



# Part I - Generative AI Holistic Overview

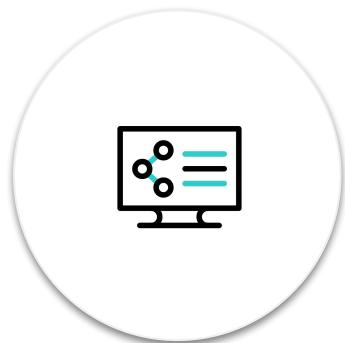
# About Datacouch



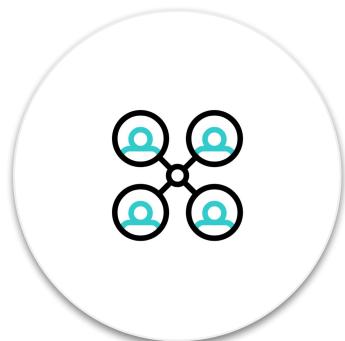
**Training for Professionals**



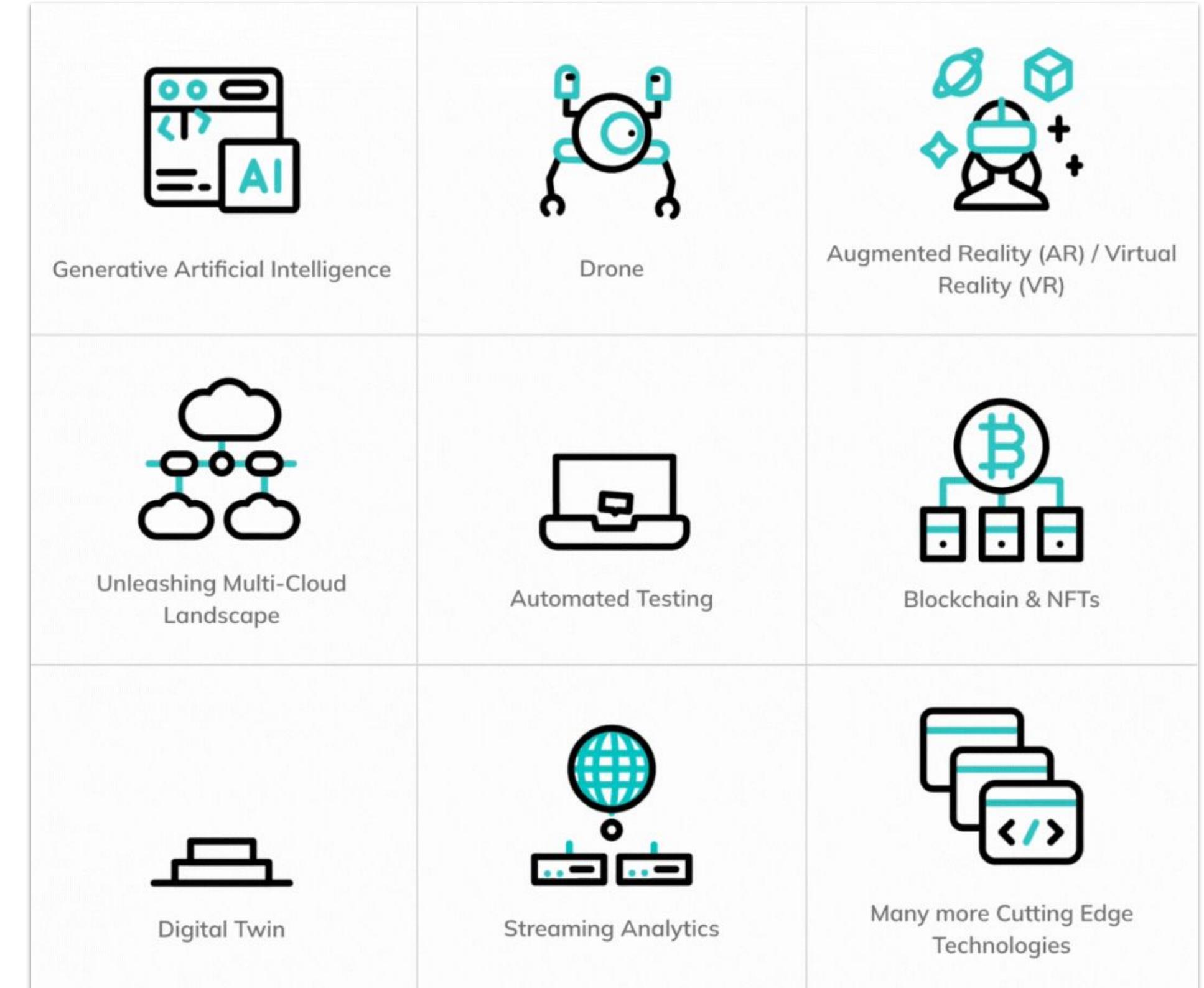
**Consulting**



**Browser-based Lab Solution**



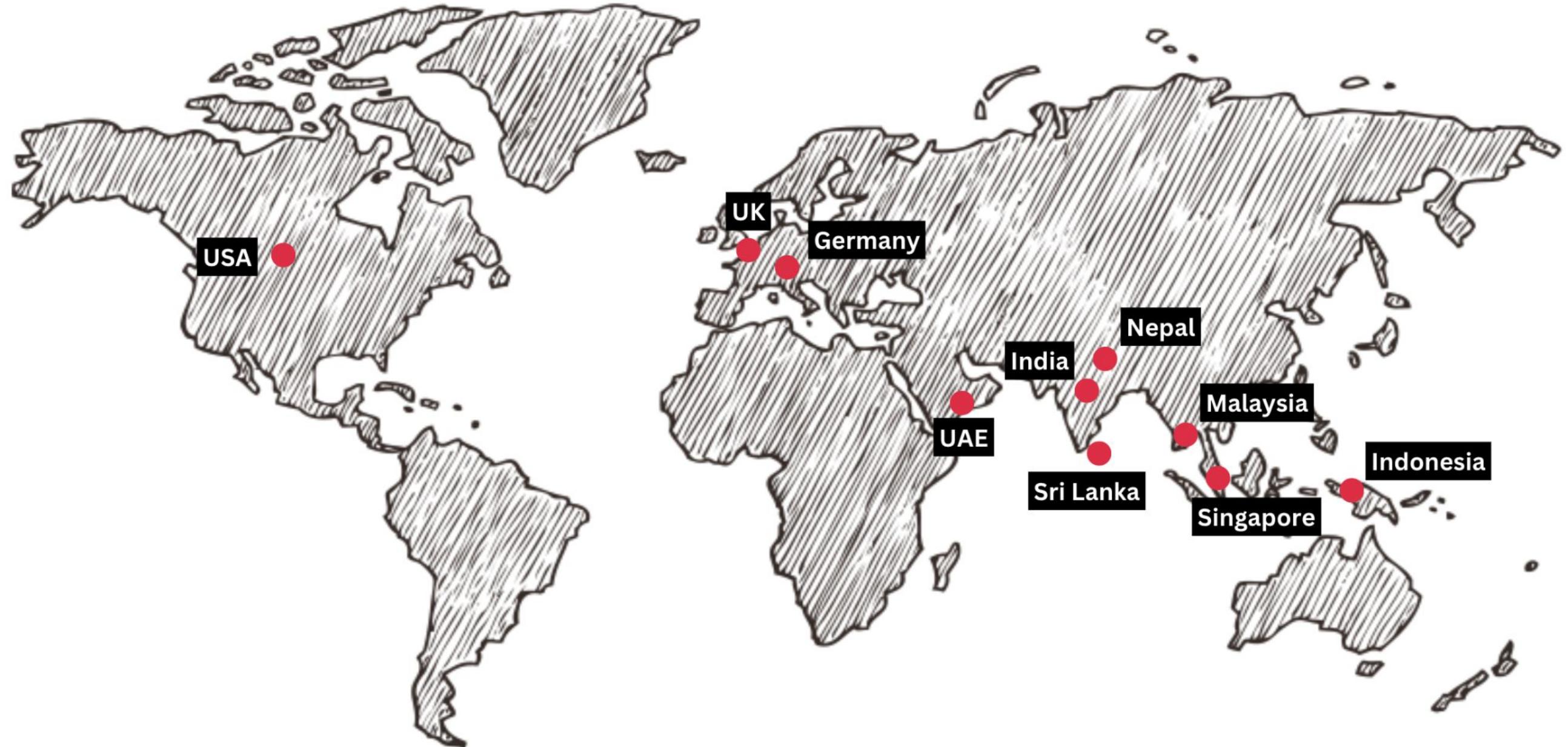
**Managed Services**



# About Datacouch

1

150K+ Participants Trained



2

100+ different technologies covered

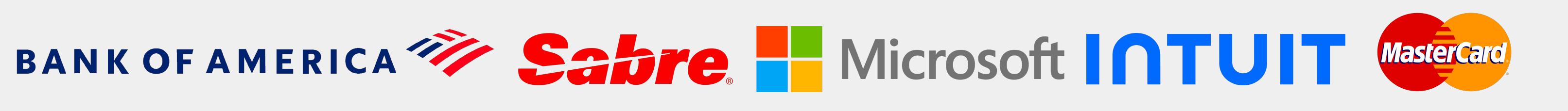
3

Focused in Web3, Generative AI,  
Data Engineering, Data Science,  
DevOps, Cloud, and many more.

4

Industry Experts / Globally Renowned  
/ Vetted by Google/Microsoft/Amazon

# Our Footprints



# Sharing Knowledge with Community!!



## AI Minds: The Hub For AI Intellectuals

📍 Bangalore, India  
👤 7,723 members · Public group  
👤 Organized by **Bhavuk Chawla** and 2 others

A group sponsored by

DATA COUCH

Pioneers in Blazing Edge Technologies



**2100+**

*Members Globally*

Share: [n](#) [f](#) [t](#) [in](#) [m](#)



A group sponsored by Datacouch for promoting Data, Machine Learning, and Artificial Intelligence awareness for social good.

## All Things Data

📍 Chandigarh, India  
👤 4,289 members · Private group  
👤 Organized by **Bhavuk Chawla** and 2 others



**100+**

*Meetup sessions conducted till now*

Share: [n](#) [f](#) [t](#) [in](#) [m](#)



## Generative AI User Group

📍 Pune, India  
👤 3,559 members · Public group  
👤 Organized by **Bhavuk Chawla** and 2 others

A group sponsored by

DATA COUCH

Pioneers in Blazing Edge Technologies



**50+**

*Various Technologies covered*

Share: [n](#) [f](#) [t](#) [in](#) [m](#)

# About Datacouch



**CONFLUENT HAS CHOSEN US AS THEIR ONLY  
GLOBAL EDUCATION  
PARTNER  
AGAIN FOR THE YEAR 2023**

WE ARE GRATEFUL TO CONFLUENT & OUR AMAZING  
TEAM FOR THIS EXCEPTIONAL FEAT - TWO YEARS RUNNING



# About Datacouch

**DATACOUCH IS CHOSEN AS THE**

**FIRST TRAINING  
DELIVERY PARTNER**

**BY**



**Starburst**



**ONEHOUSE**



# Overall objectives of the Webinar Series

- 01 AI Holistic Perspective
- 02 Scalable AI on Cloud
- 03 Prompt Engineering
- 04 Working with Large Language Models

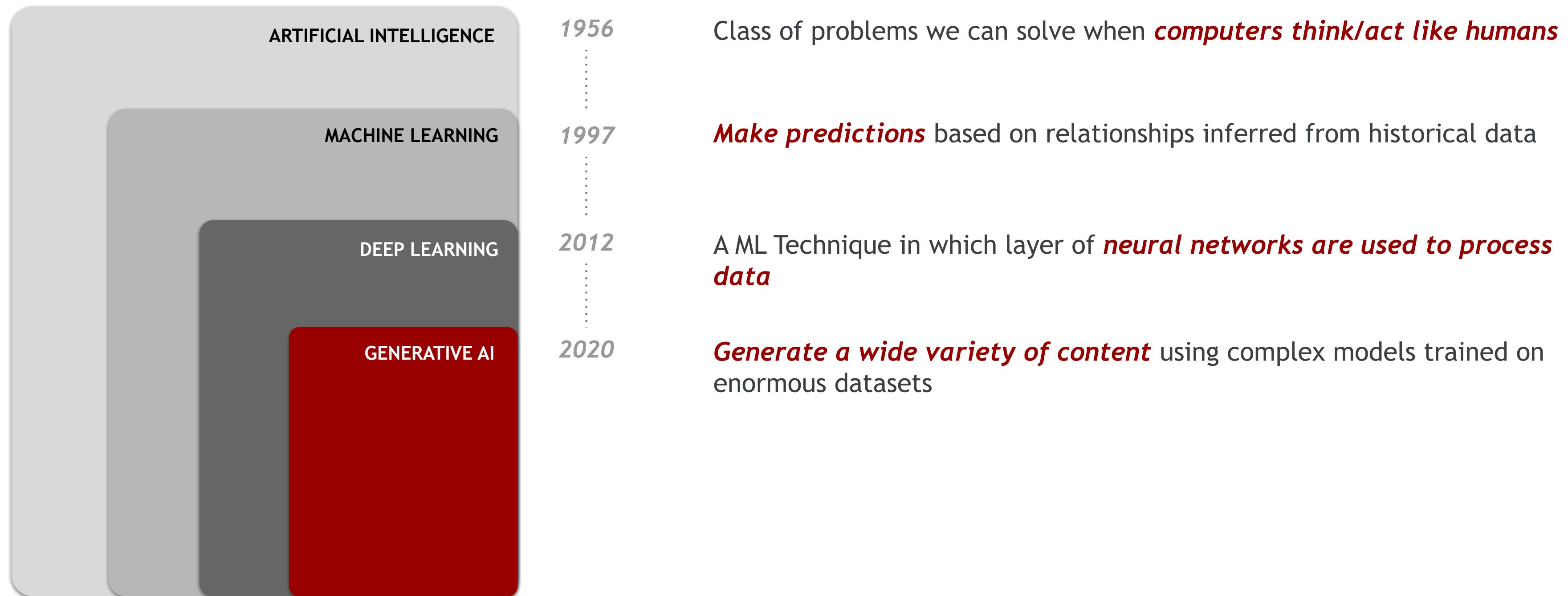


# Agenda for Today

- 01 | Understanding the fundamentals
- 02 | Generative AI & it's usecases
- 03 | An Introduction to LLMs
- 04 | Github Co-Pilot

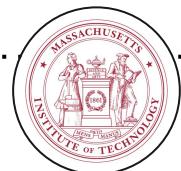


# Machine Learning is a type of AI, and Generative AI (GenAI) is a type of machine learning



# Overview - What is AI?

*It is the quest to build machines that can reason, learn, and act intelligently, and it has barely begun*



MIT

*Artificial intelligence (AI) applies advanced analysis and logic-based techniques, including machine learning, to interpret events, support and automate decisions and take action*



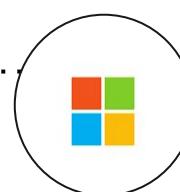
Gartner

*AI is computer programming that learns and adapts. It can't solve every problem, but its potential to improve our lives is profound*

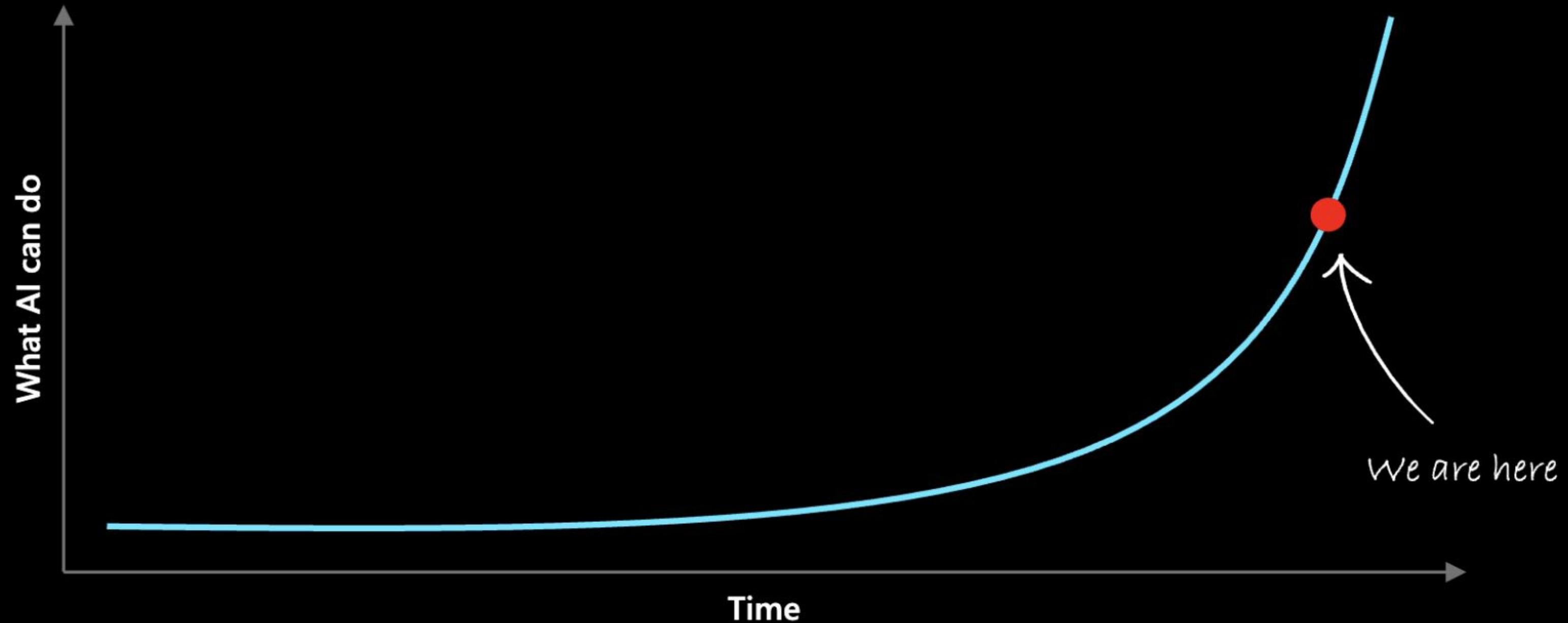


Google

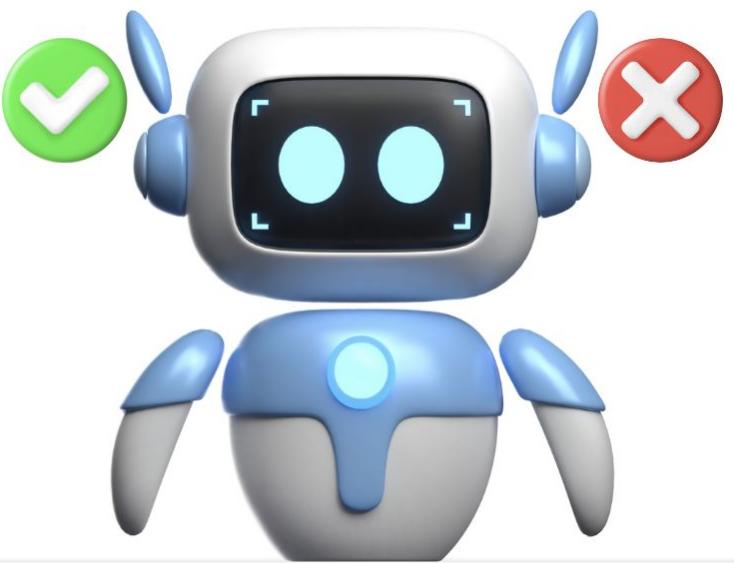
*It is the capability of a computer system to mimic human-like cognitive functions such as learning and problem-solving*



Microsoft

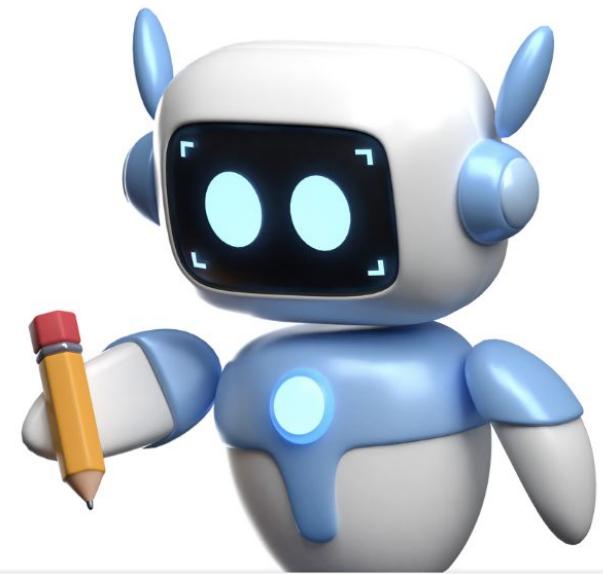


# Key ML Types



## DISCRIMINATIVE MODEL

*Used to classify or predict*

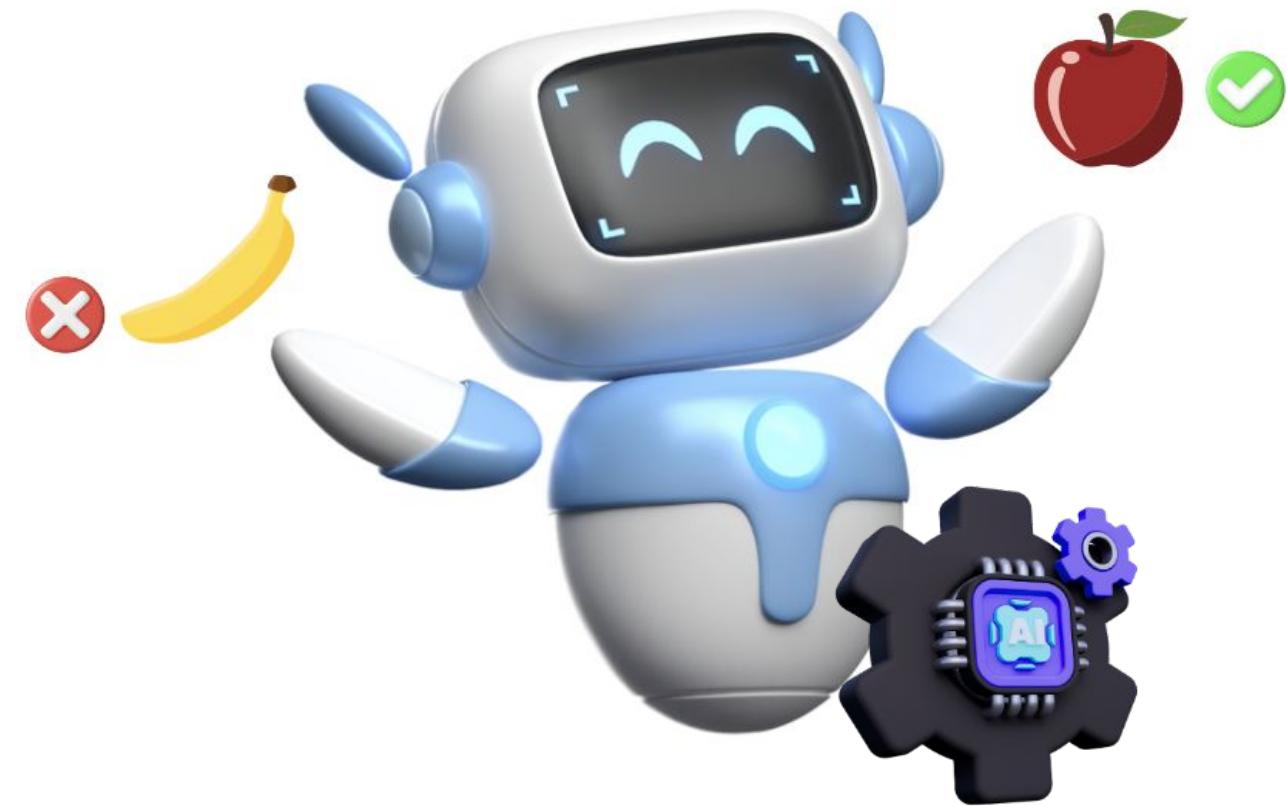


## GENERATIVE MODEL

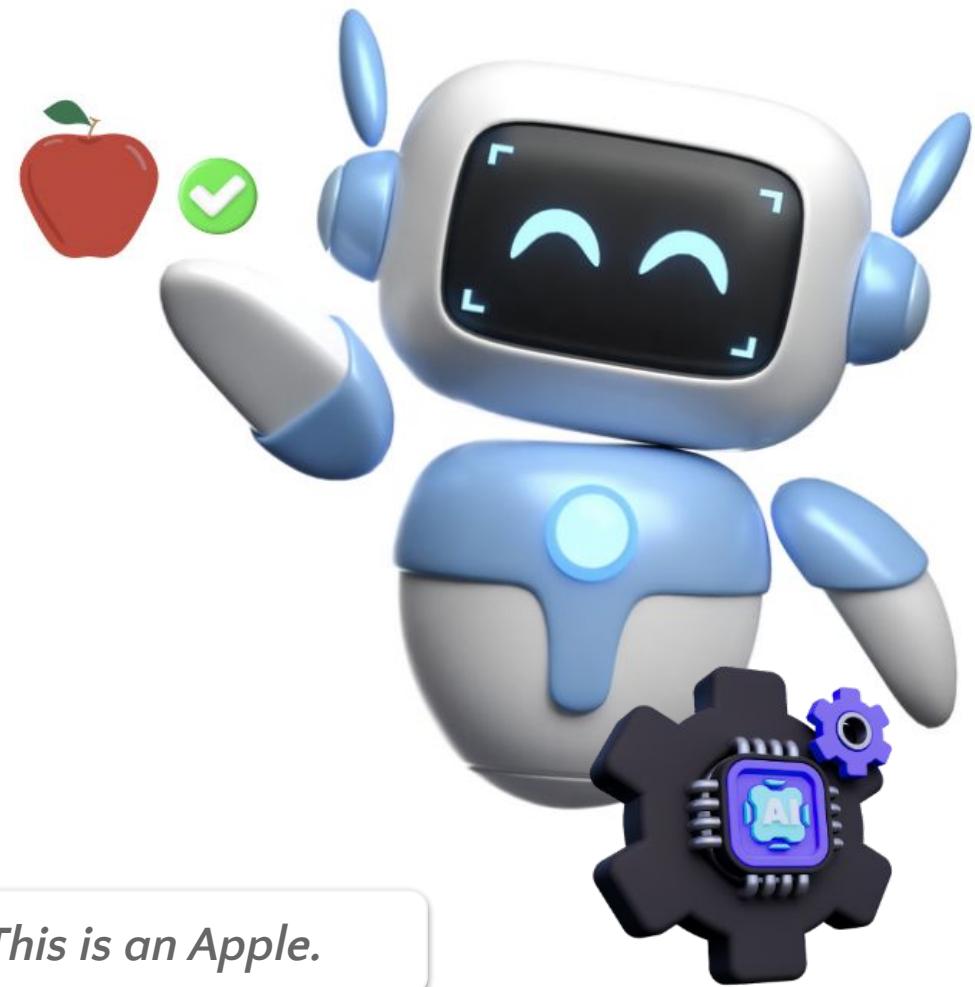
*Generates new data*

# Discriminative Model

*This is not an Apple.*



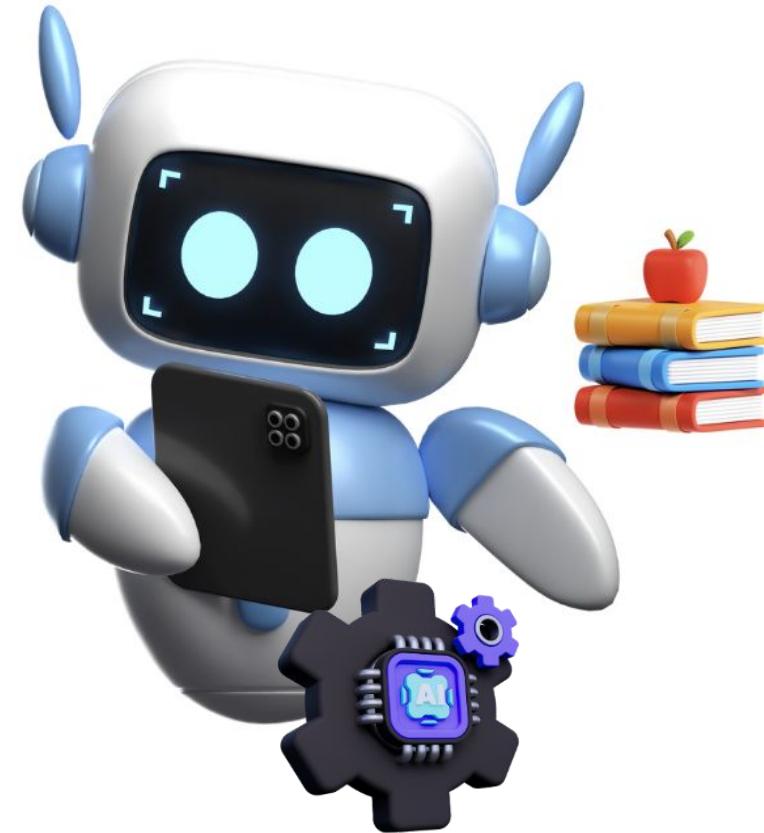
*This is an Apple.*



*This is an Apple.*

# Generative Language Model

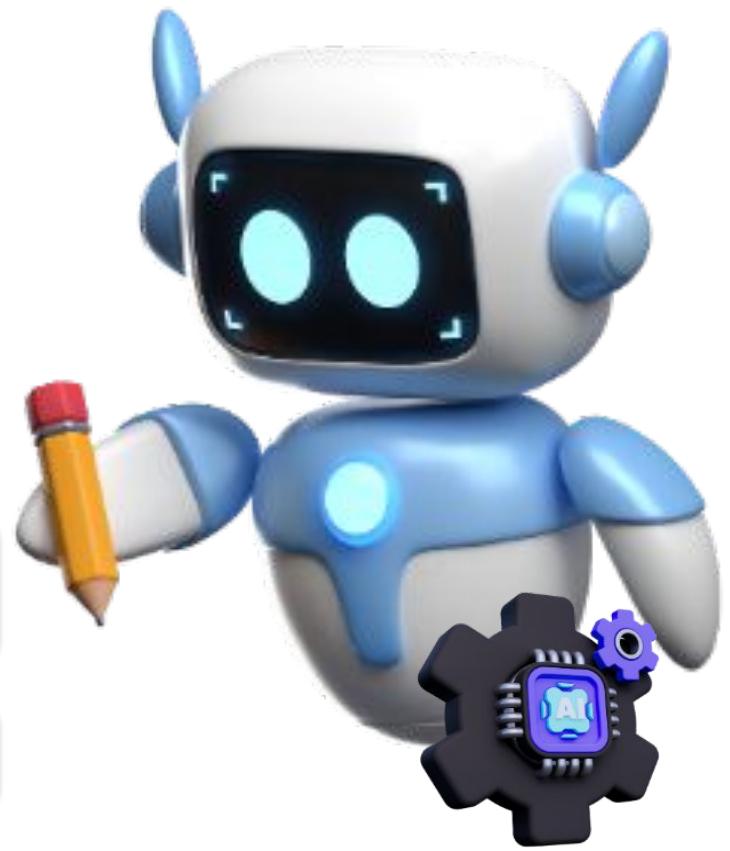
*Read this huge pile of books on apples*



*So, you have read about apple and  
its various types*

*Now tell me what is an apple?*

*An apple is a ...*

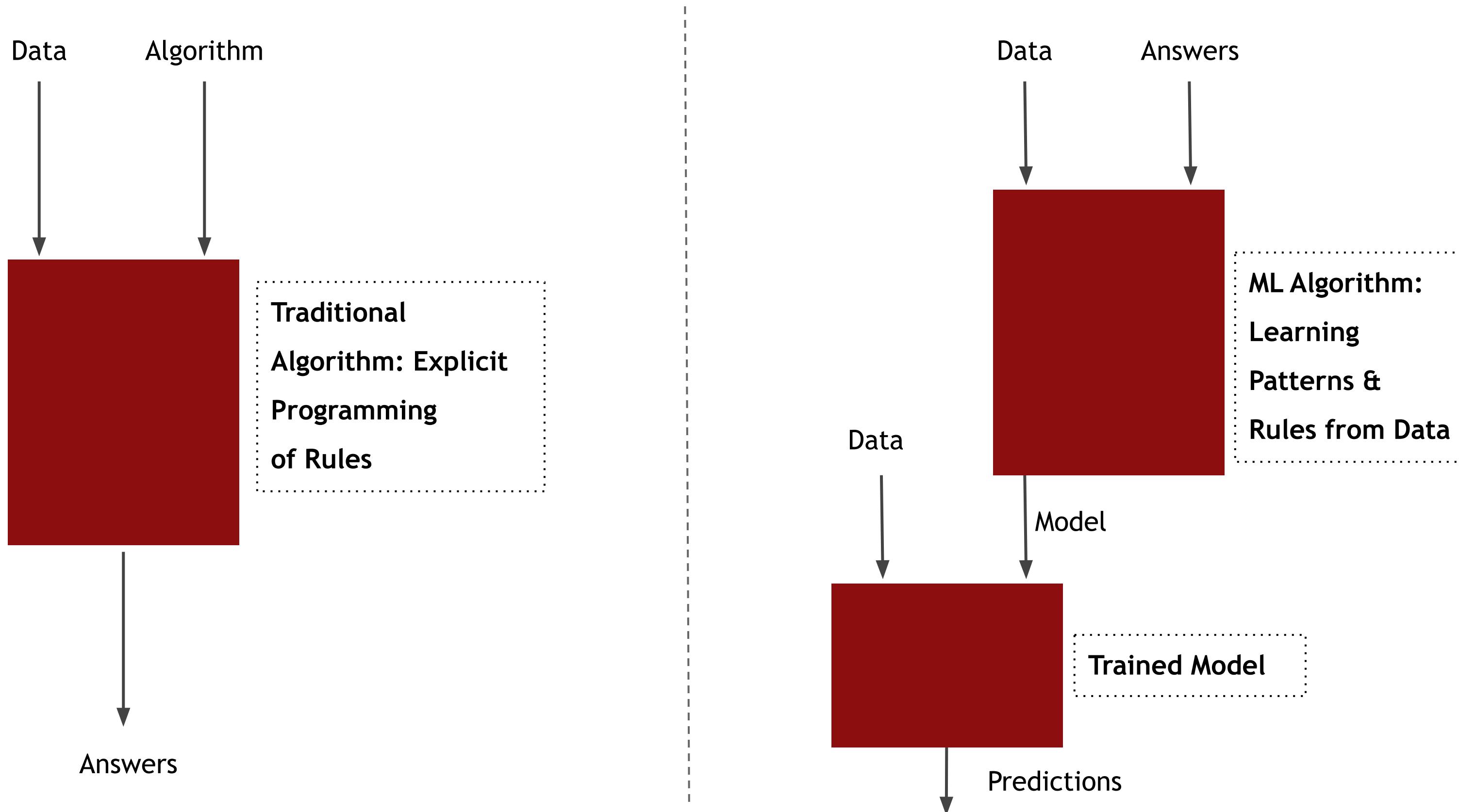


# Machine learning allows computers to learn without explicit programming

- In traditional programming, the programmer writes the code to perform a task
- In machine learning, algorithms are trained to make predictions using historical data
  - Computers iterate over the algorithm making adjustments to find the best solution



# Traditional Systems vs Machine Learning based AI Systems



# Example of Spam Classification using Traditional Systems

```
#!/usr/bin/env python

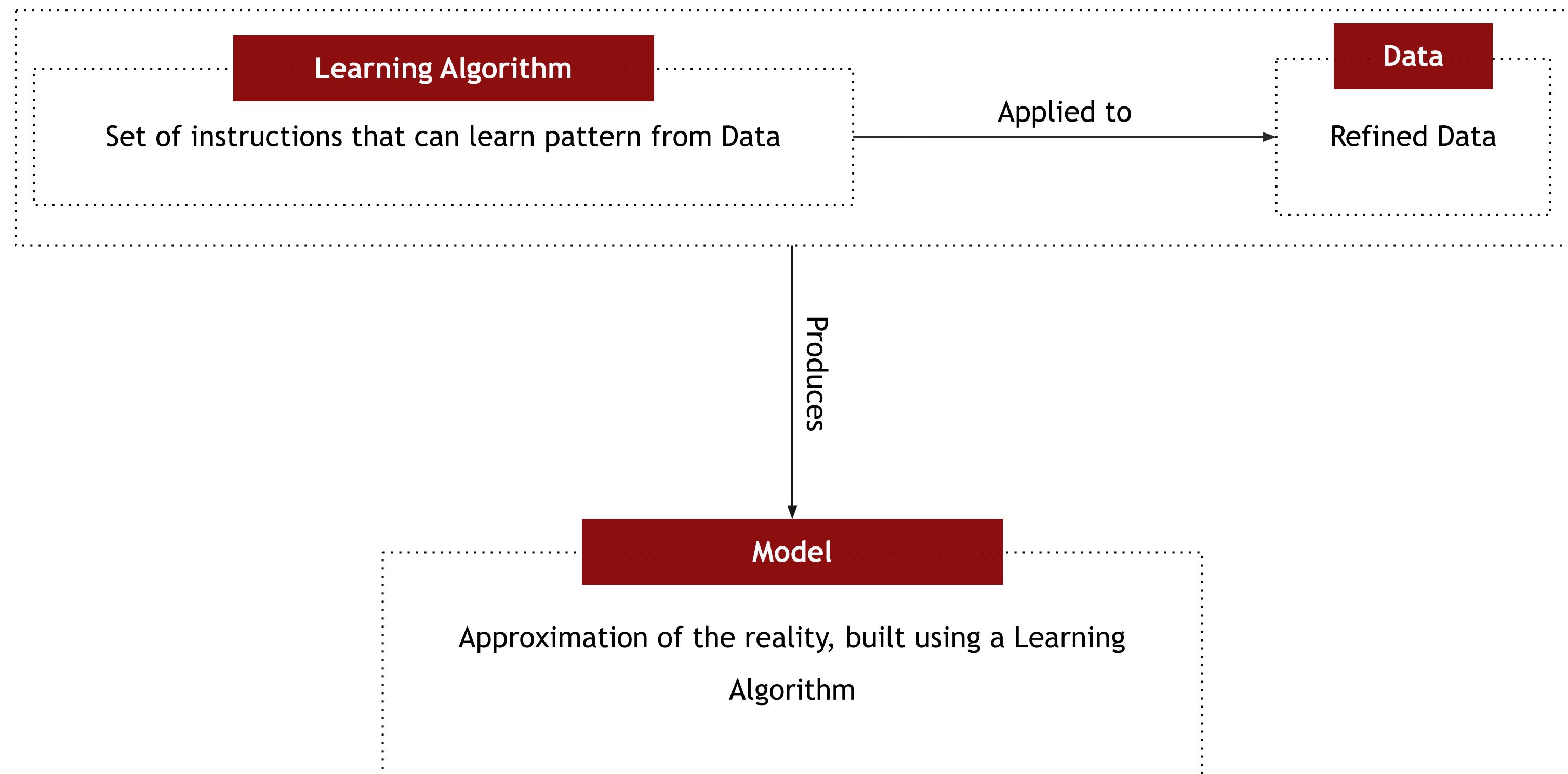
import sys
for line in sys.stdin:
    if "Make MONEY Fa$t At Home!!!" in line:
        print("This message is likely spam")
    if "Happy Birthday from Aunt Betty" in line:
        print("This message is probably OK")
```

# Impossible to solve Computer Vision Problems using Software Engineering based approach

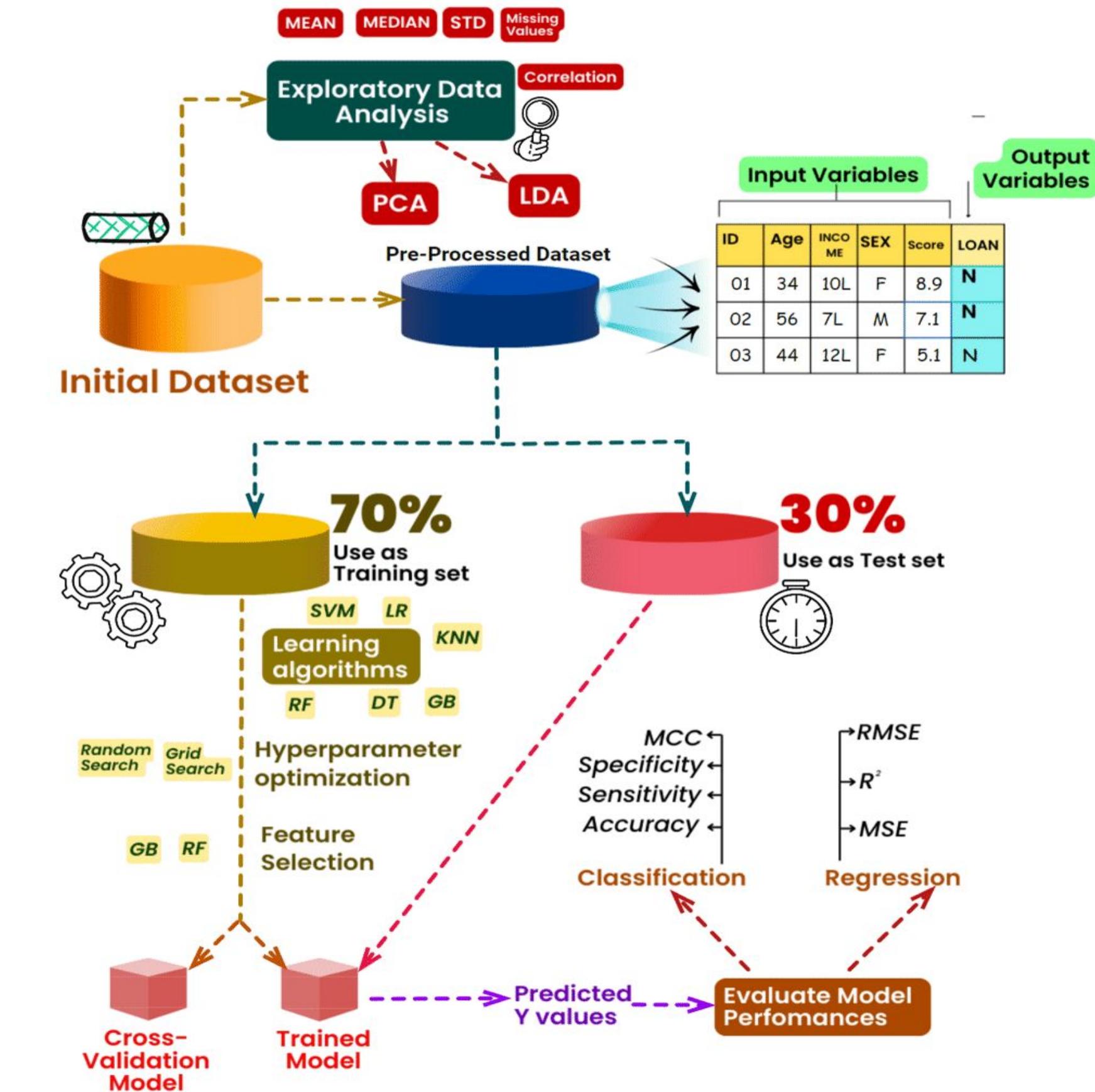


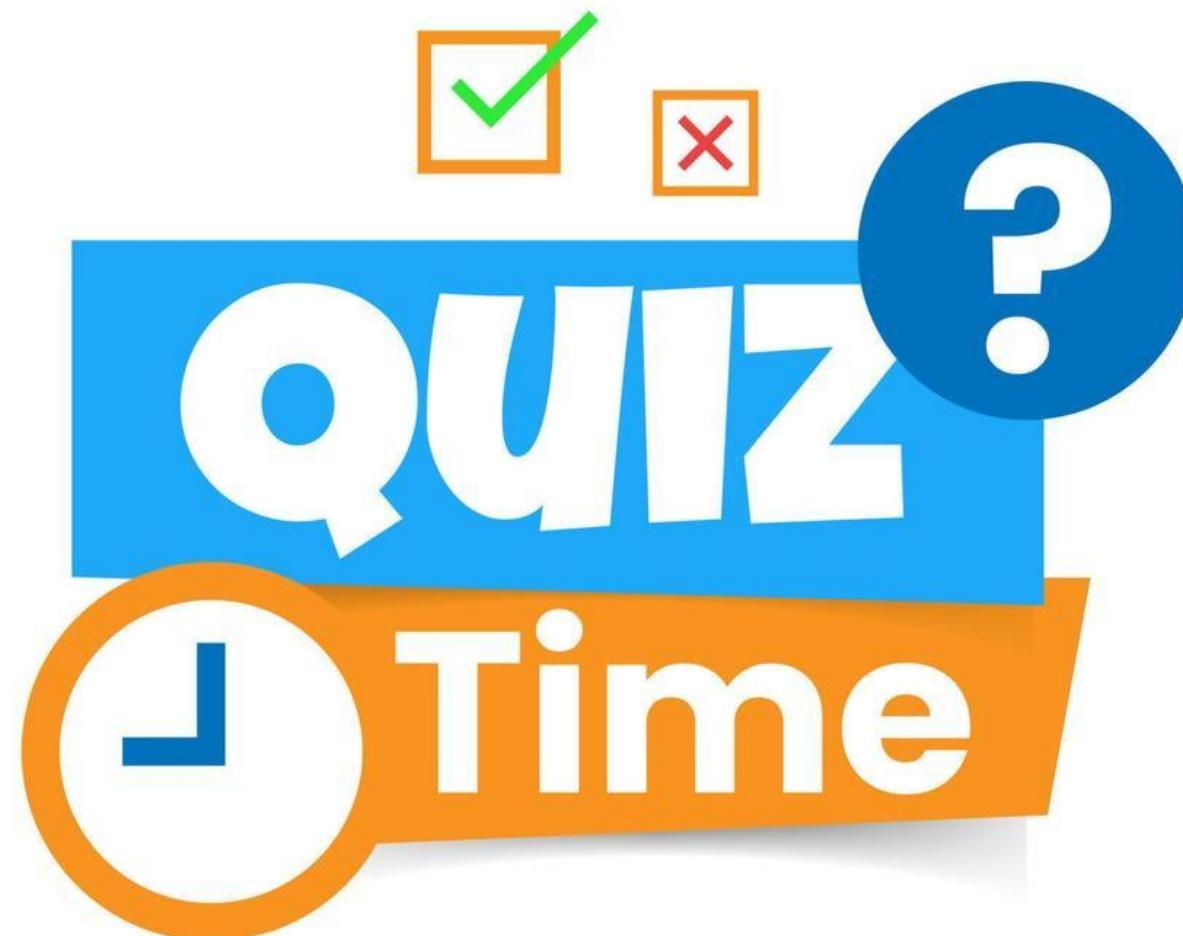
IN CS, IT CAN BE HARD TO EXPLAIN  
THE DIFFERENCE BETWEEN THE EASY  
AND THE VIRTUALLY IMPOSSIBLE.

# Learning Algorithm vs Model



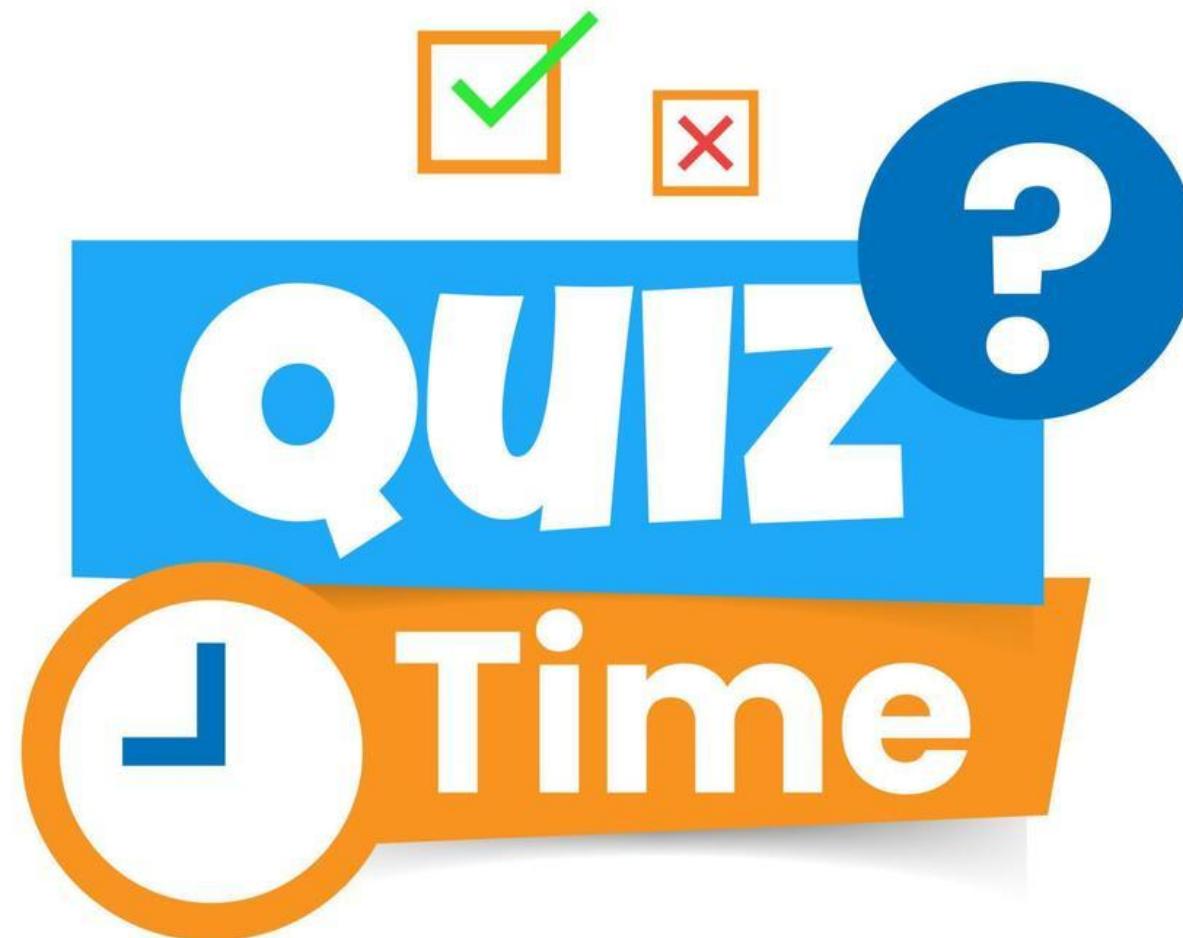
# Supervised Machine Learning at a glance!





You are a data scientist at SocialNet, a fictitious social media company similar to Facebook. The company receives millions of comments daily and wants to classify them automatically into categories such as "positive", "negative", or "neutral". Which approach would be best suited to handle the immense volume and variability of these comments?

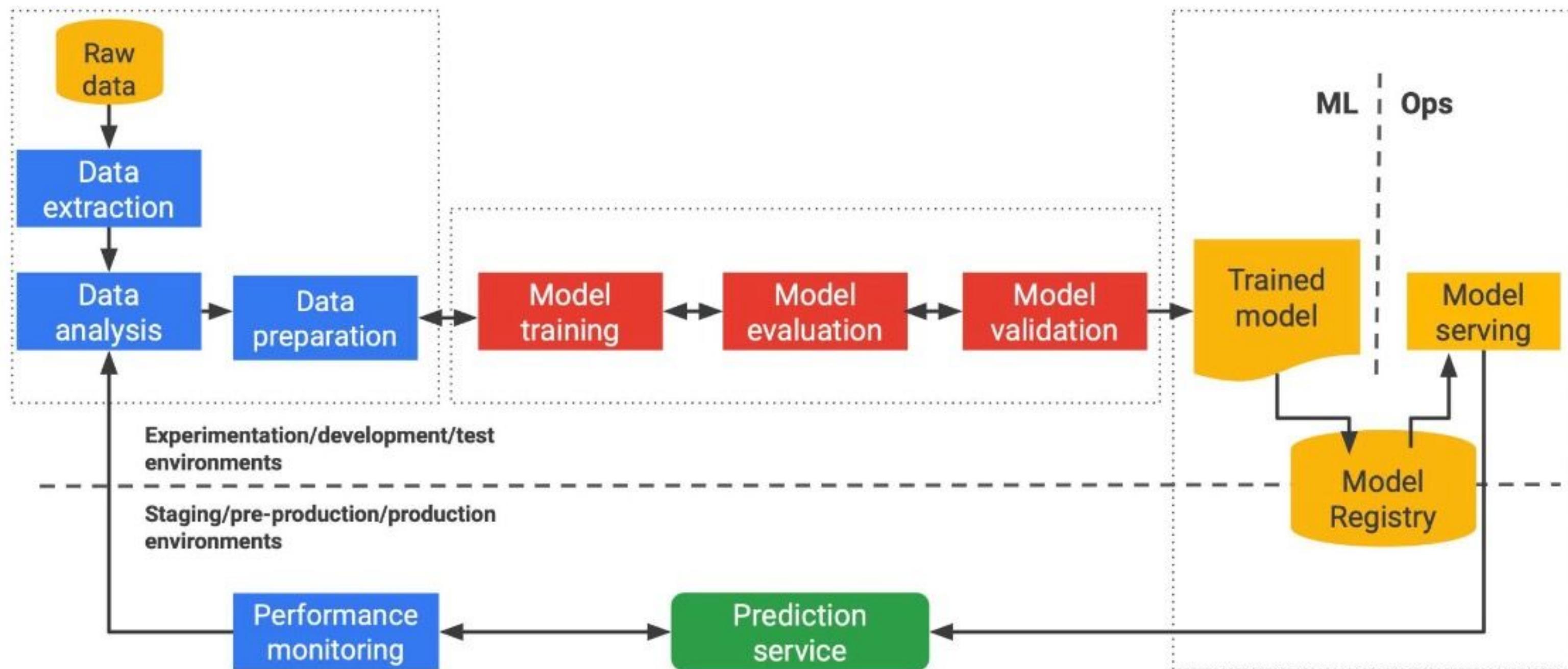
A	Designing a conventional programming algorithm that classifies comments based on predefined rules and keywords.
B	Implementing a machine learning based system that can be trained on labeled examples and adapt to new patterns over time.



You are a data scientist at SocialNet, a fictitious social media company similar to Facebook. The company receives millions of comments daily and wants to classify them automatically into categories such as "positive", "negative", or "neutral". Which approach would be best suited to handle the immense volume and variability of these comments?

A	Designing a conventional programming algorithm that classifies comments based on predefined rules and keywords.
B	Implementing a machine learning based system that can be trained on labeled examples and adapt to new patterns over time.

# Machine Learning Pipeline



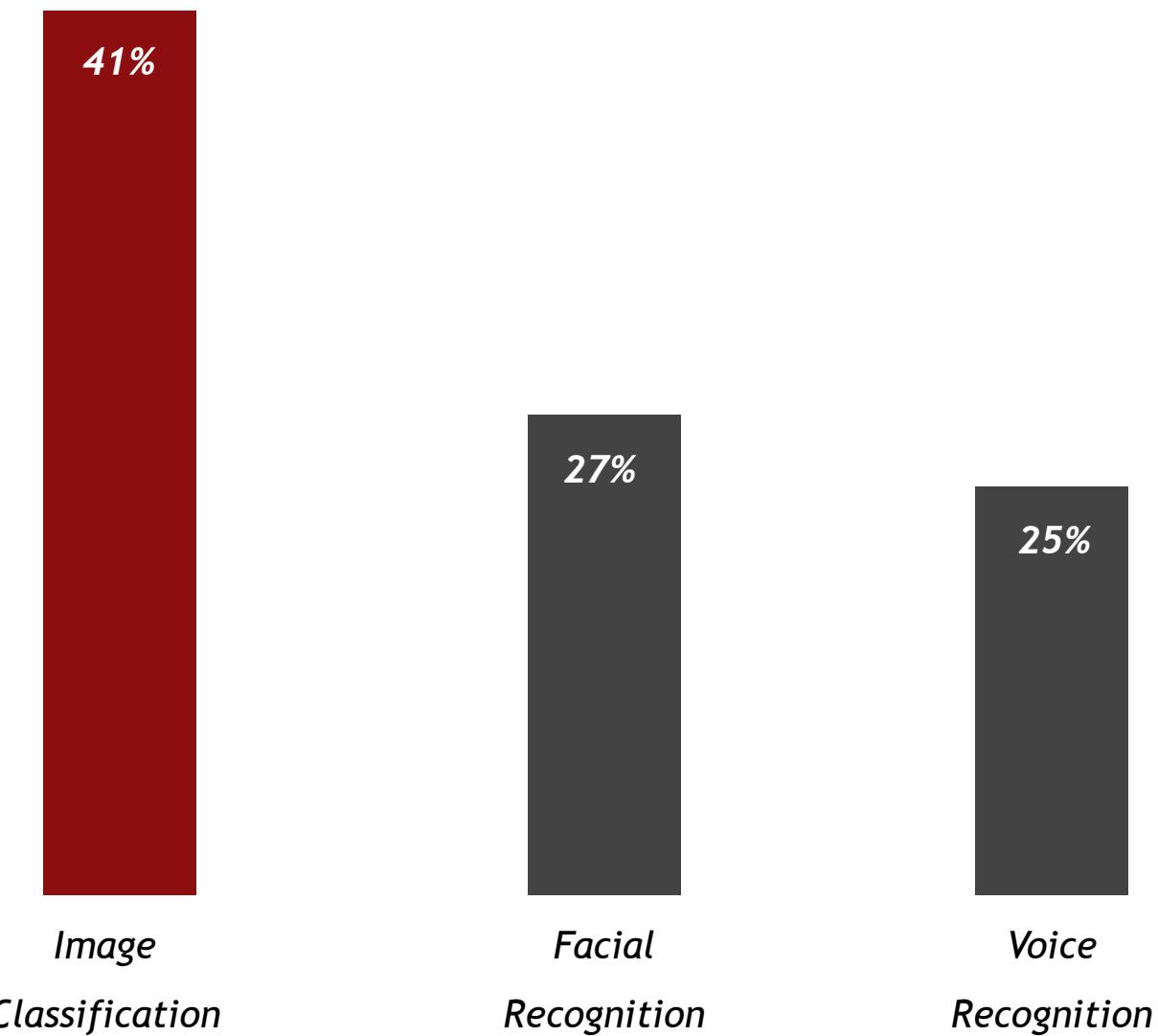
# Machine learning use cases include:

- Image recognition
- Sentiment analysis
- Speech recognition
- Fraud detection
- Customer segmentation
- Recommendation systems
- Content Generation
- Text Summarization

# Deep Learning

**Deep Learning often outperforms traditional ML methods**

*% reduction in error rate achieved by deep learning vs. traditional ML methods*

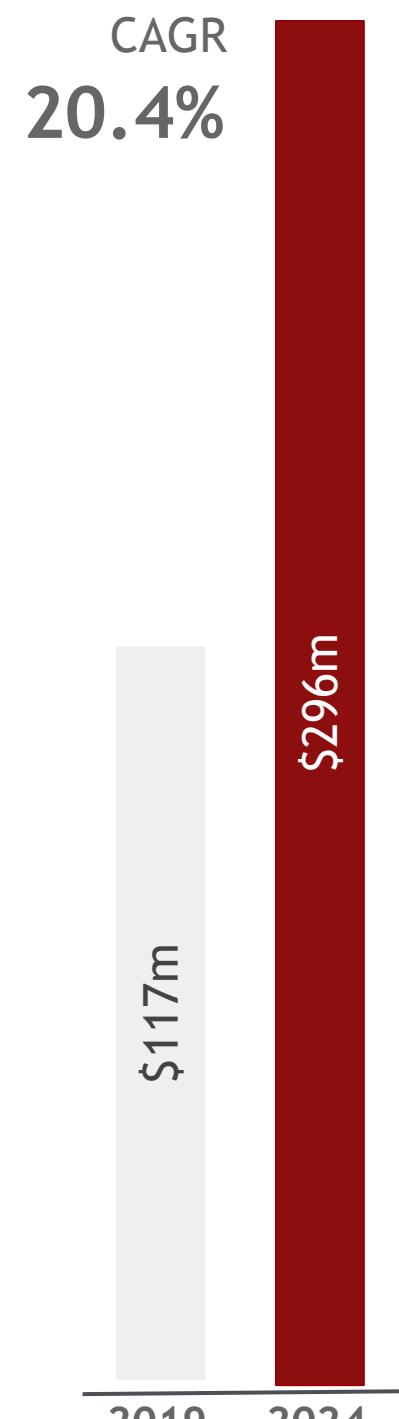


Source: Mckinsey

Deep Learning can find complex patterns from  
the data and produce more accurate results  
than traditional ML approaches

# Deep Learning - Artificial Neural Network

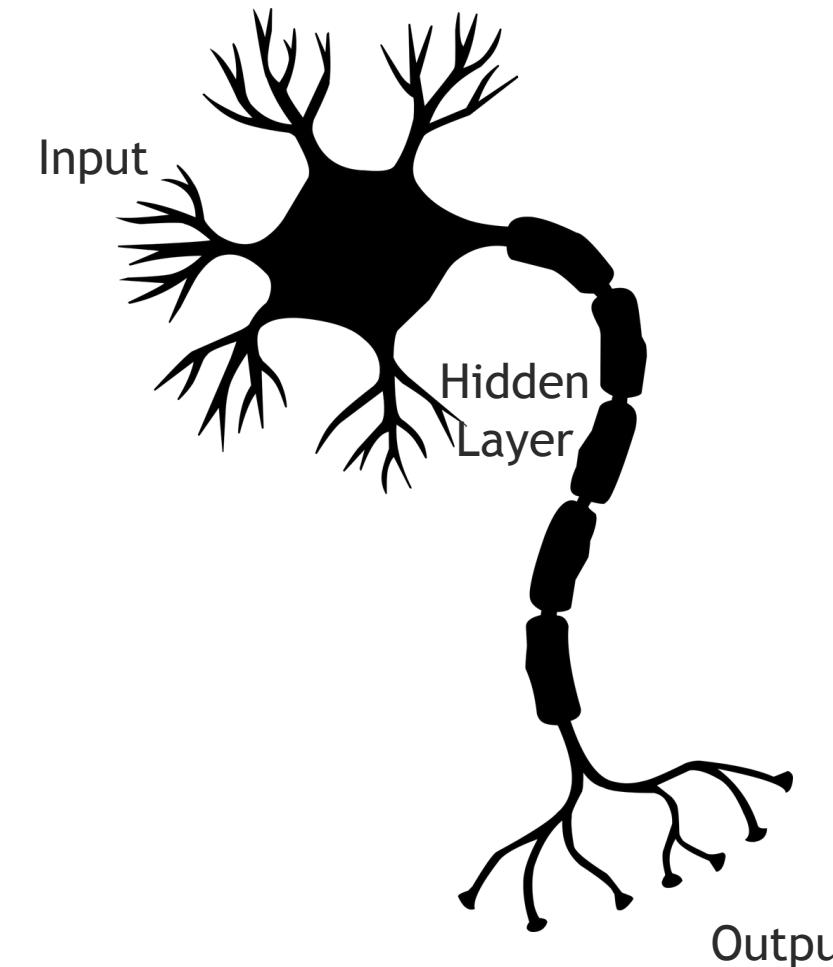
The Global market for ANN applications is projected to grow from \$177m to \$296m by 2024, at a CAGR of 20.4%.



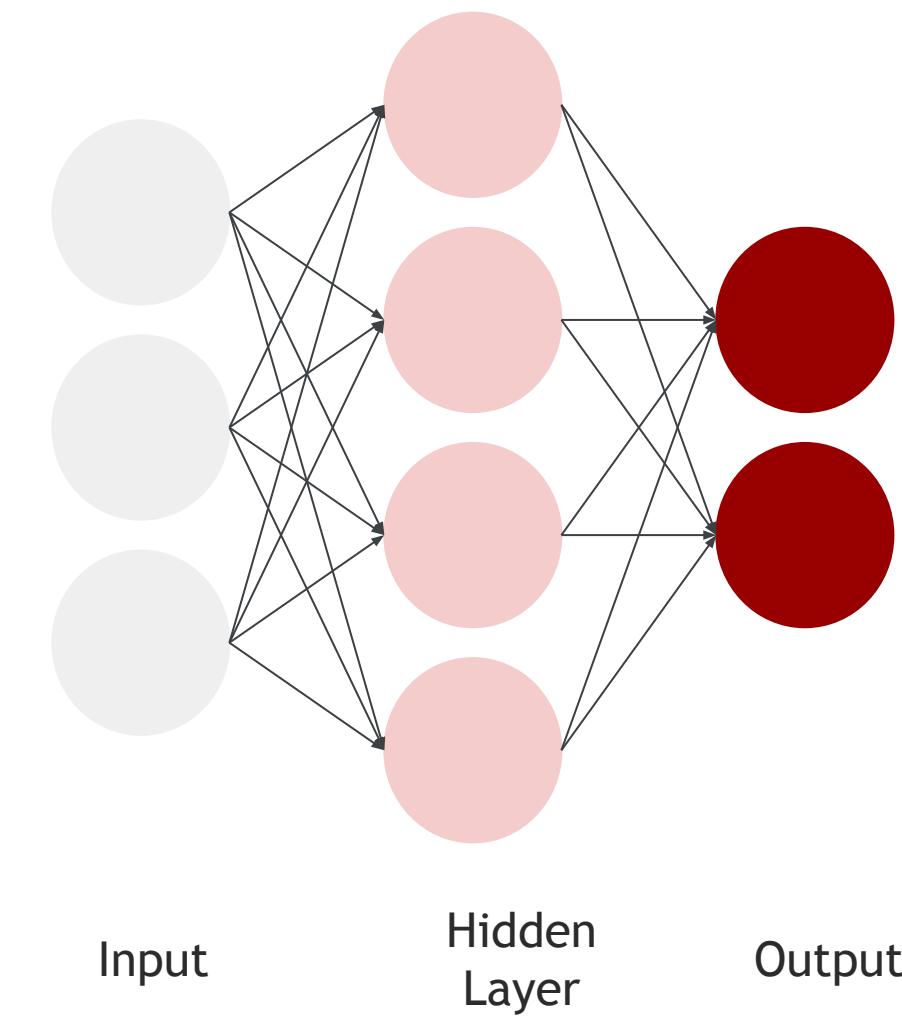
Source:  
MarketsAndMarkets

Artificial Neural Networks are a very rough imitation of the brain's structure

Human Brain Neuron

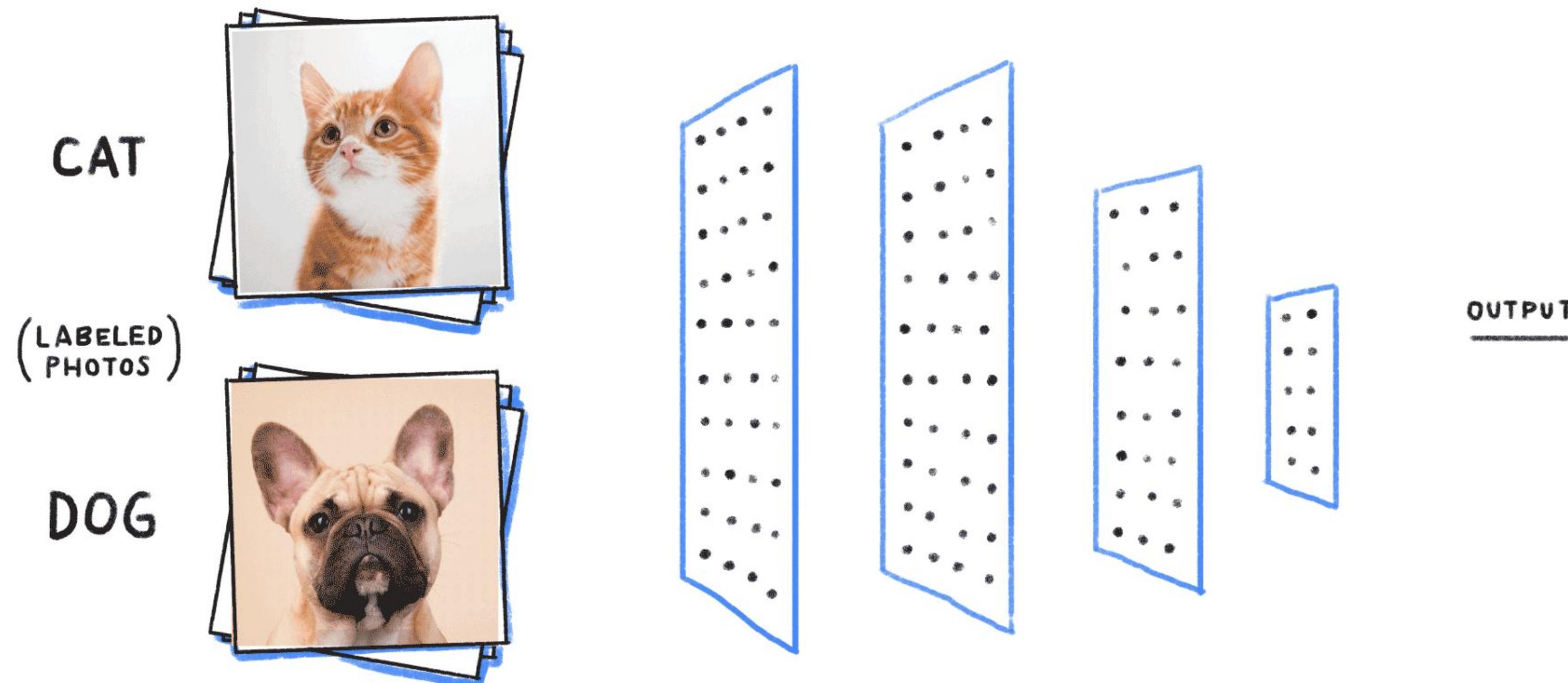


Artificial Neural Network

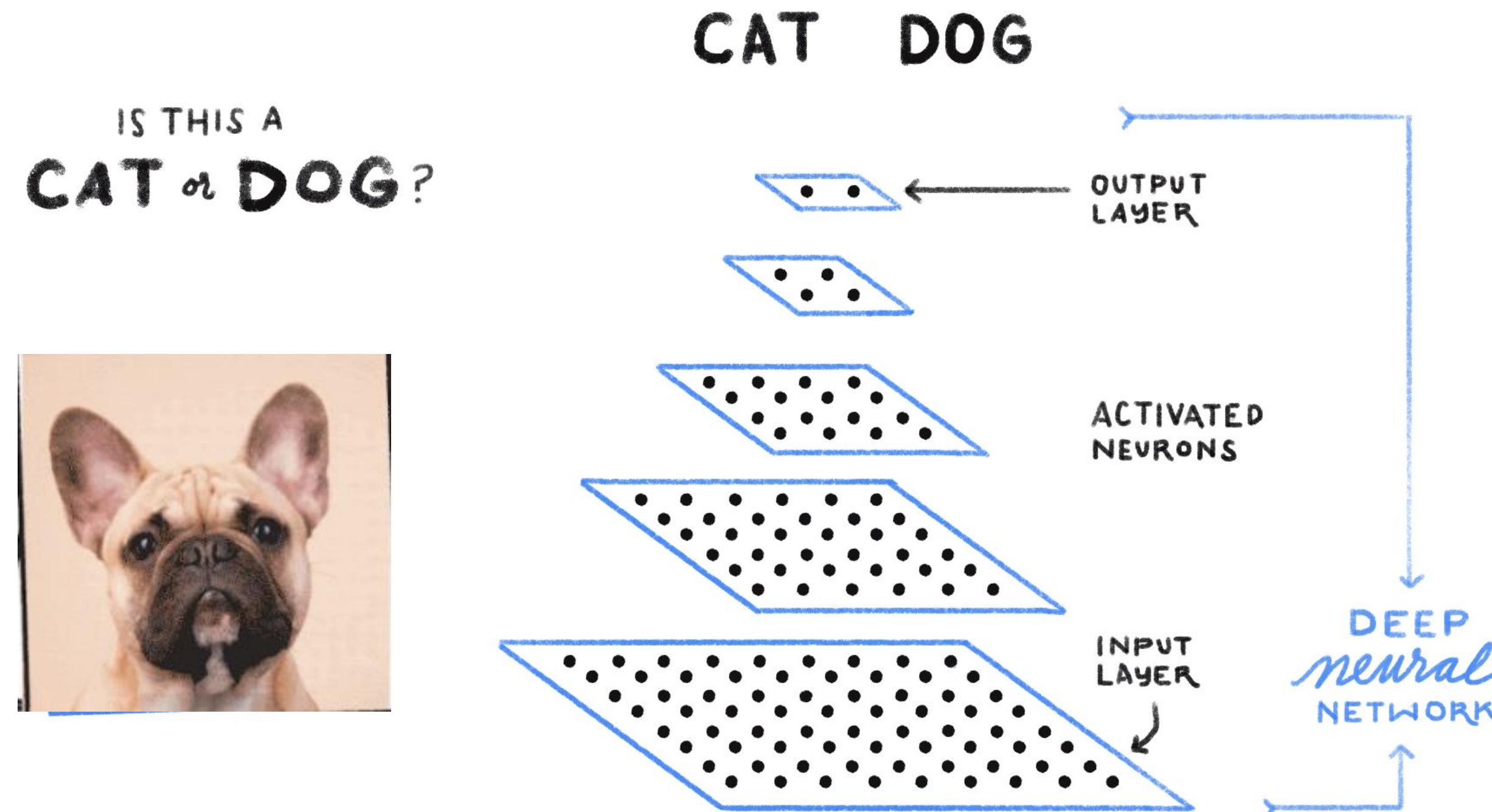


# Deep Learning - Artificial Neural Network

“Profound Learning and Artificial Neural Networks (ANN) have fueled the adoption of AI in several industries, such as aerospace, healthcare, manufacturing, and automotive. ANN is substituting conventional machine learning systems to evolve precise and accurate versions” - Grand View Research



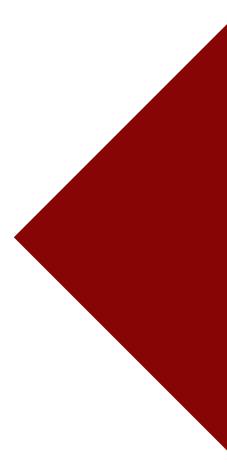
# Deep Learning - Artificial Neural Network



# Ways to perform AI/ML

Requirements, Skillset, Compute, & Availability of Data influence the choice of approach

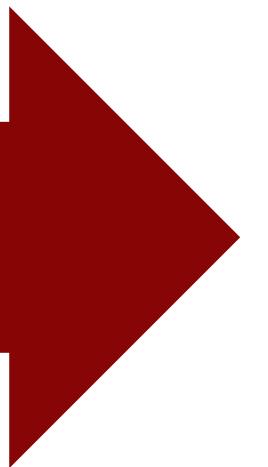
*Less Flexible*



*Easiest Approach*

Cognitive API

*More Flexible*

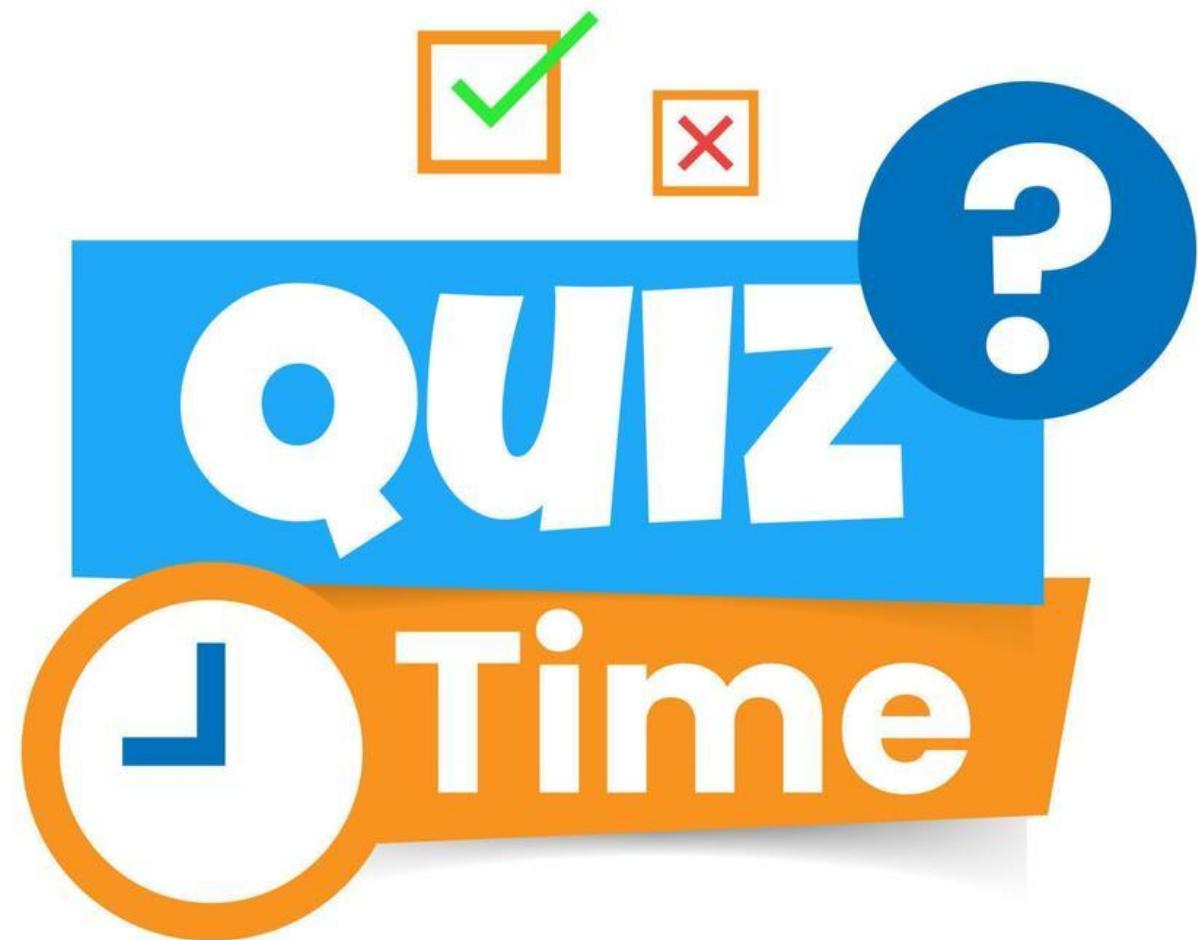


*Most complex Approach*

Auto ML

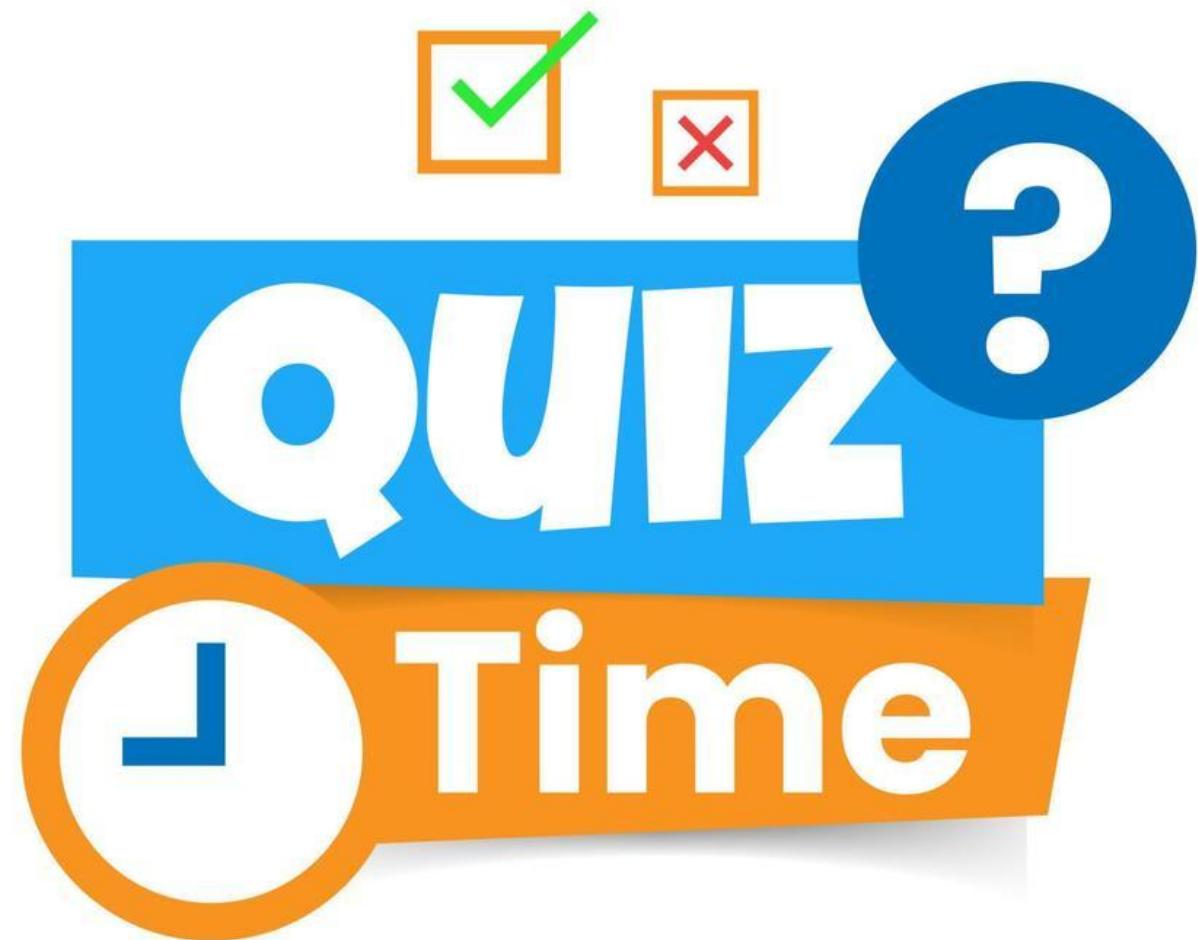
Transfer Learning

Build your own  
Model (BYOM)



You are the CTO of UrbanDrive, a ride-hailing company that operates in several large cities around the world. With traffic congestion becoming a major issue in urban areas, you are considering leveraging AI to optimize routes for your drivers, reduce commute times, and enhance the overall customer experience. Which of the following AI-based solutions would be the most effective in achieving these goals?

- |   |  |
|---|--|
| A | An AI chatbot that can communicate with drivers and provide them with traffic updates.                       |
| B | A rule-based system that suggests routes based on fixed traffic patterns and historical data.                |
| C | A deep learning model that predicts real-time traffic congestion and dynamically adjusts routes for drivers. |
| D | A recommendation system that suggests drivers the best times to work based on previous trip data.            |



You are the CTO of UrbanDrive, a ride-hailing company that operates in several large cities around the world. With traffic congestion becoming a major issue in urban areas, you are considering leveraging AI to optimize routes for your drivers, reduce commute times, and enhance the overall customer experience. Which of the following AI-based solutions would be the most effective in achieving these goals?

- |   |  |
|---|--|
| A | An AI chatbot that can communicate with drivers and provide them with traffic updates.                       |
| B | A rule-based system that suggests routes based on fixed traffic patterns and historical data.                |
| C | A deep learning model that predicts real-time traffic congestion and dynamically adjusts routes for drivers. |
| D | A recommendation system that suggests drivers the best times to work based on previous trip data.            |



# Microsoft Cognitive APIs

- Useful when you want to use pre-built models by Azure
- Make a REST Based API call to invoke it
- No Machine Learning Skills Required
- Least Flexible Option

Example - Azure Vision API

<https://azure.microsoft.com/en-in/services/cognitive-services/computer-vision/>

# Why Microsoft Cognitive Services?



Easy

Roll your own with REST APIs  
Simple to add: just a few  
lines of code required



Flexible

Integrate into the language  
and platform of your choice  
Breadth of offerings helps you  
find the right API for your app



Tested

Built by experts in their field  
from Microsoft Research, Bing,  
and Azure Machine Learning  
Quality documentation, sample  
code, and community support



# Microsoft Cognitive Services

## Give your apps a human side



### Vision

From faces to feelings, allow your apps to understand images and video

### Speech

Hear and speak to your users by filtering noise, identifying speakers, and understanding intent

### Language

Process text and learn how to recognize what users want

### Knowledge

Tap into rich knowledge amassed from the web, academia, or your own data

### Search

Access billions of web pages, images, videos, and news with the power of Bing APIs

### Labs

An early look at emerging Cognitive Services technologies: discover, try and give feedback on new technologies before general availability



# Vision



## Computer Vision API

Distill actionable information from images



## Face API

Detect, identify, analyze, organize, and tag faces in photos



## Emotion API

Personalize experiences with emotion recognition



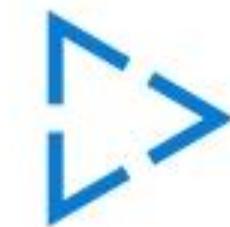
## Content Moderator

Machine-assisted moderation of text and images, augmented with human review tools



## Custom Vision Service

Customizable web service that learns to recognize specific content in imagery



## Video Indexer

Process and extract smart insights from videos



# VISION

From faces to feelings, allow your apps to understand images and video

Computer Vision | Content Moderator | Emotion | Face | Video Indexer | Custom Vision Service



# Computer Vision API

## Analyze an image

Understand content within an image

## OCR

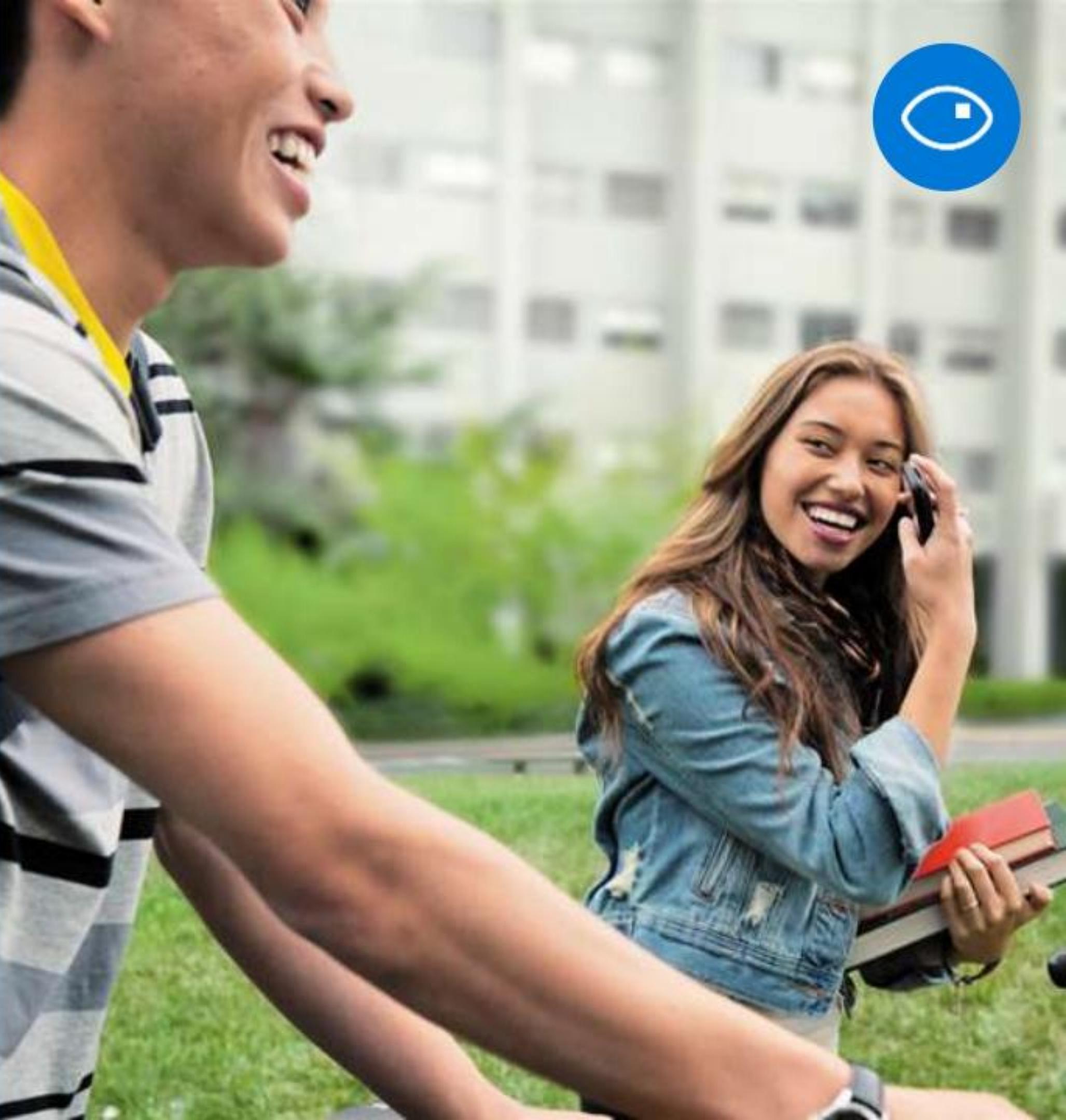
Detect and recognize words within an image

## Generate thumbnail

Scale and crop images, while retaining key content

## Recognize celebrities

Thanks to domain specific models, ability to recognize 200K celebrities from business, politics, sports and entertainment around the world



# Analyze image



## Type of image

Clip Art Type	0 Non-clipart
Line Drawing Type	0 Non-Line Drawing
Black & White Image	False

## Content of image

Categories	[{"name": "people_swimming", "score": 0.099609375}]
Adult Content	False
Adult Score	0.18533889949321747
Faces	[{"age": 27, "gender": "Male", "faceRectangle": {"left": 472, "top": 258, "width": 199, "height": 199}}]

## Image colors

Dominant Color Background	White
Dominant Color Foreground	Grey
Dominant Colors	White
Accent Color	



# Emotion API

**Recognize  
emotions**

Understand content  
within an image



# Emotion API

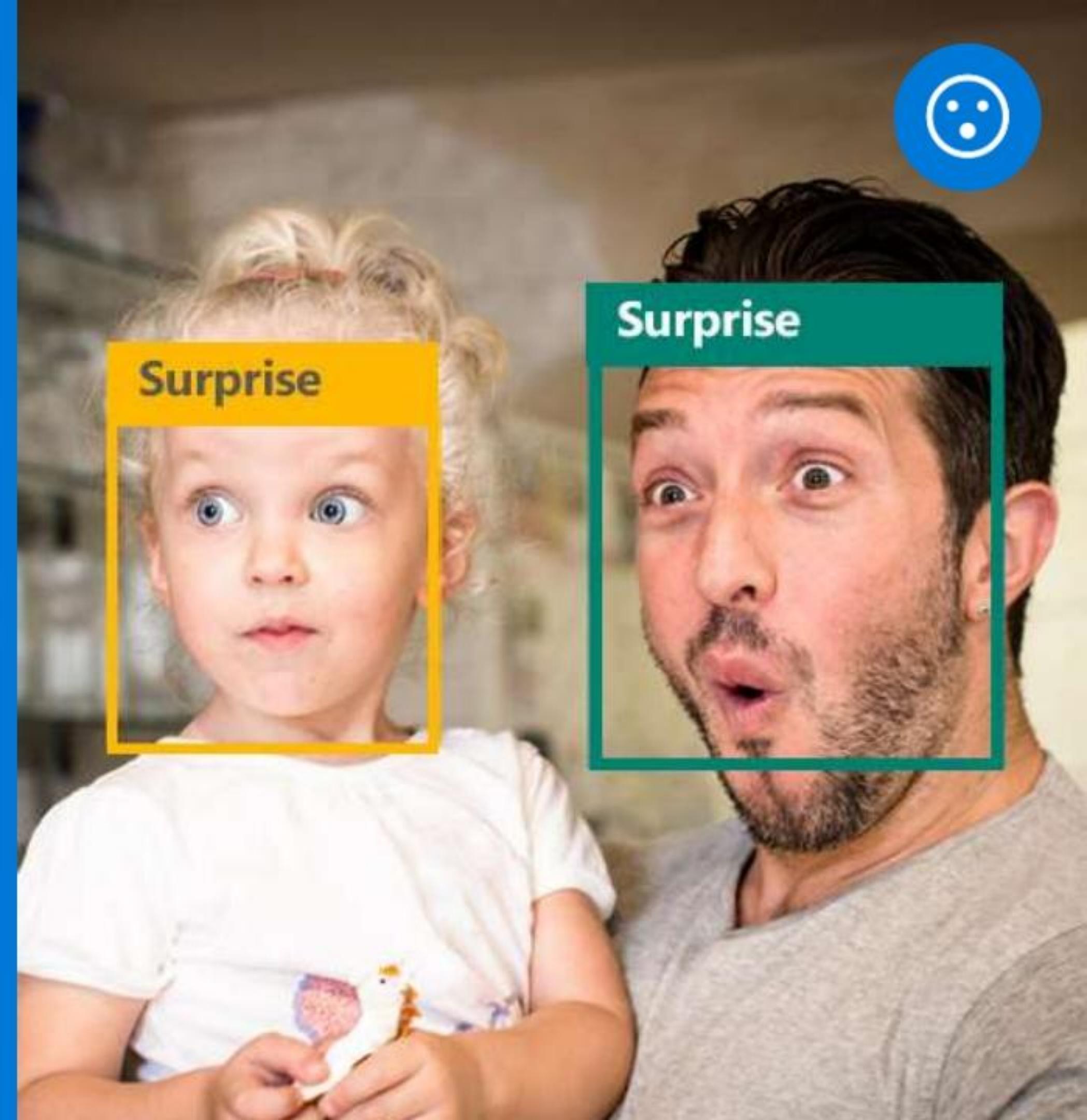


## Face detection

```
"faceRectangle": {"width": 193,  
                 "height": 193,  
                 "left": 326,  
                 "top": 204} ...
```

## Emotion scores

```
"scores": { "anger": 5.182241e-8,  
            "contempt": 0.0000242813,  
            "disgust": 5.621025e-7,  
            "fear": 0.00115027453,  
            "happiness": 1.06114619e-8,  
            "neutral": 0.003540177,  
            "sadness": 9.30888746e-7,  
            "surprise": 0.9952837}
```



# Face API



## Face detection

Detect faces and their attributes within an image

## Face verification

Check if two faces belong to the same person

## Similar face searching

Find similar faces within a set of images

## Face grouping

Organize many faces into groups

## Face identification

Search which person a face belongs to





# Face API

## Detection

```
"faceRectangle": {"width": 193, "height": 193,  
"left": 326, "top": 204}
```

...

## Feature attributes

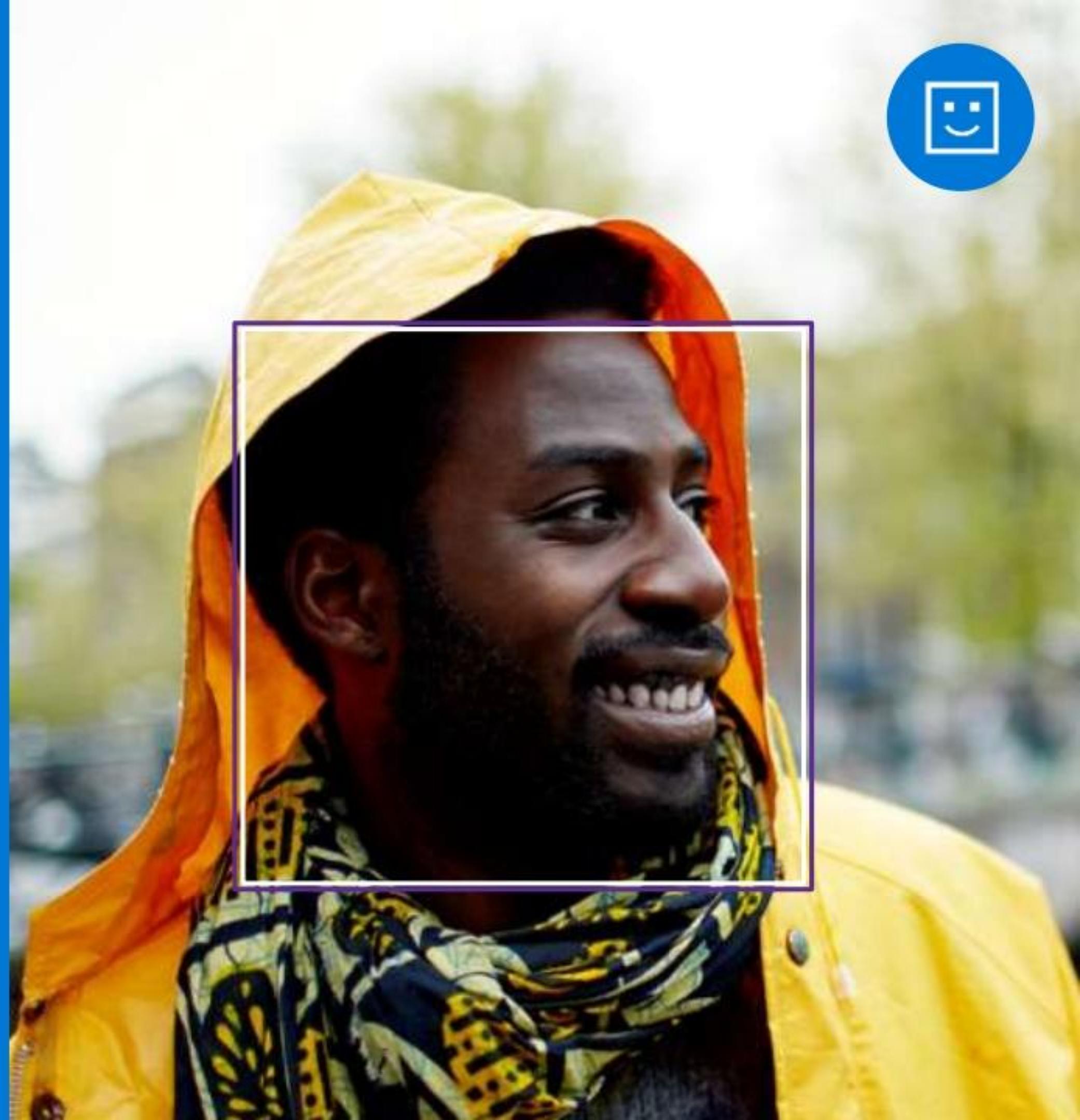
```
"attributes": { "age": 42, "gender": "male",  
"headPose": { "roll": "8.2", "yaw": "-37.8",  
"pitch": "0.0" }}
```

## Grouping



## Identification

Jasper Williams



# Content Moderator

Machine-assisted moderation of text and images, augmented with human review tools

## Image moderation

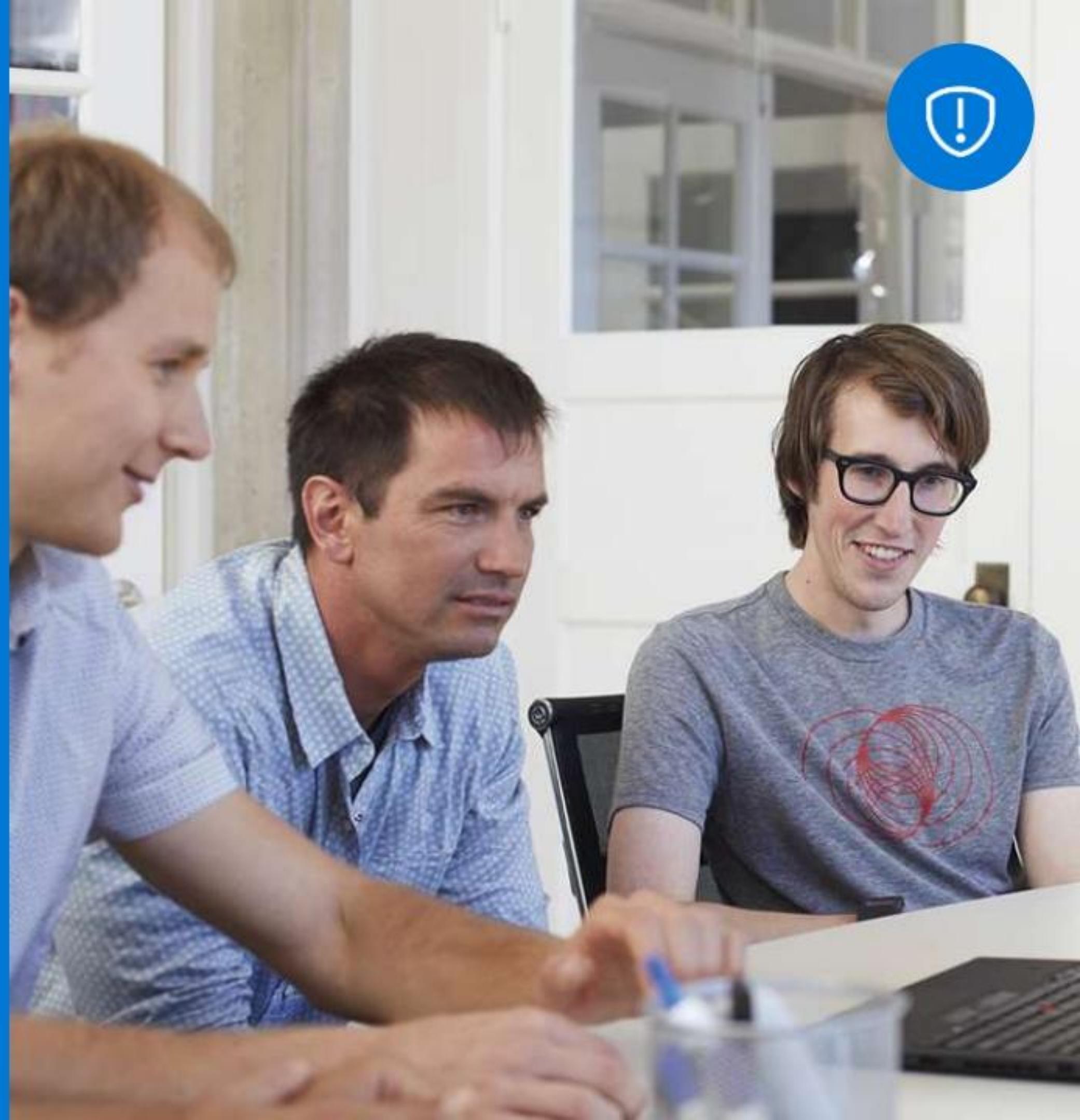
Enhance your ability to detect potentially offensive or unwanted images through machine-learning based classifiers, custom blacklists, and Optical Character Recognition (OCR)

## Text moderation

Helps you detect potential profanity in more than 100 languages and match text against your custom lists automatically. Content Moderator also checks for possible Personally Identifiable Information (PII)

## Video moderation (in Azure Media Services)

Enable the scoring of possible adult content in videos. Video moderation is currently deployed in preview on Azure Media Services



99

# LANGUAGE

Process text and learn how to recognize  
what users want

Bing Spell Check | Language Understanding |  
Linguistic Analysis | Text Analytics | Web Language Model |  
Translator Text and Speech

# Bing spell check API



## **State-of-the-art cloud-based spelling algorithms**

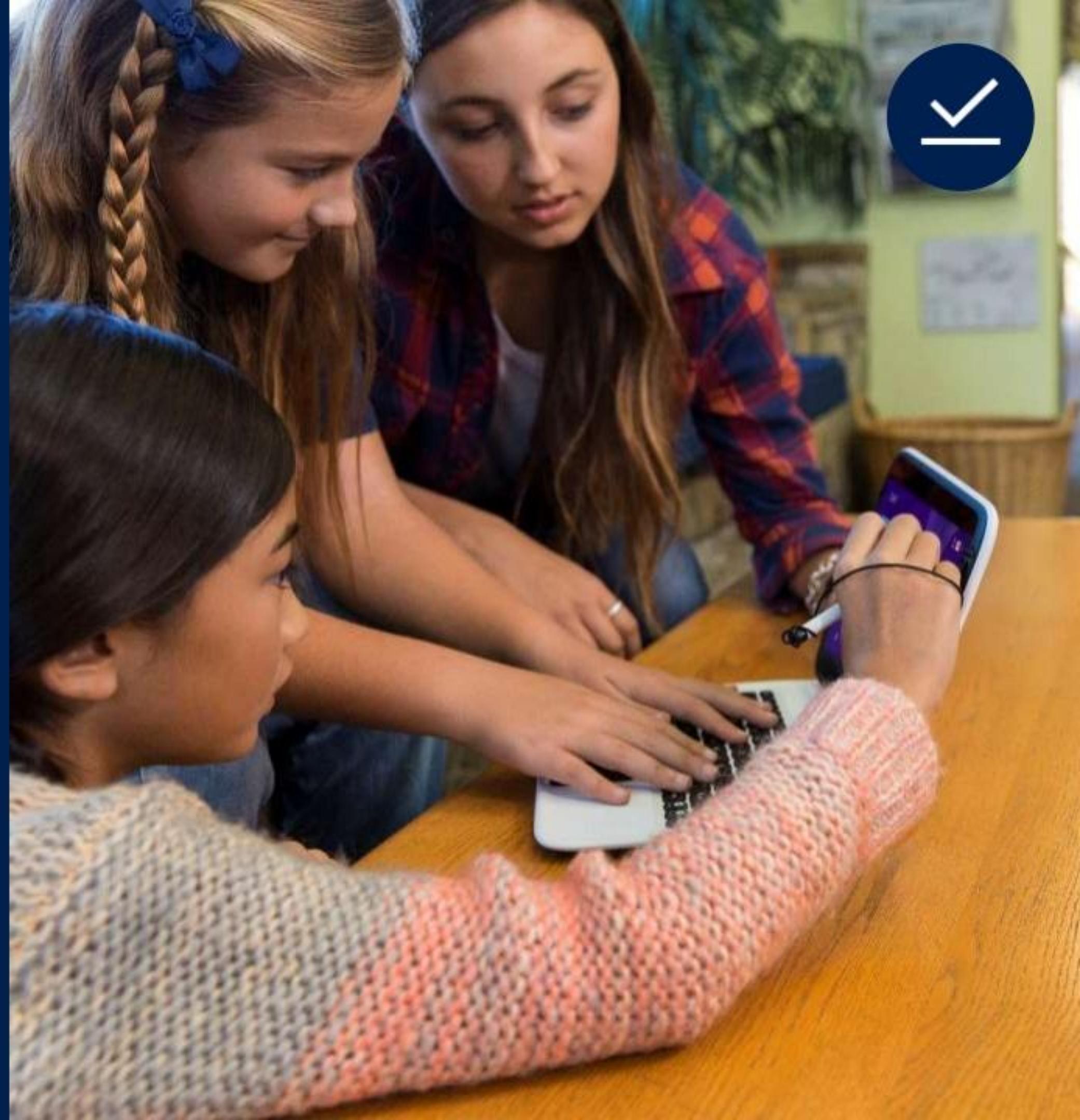
Recognizes a wide variety of spelling errors

## **Recognize name errors and homonyms in context**

Difficult to spot errors that use the context  
of the words around them

## **Updates over time**

Support for new brands and coined  
expressions as they emerge



# Bing spell check API



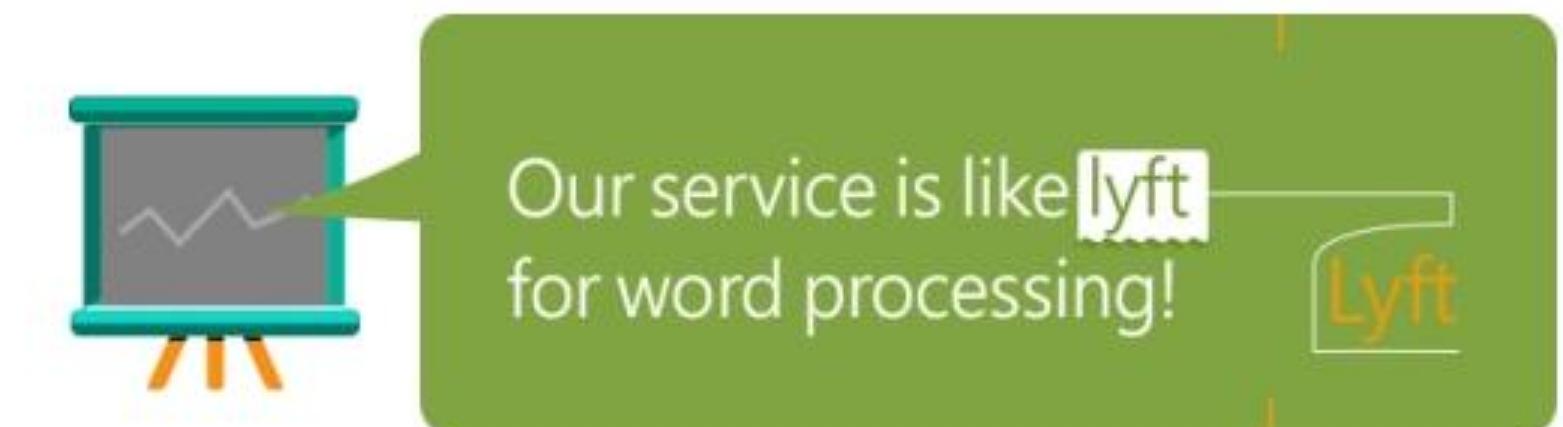
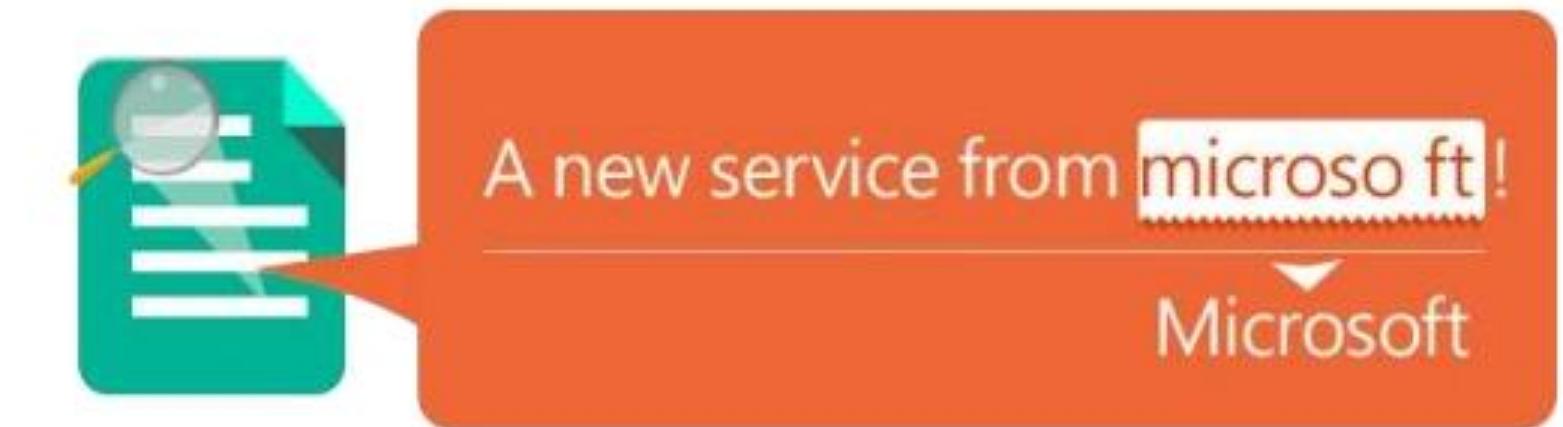
Check a single word or a whole sentence

"Our engineers developed this **four** you!"

Corrected Text: "four" → "for"

Identify errors & get suggestions

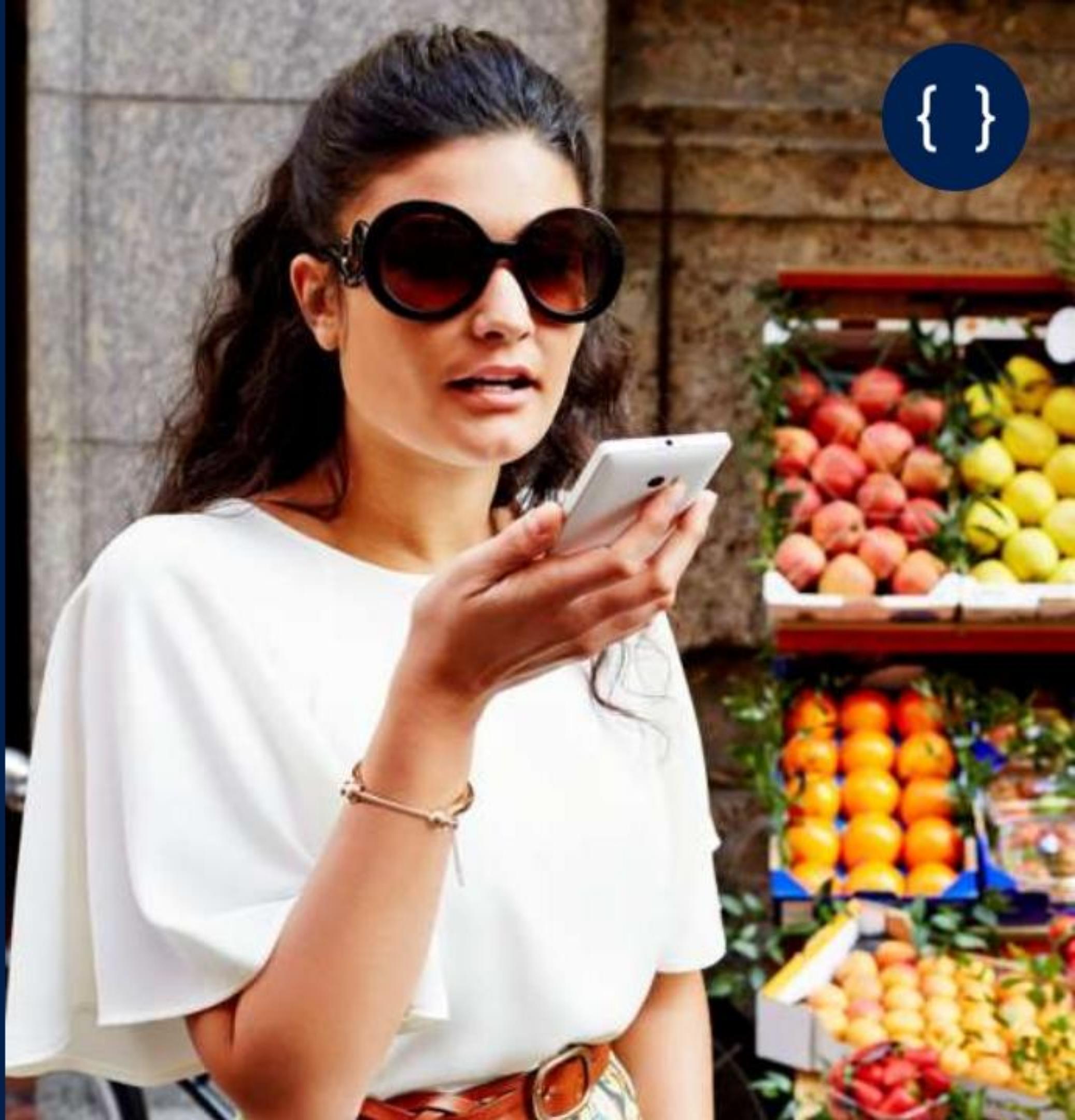
```
"spellingErrors": [  
    {  
        "offset": 5,  
        "token": "gona",  
        "type": "UnknownToken",  
        "suggestions": [  
            { "token": "gonna" }  
        ] }]
```



# Language Understanding Intelligent Service

**Understand what  
your users are saying**

Use pre-built Bing and Cortana  
models or create your own





# Text analytics

## Sentiment analysis

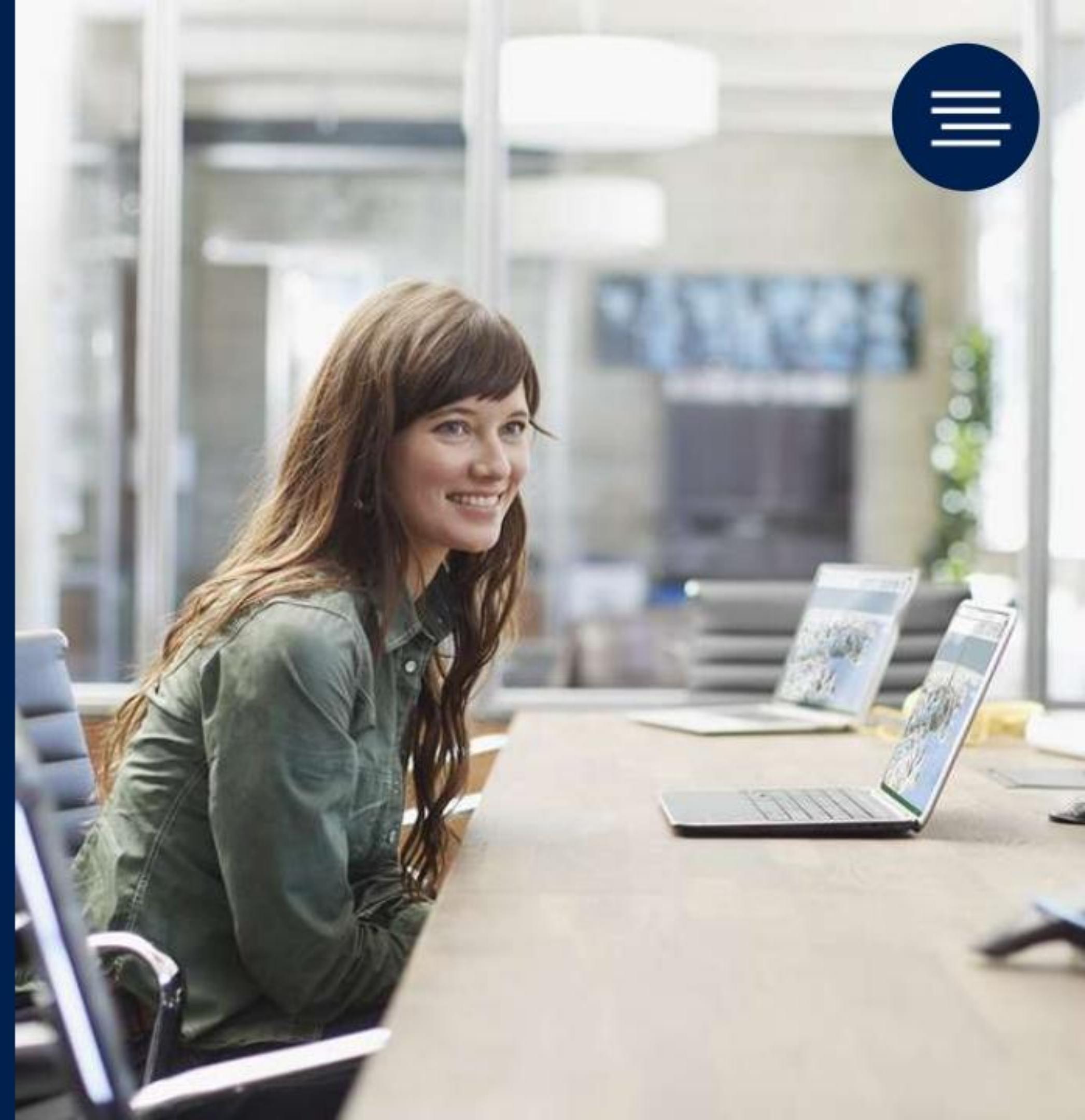
Understand if a record has positive or negative sentiment

## Key phrase extraction

Extract key phrases from a piece of text, and retrieve topics

## Language detection

Identify the language, 120 supported languages



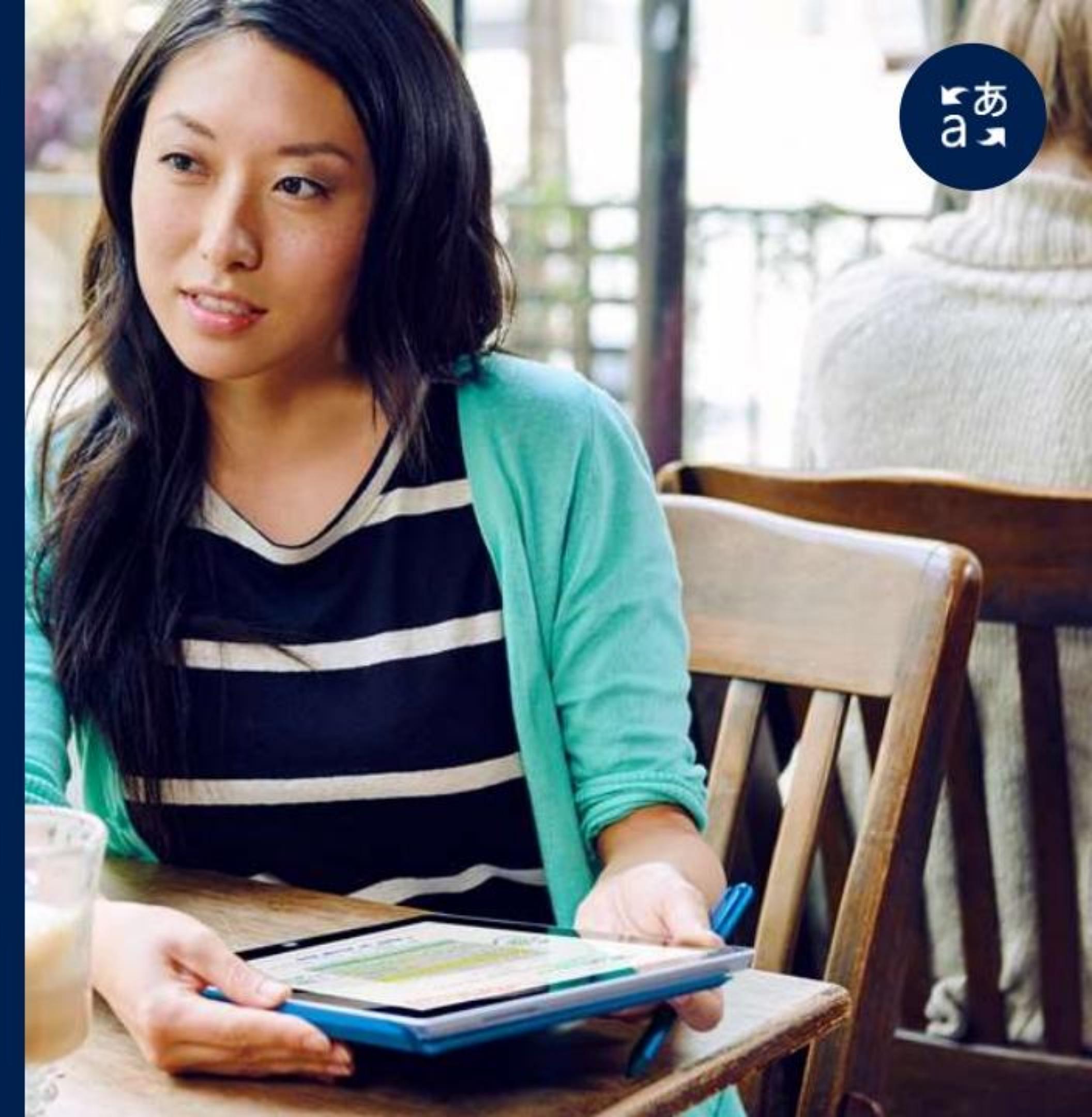
# Microsoft Translator

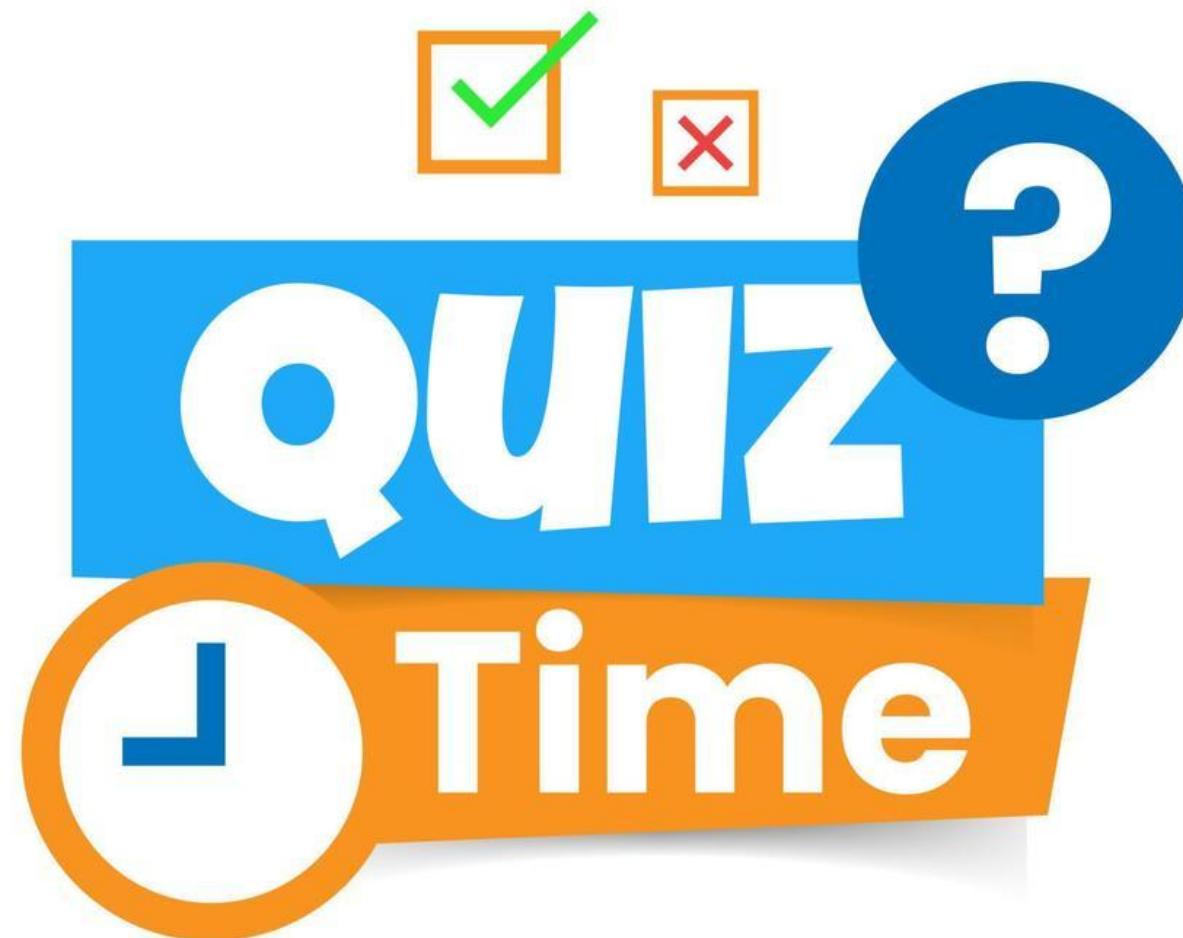
## Translator Text API

Automatically detect language  
and easily power translation to and  
from 60 supported text languages

## Translator Speech API

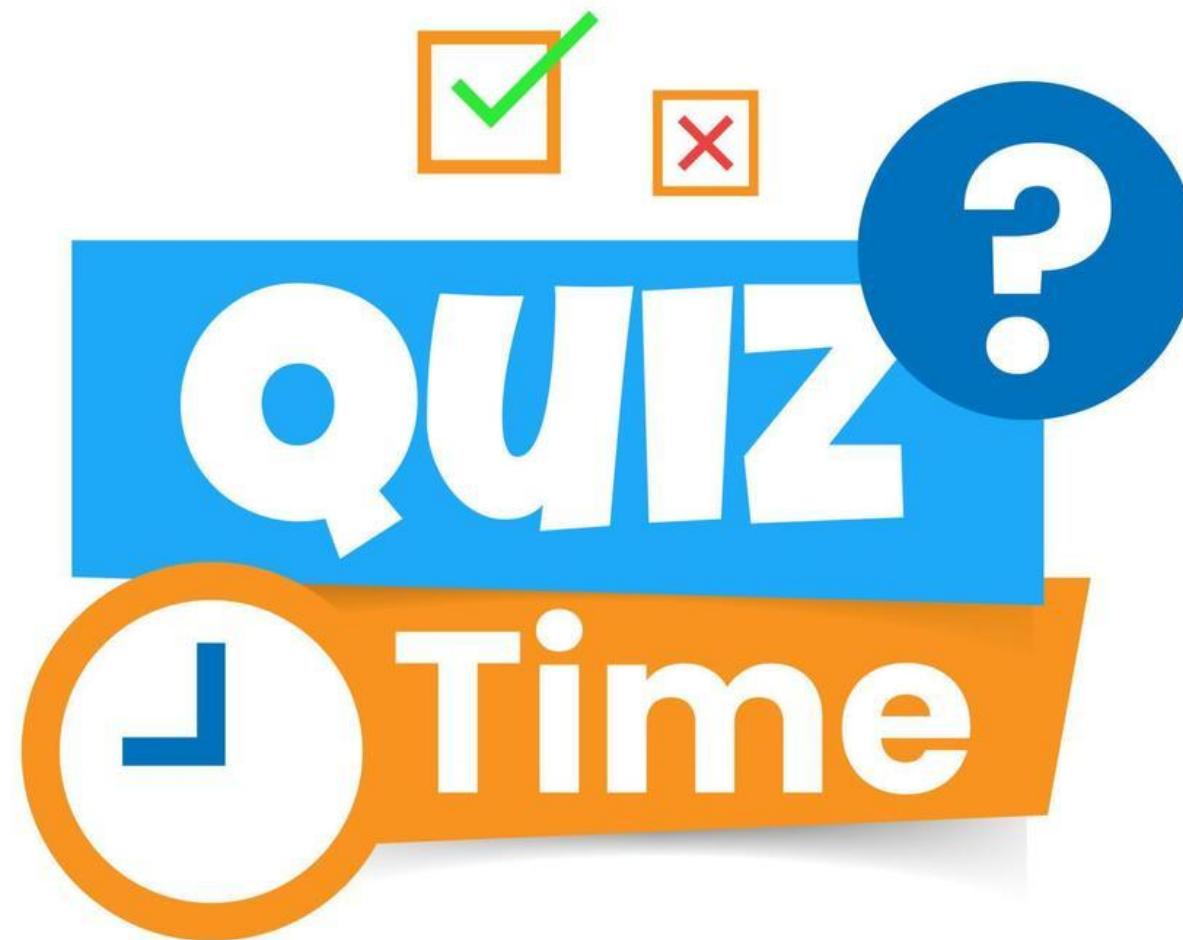
Easily translate real-time speech  
conversations in 9 support languages





You are the IT director of ArtMuseum Inc., a prominent art institution that plans to digitize its vast collection of historical artworks. To provide an interactive and rich experience for online visitors, you're considering leveraging Azure Cognitive Services. Which of the following use cases would be most impactful for the museum using Azure's AI capabilities?

<b>A</b>	Using Azure Computer Vision to automatically tag and categorize artworks based on their content, making it easier for visitors to search and discover related pieces.
<b>B</b>	Deploying Azure Text Analytics on historical documents to automatically translate them into multiple languages for international visitors.
<b>C</b>	Implementing Azure Personalizer to provide art recommendations to visitors based on their browsing behavior and preferences.
<b>D</b>	Utilizing Azure QnA Maker to set up an automated FAQ section for the museum's website, answering general questions about opening hours and ticket prices.



You are the IT director of ArtMuseum Inc., a prominent art institution that plans to digitize its vast collection of historical artworks. To provide an interactive and rich experience for online visitors, you're considering leveraging Azure Cognitive Services. Which of the following use cases would be most impactful for the museum using Azure's AI capabilities?

<input checked="" type="checkbox"/>	<b>A</b>	Using Azure Computer Vision to automatically tag and categorize artworks based on their content, making it easier for visitors to search and discover related pieces.
<input type="checkbox"/>	<b>B</b>	Deploying Azure Text Analytics on historical documents to automatically translate them into multiple languages for international visitors.
<input type="checkbox"/>	<b>C</b>	Implementing Azure Personalizer to provide art recommendations to visitors based on their browsing behavior and preferences.
<input type="checkbox"/>	<b>D</b>	Utilizing Azure QnA Maker to set up an automated FAQ section for the museum's website, answering general questions about opening hours and ticket prices.

# Today's Topics

- 01 Understanding the fundamentals
- 02 Generative AI & it's usecases
- 03 An Introduction to LLMs
- 04 Github Co-Pilot



# Generative AI

An overview of what generative AI is and its significance in the field of artificial intelligence

---

- Generative AI is a type of artificial intelligence that can create new content, such as text, images, or music
- It works by learning from large datasets of existing content and then using that knowledge to generate new content that is similar to the training data



<https://www.thispersondoesnotexist.com>

# Generative AI for Developers

Write code faster, Solve more complicated problems, and Build experiences not possible before

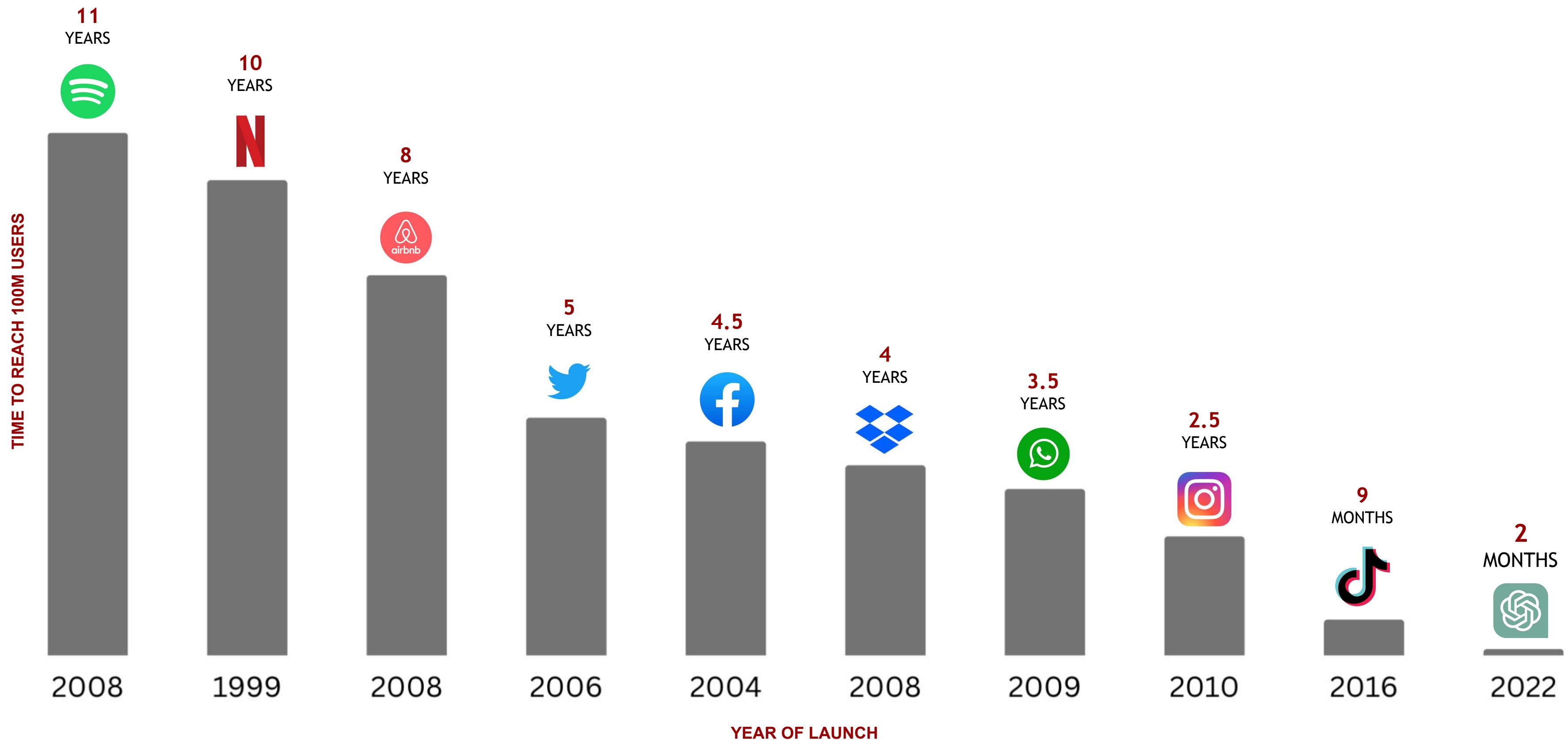
## How much of your work involves these things?

- **Finding Information** - Googling errors, Debugging
- **Explaining Concepts** - Learning new technologies
- **Generating and Fixing Code** - Writing scripts, tests, functions, components etc
- **Building new applications** - Natural language processing, speech to text

## So, it's great at...

- *Finding Information*
- *Explaining Concepts*
- *Generating and Fixing Code*
- *Building new applications*

# A New Wave!



# A New Wave!

 Kent Beck 🌸  
@KentBeck

I've been reluctant to try ChatGPT. Today I got over that reluctance. Now I understand why I was reluctant.

The value of 90% of my skills just dropped to \$0. The leverage for the remaining 10% went up 1000x. I need to recalibrate.

1:21 AM · Apr 19, 2023 · 1.4M Views

---

195  1,060  5,910  1,105  

# A New Wave!

## 90% of My Skills Are Now Worth \$0

...but the other 10% are worth 1000x



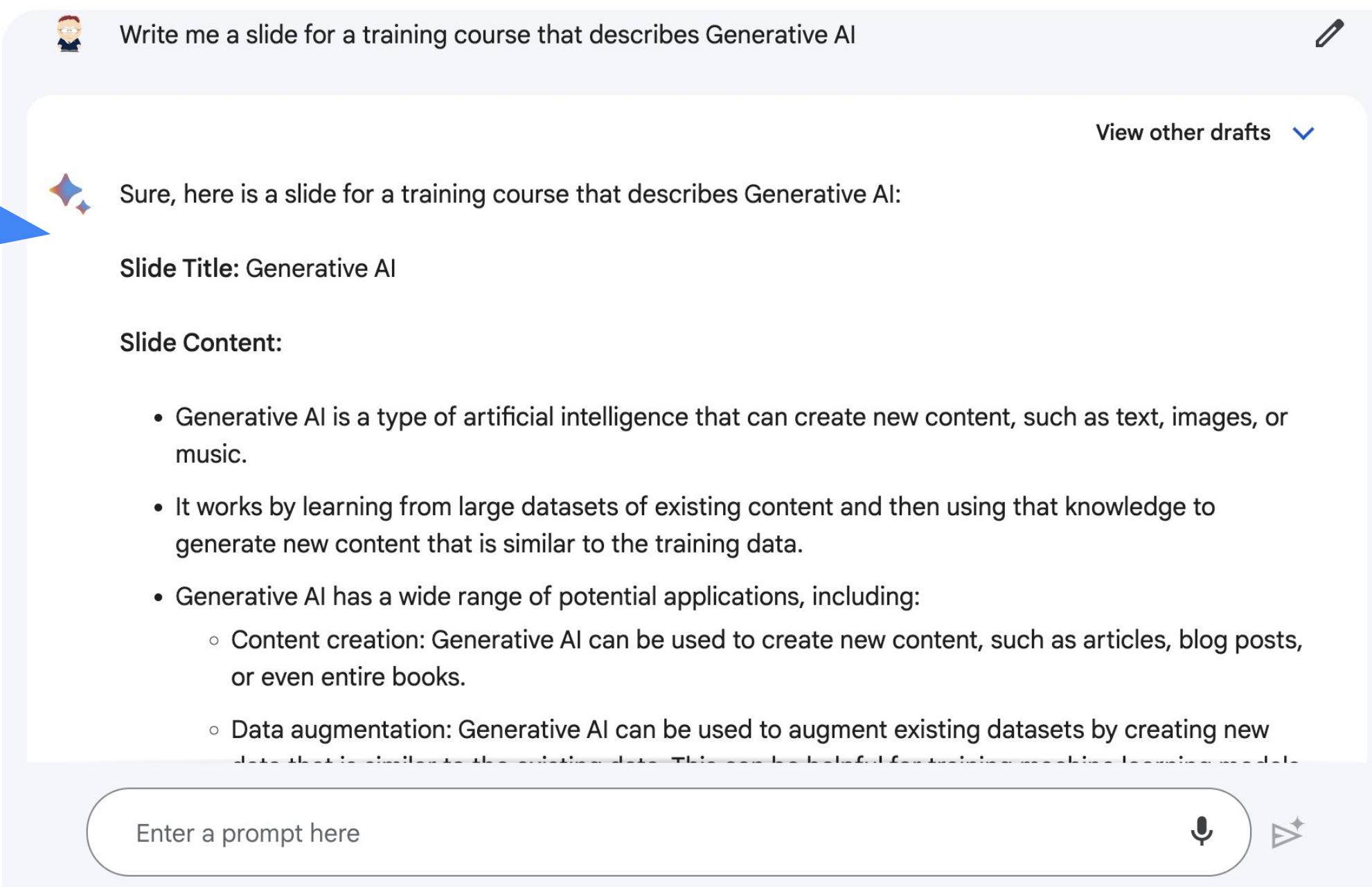
KENT BECK

19 APR 2023

In my experience, the most successful software development projects are those that are driven by a combination of human creativity and expertise, as well as the efficient use of tools like ChatGPT. **Rather than seeing the rise of AI technology as a threat to our skills, we should view it as an opportunity to enhance and augment our abilities as software developers.**

Rather than allowing our skills to become obsolete, we should focus on finding ways to integrate AI tools like ChatGPT into our work **in a way that complements our human expertise and creativity.**

# Bard is a Google consumer-oriented generative AI tool for creating custom text-based content



The slide!

Write me a slide for a training course that describes Generative AI

Sure, here is a slide for a training course that describes Generative AI:

Slide Title: Generative AI

Slide Content:

- Generative AI is a type of artificial intelligence that can create new content, such as text, images, or music.
- It works by learning from large datasets of existing content and then using that knowledge to generate new content that is similar to the training data.
- Generative AI has a wide range of potential applications, including:
  - Content creation: Generative AI can be used to create new content, such as articles, blog posts, or even entire books.
  - Data augmentation: Generative AI can be used to augment existing datasets by creating new

Enter a prompt here

View other drafts ▾

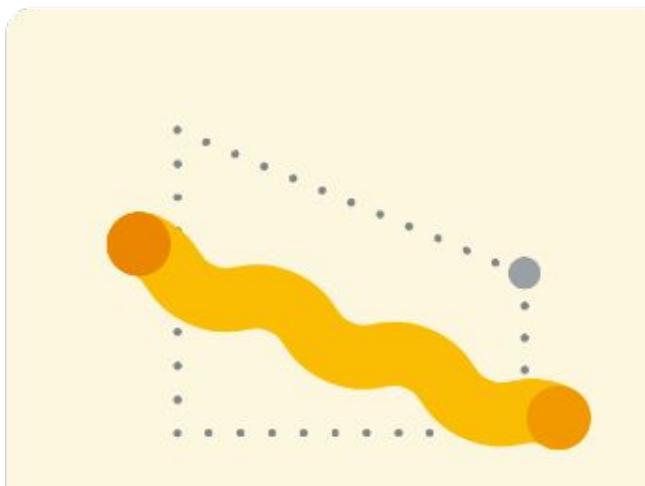
A blue box with the text "The slide!" is overlaid on the top left of the screenshot, with a blue arrow pointing from it towards the generated slide content.

<https://bard.google.com/>

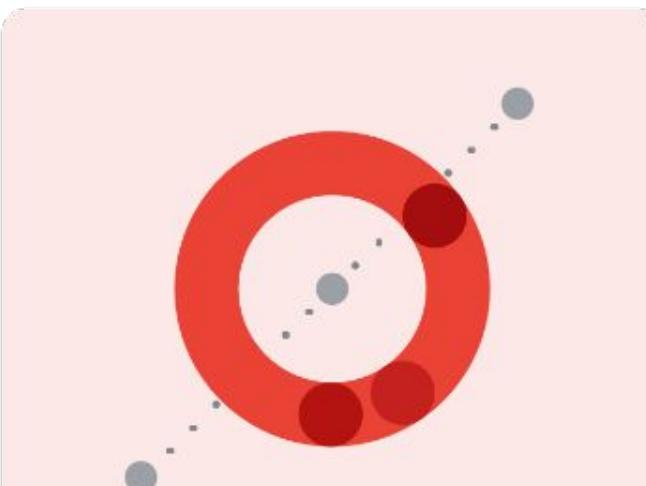
# There are huge benefits of generative AI



Increase in efficiency & productivity



Reduce costs for your organization



Automate monotonous tasks

# There are also challenges with generative AI

- Can be difficult to control the quality of generated content
- Can be difficult to ensure that generated content is accurate
  - Untrue statements can be presented in a confident manner
  - These are known as hallucinations in generative AI terms
- Can be difficult to ensure that generated content is not offensive or harmful



# Generative AI use cases with Azure AI



## Language

Writing  
Summarization  
Ideation  
Classification  
Sentiment analysis  
Extraction  
Customer chat

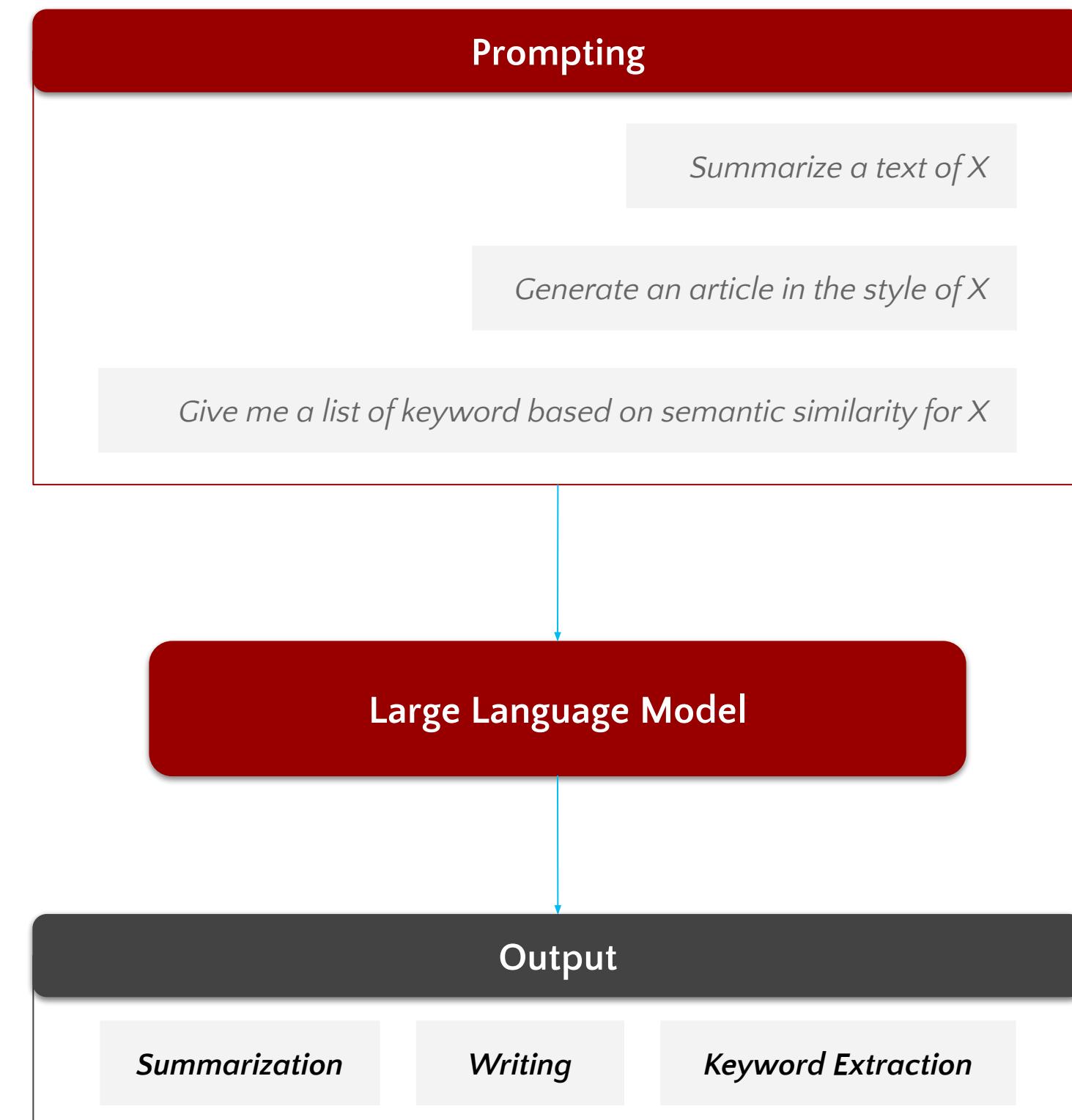
## Code

Code generation  
Code completion  
Code chat  
Code conversion

## Speech

Speech to text  
Text to speech

# Generative Language Model



# Key Concepts

## Prompts

You give the model a prompt and it responds with a series of words that fits both the content and the style of the prompt and, in some cases, even the mood. The models try to predict what you want from the prompt. If you send the words "Give me a list of cat breeds," the model wouldn't automatically assume that you're asking for a list of cat breeds.

**Key guidelines:** Show and tell, provide quality data, check your settings e.g., Temperature

## Tokens

Tokens can be thought of as pieces of words. Before the API processes the prompts, the input is broken down into tokens. 1 token = ~4char

## Embeddings

An embedding is a special format of data representation that can be easily utilized by machine learning models and algorithms. It is a compressed format of document in a vectorization form. Each embedding is a **vector of floating-point numbers**, such that the distance between two embeddings in the vector space is correlated with semantic similarity between two inputs in the original format. For example, if two texts are similar, then their vector representations should also be similar. **cosine similarity** measures the cosine of the angle between two vectors projected in a multi-dimensional space. Used in search similarities, recommendation, clustering etc.

## Completions

The completions endpoint can be used for a wide variety of tasks. It provides a simple but powerful text-in, text-out interface to any of supported [models](#).

## Responsible AI - Content Filtering

Azure OpenAI Service includes a content management system that works alongside core models to filter inappropriate contents.

## Fine-tuning

Prompt Engineering vs Fine-tuning

# Words vs Token

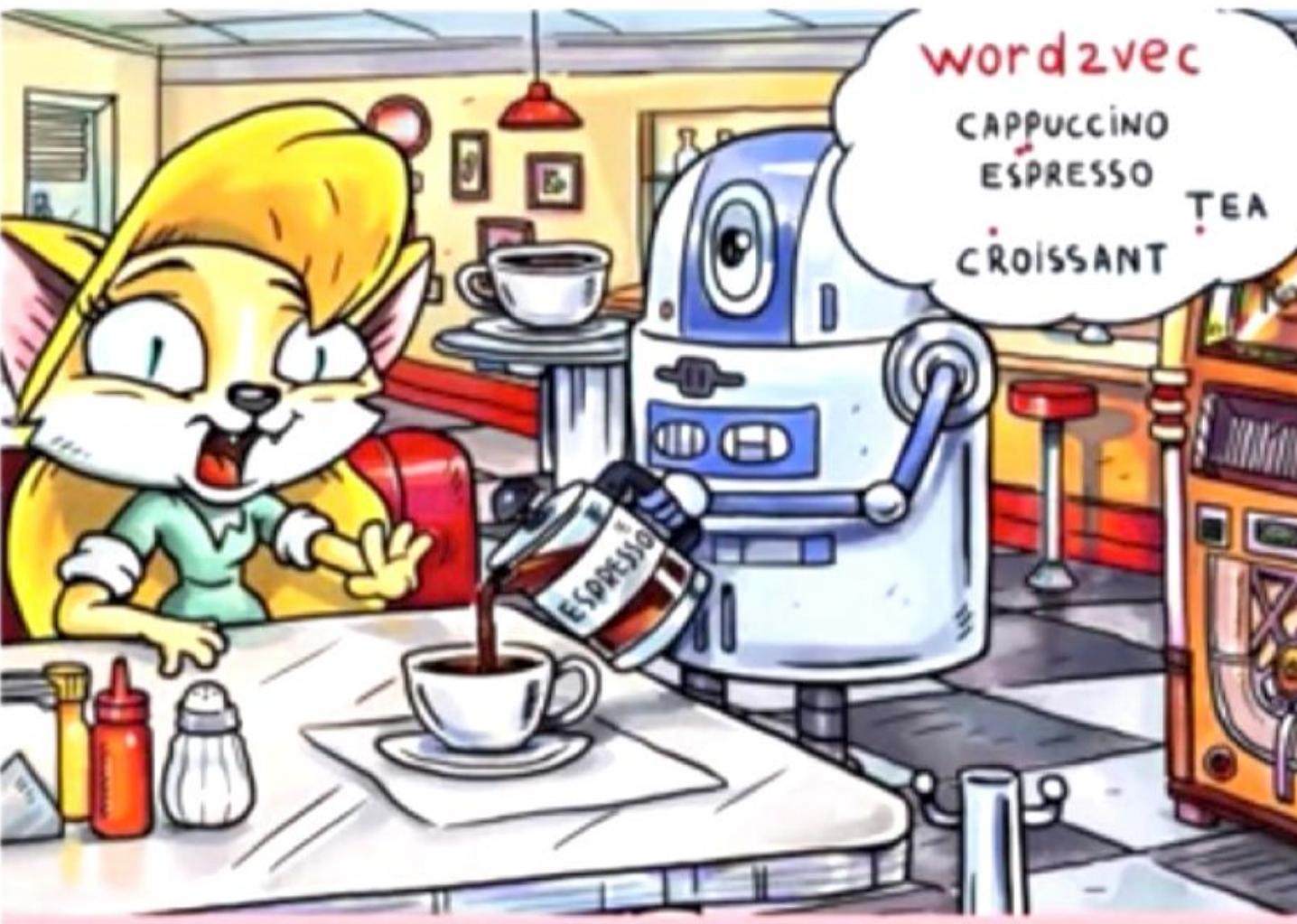
LLMs take tokens as inputs

<b>Words</b>	<b>Tokens</b>
Everyday	[Every, day]
Joyful	[Joy, ful]
I'd like	[I, 'd, like]

# Embeddings

**How to represent a meaning of a word, a sentence, or a text?**

*You shall know the word by the company it keeps. (Firth, 1957)*



- Espresso? But I ordered a cappuccino!  
- Don't worry, the cosine distance between them is so small  
that they are almost the same thing.

- Word embeddings
- Sentence embeddings
- Topic models

# Today's Topics

- 01 Understanding the fundamentals
- 02 Generative AI & it's usecases
- 03 An Introduction to LLMs
- 04 Github Co-Pilot



# Large Language Models (LLM)

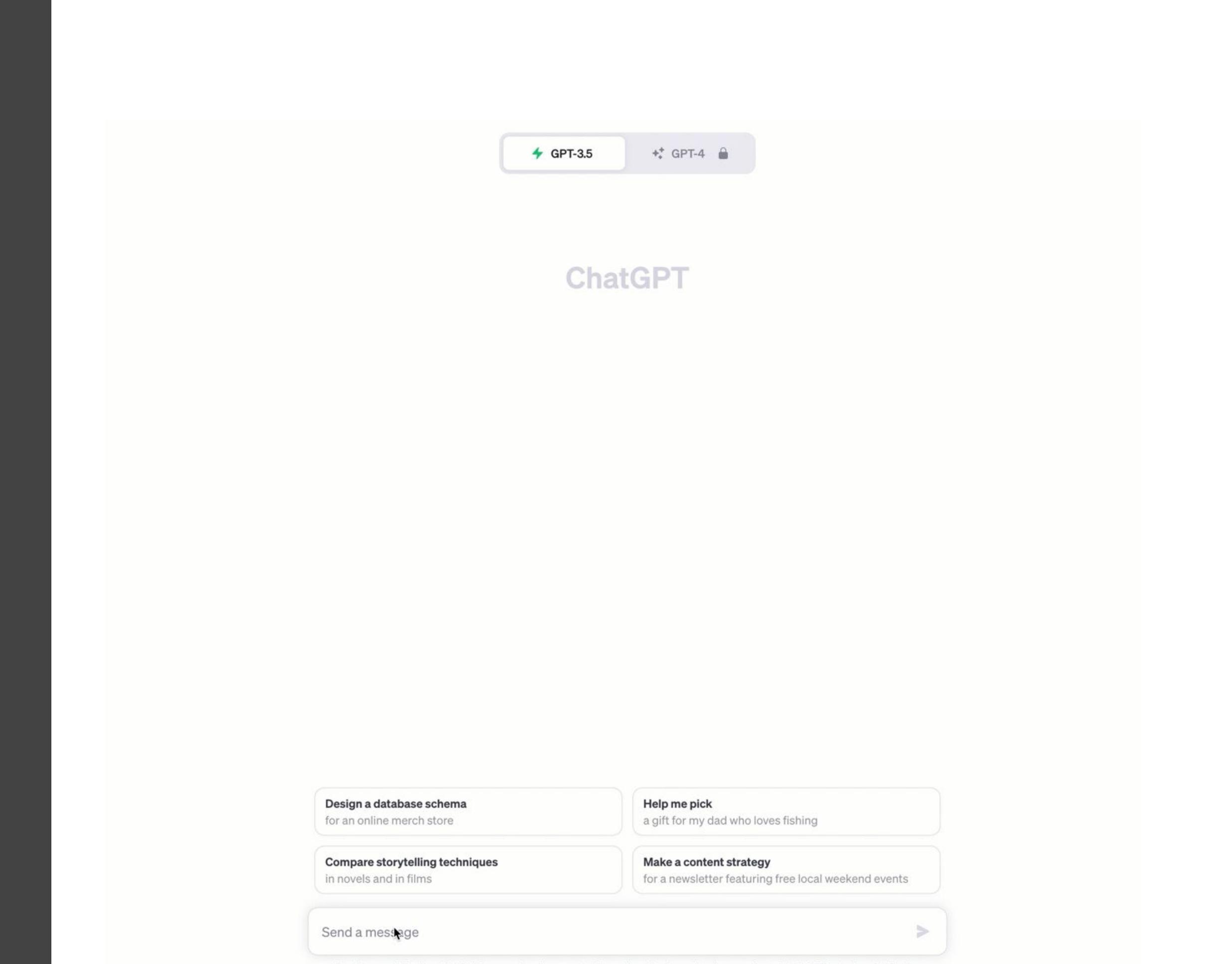
Models trained on large amounts of text data that can perform a wide variety of language tasks, including text summarization, generation, and categorization. These models can perform generative tasks like text generation and so there is some overlap between LLMs and generative AI.

## Examples of Language Models

- GPT-1, GPT-2, GPT-3
- Jurassic
- GPT-J
- Dall-E, Midjourney, Stable Diffusion
- BERT
- Bard and LaMDA

# Chat GPT

Example: Accessibility to React Application



# ChatGPT Timeline



# What does GPT mean?

**Generative Pretrained Transformer**

## **Generative**

means that the program can create new text. It's like having a virtual writer that can come up with sentences and paragraphs on its own.

## **PreTrained**

means that the program has already learned a lot about how people use language. It has studied a huge amount of text from books, articles, and websites etc

## **Transformer**

is the name of a special structure inside the program that helps it understand and generate text. It's like a powerful tool that helps the program organize and process information.

## ChatGPT

### What is ChatGPT?

- Human-like Responses
- Contextual Awareness
- Trained on large data

### GPT-3.5 vs ChatGPT : Know the difference

#### GPT-3.5: Make smarter apps



An AI model accessible through an API for on-demand intelligence



Implementing semantic text understanding



Internal Information search & extraction



Building copilot-like applications



Can be used for a lot more including what ChatGPT can do

#### ChatGPT: Get more productive



A GPT-3 model used to build Chatbots you can interact with and ask to perform tasks

#### Can be used for:



Ideation for content creation



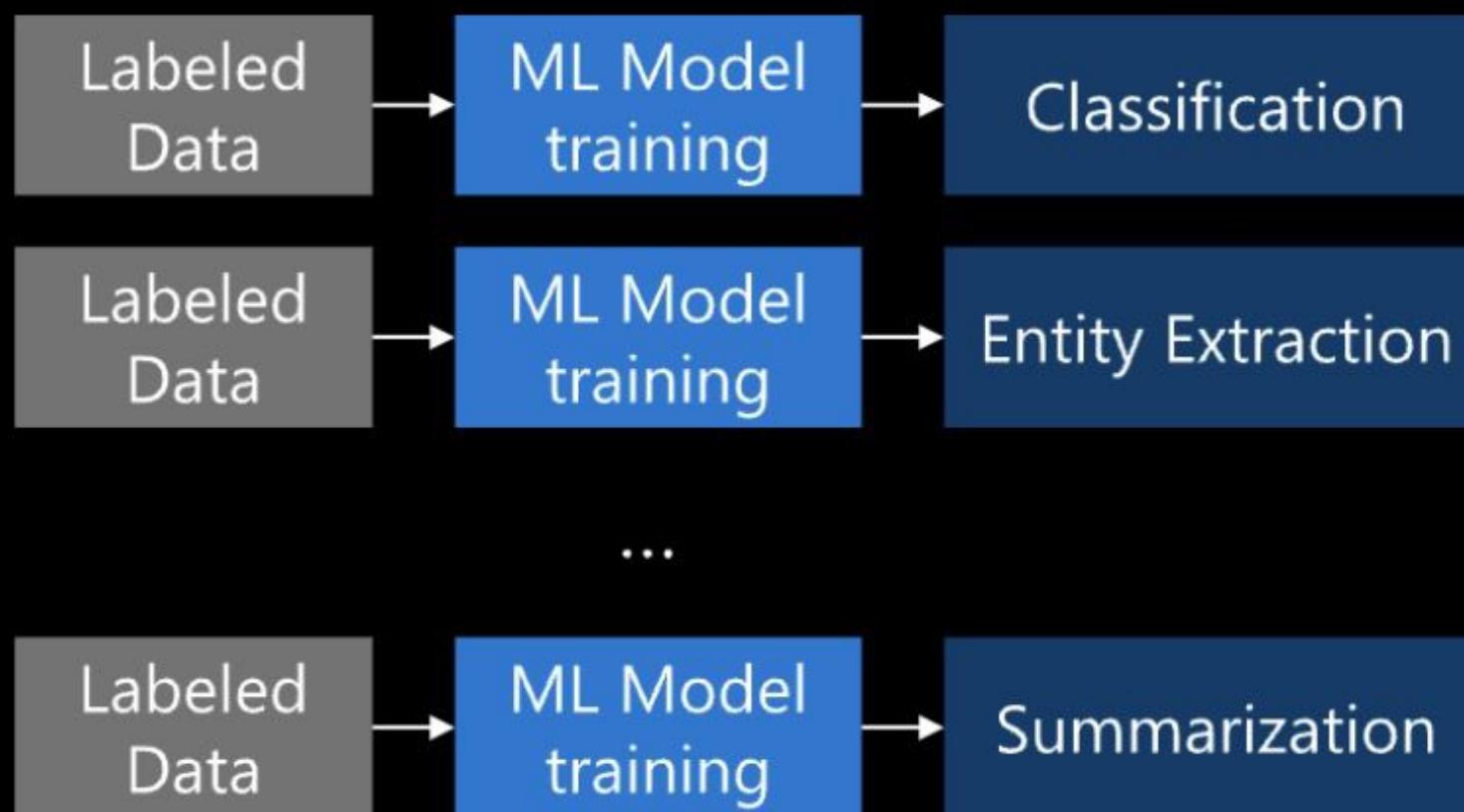
General question answering



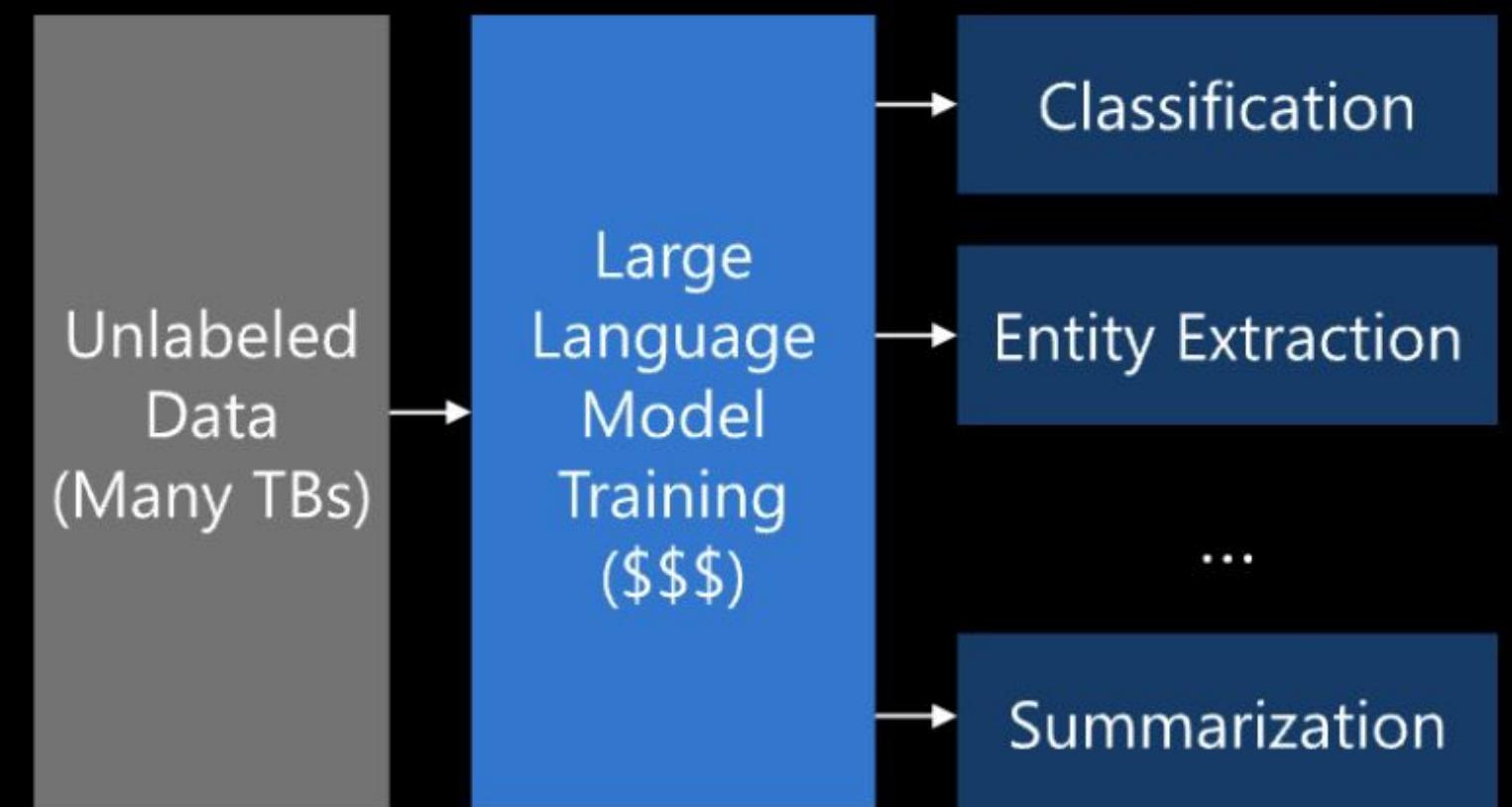
Assistance in code generation and conversation

# Large Language Models

## Typical ML for NLP



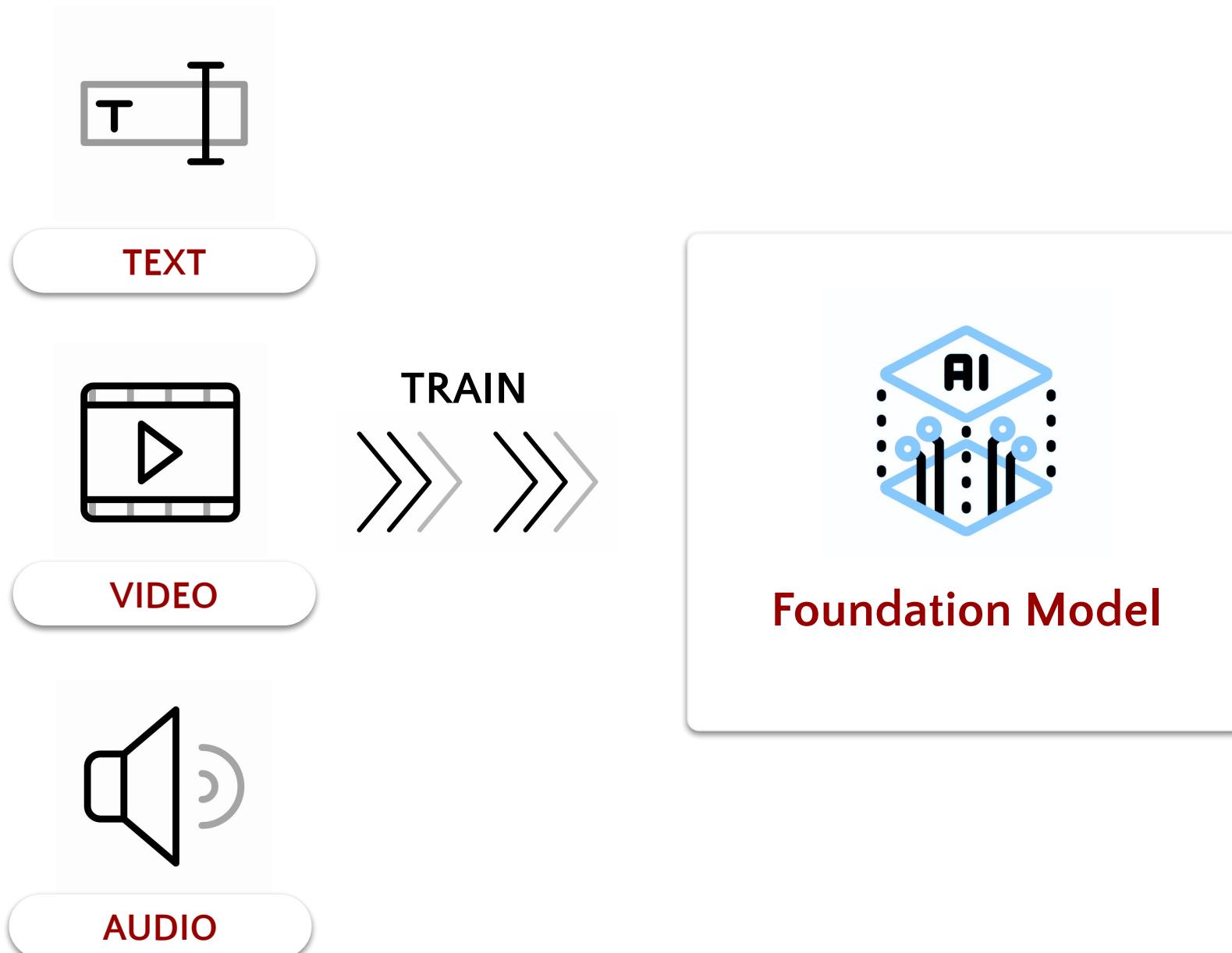
## Large Language Models



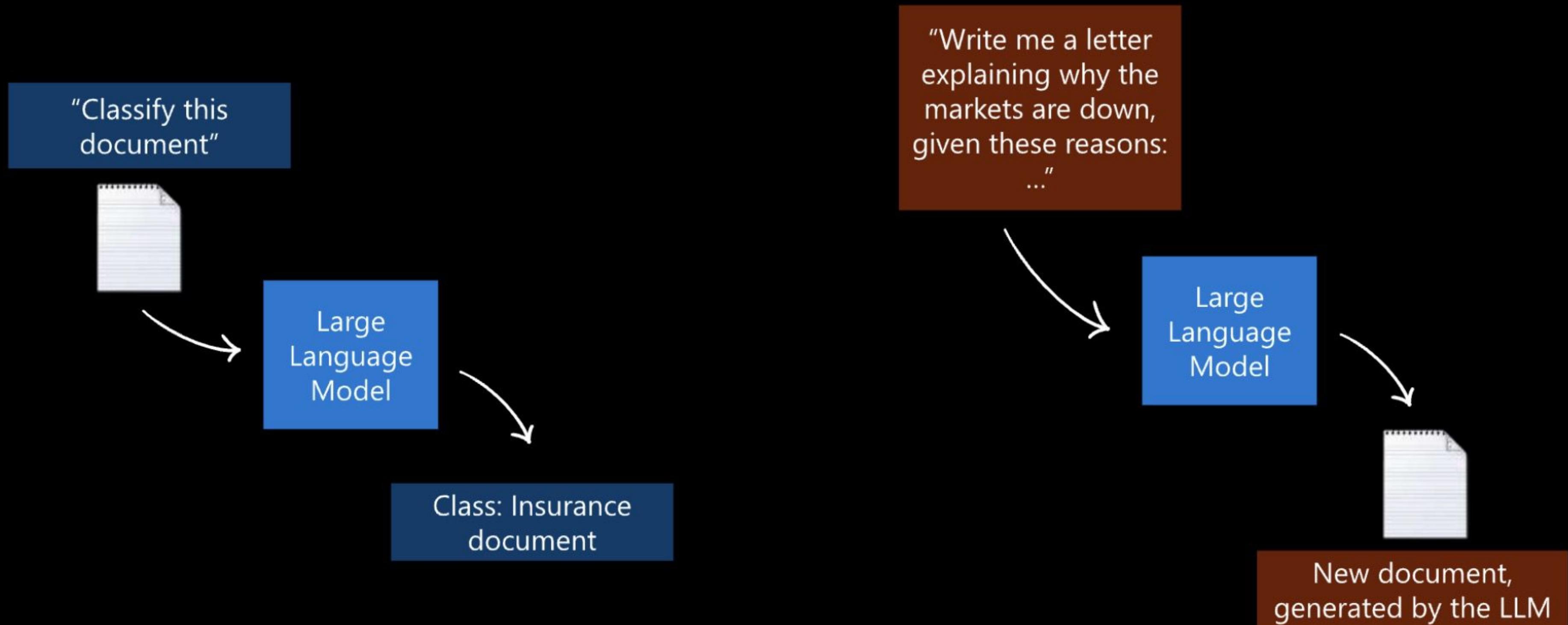
One model per capability  
Labeled data to train  
Highly optimized for use case

Single model for all use cases  
Describe in natural language what it should do

# Foundation Model



# LLMs can also “generate things”

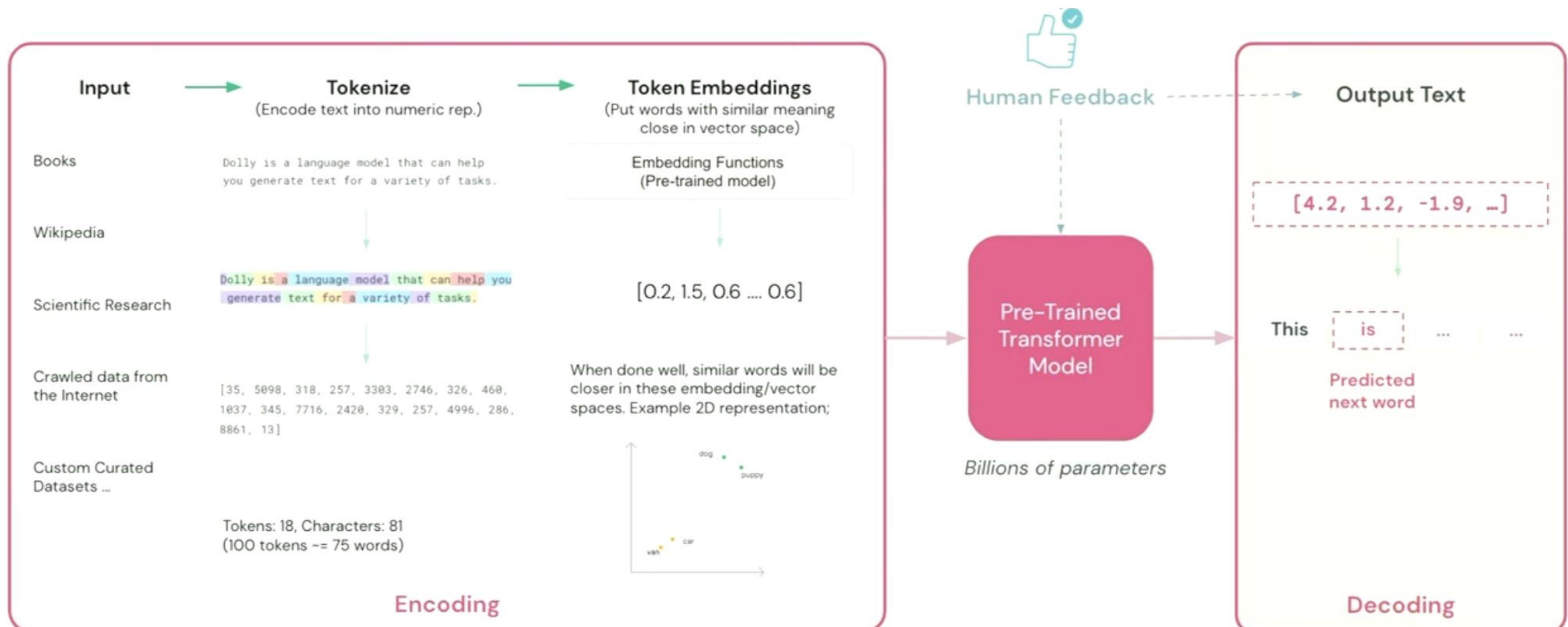


LLMs can do typical ML tasks easily (often zero-shot learned)

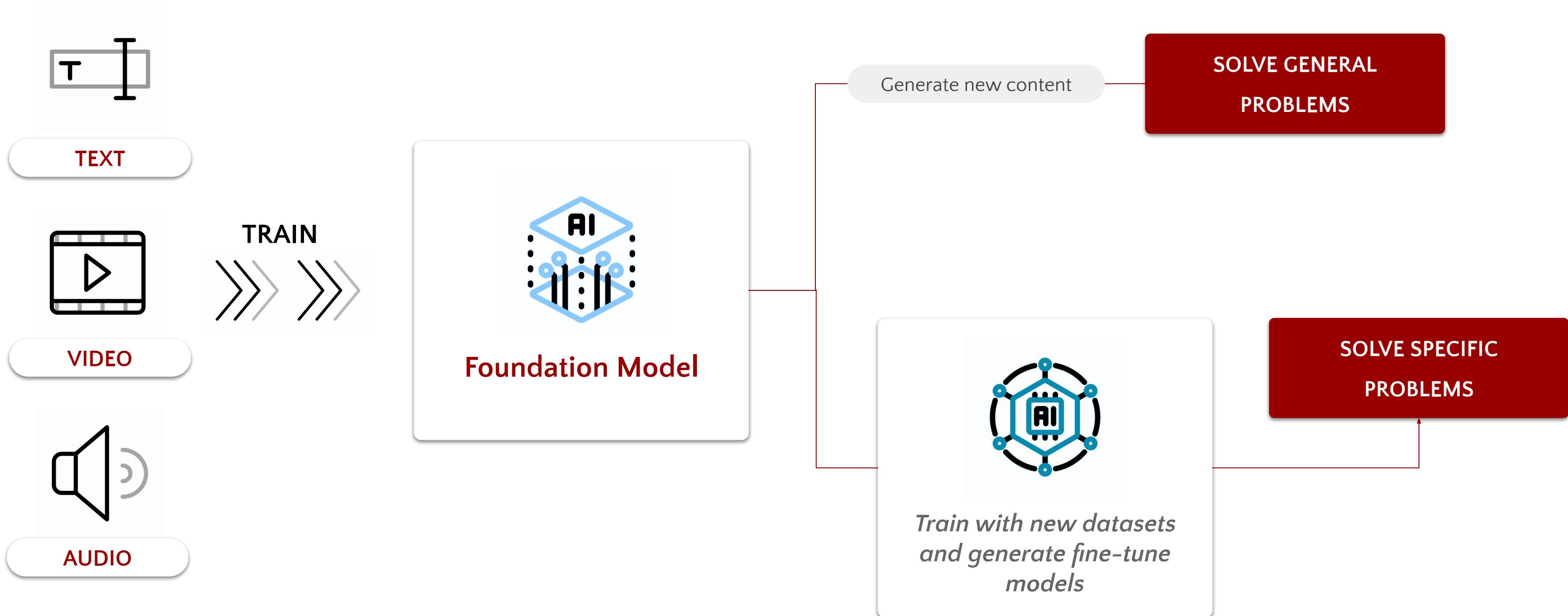
LLMs can also machine-read (understand) and generate new documents

# How Do LLM's Work?

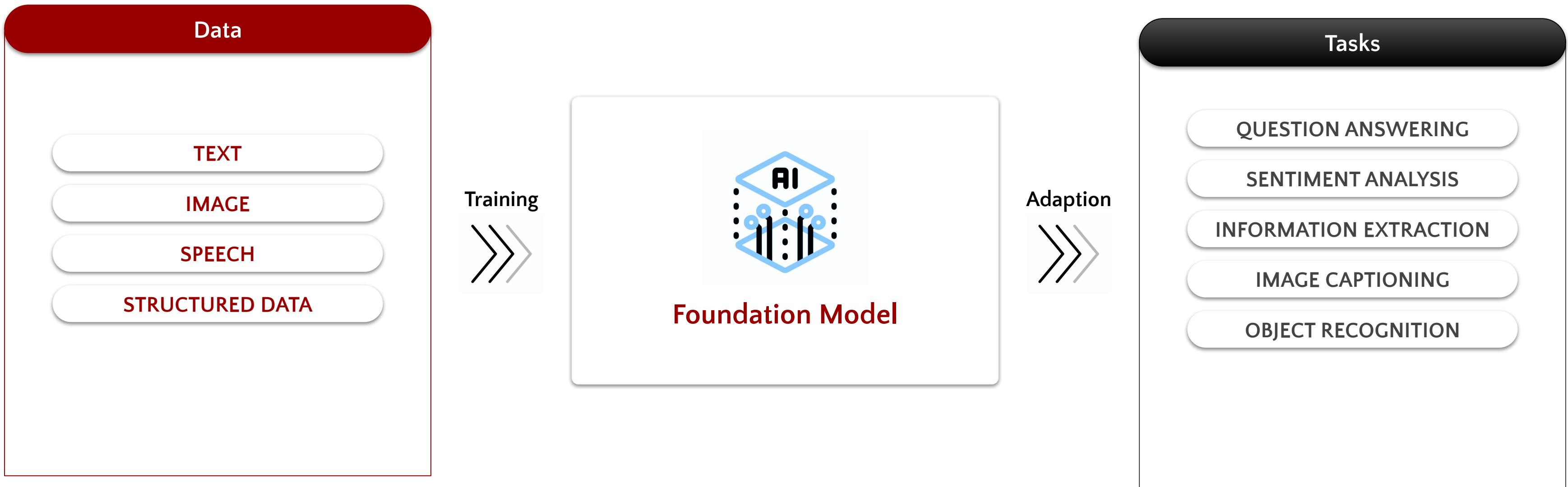
A Simplified version of LLM training process



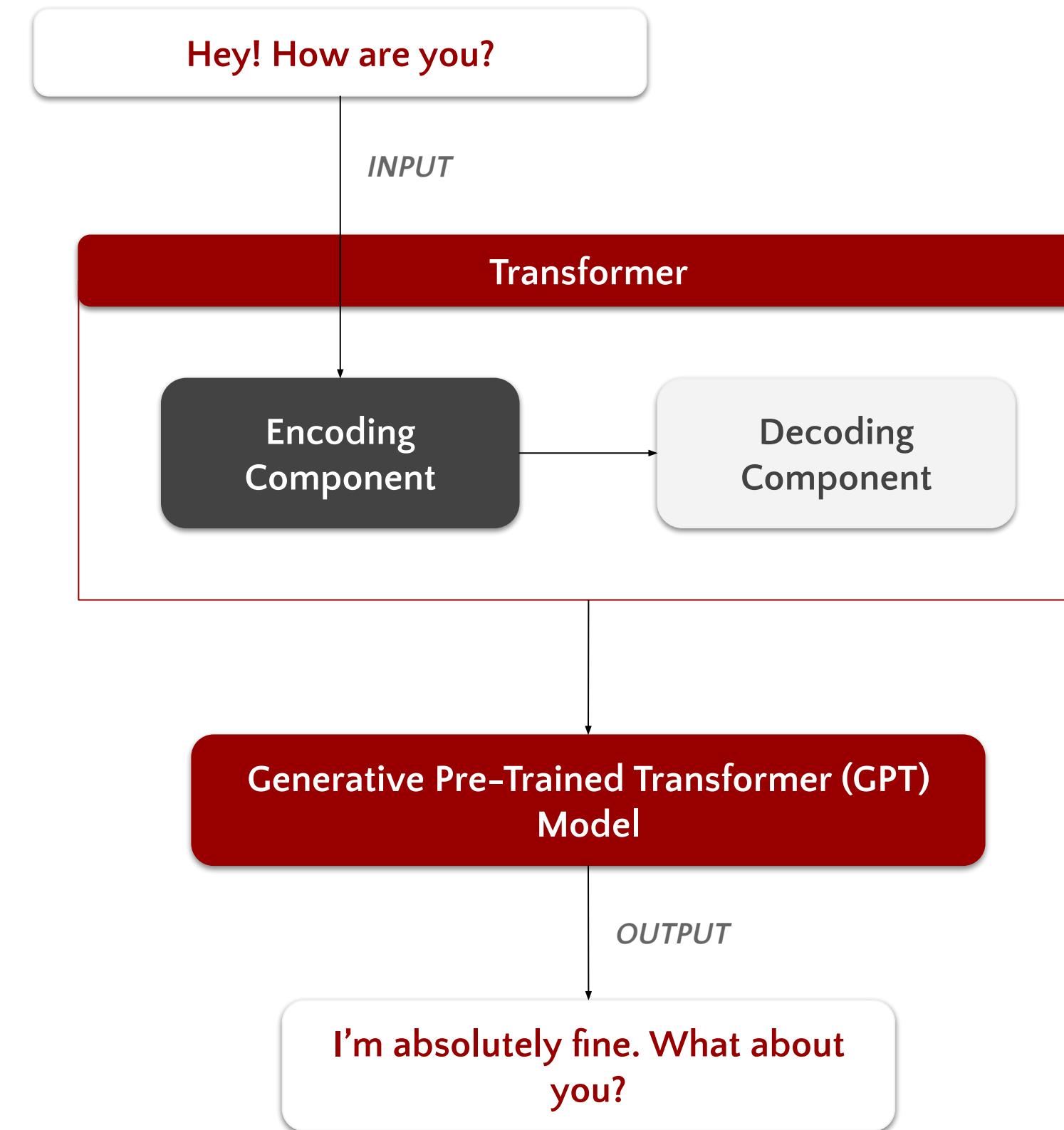
# Foundation Model



# Foundation Model

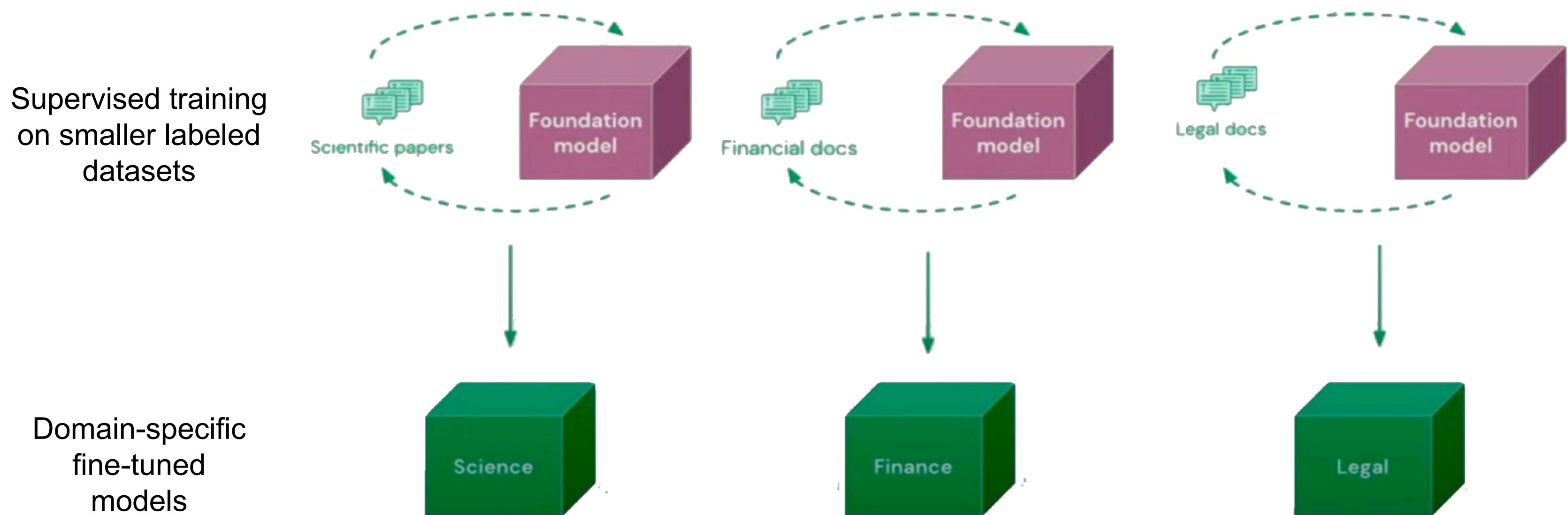


# How it works?



# Fine Tuned Model

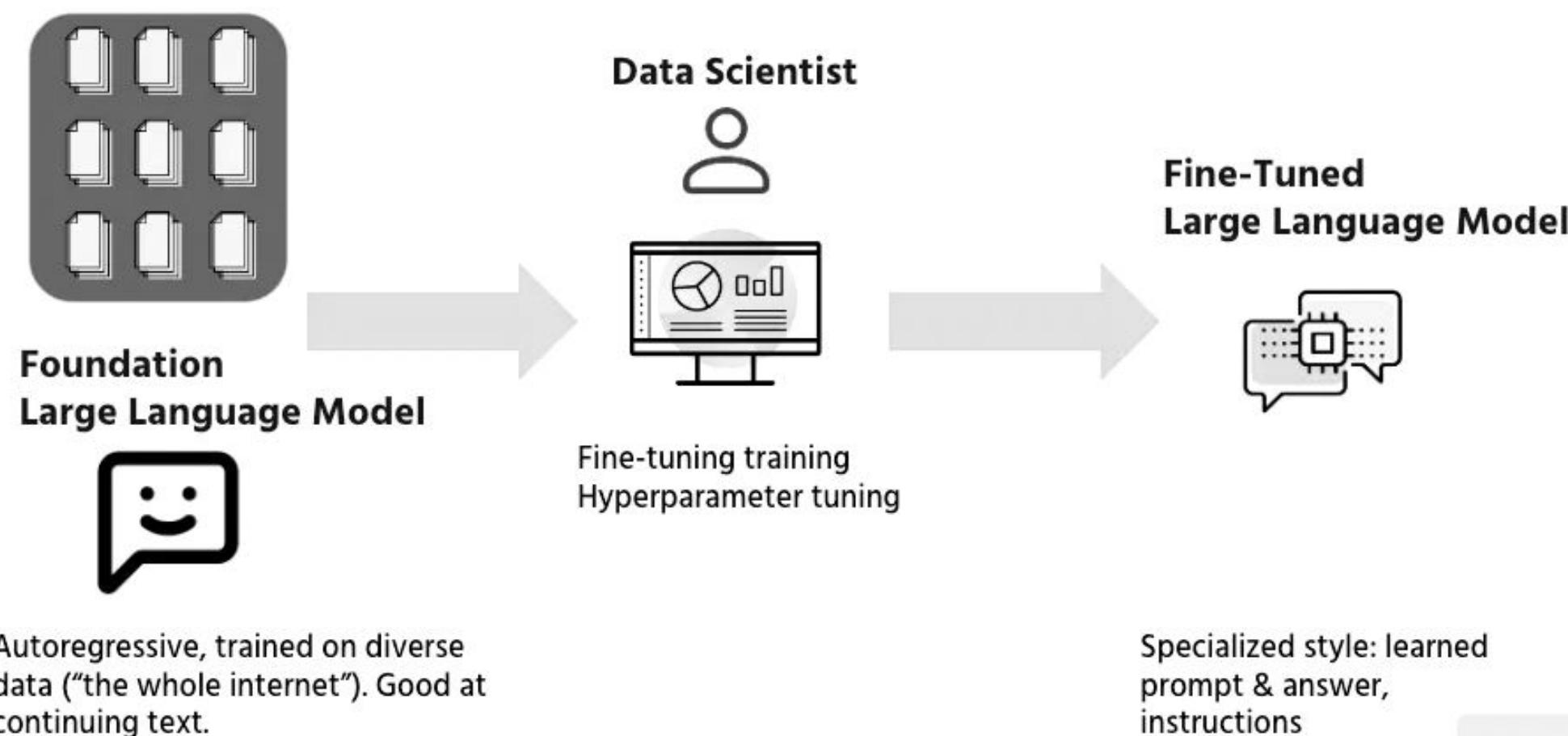
Foundation models can be fine-tuned for domain adaptation



# Fine Tuned Model

## What is fine - tuning and how it works

**Fine-tuning:** The process of further training a pre-trained model on a specific task or dataset to adapt it for a particular application or domain.



# LLMs Business Use Cases

## Customer Engagement

- Personalization and customer segmentation:
  - Provide personalized product/content recommendation based on customer behaviour and preferences
- Feedback Analysis
- Virtual assistants

What are the top 5 customer complaints based on the provided data?



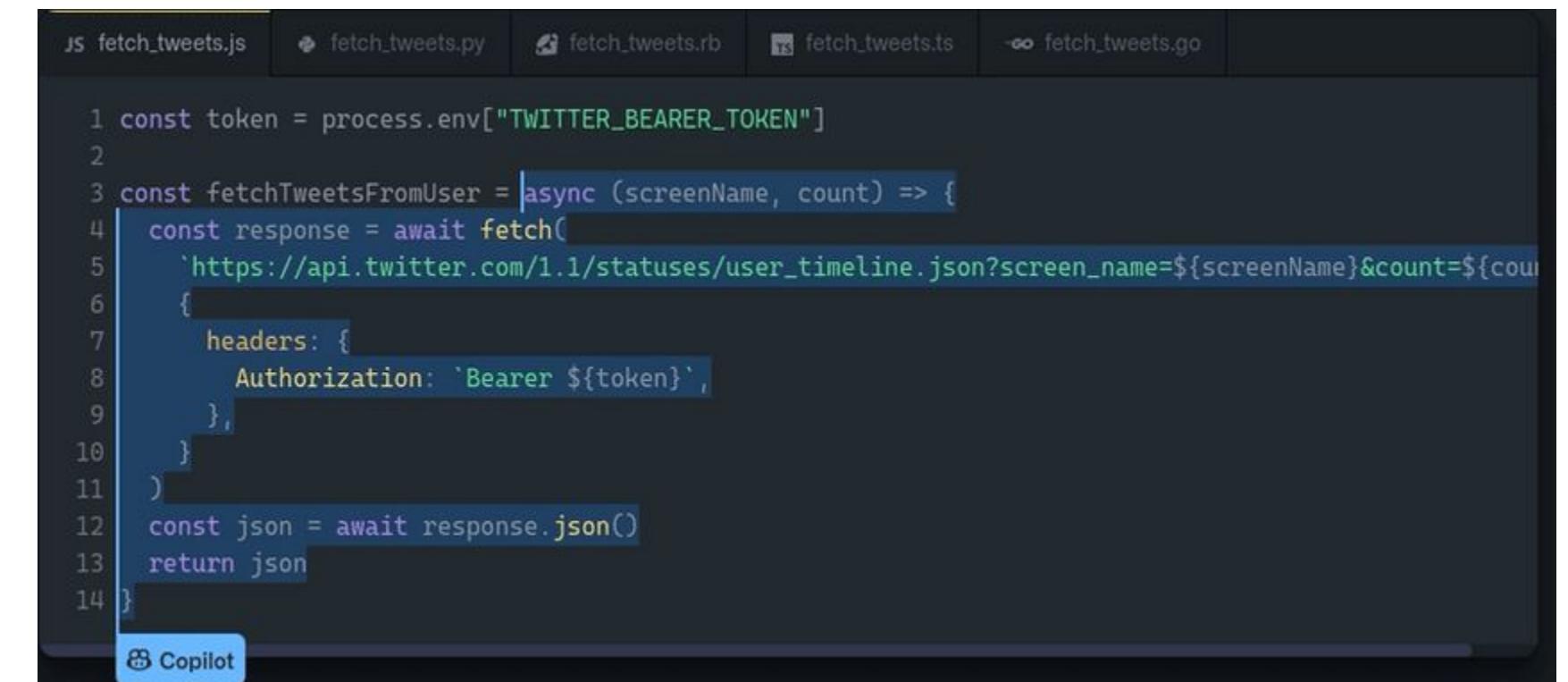
Based on the *customer review dataset*, the top 5 customer complaints are as follows:

1. **Shipping Delays** - 25% of customers expressed frustration with delayed deliveries.
2. **Product Quality** - 20% of customers reported issues with the quality of the received products.
3. **Customer Service Responsiveness** - 18% of customers felt that the response time from customer service was slow.
4. **Billing and Payment Errors** - 15% of customers encountered errors and discrepancies in their bills and payments.
5. **Order Inaccuracies** - 12% of customers received incorrect or incomplete orders.

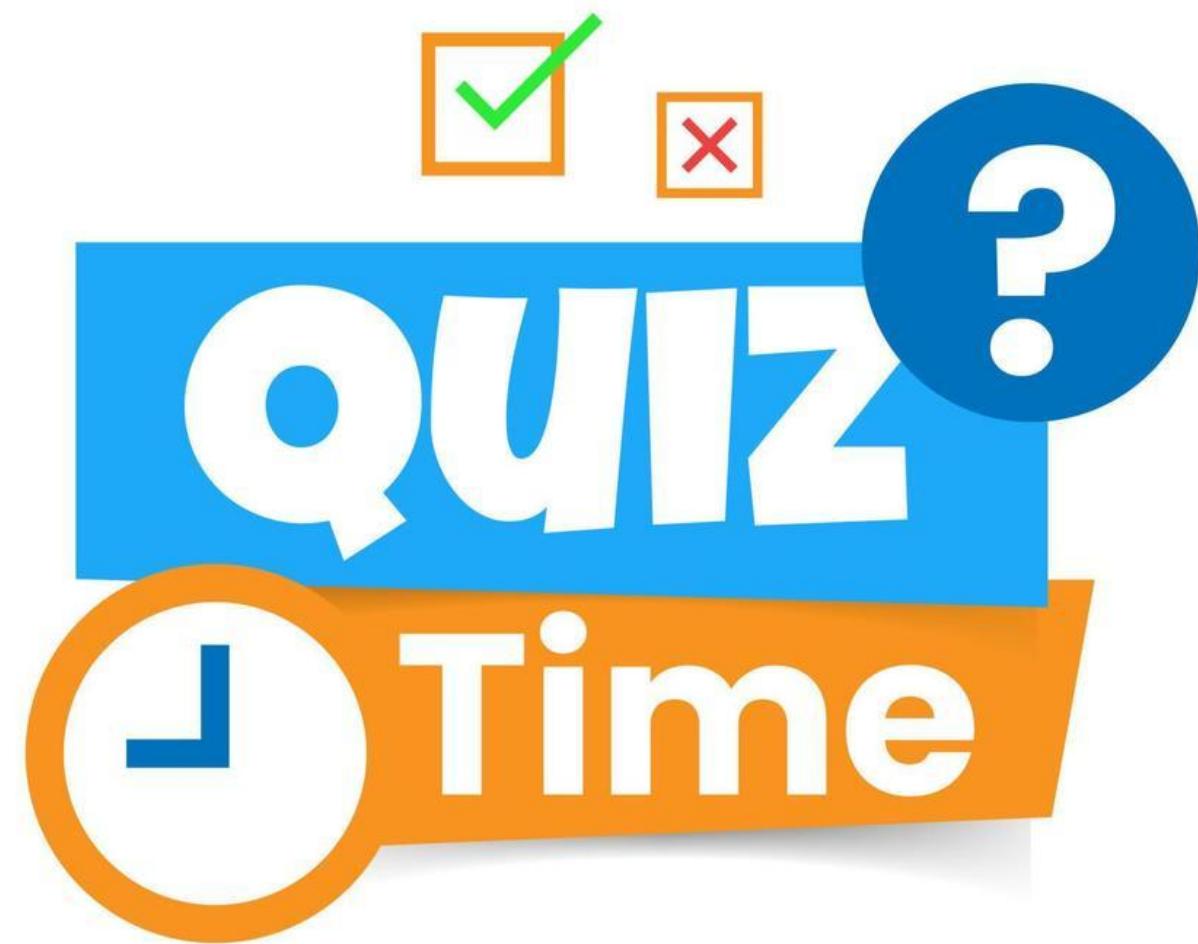
# LLMs Business Use Cases

## Code generation and developer productivity

- Code completion, boilerplate code generation
- Error detection and debugging
- Convert code between languages
- Write code documentation
- Automated testing
- Natural language to code generation
- Virtual code assistant for learning to code



```
js fetch_tweets.js fetch_tweets.py fetch_tweets.rb fetch_tweets.ts fetch_tweets.go
1 const token = process.env["TWITTER_BEARER_TOKEN"]
2
3 const fetchTweetsFromUser = async (screenName, count) => {
4   const response = await fetch(
5     `https://api.twitter.com/1.1/statuses/user_timeline.json?screen_name=${screenName}&count=${count}`,
6     {
7       headers: {
8         Authorization: `Bearer ${token}`,
9       },
10    }
11  )
12  const json = await response.json()
13  return json
14}
```



Imagine you're an advisor at MedCorp, a leading healthcare institution. The company wants to leverage Large Language Models (LLMs) like ChatGPT to improve patient experience and administrative efficiency. How might MedCorp best utilize an LLM for its operations?

A	Implementing the LLM to autonomously diagnose patient illnesses based solely on the symptoms they describe.
B	Using the LLM to assist customer service representatives by providing real-time information and answers to common patient queries.
C	Deploying the LLM as a tool for doctors to generate comprehensive patient reports, taking raw data and translating it into understandable language.
D	Integrating the LLM to automate all managerial decisions based on the text data from past records.



Imagine you're an advisor at MedCorp, a leading healthcare institution. The company wants to leverage Large Language Models (LLMs) like ChatGPT to improve patient experience and administrative efficiency. How might MedCorp best utilize an LLM for its operations?

A	Implementing the LLM to autonomously diagnose patient illnesses based solely on the symptoms they describe.
B	Using the LLM to assist customer service representatives by providing real-time information and answers to common patient queries.
C	Deploying the LLM as a tool for doctors to generate comprehensive patient reports, taking raw data and translating it into understandable language.
D	Integrating the LLM to automate all managerial decisions based on the text data from past records.

# Today's Topics

- 01 Understanding the fundamentals
- 02 Generative AI & it's usecases
- 03 An Introduction to LLMs
- 04 Github Co-Pilot



# Github Copilot

## GitHub Copilot: A Revolution in AI-Powered Coding Assistance

- Cloud-based Generative AI tool
- Turns natural language prompts into coding suggestions
- Works across dozens of languages



- Trained on billions of lines of code
- Functions similar to an AI pair programmer\*
- Works on multiple IDEs
- Not intelligent - may seem that way - or not...
- Sometimes more like a magnified autocomplete

A screenshot of a code editor interface. At the top, there are tabs for "sentiments.ts", "write\_sql.go", "parse\_expenses.py", and "addresses.rb". The main area shows a snippet of TypeScript code with line numbers from 1 to 17. Lines 1 and 2 are empty. Line 3 contains the code "#!/usr/bin/env ts-node".

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5
6
7
8
9
10
11
12
13
14
15
16
17
```

# Github Copilot

Your AI Pair Programmer

**CONVERT COMMENTS TO CODE**

```
// write a binary search algorithm
const binarySearch = (arr, target) => {
  let left = 0;
  let right = arr.length - 1;
  let middle = Math.floor((left + right) / 2);
  while (arr[middle] !== target && left <= right) {
    if (target < arr[middle]) {
      right = middle - 1;
    } else {
      left = middle + 1;
    }
    middle = Math.floor((left + right) / 2);
  }
  return arr[middle] === target ? middle : -1;
}
```

**AUTOFILL REPETITIVE CODE**

```
1 const seconds = 3600
2 const minutes = seconds / 60
3 const hours = minutes / 60
4 const days = hours / 24
5 const weeks = days / 7
6 const months = days / 30
7 const years = months / 12
```

 Copilot

**SHOW ALTERNATIVE**

```
1 def max_sum_slice(xs):
2     max_endng = max_xs_far = 0
3     begin = end = 0
4     for i, x in enumerate(xs):
5         if max_endng + x > 0:
6             max_endng += x
7         else:
8             max_endng = 0
9             begin = i
10            if max_endng > max_xs_far:
11                max_xs_far = max_endng
12                end = i
13    return max_xs_far
14
15 def max_sum_slices():
16     if not xs:
17         return 0
18
19     max_endng = max_slice = 0
20     for i, x in enumerate(xs):
21         max_endng = min(0, max_endng + x)
22         max_slice = max(max_slice, max_endng)
23
24    return max_slice
```



**INCREASE DEVELOPER  
PRODUCTIVITY**



**ACCELERATE INNOVATION**



**BRIDGE SKILL GAP**

**96%**

say they are **faster** with repetitive tasks

**88%**

say they are **more productive**

**74%**

focus on more satisfying work

# GitHub Copilot



---

**GitHub Copilot (Maintenance)**  
Refactoring code (code translate)  
Reviewing code (code explain)  
Documentation

---

**GitHub Copilot (Implementation)**  
Convert comments to code  
Autofill for repetitive code  
Show alternatives

---

**GitHub Copilot (Testing)**  
Unit testing  
Finding code errors  
Debugging  
Code review  
AI Pull Requests

# Write Code



**GitHub**  
Copilot



tabnine

OSS Tools

<https://github.com/CodedotAI/gpt-code-clippy>  
<https://github.com/samrawal/emacs-secondmate>



# Demo of Github Copilot



**10 Minutes**

GitHub Copilot is a powerful AI coding assistant developed by GitHub and OpenAI, designed to boost coding productivity.



# Today's Topics

- 01 Understanding the fundamentals
- 02 Generative AI & it's usecases
- 03 An Introduction to LLMs
- 04 Github Co-Pilot

