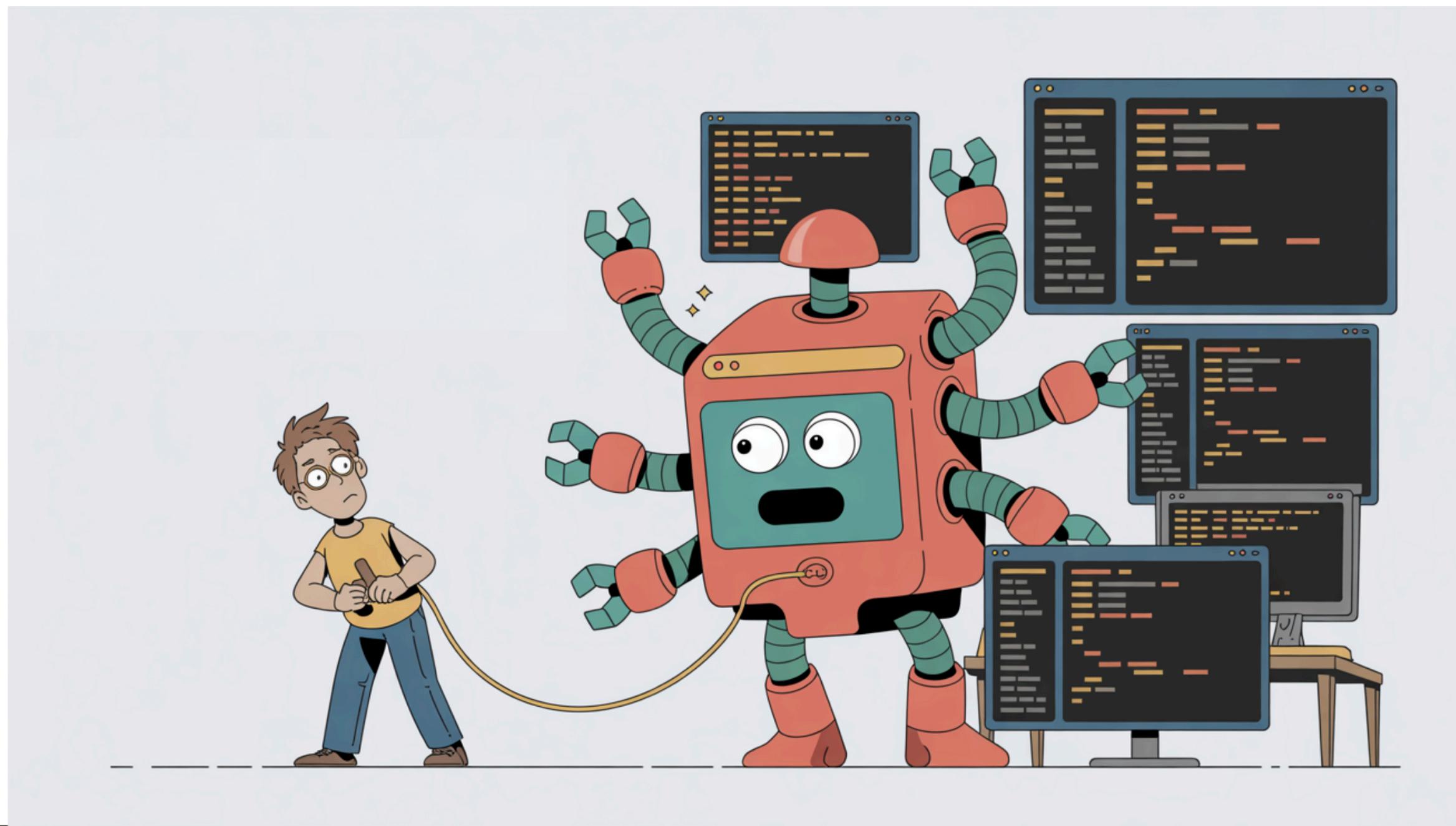
Software 2.0

Neural Networks



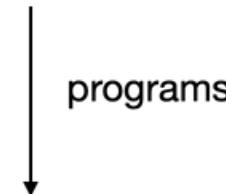
Software 3.0

Generative AI → Prompts



Software 1.0

computer code



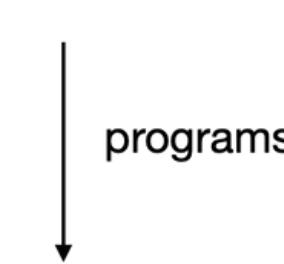
computer



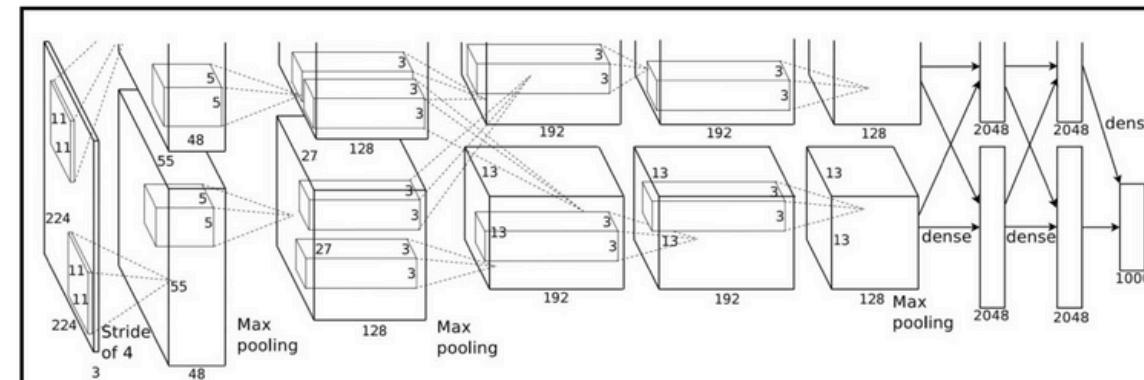
became programmable in ~1940s

Software 2.0

weights



neural net

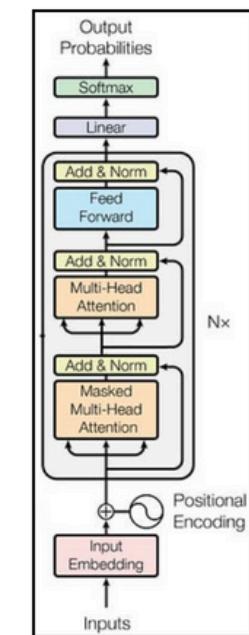


Software 3.0

prompts



LLM



LLM = programmable neural net!

Example: Sentiment Classification

Software 1.0

```
python  
  
def simple_sentiment(review: str) -> str:  
    """Return 'positive' or 'negative' based on a tiny keyword lexicon."""  
    positive = {  
        "good", "great", "excellent", "amazing", "wonderful", "fantastic",  
        "awesome", "loved", "love", "like", "enjoyed", "superb", "delightful"  
    }  
    negative = {  
        "bad", "terrible", "awful", "poor", "boring", "hate", "hated",  
        "dislike", "worst", "dull", "disappointing", "mediocre"  
    }  
  
    score = 0  
    for word in review.lower().split():  
        w = word.strip(",.!?:;")      # crude token clean-up  
        if w in positive:  
            score += 1  
        elif w in negative:  
            score -= 1  
  
    return "positive" if score >= 0 else "negative"
```

Software 2.0

10,000 positive examples
10,000 negative examples
encoding (e.g. bag of words)

train binary classifier

parameters

Software 3.0

You are a sentiment classifier. For every review that appears between the tags

<REVIEW> ... </REVIEW>, respond with **exactly one word**, either POSITIVE or NEGATIVE (all-caps, no punctuation, no extra text).

Example 1

<REVIEW>I absolutely loved this film—the characters were engaging and the ending was perfect.</REVIEW>

POSITIVE

Example 2

<REVIEW>The plot was incoherent and the acting felt forced; I regret watching it.</REVIEW>

NEGATIVE

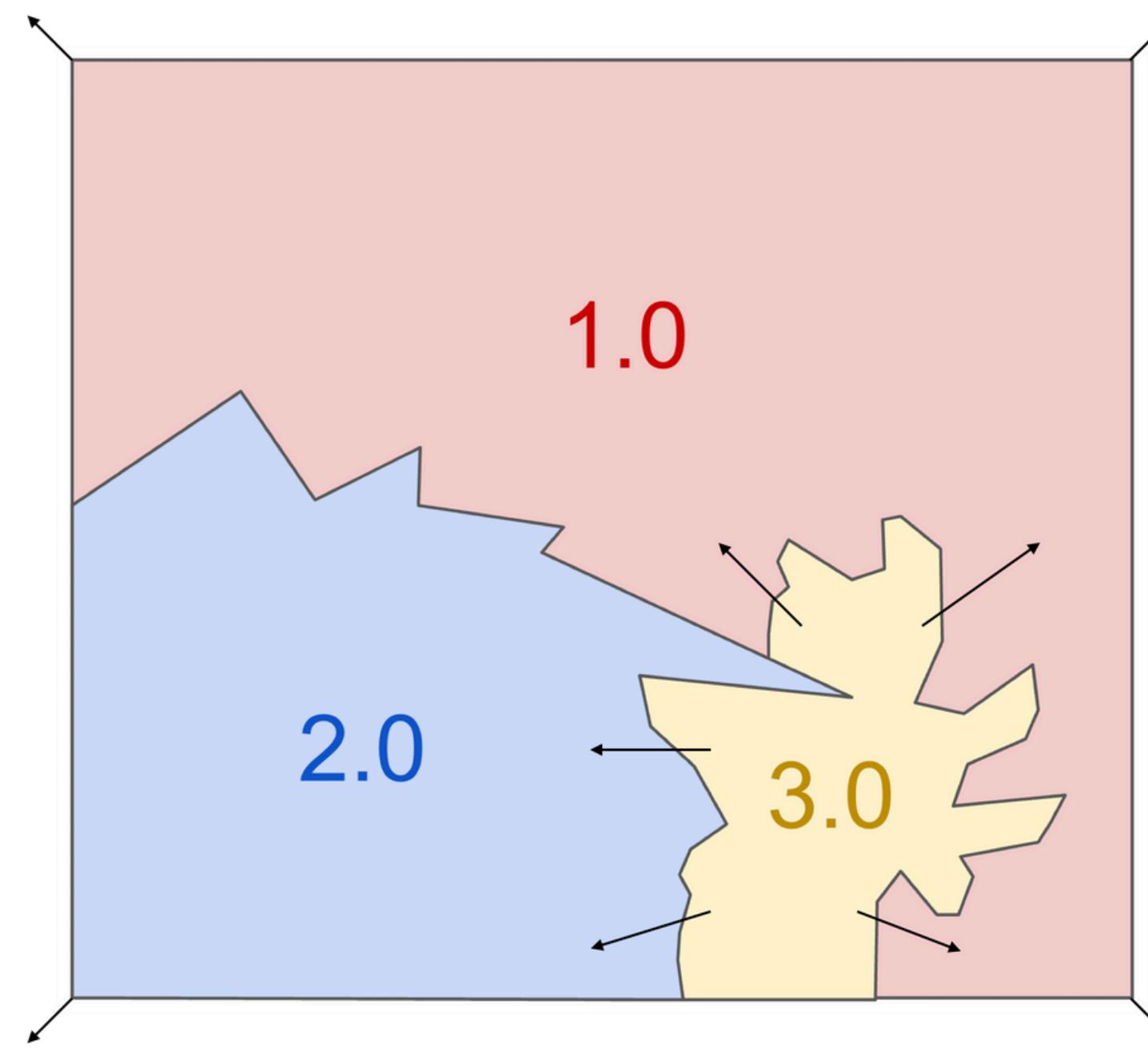
Example 3

<REVIEW>An energetic soundtrack and solid visuals almost save it, but the story drags and the jokes fall flat.</REVIEW>

NEGATIVE

Now classify the next review.

A huge amount of software will be (re-)written.



Industry Use Cases of Gen AI



- **Create Content** - Images, Videos, Music
- **Programming** - Write code, fix bugs
- **Customer Support** - Make Chatbots
- **Research** - Write Reports and Insights
- **Education** - Make notes, take quizzes
- **Marketing** - Create Ad copies & campaigns
- **Content Writing** - Write articles, blogs

FOUNDATION MODELS & TRANSFORMERS



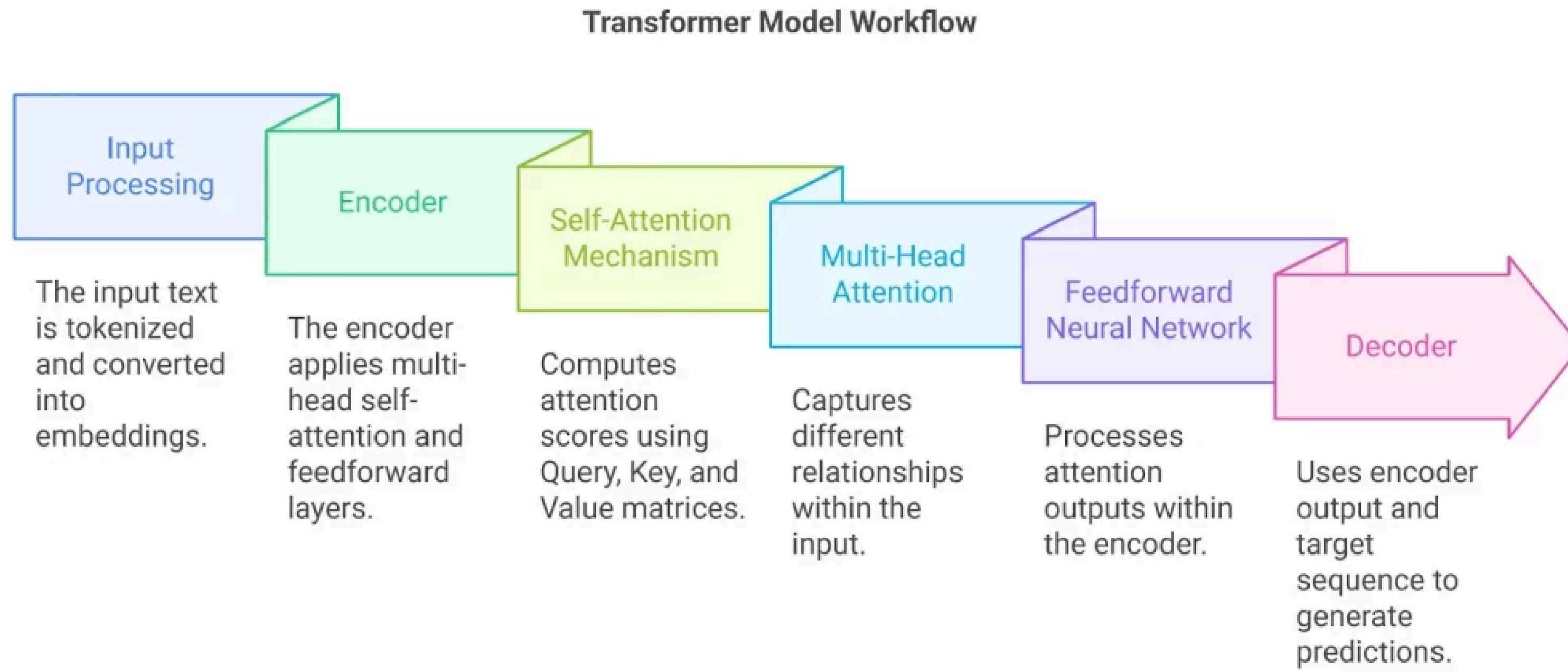
FOUNDATION MODELS & TRANSFORMERS

- Introduced in 2017 (Attention is All You Need)
- Key Innovation (Self Attention)
- Architecture
 - Encoder → understands input (used in BERT)
 - Decoder → generates output (used in GPT/ChatGPT)
 - Encoder - Decoder → translation, summarization

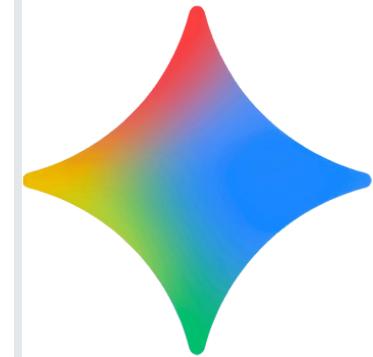
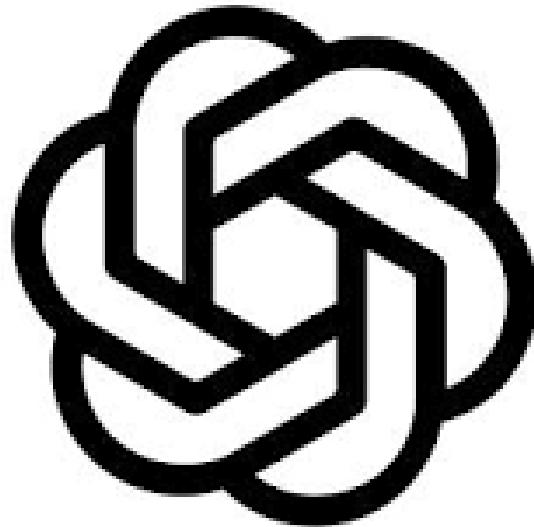
FOUNDATION MODELS & TRANSFORMERS

- Why Powerful?
 - Processes text in parallel
 - Faster than RNNs / LSTMs
 - Scales to billions of parameters
 - Adaptable to text, images, audio, code
- ChatGPT = Decoder-only Transformer
 - Specialized for text generation

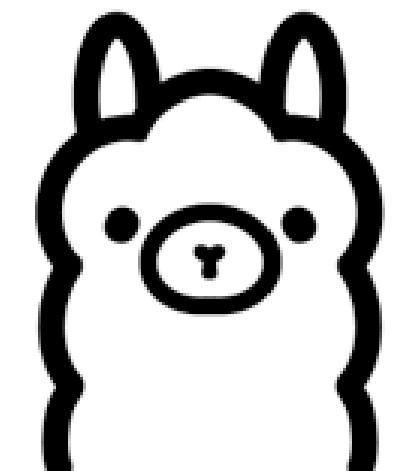
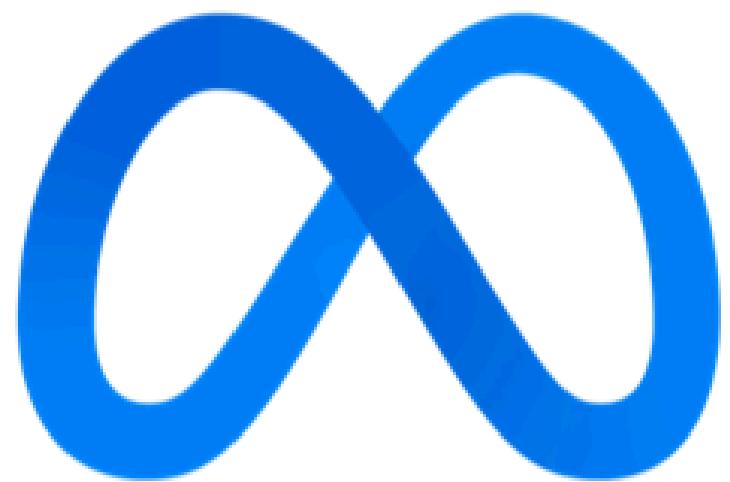
FOUNDATION MODELS & TRANSFORMERS



LARGE LANGUAGE MODELS



Gemini



LARGE LANGUAGE MODELS

RC which number is greater 1.11 or 1.2?

1.2 is greater than 1.11.

To see why, you can compare them by looking at the decimal places:

- $1.11 = 1.11$
- $1.2 = 1.20$

Since $1.20 > 1.11$, we have $1.2 > 1.11$.

Another way to think about it: 1.2 is equivalent to 1 and $2/10$, while 1.11 is equivalent to 1 and $11/100$. Since $2/10 = 20/100$, and $20/100 > 11/100$, therefore $1.2 > 1.11$.

RC give me only answer

1.2

TEXT GENERATION

LLMs are built by using supervised learning ($A \rightarrow B$) to repeatedly predict the next word

My favorite food is a bagel with cream cheese

Input (A)

Output (B)



TEXT GENERATION

Hands-On

Write a simple explanation of a motor insurance claim rejection letter in customer-friendly language.

Generate an advisory note for farmers on water management during drought season.

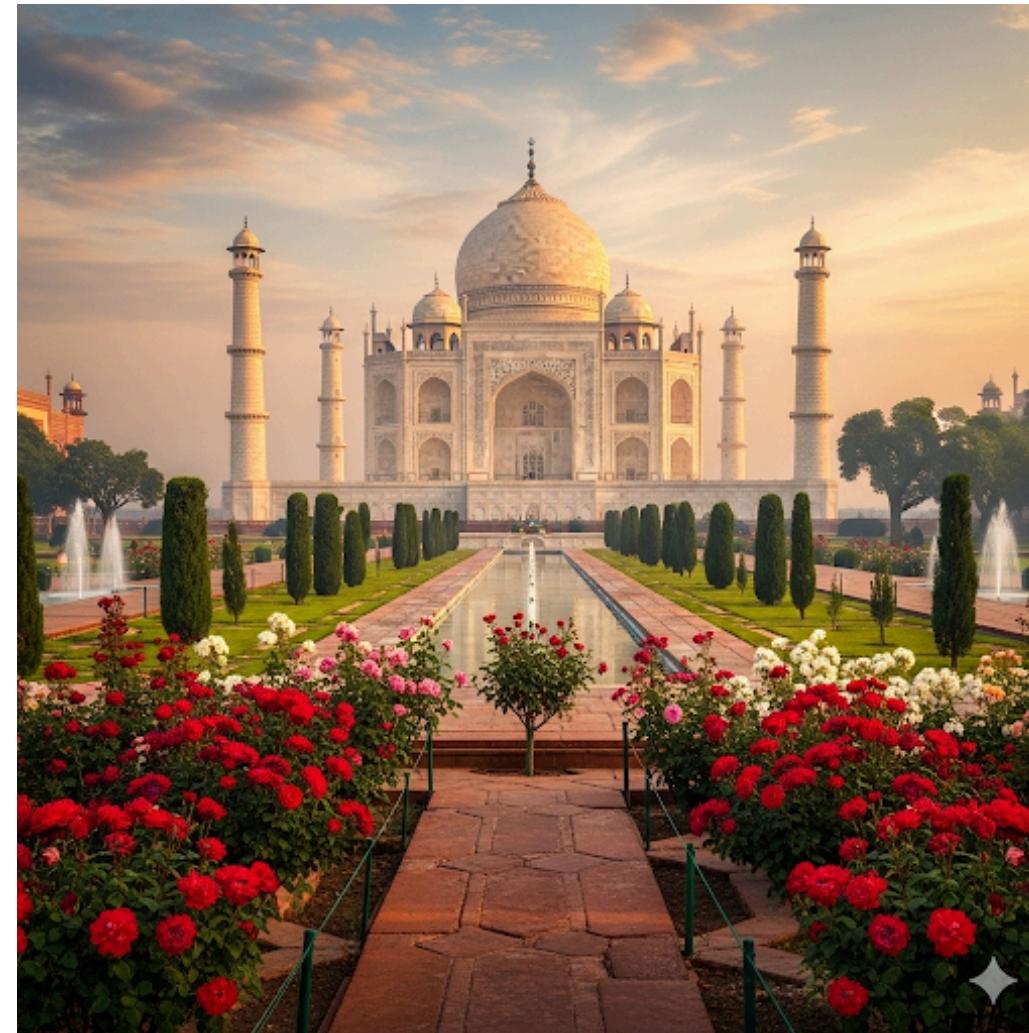


IMAGE GENERATION

Example Generated Images



A Picture of Women Smiling



Taj Mahal with a rose garden in front



Taj Mahal in night view

IMAGE GENERATION

It works by using Diffusion Model



Image 1,
"red apple"



Image 2

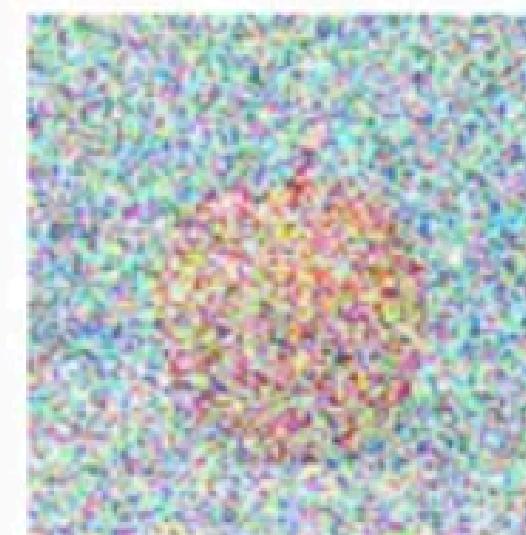


Image 3

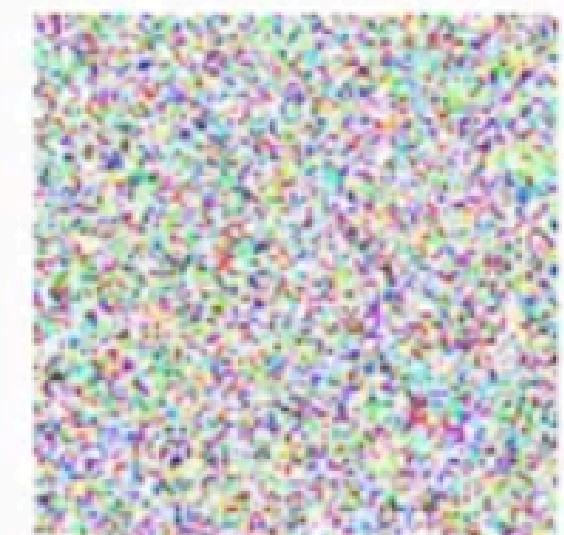


Image 4

Input (A)

Noisy image
and caption



Output (B)

Slightly
less noisy
image

Input (A)

Image 2, "red apple"
Image 3, "red apple"
Image 4, "red apple"

Output (B)

Image 1
Image 2
Image 3

IMAGE GENERATION

It works by using
Diffusion Model

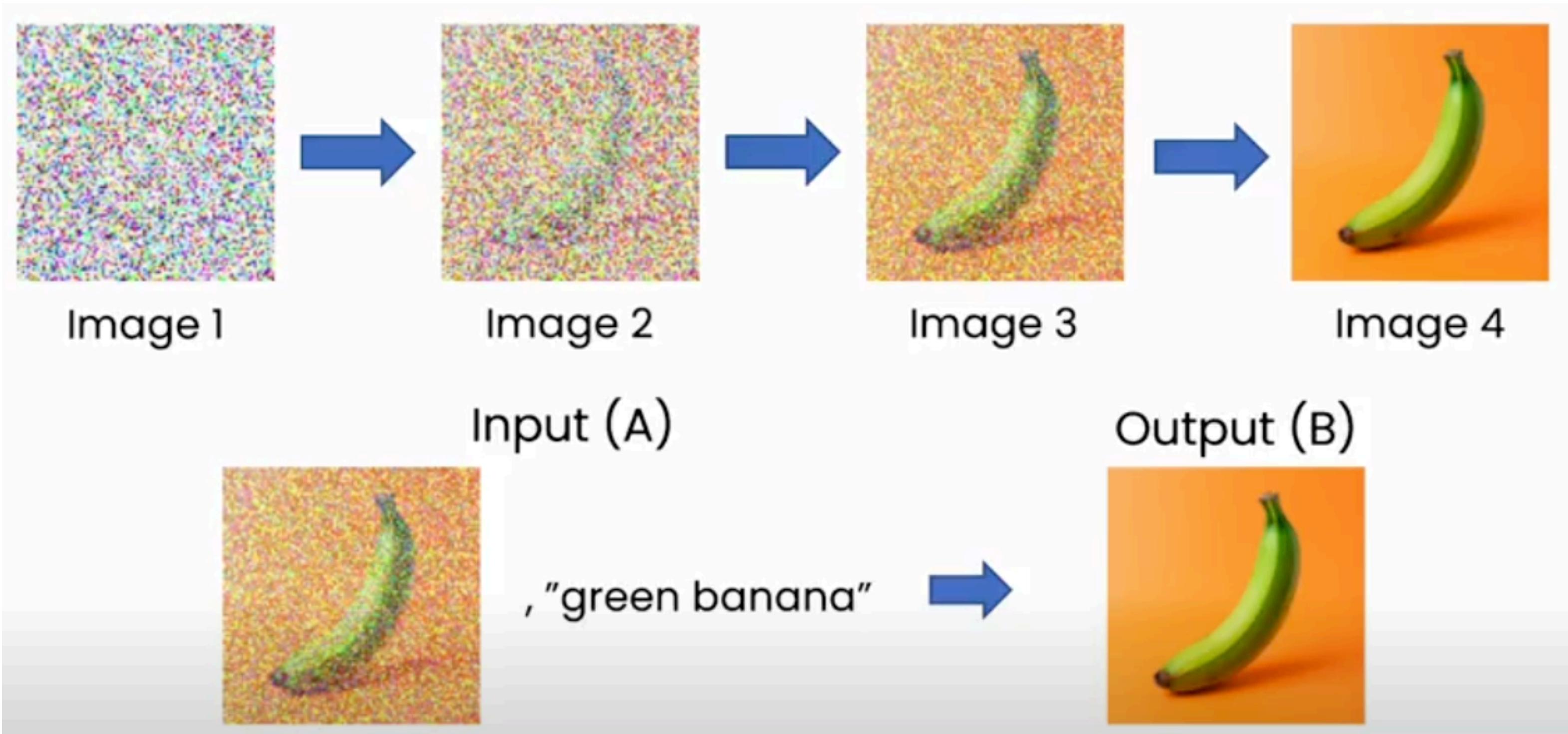


IMAGE GENERATION

Hands-On

Generate an illustration of a car accident claim scene with two cars and an insurance officer.



IMAGE GENERATION

Hands-On

Generate an illustration of a car accident claim scene with two cars and an insurance officer.





BREAK



Application of Generative AI

Education

1

Healthcare

2

Finance

3

Entertainment
and media

4

Virtual assistant

8

Gaming

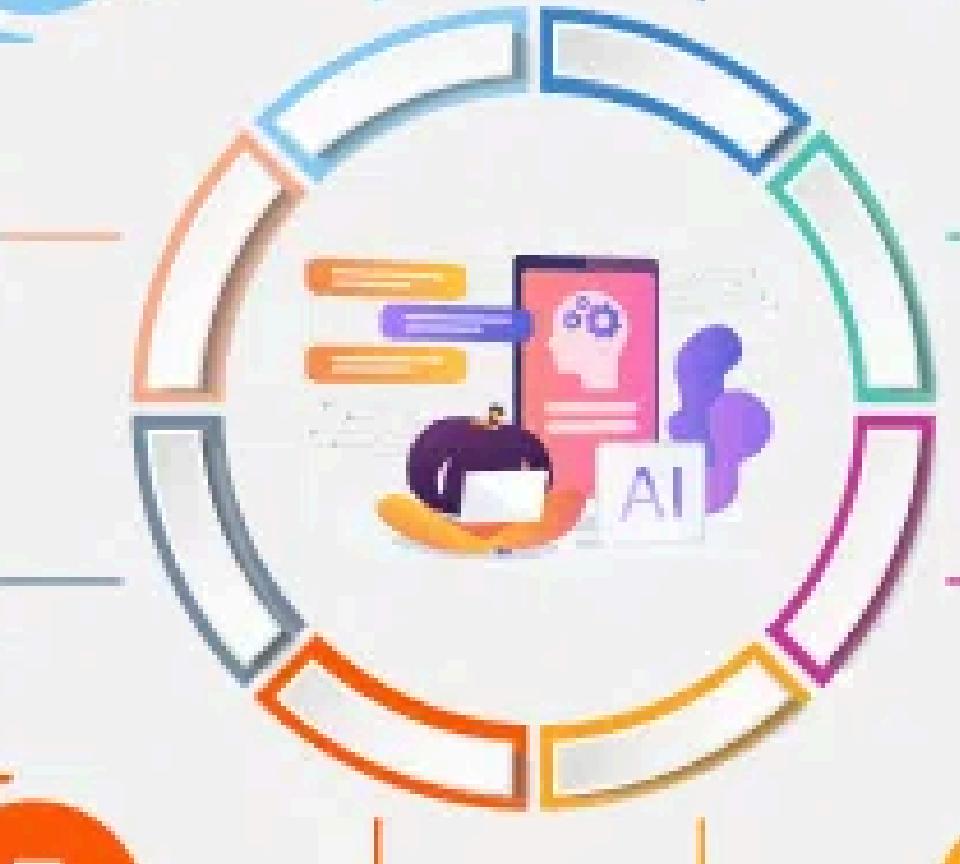
7

Content
Creation

6

Cybersecurity

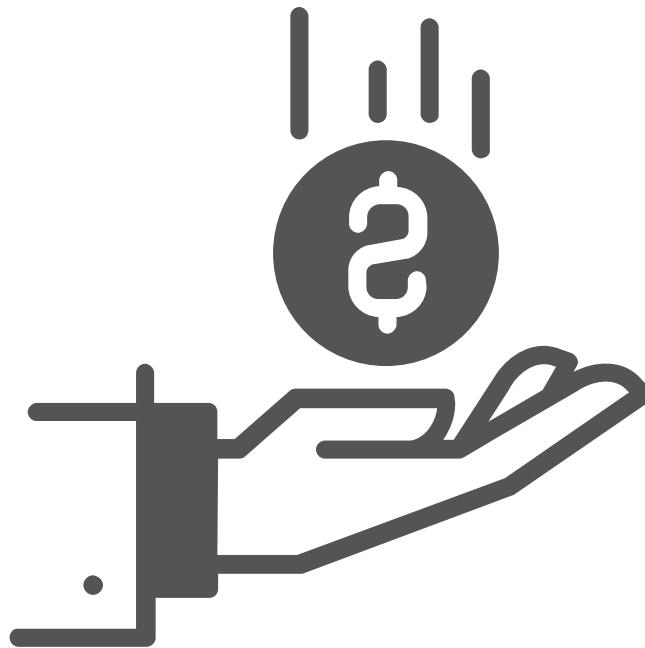
5



GENERATIVE AI APPLICATION

In Finance

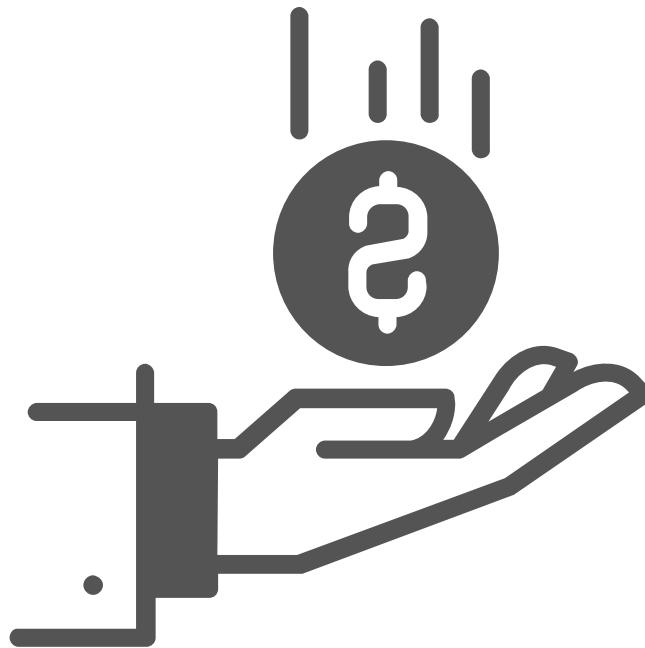
- Finance Report Summarization
- Automated compliance & Regulatory Checks
- Conversational Banking Assistants
- Fraud Detection & Customer Alerts
- Risk Assessment Reports



GENERATIVE AI APPLICATION

In Insurance

- Personalized Policy Recommendations
- Automated Claims Processing
- Fraud Detection In Insurance claim
- Image & Video Analysis for the claim process
- Personalized Campaign Content
- Virtual Insurance Advisors



GENERATIVE AI APPLICATION

In Agriculture

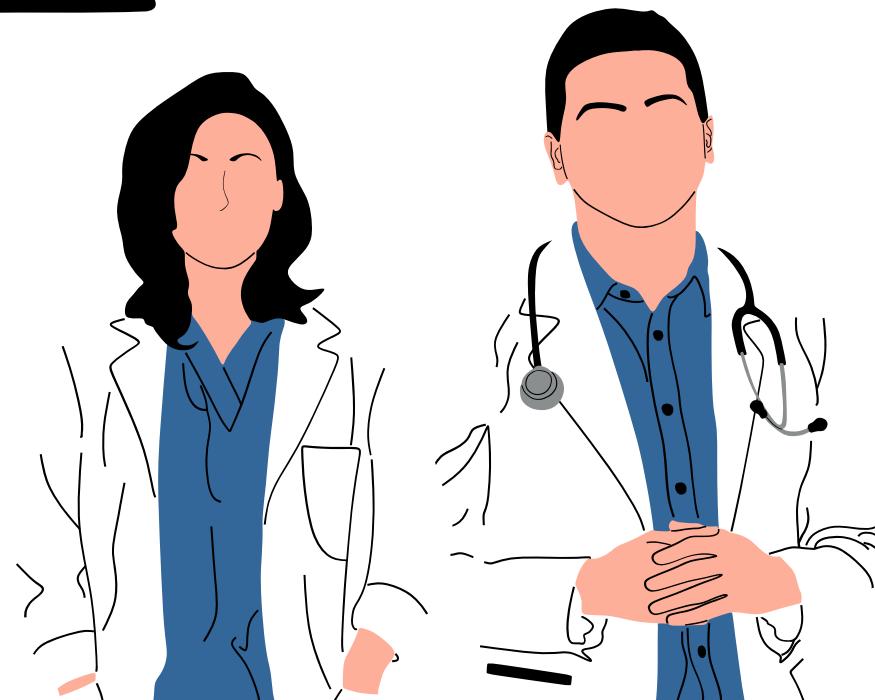
- Precision Farming Insights.
- Market Price Forecasting & Reports
- Voice bot in the local language
- Suggestions Weather forecasting
- Suggestions to improve the health of corp



GENERATIVE AI APPLICATION

In Healthcare

- Clinical Documentation & Summarization.
- Medical Research & Drug Discovery
- Patient Communication & Virtual Assistant
- Personalized Treatment & Simulation
- Healthcare Training & Education



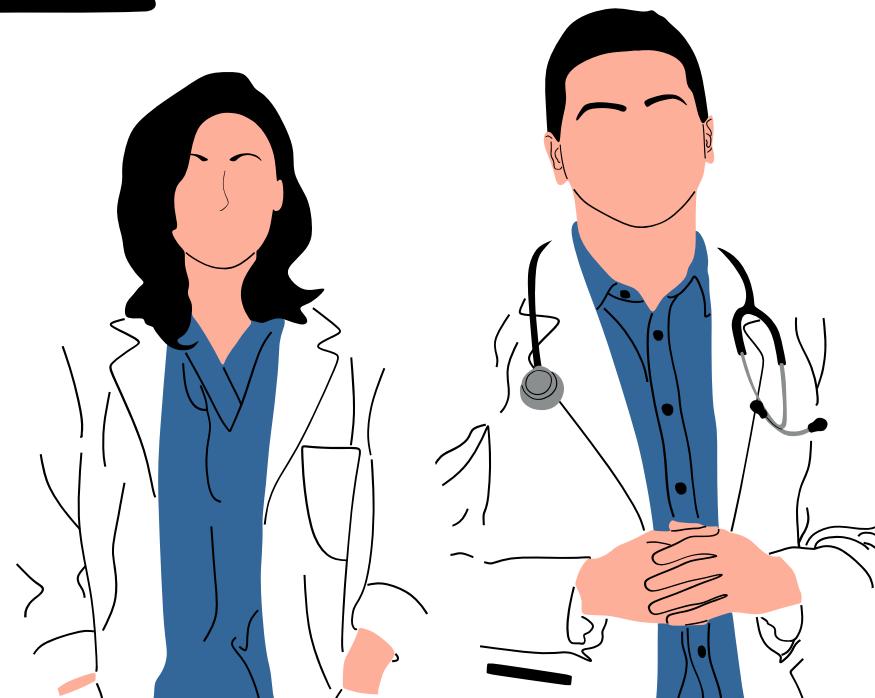
GENERATIVE AI APPLICATION

Impact In Healthcare

Gen AI is not just text/image generation, but transforming healthcare workflows end-to-end

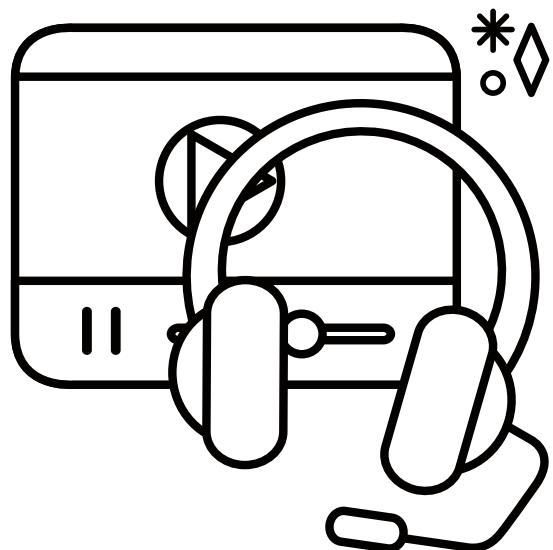
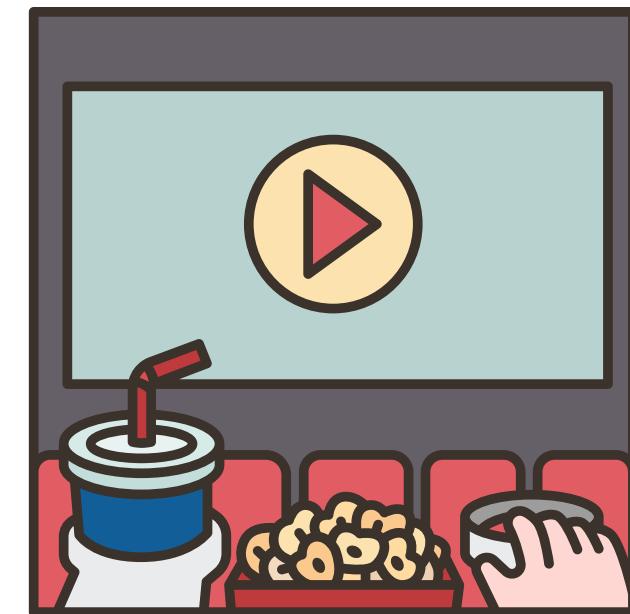


- Doctors → Faster Documentation
- Researchers → Faster Drug Discovery
- Patients → Better communication & engagement



GENERATIVE AI APPLICATION In Entertainment

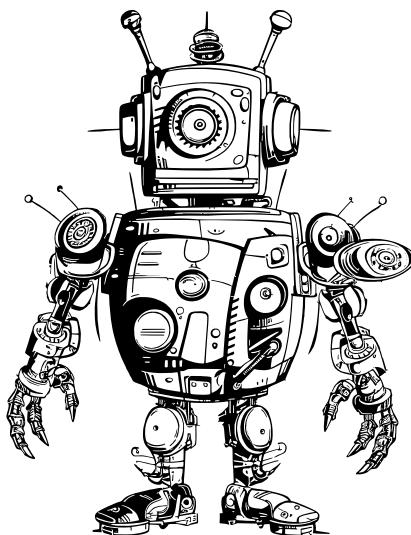
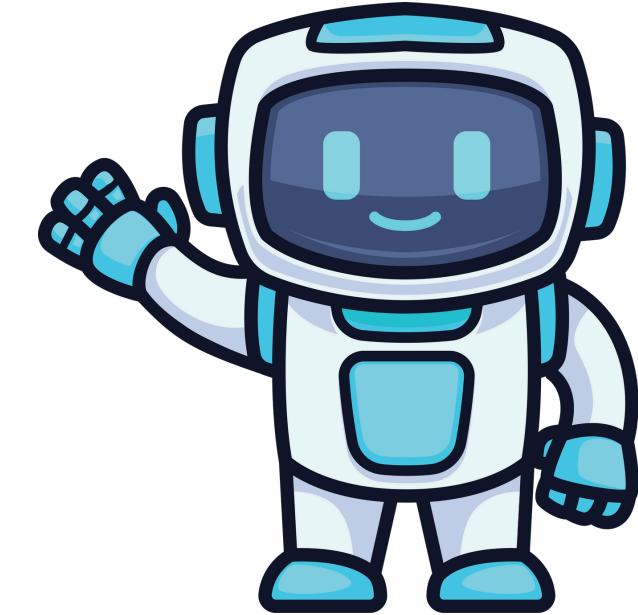
- Script & Story Generation
- Music & Audio Generation
- Gaming Content Creation
- Visual Effects & Animation
- Personalized Content



GENERATIVE AI APPLICATION

In Robotics

- Robotics in Healthcare
 - Example: A hospital robot that explains medicine schedules in simple terms
- Human-Robot Interaction
 - Example: A service robot in a bank answering customer queries in plain language.
- Simulation & Training Data
 - Example: A warehouse robot trained in AI-generated 3D layouts before deployment





FINE-TUNING & CUSTOMIZATION

FINE-TUNING & CUSTOMIZATION

- Prompt Engineering
- RAG
- Fine-Tuning with Hugging Face
- Business Applications

PROMPT ENGINEERING

Prompt Engineering is the practice of designing and optimizing input instructions (prompts) to guide a Generative AI model (like ChatGPT, GPT-4, or Stable Diffusion) to produce accurate, relevant, and useful outputs.

PROMPT ENGINEERING

- It's like "**programming with natural language**".
- Good prompts = better responses (structured, role-based, contextual).
- Techniques include:
 - **Zero-shot prompting** (no examples, just instructions).
 - **Few-shot prompting** (give examples to guide the model).
 - **Chain-of-thought prompting** (ask model to reason step by step).
 - **Role prompting** ("You are a financial advisor...").

PROMPT ENGINEERING

FEW SHOT PROMPTING

Classify the sentiment of the following customer reviews as Positive, Negative, or Neutral.

<Review> The insurance claim process was quick and hassle-free. </Review> → Positive

<Review> The bank's customer support was unhelpful and rude. </Review> → Negative

<Review> The SAP invoice system works fine, nothing special. </Review> → Neutral

Now classify the sentiment of this review:

<Review> The loan approval process was smooth but took longer than expected. </Review>

Output: Neutral

PROMPT ENGINEERING CHAIN OF THOUGHTS

A customer bought an insurance **policy for** ₹5,000.

The **policy** covers hospital bills up to ₹2,00,000.

Last month, the customer had a hospital expense of ₹1,50,000.

How much will the insurance company pay?

Think **step by step** before giving the final answer.

PROMPT ENGINEERING

CHAIN OF THOUGHTS

- Policy cost = ₹5,000 (not relevant for claim amount).
- Coverage limit = ₹2,00,000.
- Actual hospital bill = ₹1,50,000.
- Since bill < coverage limit, insurance pays full bill.

The insurance company will pay ₹1,50,000.

PROMPT ENGINEERING



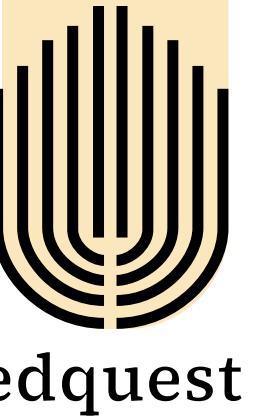
HANDS-ON

A farmer sells 120 kg of wheat at ₹25 per kg.
He then spends ₹1,500 on fertilizer for his field.
How much profit did the farmer make in total?

BENIFITS OF PROMPT ENGINEERING

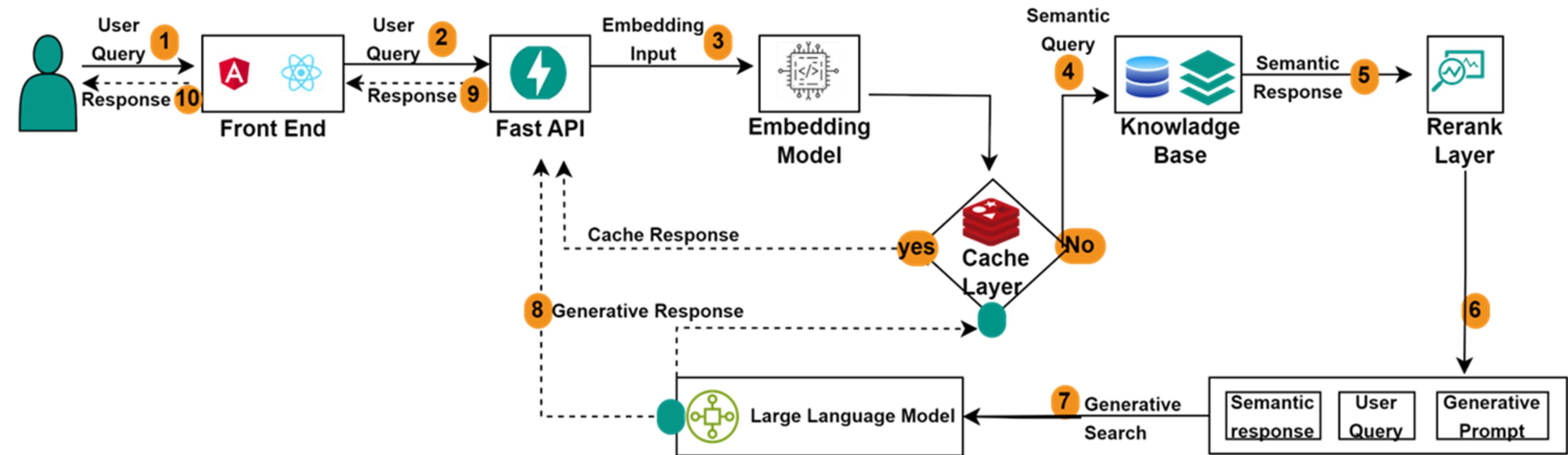
- **Improves output quality** – clearer, more accurate responses.
- **Customizes model behavior** – control tone, style, and format.
- **Adapts models to domains** – use in finance, insurance, healthcare without retraining.
- **Faster prototyping** – build & test AI solutions quickly.
- **Cost-effective** – avoids expensive fine-tuning while still improving results.

FEEDBACK & SUGGESTIONS



RAG

RETRIEVAL-AUGMENTED GENERATION



WHAT IS FINE-TUNING?

- Fine-tuning = **adapting a pre-trained model** to your domain/task.
- Starts from a large foundation model (GPT, BERT, etc.).
- Improves performance on domain-specific data (insurance policies, banking regulations, agriculture reports, etc.).

Example:

- Base model: GPT (trained on general internet text).
- Fine-tuned model: Insurance claim assistant trained on claim forms & FAQs.

WHEN TO USE RAG?

 Use **Prompt Engineering** when:

- Quick experiments
- General use cases
- Style/tone control

 Use **RAG** when:

- Model **fails with prompting**
- You need **domain accuracy**
- Business requires **consistent outputs**
- Sensitive domain (finance, healthcare, SAP ERP data)

APPROACHES & BENEFITS OF FINE-TUNING?

Approaches:

- **Full fine-tuning** → retrain all parameters (expensive).
- **Parameter-efficient tuning (PEFT)** → LoRA, adapters (cheaper).
- **Transfer learning** → reuse base model + add small domain dataset.

Benefits:

- Better domain accuracy
- Consistency in outputs
- Reduces need for long prompts

GENERATIVE AI ETHICS

GENERTIVE AI ETHICS

Why It Matters?

- AI is powerful but can create **risks** if unchecked.
- Ethical use ensures **trust, safety, and fairness**.
- Key issues in GenAI:
 - **Bias** → unfair outcomes
 - **Hallucinations** → wrong but confident answers
 - **Security risks** → misuse or data leaks

GENERTIVE AI ETHICS

Bias & Fairness

- **Bias:** Model reflects unfair patterns from training data.
 - Example: Loan approval model favors certain demographics.
- **Fairness:** Ensuring equal treatment across all users.

Mitigation:

- Train on **diverse datasets**
- Apply **bias detection tools**
- Keep **human-in-the-loop**

GENERTIVE AI ETHICS

Hallucinations in AI

- **Definition:** AI generates incorrect or fabricated information.
- **Example:** AI chatbot fabricates RBI rules in banking compliance.

Mitigation:

- **Retrieval-Augmented Generation (RAG)** for grounding
- Clear **confidence scoring**
- Encourage **human validation**

GENERTIVE AI ETHICS

Security Concerns

- **Prompt Injection Attacks** → user manipulates AI to reveal secrets.
- **Data Leakage** → sensitive info exposed in training or output.
- **Misuse** → Deepfakes, fraud, misinformation.

Mitigation:

- **Guardrails & input filters**
- Access control & encryption
- **Regular audits & monitoring**

PRICE & TOKEN LIMIT

Realtime API

Build low-latency, multimodal experiences including speech-to-speech.

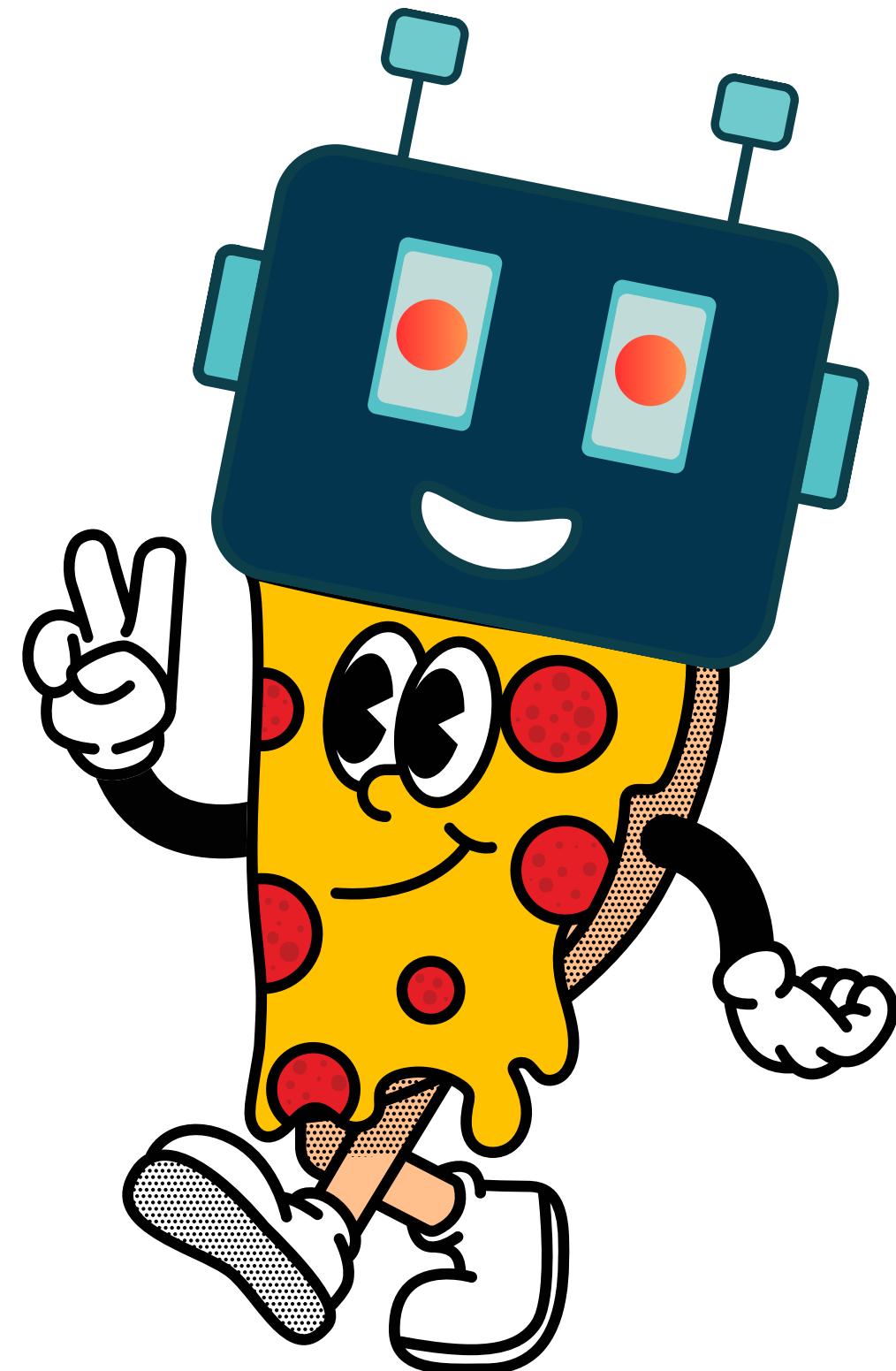
		\$4.00 / 1M input tokens	\$0.40 / 1M cached input tokens	\$16.00 / 1M output tokens
Text	gpt-realtime	\$4.00 / 1M input tokens	\$0.40 / 1M cached input tokens	\$16.00 / 1M output tokens
	GPT-4o mini	\$0.60 / 1M input tokens	\$0.30 / 1M cached input tokens	\$2.40 / 1M output tokens
Audio	gpt-realtime	\$32.00 / 1M input tokens	\$0.40 / 1M cached input tokens	\$64.00 / 1M output tokens
	GPT-4o mini	\$10.00 / 1M input tokens	\$0.30 / 1M cached input tokens	\$20.00 / 1M output tokens
Image	gpt-realtime	\$5.00 / 1M input tokens	\$0.50 / 1M cached input tokens	-

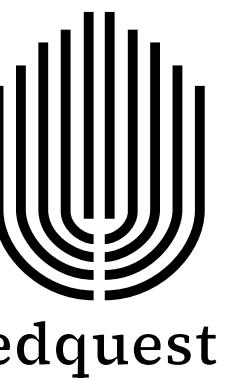
<https://openai.com/api/pricing/>

HANDS-ON

PIZZA BOT

AI-powered Burger Bot that takes customer orders in natural language and converts them into structured order details (burger type, add-ons, sides, drinks, quantity).





RESOURCES

C
C
C

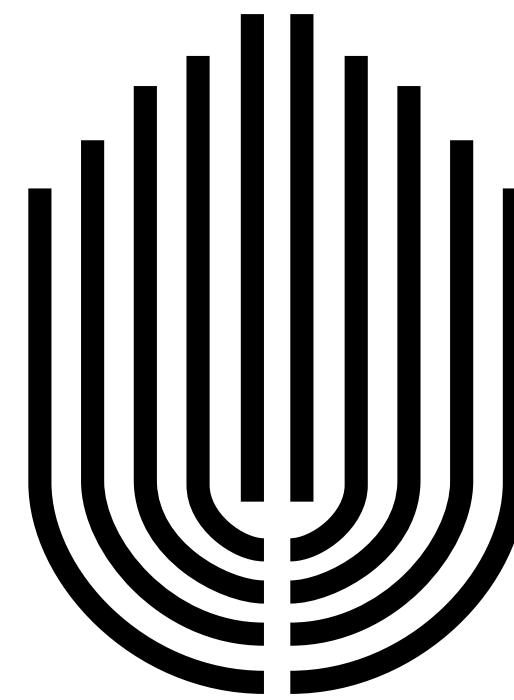


edquest

Our Services

- **AI Product Development** – Build domain-specific GenAI tools (chatbots, report summarizers, dashboards).
- **AI Resource Augmentation** – On-demand experts (model tuning, prompt engineering, RAG, Agentic AI, MCP, integrations).
- **Hands-On AI Training** – Cohort-based, project-driven programs for rapid team upskilling.
- **Tool & Platform Expertise** – OpenAI, Hugging Face, LangChain, Pinecone, Chroma DB, FastAPI.

<https://edquest.co.in>



edquest

Contact Us



@edquestofficial



info@edquest.co.in



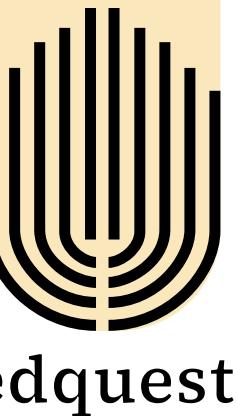
@edquestofficial



+91 - 9467752177



www.edquest.co.in



THANK
YOU

