ML Products

taxonomy, application areas, engineering & tech stack

Most Famous Applications

recommendations

chatbots

ChatGPT Bard

autonomous driving

WAYMO TESLA

translation

DeepL

robotics

OCR

assistants (speech recognition)

but many more …

When To Use ML

automation

all applications from previous slide

complexity

protein structure predictions: AlphaFold (GNN)

Taxonomy of ML Models

Supervised Learning

Target Quantity

- **known in training**: labeled samples or observations from past
- to be **predicted** for unknown cases (e.g., future values)

Features

input information that is

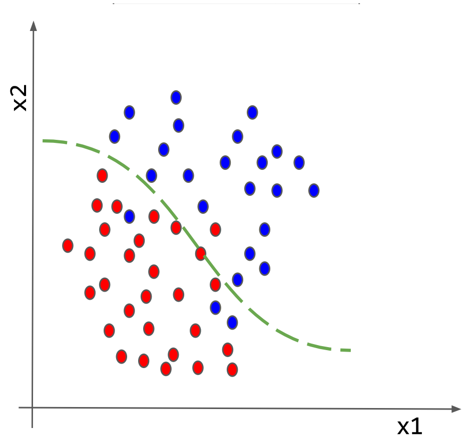
- correlated to target quantity
- known at prediction time

Example: Spam Filtering

Classify emails as spam or no spam

use accordingly ***labeled emails as training set***

use information like **occurrence of specific words** or **email length** as **features**



features x1 and x2

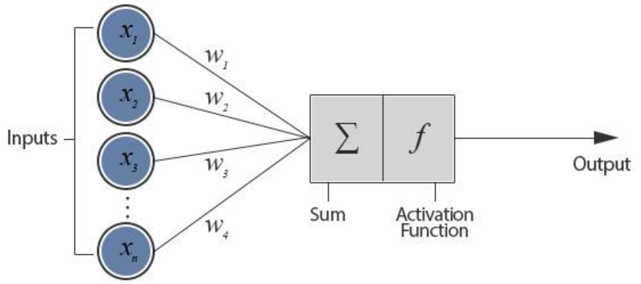
spam, no spam

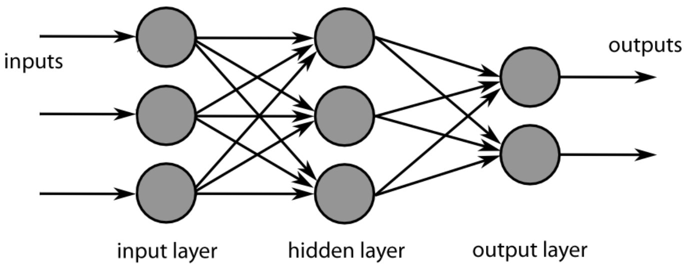
usually rather narrow tasks

Algorithmic Families and Linear Building Blocks

linear (parametric) models

neural networks: non-linear just by means of activation functions

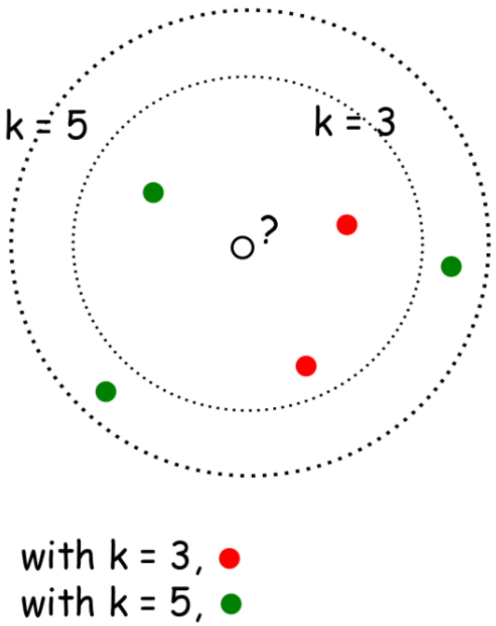




from wikipedia

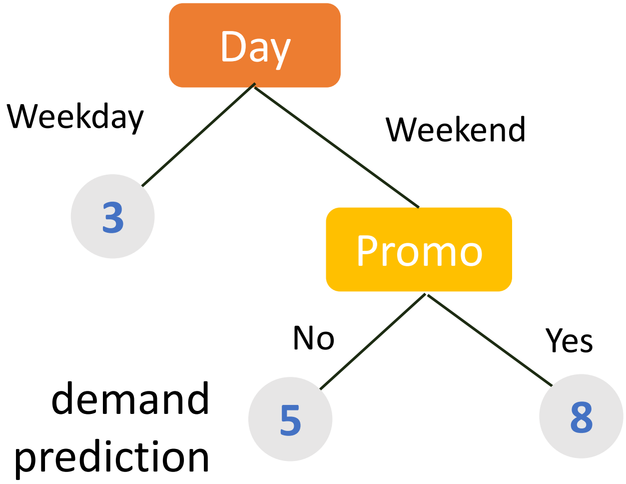
deep learning: many hidden layers
computer vision: CNN
NLP: transformer

nearest neighbors (local methods, instance-based learning) – non-parametric models



kernel/support-vector machines: linear model (maximum-margin hyperplane) with kernel trick

decision trees: rule learning



often used in ensemble methods

- bagging: random forests
- boosting: gradient boosting

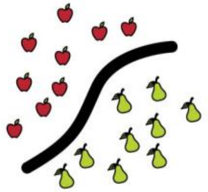
mainly used for structured data

MACHINE LEARNING

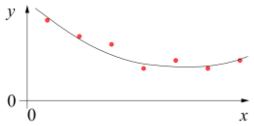
training target available (labeled or past data)

SUPERVISED

CLASSIFICATION



REGRESSION



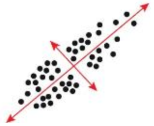
data not labeled in any way

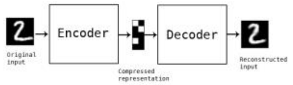
UNSUPERVISED

CLUSTERING

ASSOCIATION

DIMENSIONALITY REDUCTION





no supervision, but goal-based interaction with environment

REINFORCEMENT LEARNING

LEARN STATE OR ACTION VALUES

LEARN POLICY DIRECTLY

learning by teacher
(high-dimensional curve fitting)

learning by observation
(pattern recognition)

learning by trial-and-error
(sequential decision making)

unsupervised and reinforcement learning can both be cast as supervised-learning setup

Classic (Discriminative) Models

discriminative models:

prediction/estimation of labels (classification) or numerical values (regression)

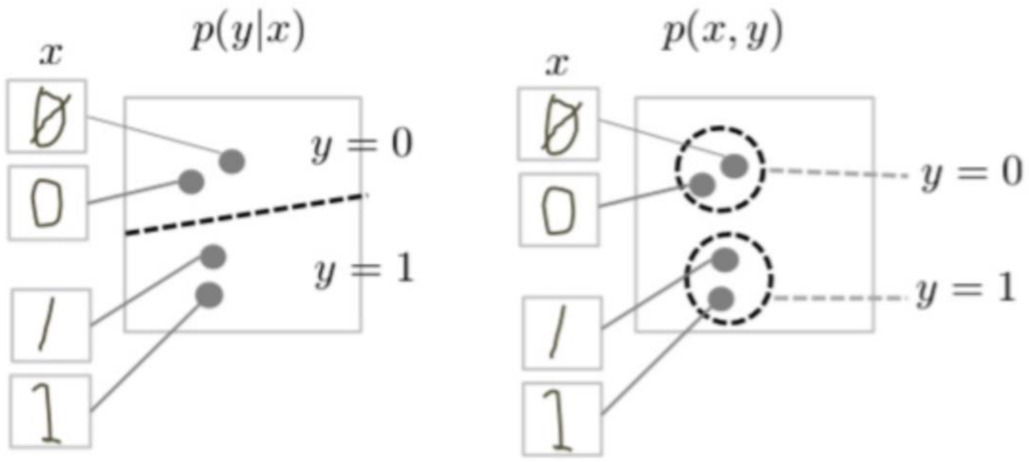
examples for discriminative tasks:

- object recognition
- demand forecasting

generative models:

generation of new data according to data distribution seen in training

discriminative model generative model



source

Generative Models

new paradigm for ML workflow:

prompt engineering (low/no code)

instead of classic training (fit) and inference (predict) steps

→ multi-task (and multi-modal) models

instead of narrow use cases of classic models

Under The Hood: Foundation Models

LLMs: transformer models with hundreds of billions of parameters

self-supervised (pre-)training on vast data sets

→ huge foundation models (impossible to train yourself)

usage options:

- typical: **in-context** learning via prompt (potentially giving few examples for task at hand and add retrieval augmentation or tool usage)
- for special case and high-quality requirements: **fine-tuning** on specific tasks and data sets (example: chatbot like ChatGPT via reinforcement learning from human feedback)

Depending on the application, there are currently two dominant approaches for generative AI:

- text generation: large language models (LLM)
- image synthesis: diffusion models

Text Generation

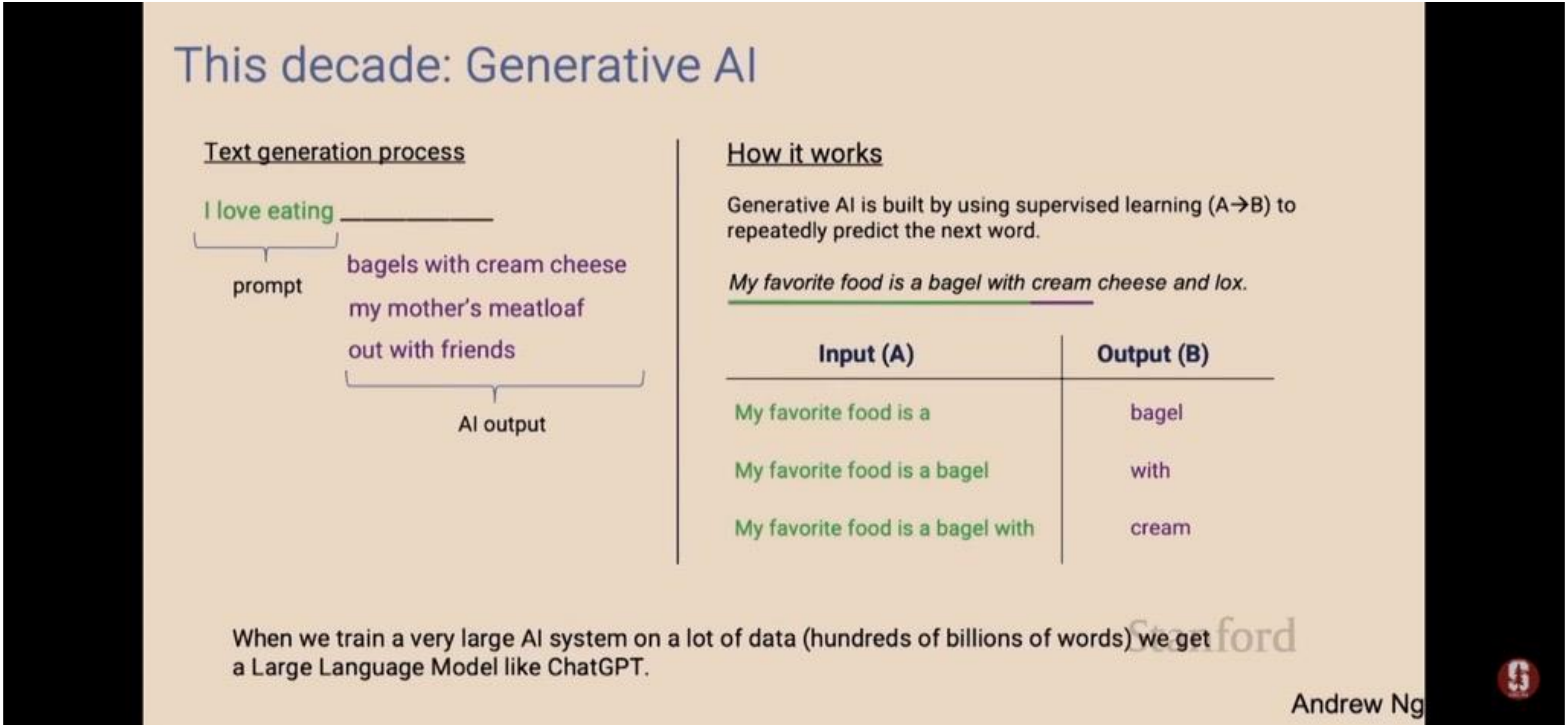
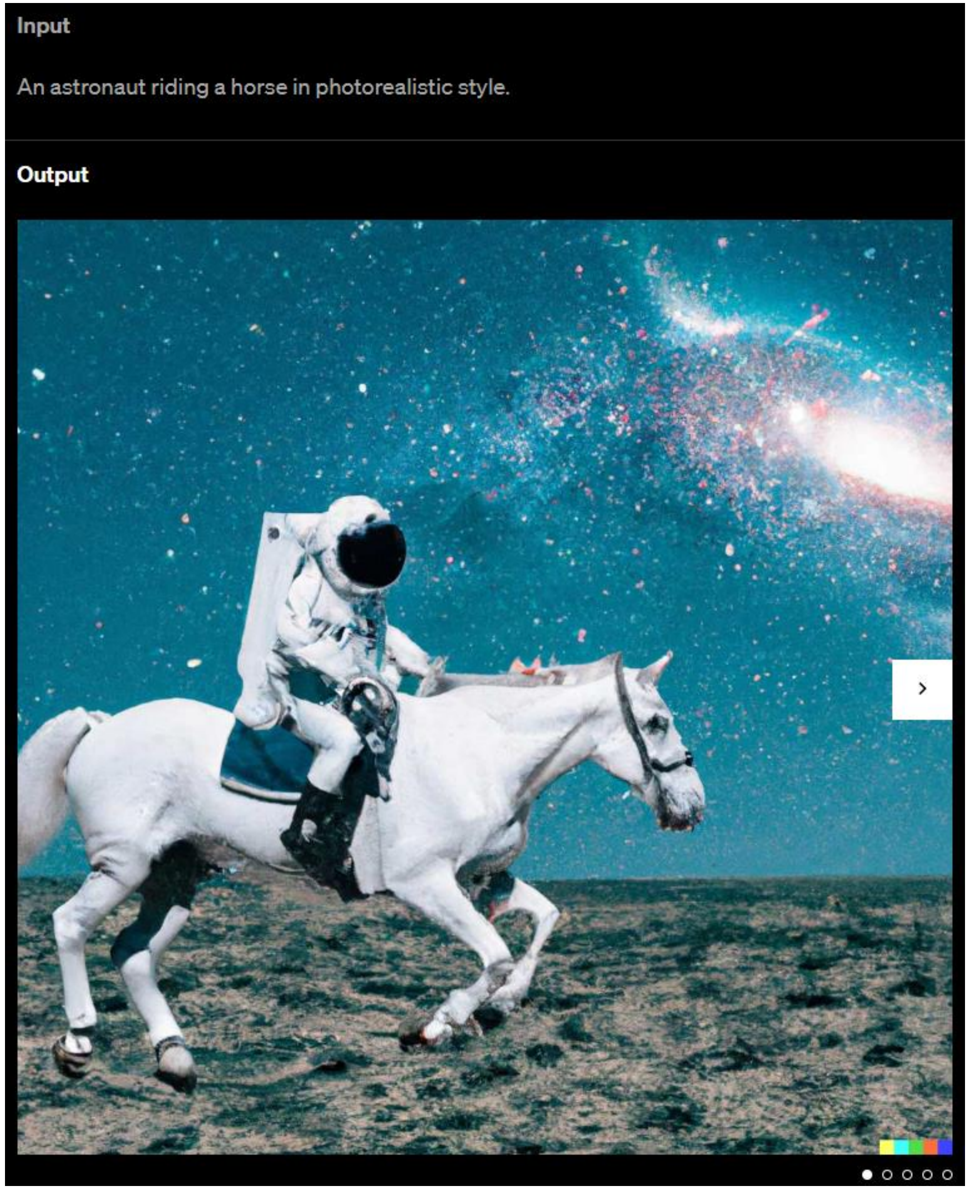


Image Synthesis



idea: generate new images as variations of training data

condition generation on text prompts: text-to-image

trade-off between diversity and fidelity

SOTA: (guided) diffusion models

classic/discriminative models

- effective for performing numerical and optimization tasks (predictions)
- continue to account for majority of AI value in wide range of industries (e.g., supply chain)

generative models

- not suitable for classical use cases like numerical and optimization tasks
- but complimentary: drive value across entire organizations by revolutionizing internal knowledge management systems

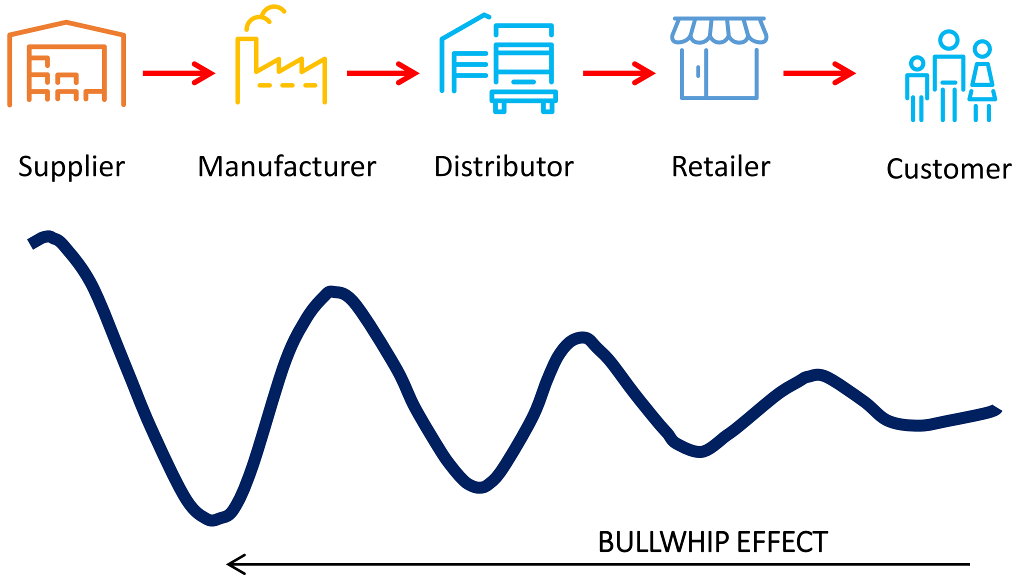
Application Areas

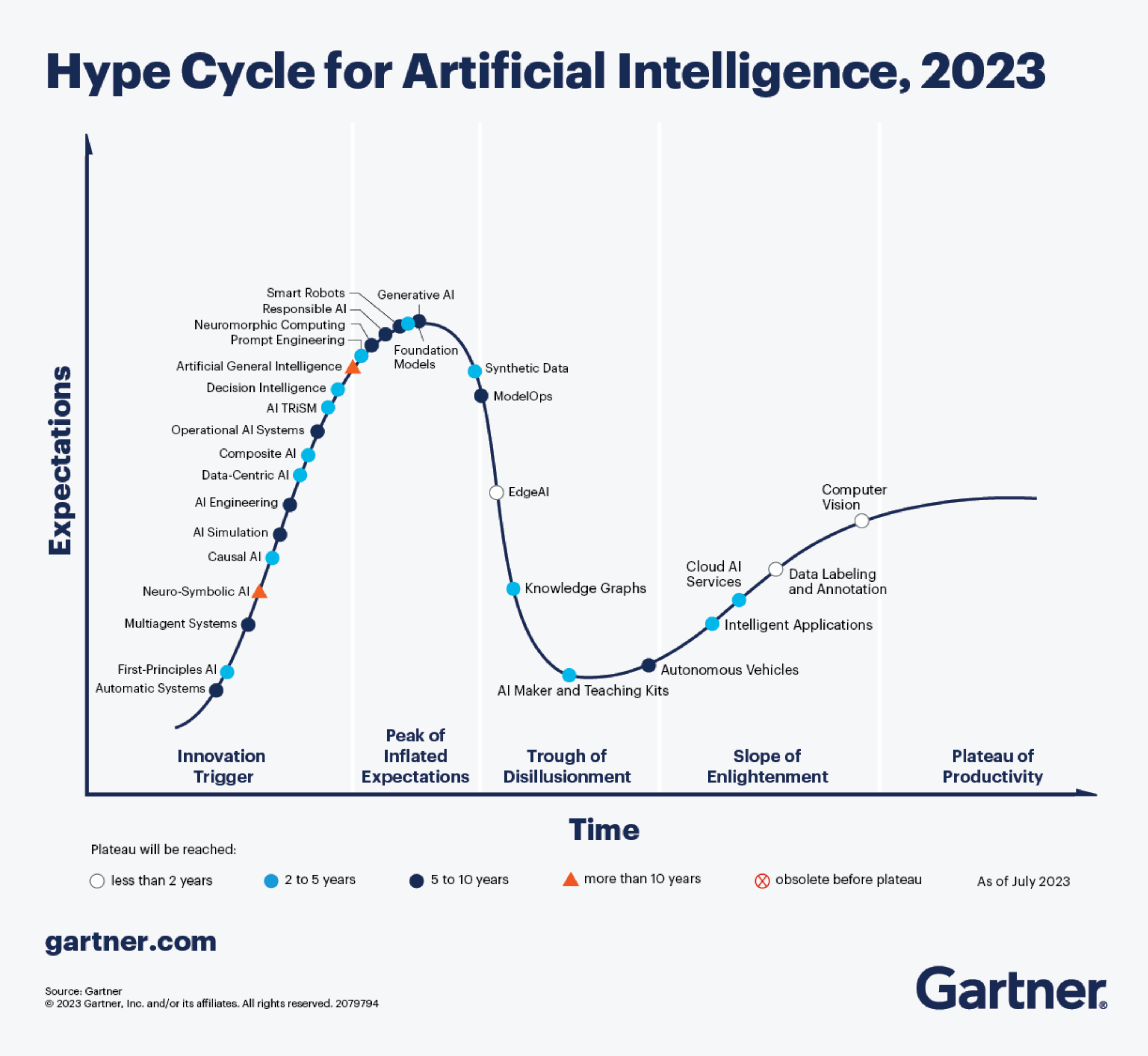
Example for Classic Models: Supply Chain

support of operations research

some examples:

- warehouse operations and yard management: slotting (e.g., via item affinity), labor management, tracking (computer vision)
- transportation (logistics and mobility): routing, ETA (e.g., in Google Maps via GNN), disruption predictions (e.g., stock-outs)
- retail: demand forecasting, replenishment, pricing, targeting





Generative AI at peak of inflated expectations

still: there is now something to play with

Biggest Business Impact of Generative AI

customer operations

interactions with customers

marketing & sales

generation of creative content

Figure 1. Magic Quadrant for Enterprise Conversational AI Platforms

Coding Assistant

LLM: text-to-code

prominent example: GitHub Copilot

```python
import tweepy, os # secrets in environment variables

def fetch_tweets_from_user(user_name):
    # authentification
    auth = tweepy.OAuthHandler(os.environ['TWITTER_KEY'], os.environ['TWITTER_SECRET'])
    auth.set_access_token(os.environ['TWITTER_TOKEN'], os.environ['TWITTER_TOKEN_SECRET'])
    api = tweepy.API(auth)

    # fetch tweets
    tweets = api.user_timeline(screen_name=user, count=200, include_rts=False)
    return tweets
```

Copilot

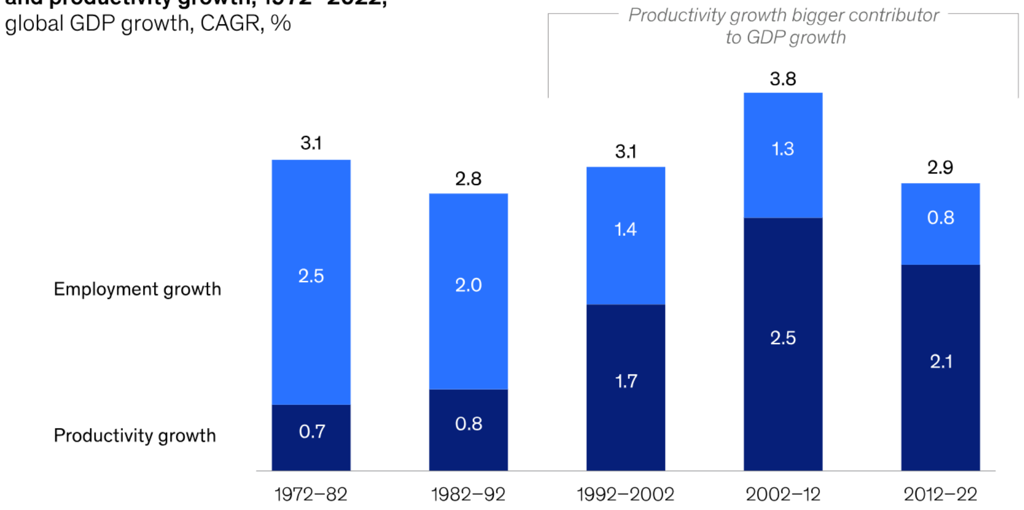
AI Potential

(generative) AI could free up 70% of employees' time

half of today's work activities could be automated between 2030 and 2060

Productivity growth, the main engine of GDP growth over the past 30 years, slowed down in the past decade.

Real GDP growth contribution of employment and productivity growth, 1972–2022,
global GDP growth, CAGR, %



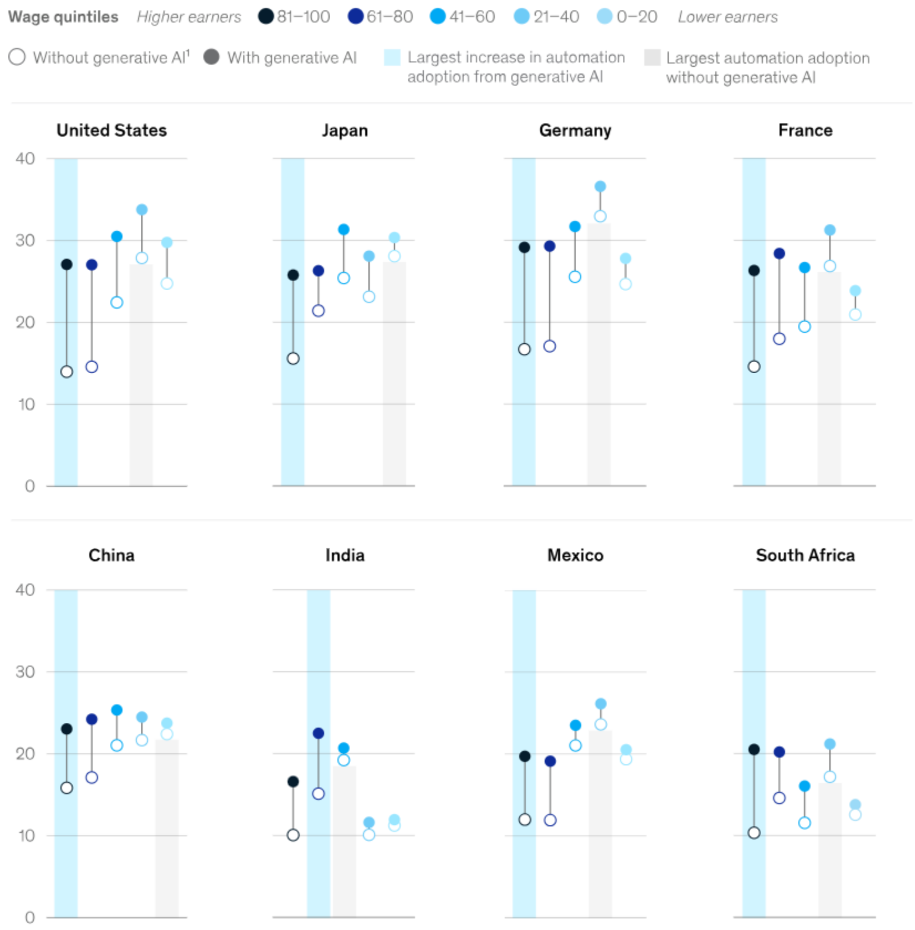
Source: Conference Board Total Economy database; McKinsey Global Institute analysis

McKinsey & Company

automate white-collar more than blue-collar jobs

Generative AI could have the biggest impact on activities in high-wage jobs; previously, automation's impact was highest in lower-middle-income quintiles.

Automation adoption per wage quintile, % in 2030, midpoint scenario



[1]Previous assessment of work automation before the rise of generative AI.
Source: McKinsey Global Institute analysis

McKinsey & Company

Engineering & Tech Stack

Scientific Python Stack

NumPy

pandas

scikit learn

SciPy

matplotlib

Deep Learning Frameworks

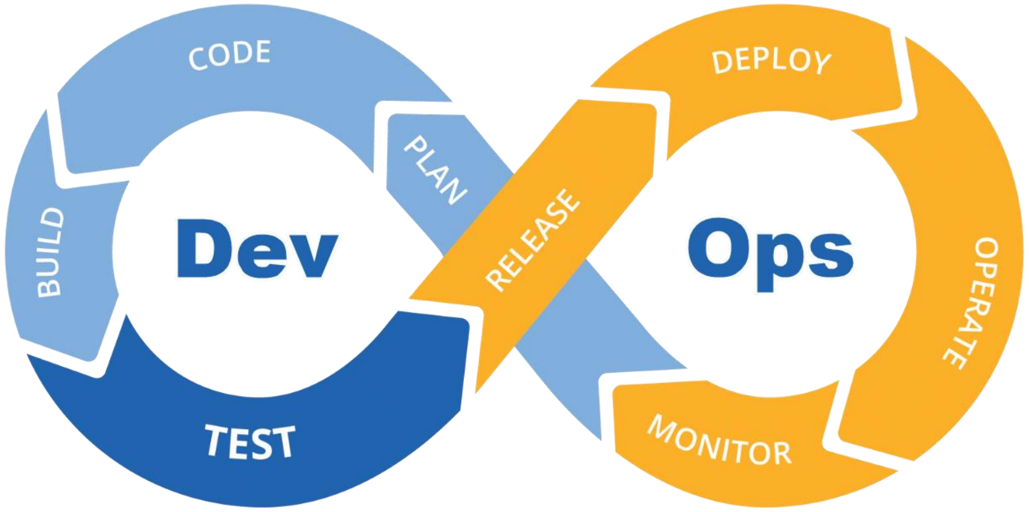
PyTorch TensorFlow

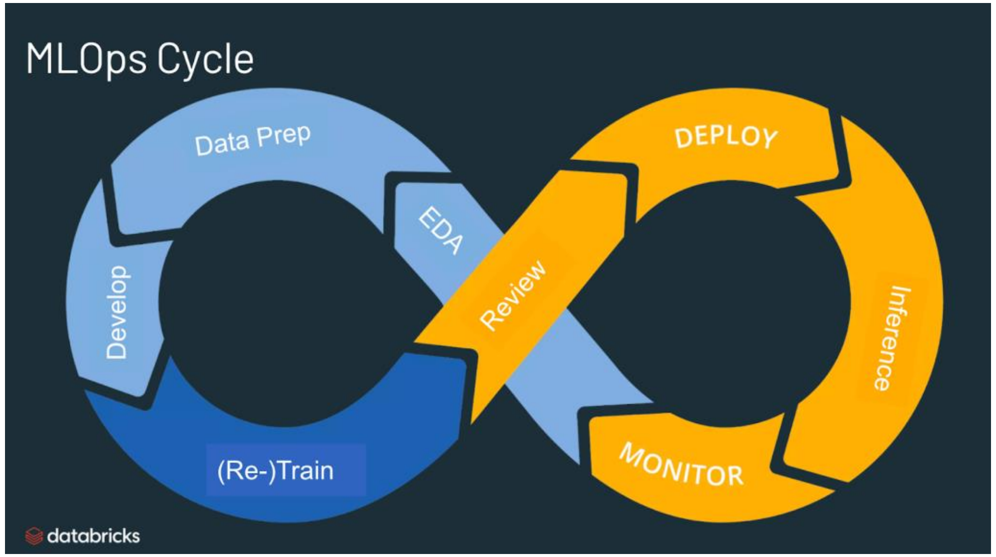
need for lots of memory and compute …

IaaS, PaaS, SaaS

example for managed service:
Azure Kubernetes Service (AKS)
for container orchestration

ML in Production





examples for managed ML services: Azure ML, SageMaker

Data Management

not only compute but also cloud-based data storage

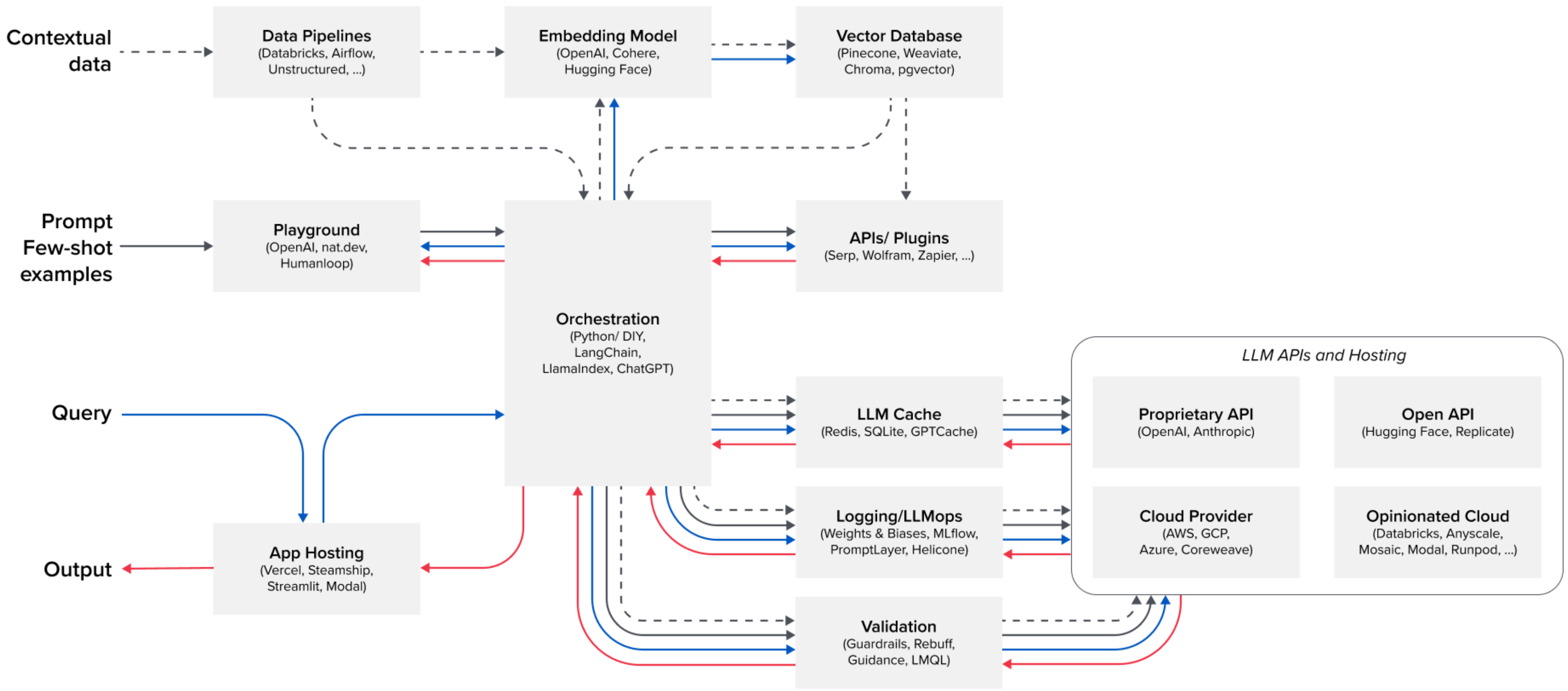
data lake: raw data

data warehouse: integrated data

unification of data lake and warehouse: ETL → ELT

NoSQL example: graph database

Emerging LLM App Stack



LEGEND

Gray boxes show key components of the stack, with leading tools/systems listed

Arrows show the flow of data through the stack

- - - → Contextual data provided by app developers to condition LLM outputs
——→ Prompts and few-shot examples that are sent to the LLM
——→ Queries submitted by users
——→ Output returned to users

a16z Enterprise

What The Future Holds: Autonomous Agents

learning from data