Democratizing AI Training and Access Statewide

A NAIRR Pilot Expansion Project

Take a Moment, Please Fill Out Our Pre-Workshop Survey:

NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

University of Kentucky

Democratizing AI Training and Access Statewide

A NAIRR Pilot Expansion Project

Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

Disclaimer

This material is based upon work supported in part by the National Science Foundation Award *#2528161*. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Workshop Goals

- Understand core AI concepts and how AI is used in real-world settings
- Explore how AI is shaping research and the workforce in Kentucky
- Gain confidence in foundational ideas like machine learning, large language models, and multimodal AI
- Experience applications of AI through the lens of research workflows
- Learn how to access local and national AI computing resources through the NAIRR Pilot
- Spark new ideas and collaborations across AI and related research fields
- Connect Kentucky researchers and innovators to grow a shared AI ecosystem

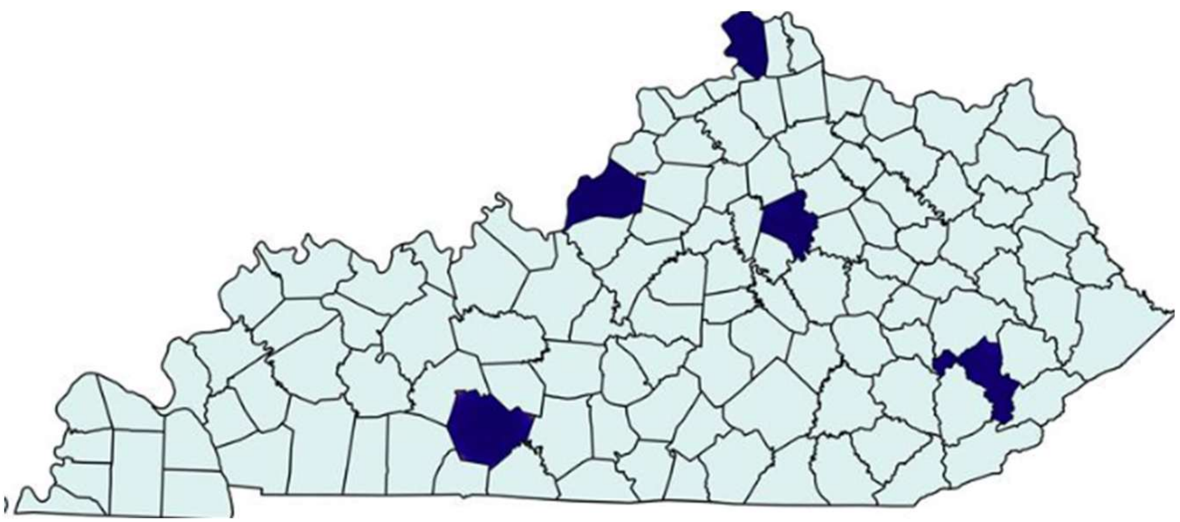
Why This Matters for Kentucky

Empowering researchers and innovators across the Commonwealth

Reducing barriers to AI (knowledge, access, collaboration)

Leveraging statewide partnerships for a connected AI ecosystem

Positioning Kentucky to shape the future of AI



NAIRR Pilot National Artificial Intelligence Research Resource Pilot

Advancing US Innovation in Artificial Intelligence

The NAIRR Pilot aims to connect U.S. researchers and educators to computational, data, and training resources needed to advance AI research and research that employs AI.

nairrpilot.org

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

NAIRR Pilot Expansion Projects

NAIRR Pilot Expansion Projects are multidisciplinary efforts to build capacity and broaden access to, and training on, NAIRR Pilot resources. Expansion Projects are aimed to expand the NAIRR Pilot community and enhance utilization of resources through targeted outreach, training, and inclusive community experiences.

Who We Are: Center for Applied AI (CAAI)

Accelerating Discovery with Applied AI & Cross-Disciplinary Collaboration

What We Do:

- Build secure, interoperable AI toolsets that can be adapted across domains
- Identify and overcome bottlenecks in data, modeling, and analytical workflows
- Work on cross-disciplinary teams to combine AI expertise with domain expertise

Center for Applied Artificial Intelligence
Pro Scientia et Humanitate

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

CAROLINE LEACH

DATA MANAGEMENT SPECIALIST

AARON MULLEN

SOFTWARE ENGINEER

VAIDEN LOGAN

SOFTWARE ENGINEER

MITCHELL KLUSTY

SOFTWARE ENGINEER

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

DR. CHRISTOPHER LEDFORD

DIRECTOR DATA AND ADVANCED ANALYTICS
COUNCIL FOR POST SECONDARY EDUCATION

DR. ADEL ELMAGHRABY

WINNIA PROFESSOR OF COMPUTER SCIENCE & ENGINEERING
DIRECTOR OF RESEARCH, DIGITAL TRANSFORMATION CENTER
CYBERHIVE DIRECTOR
UNIVERSITY OF LOUISVILLE

Time	Topic	Speaker
9:15 – 9:20	Welcome	Dr. Ty Handy
9:20 – 9:45	Workshop Goals & Overview of NAIRR	Caroline Leach
9:45 – 10:30	Introduction to AI & AI-Ready Data	Aaron Mullen, MS
10:30 – 11:30	Featured Host Region Presentation: Digital Innovation, AI Perspectives, and Cybersecurity	Dr. Adel Elmaghraby
11:30 – 12:00	AI & The Workforce	Dr. Christopher Ledford
12:00 – 12:45	Lunch Break *Boxed lunches will be served in the atrium/cafe*	
12:45 – 1:15	Machine Learning	Aaron Mullen, MS
1:15 –1:45	Generative AI & Vibe Coding	Vaiden Logan
1:45 – 2:45	Foundation Models, LLMs and Computer Vision	Mitchell Klusty & Vaiden Logan
2:45 – 3:00	Coffee Break: Peer Networking	
3:05 – 3:30	AI Pipelines & Demo	Mitchell Klusty
3:30 – 4:00	Applied AI Panel	Subject Matter Experts
4:00 - 4:10	Conclusion	Caroline Leach

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

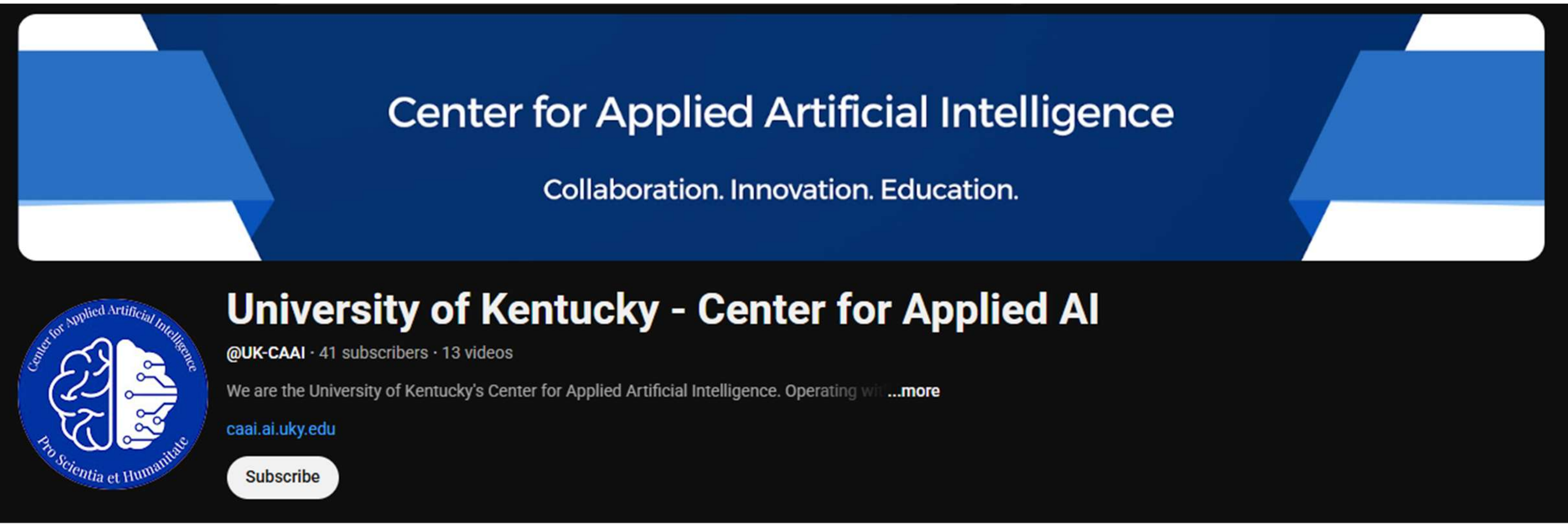
Ongoing resources and support for Kentucky researchers and innovators

To support continued learning and collaboration, we're offering:

- Videos
- Online Forum
- NAIRR Virtual Assistant
- Technical Demos
- Office Hours

Videos and Workshop Recording

https://www.youtube.com/@UK-CAAI



NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

University of Kentucky

Online Forum

Scan to create your account on the forums website:

Jefferson Community and Technical College

NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

University of Kentucky

Online Forum

Online Forum

Hazard Community and Technical College

NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

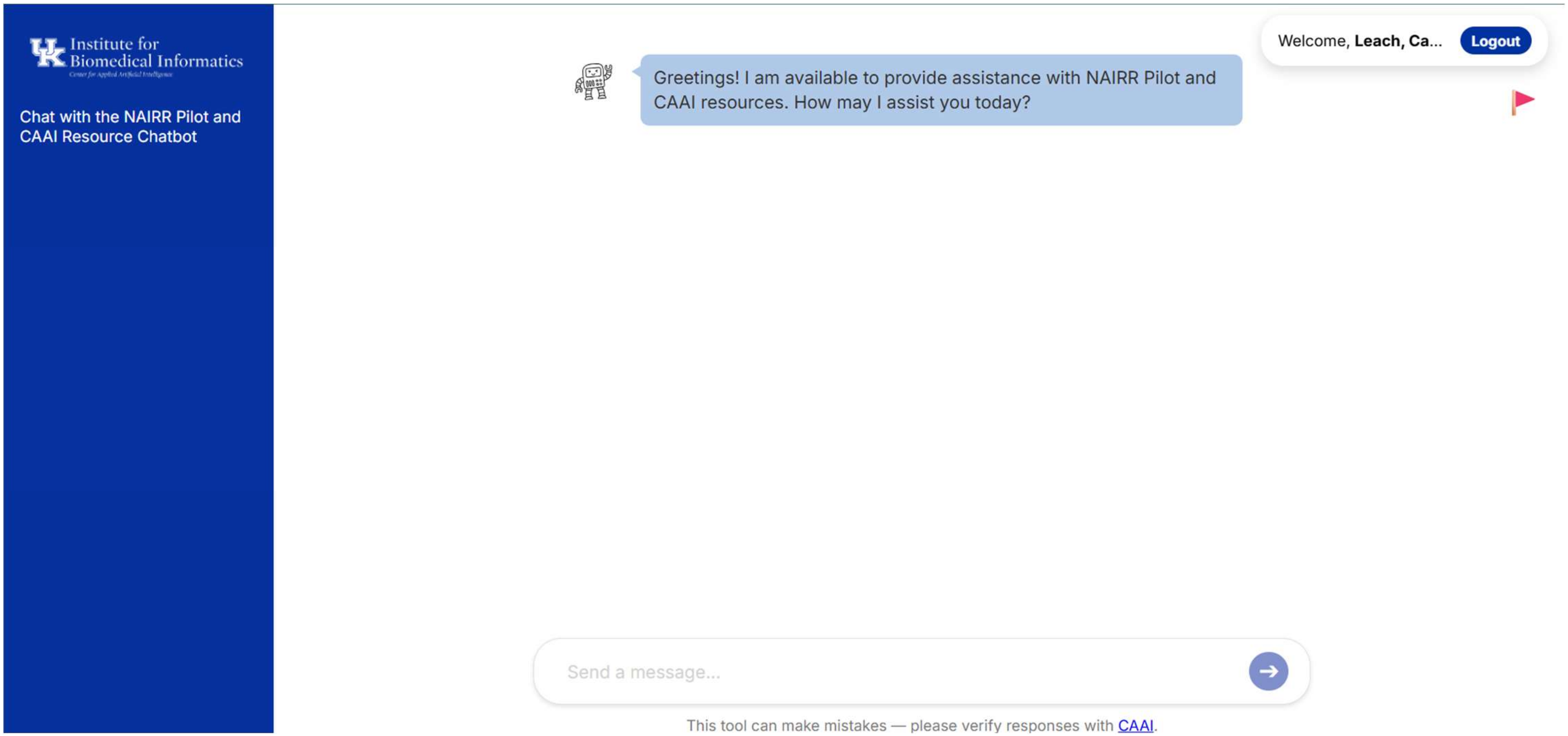
CPE Higher Education Matters

University of Kentucky

Online Forum

Henderson Community and Technical College

Virtual Assistant – *Coming Soon!*



NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

Demo Repo

GitHub

- 1_AI_Ready_Datasets.ipynb
- 2_ML_Fundamentals.ipynb
- 3_Computer_Vision.ipynb
- 4_LLMs.ipynb
- 5_Applications_and_Pipelines.ipynb
- 6_Multimodal.ipynb

github.com/Kentucky-Open-Science/CAAI-NAIRR-Pilot-Jupyter-Notebooks

NAIRR PILOT EXPANSION PROJECT: DEMOCRATIZING AI TRAINING AND ACCESS STATEWIDE

OFFICE HOURS

SECOND MONDAY EACH MONTH

11:00 AM - 12:30 PM

FEBRUARY 9, 2026

MARCH 9, 2026

APRIL 13, 2026

MAY 11, 2026

Registration required

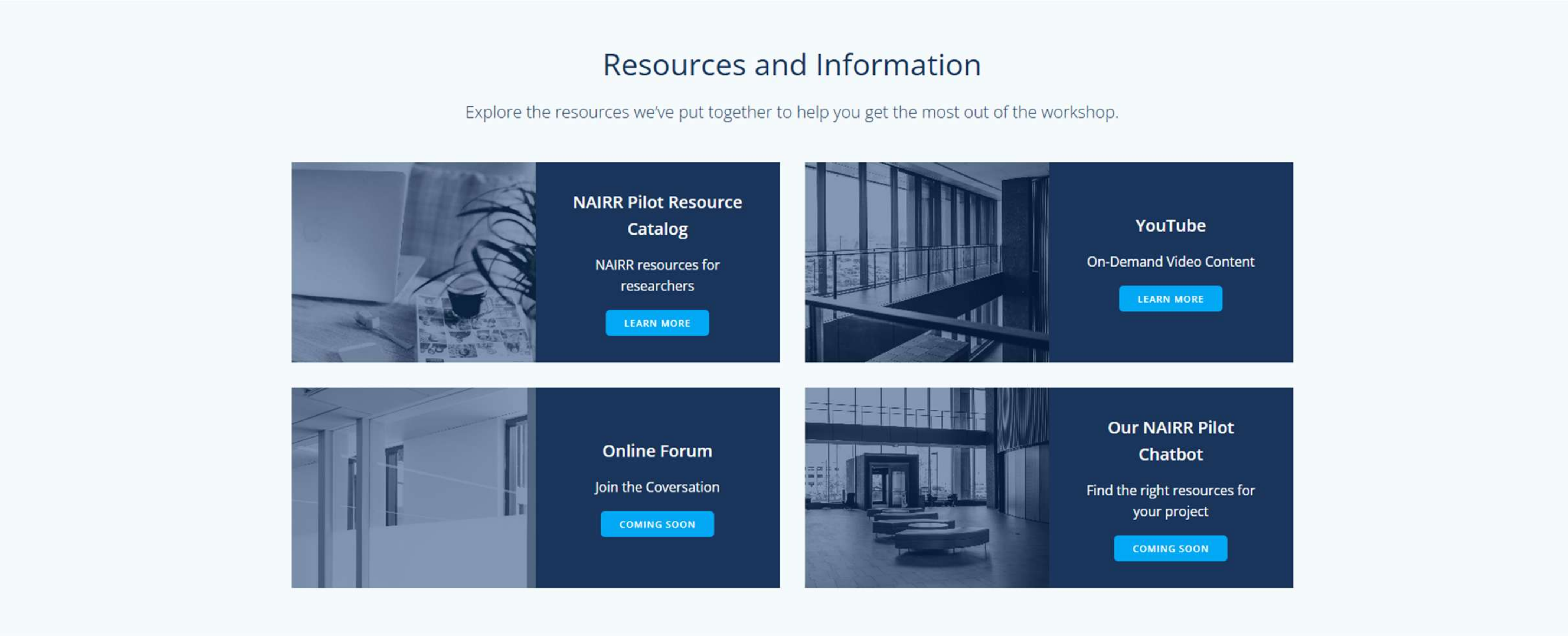
NAIRR Pilot · KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM · CPE Higher Education Matters · University of Kentucky

Open-Source

- CLASSify – Making Machine Learning Accessible
- LLM Factory – Institutionally Controlled Access to LLMs
- CAT-Talk – Automated Transcriptions with Timestamps and Speaker-Labels
- Forecaster – Evaluate and Explore Timeseries Data with Machine Learning
- Vision Foundry – Pretrain, Adapt and Deploy Foundational Vision Models

Democratizing AI Training & Access Statewide Webpage

https://caai.ai.uky.edu/about-us/news-and-events/nairr-pilot-expansion-project/



NAIRR Pilot | Kentucky Community & Technical College System | CPE Higher Education Matters | University of Kentucky

NAIRR Pilot

National Artificial Intelligence Research Resource Pilot

- Multi-agency, public-private *pilot* coordinated by the NSF
  new.nsf.gov/focus-areas/artificial-intelligence/nairr
- Connects U.S. researchers and educators to computing capacity, tools, data, and training resources for AI research
  nairrpilot.org/about
- Equitable access regardless of academic or non-profit organization size or type, funding sources, or academic domain of application
- Visit the NAIRR Pilot website for the latest information and to subscribe to the NAIRR Pilot monthly newsletter
  https://nairrpilot.org/

14 agencies and 28 industry and non-profit partners are contributing *in kind* state-of-the-art resources

DARPA
DOD
DOE
DoEd
FDA
NASA
NIH
NIST
NOAA
NSF
USDA
USGS
USPTO
VA

- AI2: Allen Institute for AI
- AMD
- Amazon Web Services
- Anthropic
- Cerebras
- Databricks
- Datavant
- EleutherAI
- Google
- Groq
- Hewlett Packard Enterprise
- Hugging Face
- IBM
- Intel
- **Lexset**
- Meta
- Microsoft
- MLCommons
- NVIDIA
- Omidyar Networks
- OpenAI
- OpenMined
- Palantir
- Regenstrief Institute
- SambaNova Systems
- Vocareum
- **Voltage Park**
- Weights & Biases

Research Resources

Computing

Datasets

Models, software, platforms

Educational/training opportunities

Collaborations

Courtesy of NAIRR Pilot Program

nairrpilot.org/pilotresources

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

NAIRR removes cost and access barriers so researchers can experiment, learn, and innovate

- Great ideas don’t always come with the required resources
- Individual subscriptions don’t always equate to research-scale infrastructure

NAIRR removes cost and access barriers so researchers can experiment, learn, and innovate

- "Tokens" – Small pieces of text that language models read and generate; measures how much a model is used
- "Models" – Powerful, pre-trained systems, like large language or vision models, that perform tasks such as reasoning, transcription, classification, or image analysis
- "Compute" – Infrastructure that run AI models; far more capacity than a typical laptop

What resources are available?

These resources represent leading-edge offerings from corporate and non-profit organizations

- Model & Inference Services
- Cloud Training Environments
- High-Performance GPU

Who are these resources for?

You!

No experience necessary!
No cost to you!

- Researchers and instructors from U.S. 2- or 4-year academic institutions or non-profit organizations *(and private sector if they have a federal research grant)*
- Any, or no, source funding
- Any application domain of AI in line with NAIRR and NSF priorities

Available Resources

The set of resources available through the NAIRR Pilot has been expanded to include additional resources funded by federal agencies as well as resources contributed by private and non-profit sector partners. Over time, we expect new resources will be added to the NAIRR Pilot, and some resources may be removed from the pilot as their available contributions are committed to projects.

Researchers are strongly encouraged to review the most up-to-date list of resources, which will always be available via the NAIRR Pilot Resource Catalog.

Filters

AI Capabilities

- [ ] Model inference services
- [ ] Model training services (GPU)
- [ ] Model training services (non-GPU)
- [ ] Research Collaboration
- [ ] AI tools and support

Resource Category

- [ ] Federal agency systems
- [ ] Private sector computational resource
- [ ] Private sector model access
- [ ] Other private sector contribution
- [ ] Classroom Platform

Resource Type

- [ ] Cloud
- [ ] GPU Compute

Resources

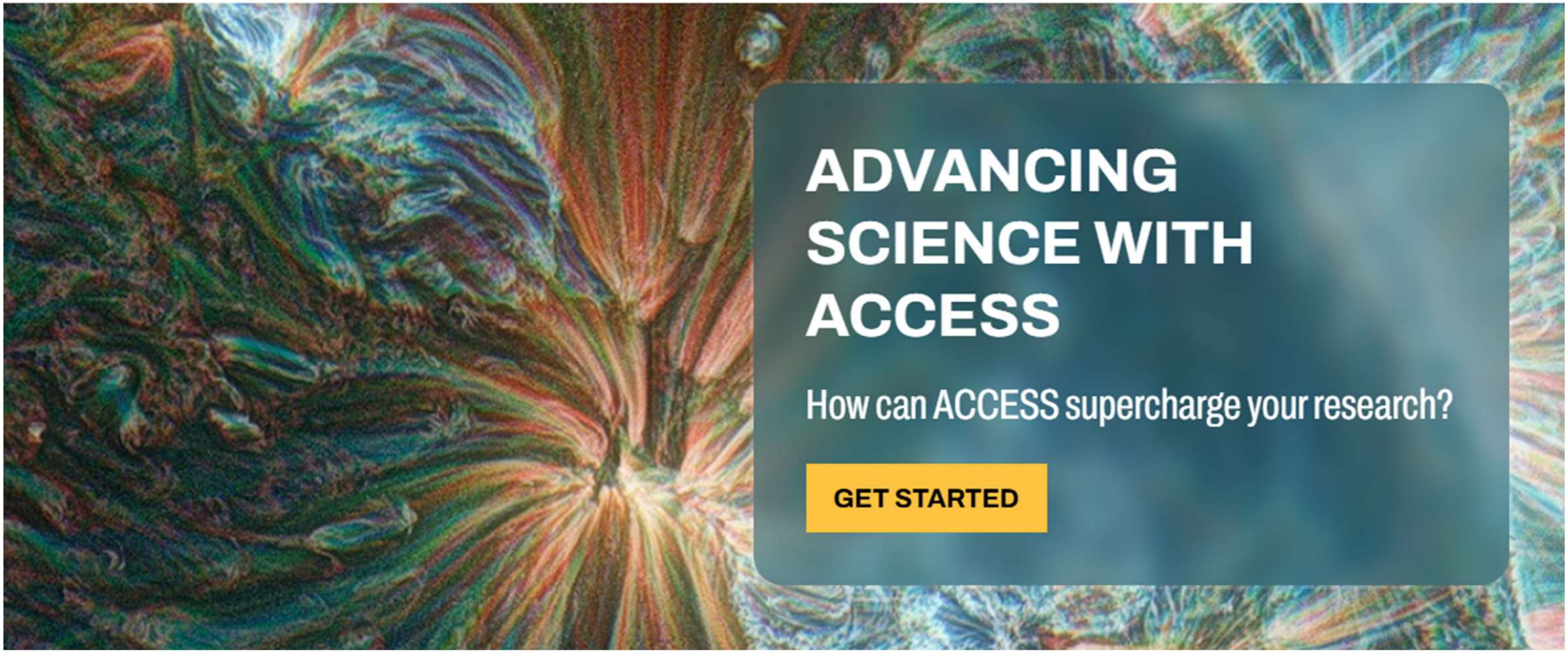
- Amazon Web Services
- Anthropic Model API
- Cerebras Wafer-Scale Engine 2 (CS-2) AI Accelerator
- Chameleon

Resource Type:	Cloud Platform
Organization:	University of Chicago
Units:	SUs
User Guide:	Link to User Guide
Features Available:	• Cloud

submit-nairr.xras.org/resources

NAIRR Pilot · Kentucky Community & Technical College System · CPE Higher Education Matters · University of Kentucky

NSF | ACCESS



https://access-ci.org/

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

How to request resources

- Visit https://nairrpilot.org/
- Under "*Current Opportunities*," select
  - Researcher Resources, for research projects
  - Classroom/Educators Resources, for teaching activities
- Full-Scale Requests require a 3-page proposal
  - Mention you attended this NAIRR Pilot Expansion workshop in your application!

Are there requirements?

- Research or course must have an AI component.
- Principal Investigator / Lead Instructor and all other users must perform work via an eligible organization physically based in the U.S.

nairrpilot.org/help/faq

NAIRR pilot focus areas

- Advancing AI methods that enable scientific discovery and improve AI interpretability, security and trust.
- Accelerating time to science and innovation through AI enabled automation, autonomy and novel design and control processes.
- Applying AI to use, share, or integrate sensitive data from multiple sources to enable new experimental methods and discovery.
- Advancing approaches for integrating simulations and AI.
- Creating or developing open-source AI tools, models, datasets, and methods.
- Training and educating the next generation AI-savvy workforce.

Other projects in other areas of AI research and applications, may secondarily be considered for allocation.

nairrpilot.org/opportunities/allocations

NAIRR Community Resources

YouTube @NAIRRPilot

Recordings of past events & showcases
Youtube.com/@NAIRRPilot

NAIRR Pilot

Registration for NAIRR Pilot Office Hours
nairrpilot.org/help/office-hours

GitHub NAIRRProgram

Hosts the community-learning-resources repository
github.com/NAIRRProgram

Workshop Goals

- Understand core AI concepts and how AI is used in real-world settings
- Explore how AI is shaping research and the workforce in Kentucky
- Gain confidence in foundational ideas like machine learning, large language models, and multimodal AI
- Experience applications of AI through the lens of research workflows
- Learn how to access local and national AI computing resources through the NAIRR Pilot
- Spark new ideas and collaborations across AI and related research fields
- Connect Kentucky researchers and innovators to grow a shared AI ecosystem

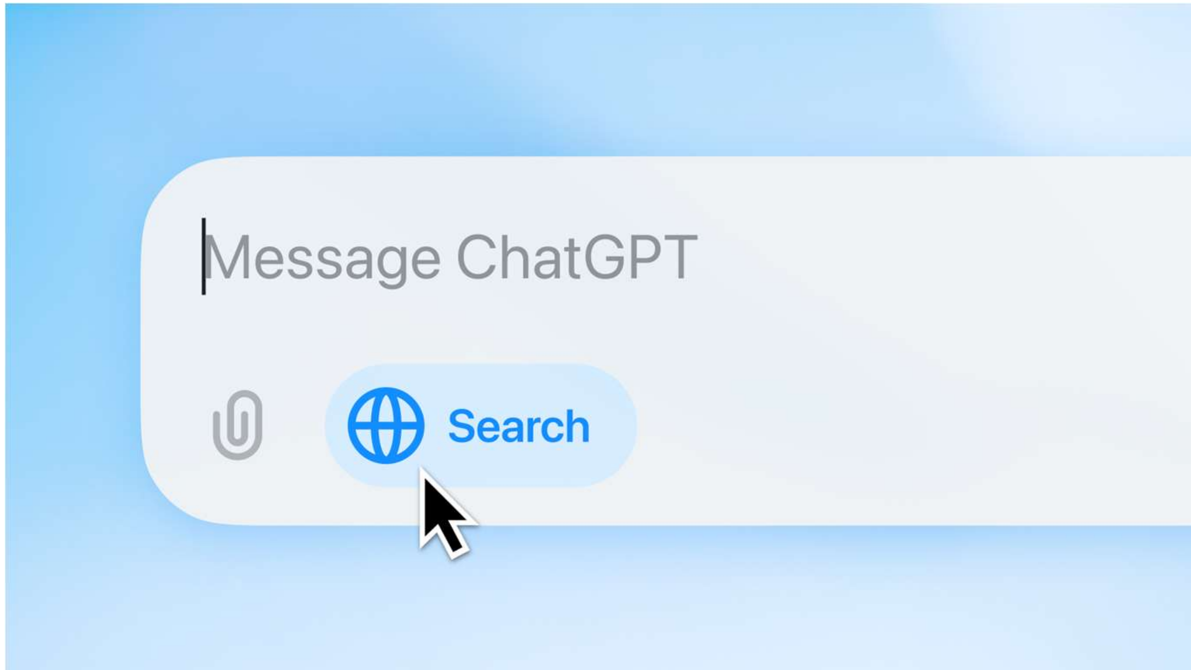
Introduction to AI & AI-Ready Data

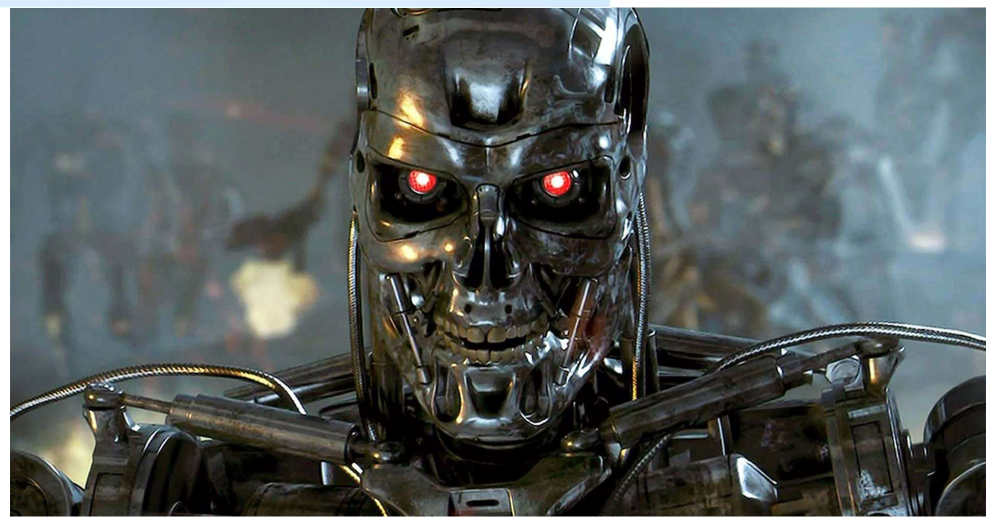
Aaron Mullen, MS

Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

What do you think of when you think of AI?





NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

AI is a Moving Target

- AI is more of a broad category than a specific goal
- Optical Character Recognition (OCR) can convert images of text/tables into raw information
  - –Example: scanning a check
  - –When OCR was developed, it seemed like “magic AI”
  - –Now, it’s more or less just a software algorithm
- Thus, AI can have many different meanings and applications

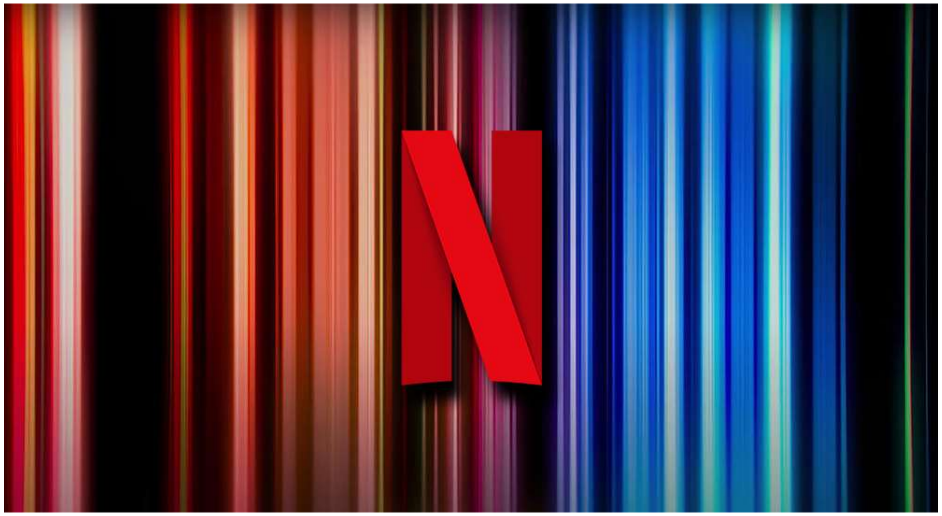
Traffic Predictions

- Route planning apps don't just use algorithms to find the shortest route between two places
  - This could be inaccurate because traffic patterns can change during your drive
- Historical traffic patterns and real time user updates can make predicted ETAs more accurate by forecasting future conditions

https://blog.google/products-and-platforms/products/maps/google-maps-101-how-ai-helps-predict-traffic-and-determine-routes/

Content Personalization

- Streaming services use AI to personalize recommendations to users based on their interests
  - But this goes beyond just modifying what shows are displayed on the front page
- To promote the last season of House of Cards, Netflix created 10 different trailers that emphasized different aspects of the show
  - The trailer that was shown to a user on the interface depended on that user's interests



https://research.netflix.com/research-area/recommendations

Improved Crop Management

- Modern John Deere tractors are equipped with cameras that utilize advanced computer vision systems
- These systems have been trained to identify weeds and filter out regular crop plants
  - Herbicide can be intelligently sprayed only where its needed, protecting crops and reducing chemical usage

https://about.deere.com/en-us/our-company-and-purpose/technology-and-innovation/sense-and-act

So overall, what is AI used for?

- AI is common in both everyday uses (home voice assistants) and groundbreaking technical advances (predicting protein structures to accelerate drug discovery)
- What do all of these uses have in common?
  - Analyzing and making sense of data
  - Responding to unique and varied situations
  - Automating manual burdens to improve outcomes

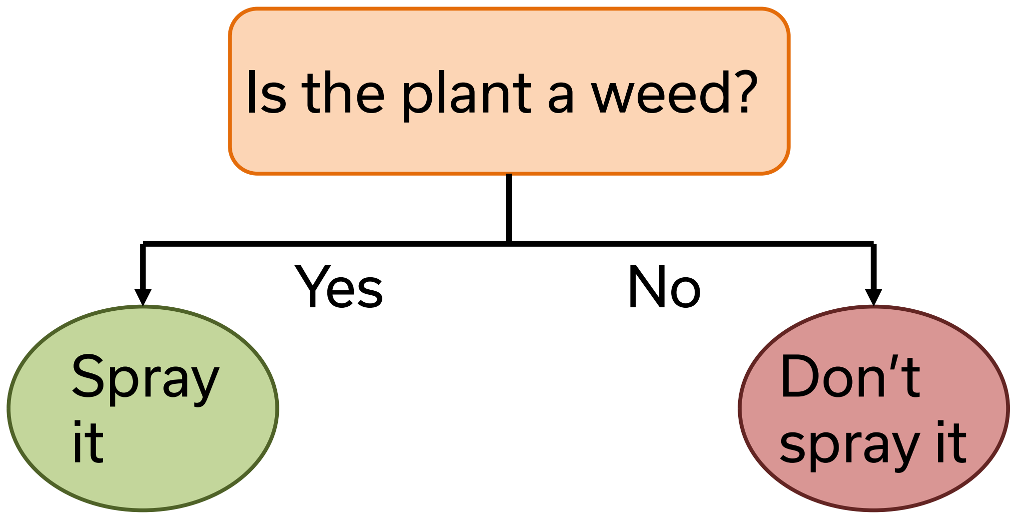
How did we get here?

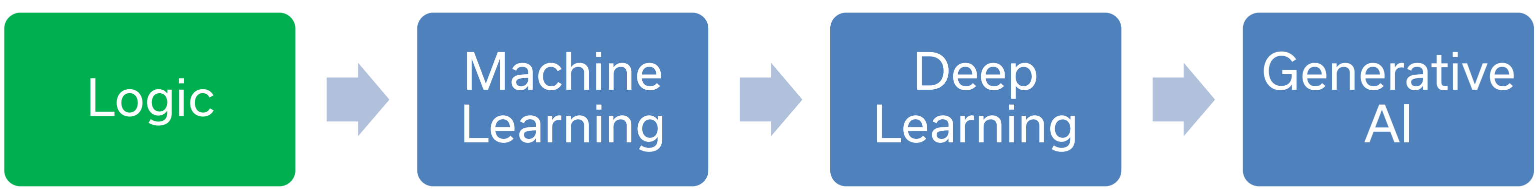
- Let's look at how decision-making systems were developed over time to lead to modern AI

Logic → Machine Learning → Deep Learning → Generative AI

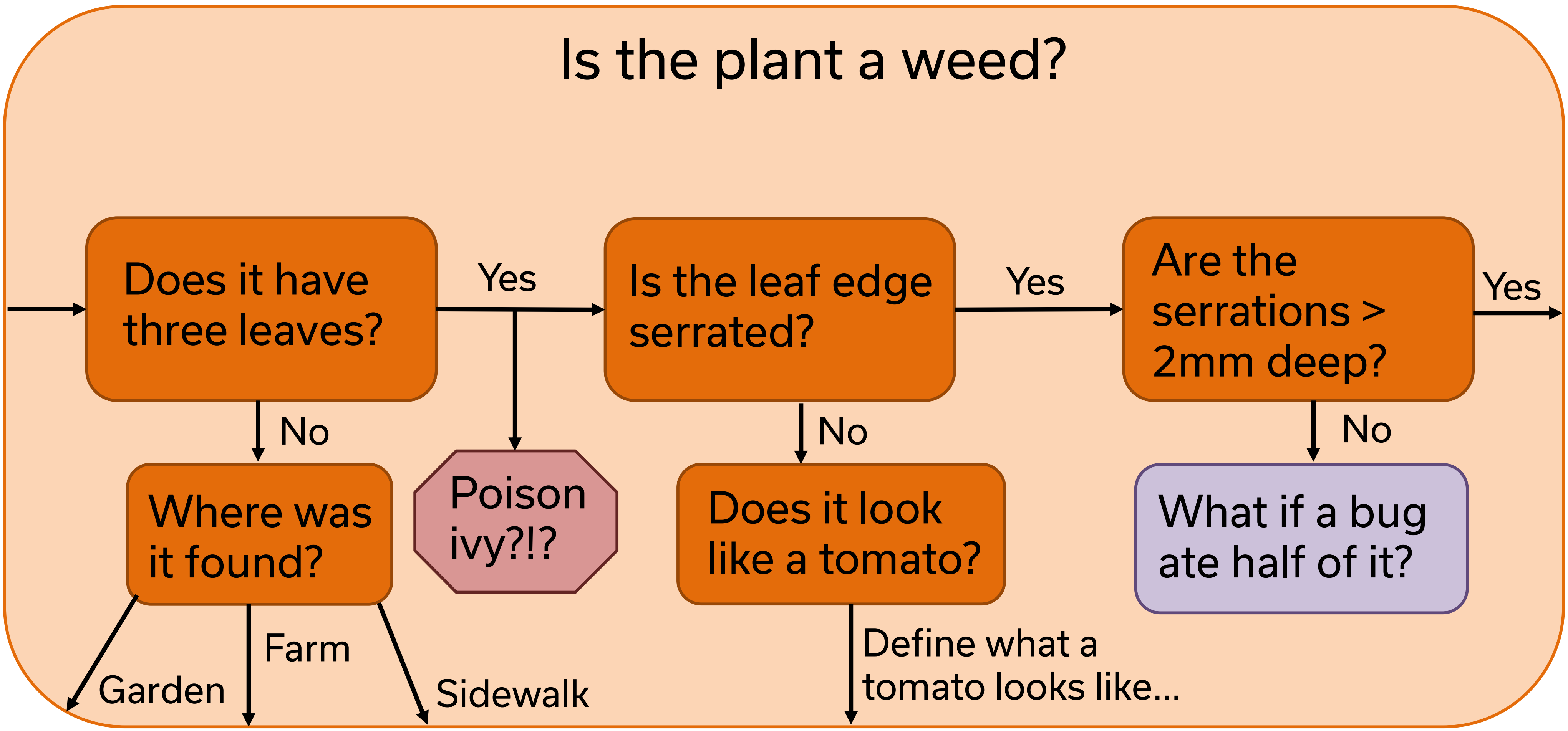
The Age of Logic (1950s-1980s)

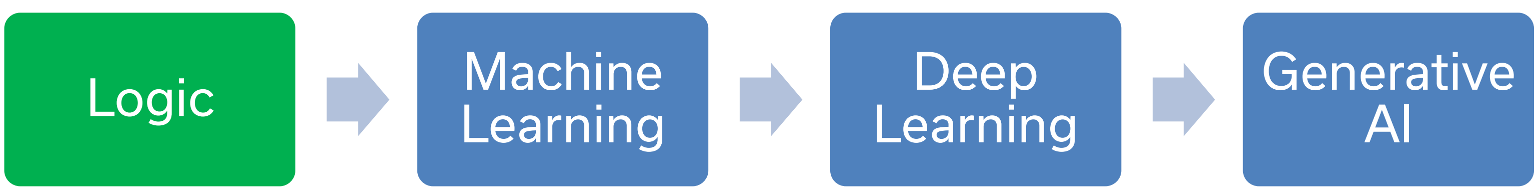
- The first computers were being developed, and with them, techniques for computer-based decision-making
- Systems were meticulously coded in an "if-then" format of explicit rules





The Age of Logic (1950s-1980s)

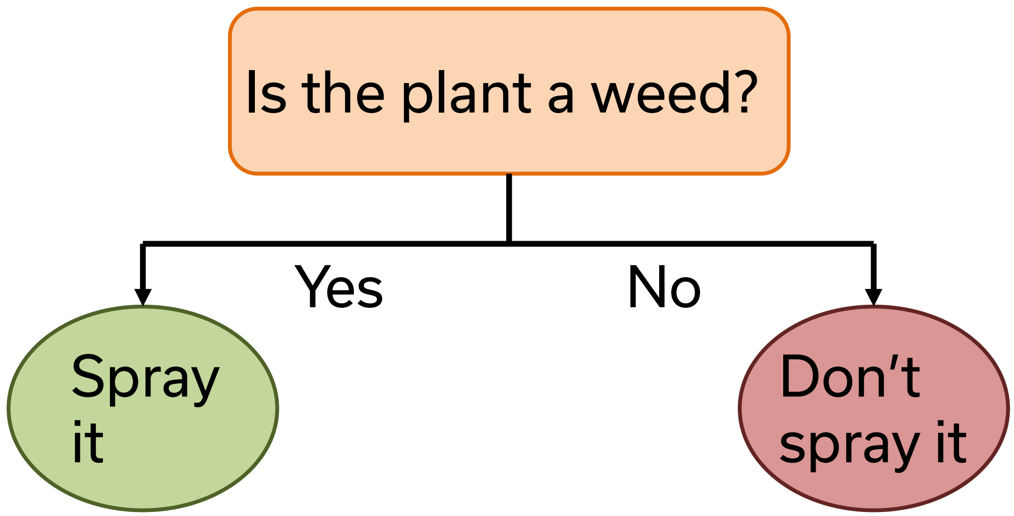


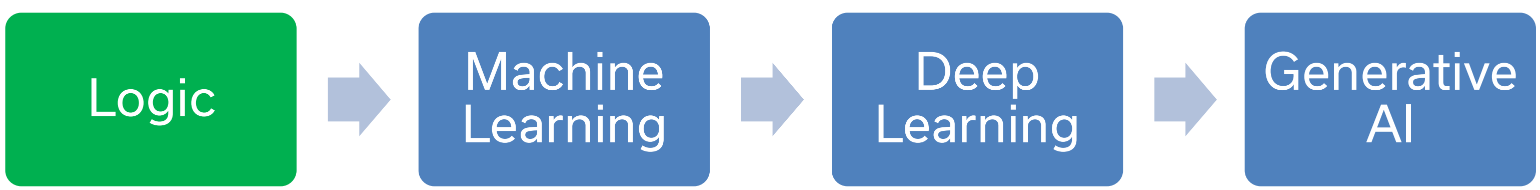


NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

The Age of Logic (1950s-1980s)

- These systems are very rigid and require definitions ahead of time for every possible scenario that could be encountered
- But they proved that computers can make logical decisions in narrow fields using human knowledge



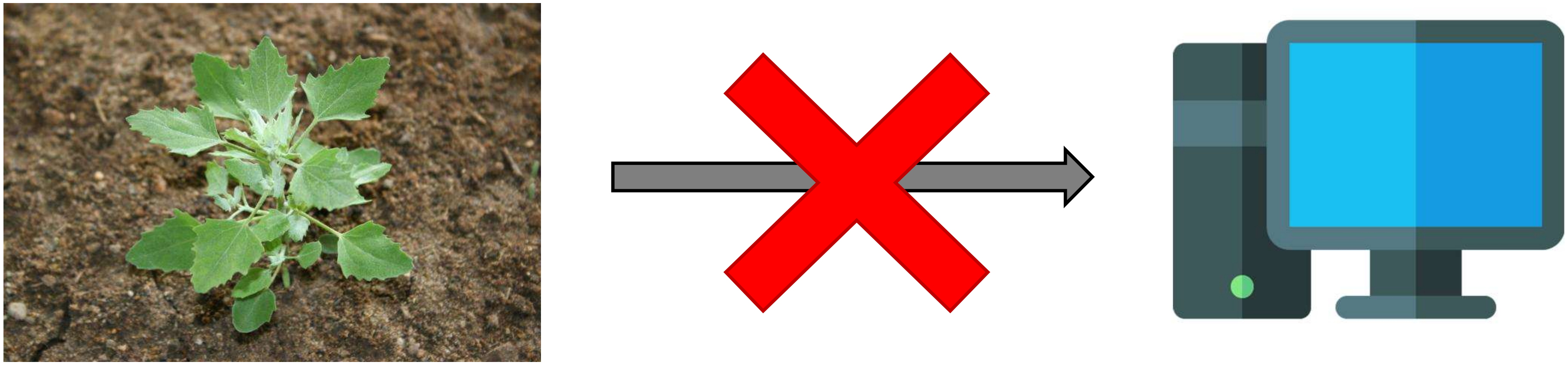


The Age of Machine Learning (1980s-2010s)

- Computers have progressed more and can make determinations without definitions of specific rulesets
- However, computers are too slow to process much raw data
- So, feature engineering is still required to limit the data the computer has to analyze
  - –Must define what aspects of the problem are important for the computer to consider

Logic → **Machine Learning** → Deep Learning → Generative AI

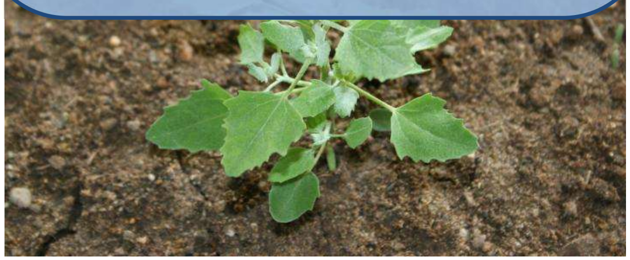
The Age of Machine Learning (1980s-2010s)

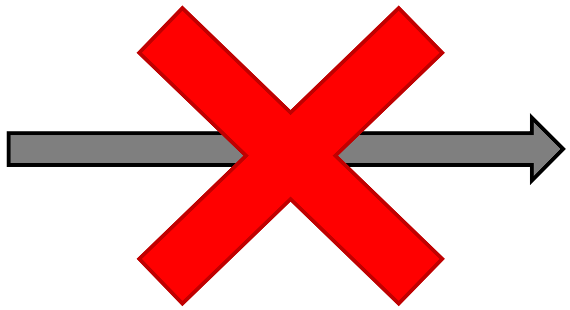


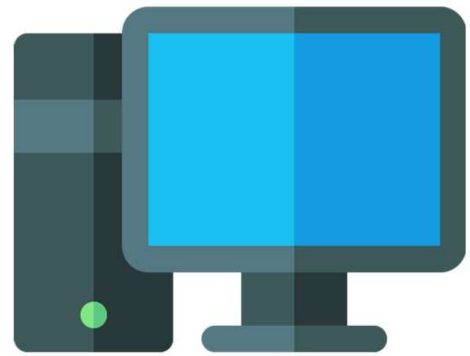
Logic → Machine Learning → Deep Learning → Generative AI

The Age of Machine Learning (1980s-2010s)

Stem Length: 6 in
Number of Leaves: 8
Leaf Serrations: Yes
Location: Garden
Taproot: Yes
...



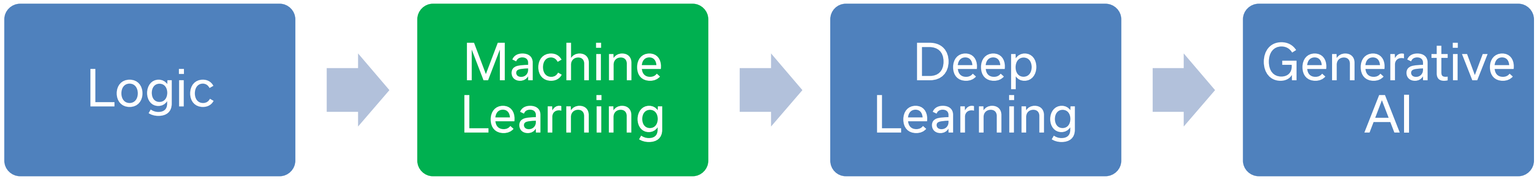




Logic → **Machine Learning** → Deep Learning → Generative AI

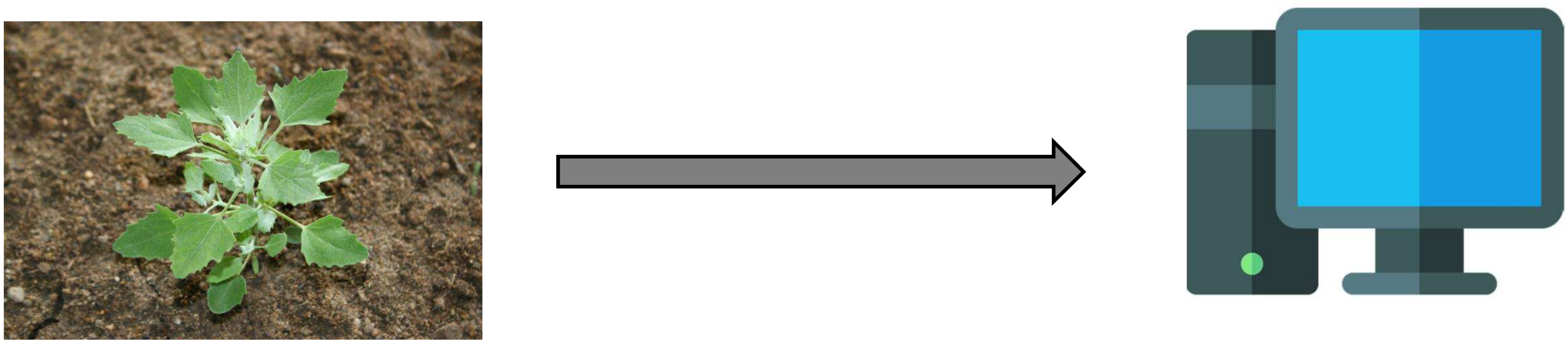
The Age of Machine Learning (1980s-2010s)

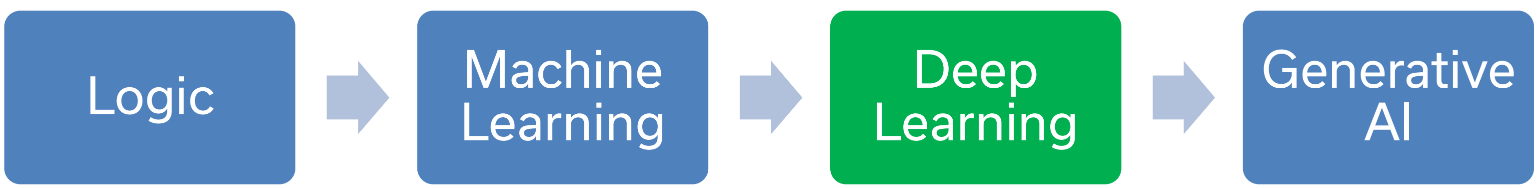
- More flexibility than rule-based systems because computers can determine which features are important for the task
  - Conditions for every possible scenario no longer need to be defined
  - New models like support vector machines and simple neural networks
- Human involvement is still required to define features of data



The Age of Deep Learning (2010s-2020s)

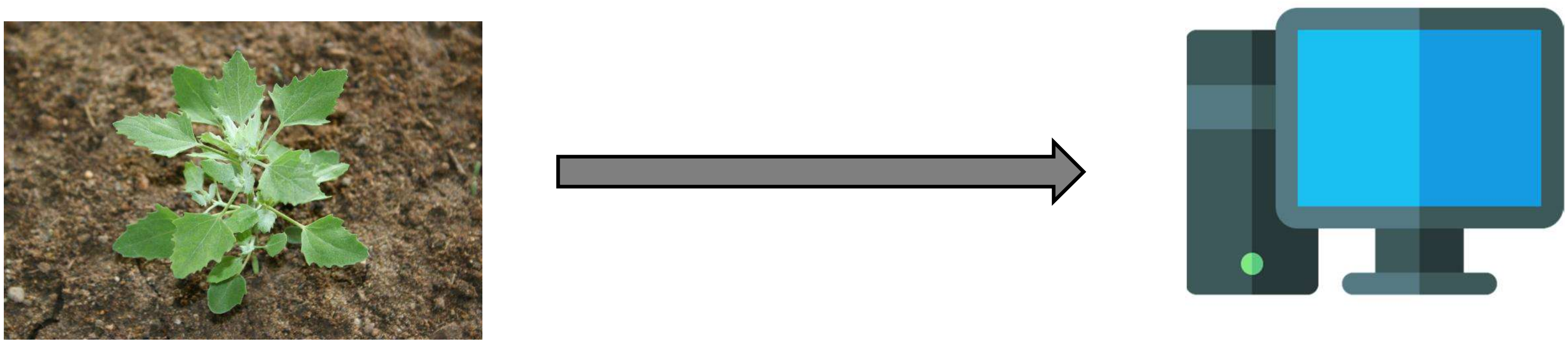
- Computers and GPUs have advanced to allow for much more complex interpretations of data
- Now, raw data can be fed directly into a model
  - Deep neural networks and transformers can determine their own sets of features to maximize performance





The Age of Deep Learning (2010s-2020s)

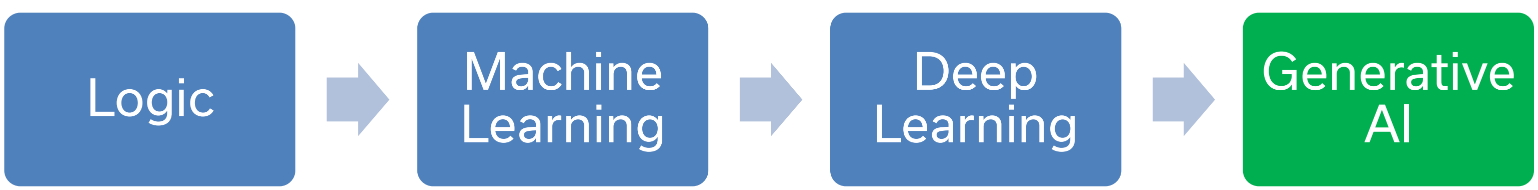
- The ability of computers to now process and understand these huge quantities of data prompted a new idea:
  - If our model understands everything about what defines a plant as a weed based on images, would it be able to generate its own image of a weed?



Logic → Machine Learning → **Deep Learning** → Generative AI

The Age of Generative AI (2020s)

- New types of models could be trained using Generative Adversarial Networks (GANs)
  - Two models: a generator (creates fake images/text) and a discriminator (tries to classify which objects are real or fake) pitted against each other
- Transformers developed with "attention" mechanisms
  - A model can look at an entire sentence or image and understand how every aspect relates to each other at the same time
  - Models can be trained on huge amounts of data and learn from all of it



Where do we go from here?

- The meaning of AI and machine learning has changed significantly over time
  - Techniques possible today seemed like science fiction 10 years ago
- The development of new AI techniques has only gotten faster
- How will AI continue to change?

AI Today is Narrow

- Narrow AI: Designed for one specific task
- Can sometimes outperform human expertise/ability at those tasks
  - AI systems tasked with detecting breast cancer from X-rays have been able to reduce both false positives and false negatives compared to human doctors
- However, these systems have no intelligence outside of their own domains
  - AI that detects breast cancer cannot be used to identify weeds

General AI

- Artificial General Intelligence (AGI): AI that can accomplish new tasks in completely unfamiliar contexts without human intervention
  - Able to reason, plan, and solve problems with same flexibility as a human
- AGI is purely theoretical and does not exist yet
- However, multimodal systems are expanding the definition of "narrow" AI

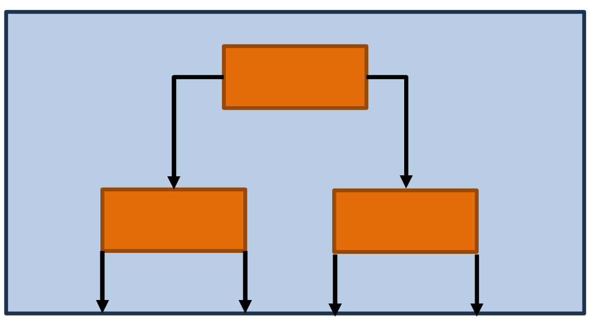
Important Caveats About AI

AI doesn't "know" anything!

- AI that generates text doesn't "understand" what it's saying in the same way a human does
  - As it's writing, it's using an algorithm to predict the most likely next word at each step, based on the data it was trained with
- AI that identifies breast cancer doesn't know what breast cancer is
  - It knows the data features that make a certain response more or less likely to be rewarded as "correct" based on training data
- This doesn't mean that AI can't be useful, just that we must be cautious with interpretations

Transparency and Trust with AI

- Ethical use of AI is crucial as this technology develops
- Training data:
  - –When AI is trained, it "copies" the data it is developed with
  - –Thus, human biases present in the data will be "copied" as well
  - –This will be discussed in more detail later
- AI tends to lack explainability (why does it predict something?)
  - –This is a problem if AI is to be used with high stakes decision-making



vs.

Black Box

Hallucinations

- A chatbot on the Air Canada website told a passenger that he could apply for refunds for a bereavement fare after traveling
- After attempting to do so, he was told bereavement rates could not retroactively apply to completed travel by the company
- Air Canada was forced to pay the customer the difference because the chatbot had misled him
- Raises interesting questions: who is responsible for the content AI produces?

https://www.theguardian.com/world/2024/feb/16/air-canada-chatbot-lawsuit

Ethics of Generative AI

- Deepfakes: how can we trust media we view?
  - –Believing something to be real that is fake
  - –Believing something to be fake that is real
- Copyright conflicts
  - –OpenAI was trained on millions of New York Times articles
  - –Is that learning or plagiarizing?
- AI is neither good or bad, but how we use it can be
- Today, we will learn much more about how to use AI effectively and ethically

Ethics of Generative AI

- AI is powerful which makes it frightening
  - It's important to remember that this power can be and is used for good too

Giving independence to blind people, allowing them to take pictures of anything and listen to conversational descriptions

Allowing patients with ALS to record their voice before losing it to preserve their identity with generative speech

Providing useful tutors to students who can't afford private help

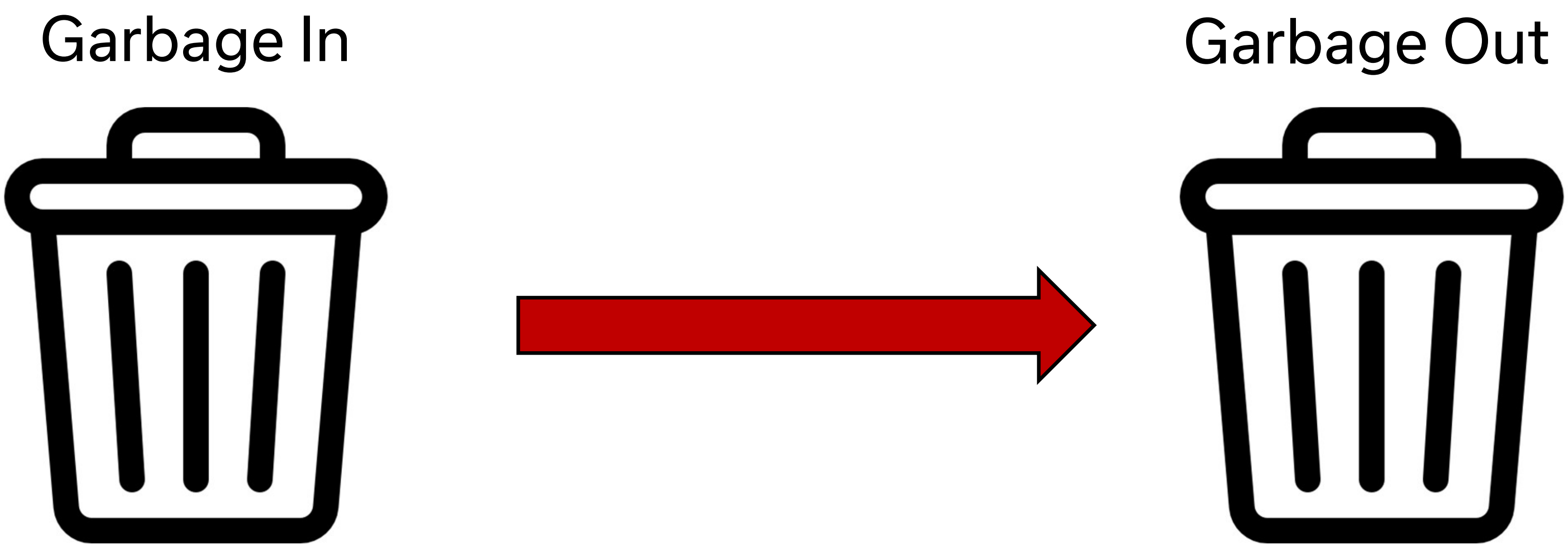
Forecasting floods and natural disasters in developing countries to provide equity in safety

How are these AI systems created?

Data!

Data is important!

- All AI systems require data for training and evaluation
  - –AI learns from data rather than pre-defined rules
- Bad data will lead to a bad AI model



- So what makes data bad?

Rule 1: Data must be sufficient

- If you don't have enough (good) data, any AI model trained with that data will underperform
  - Won't generalize well to unexpected scenarios
  - Systems only have knowledge of what they were trained with

Rule 1: Data must be sufficient

- In 2016, Tesla' autopilot caused a fatal accident when it failed to detect a tractor trailer turning in front of the car
  - –The white side of the trailer blended in with the brightly lit sky
  - –This rare condition was not accounted for in the autopilot's training data

https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk

Rule 1: Data must be sufficient

- So how much data is sufficient?
  - It depends!

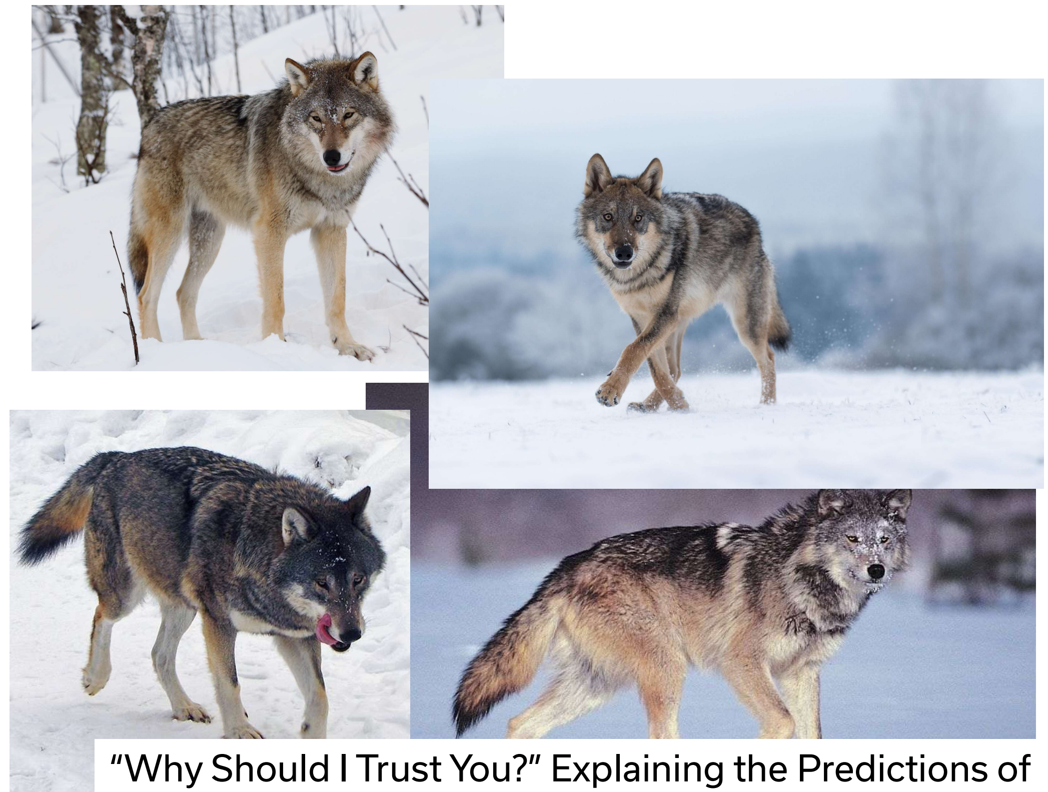
How narrow is the AI's task?	• Self driving car models must account for many more possibilities than models that scan X-rays
What's the domain of the AI?	• Images vs text vs numbers
How complex is the problem?	• More data is required to account for uncommon/difficult tasks

Rule 2: Data must be free of confounding variables

- Researchers built a model to classify images of huskies and wolves

Model says: HUSKY

Model says: WOLF



"Why Should I Trust You?" Explaining the Predictions of Any Classifier. Ribeiro M, Singh S, Guestrin C.

Rule 2: Data must be free of confounding variables

- They found that in testing, the model did not work great

Model says: WOLF

Model says: HUSKY

Rule 2: Data must be free of confounding variables

- Could the model be learning something unintended?

Model says: HUSKY

Model says: WOLF

Rule 2: Data must be free of confounding variables

- Rather than being able to tell huskies from wolves, the model could just tell snowy landscapes from grassy ones
- Lesson: closely examine all aspects of your data
  - –Consider factors aside from your primary focus that could affect results

Model says: WOLF

Model says: HUSKY

Rule 3: Data must be unbiased

- AI will reflect any biases present in the training data
- These biases must be considered and removed to create fair and equitable AI systems
- AI requires human data, and unfortunately, human data is very rarely completely unbiased

Rule 3: Data must be unbiased

- Amazon wanted to automate recruiting new employees
  - Candidates would be ranked based on their resumes
  - Training data consisted of past employee performance, as well as who got hired during previous recruitments vs who did not
- They discovered this recruitment tool immediately favored male candidates over female candidates
  - Some resumes would be discarded simply because the candidates went to women's colleges or participated in women's clubs
- The AI system itself was not inherently discriminatory
  - But the data that the system was trained on (Amazon's past hiring practices) contained these biases, and the system simply reflected those

https://www.hubert.ai/insights/why-amazons-ai-driven-high-volume-hiring-project-failed

Rule 4: Data must be appropriate in scope

Rule 4: Data must be appropriate in scope

- IBM wanted Watson's capabilities to extend to real-world problems
- They developed Watson for Oncology to analyze medical literature, treatment protocols, and emerging research
  - It could recommend treatment plans and assist oncologists
- It was trained on data from Memorial Sloan Kettering Cancer Center
  - High quality data, but reflected only patient demographics and treatment practices for that specific area

https://www.henricodolfing.ch/case-study-20-the-4-billion-ai-failure-of-ibm-watson-for-oncology/

Rule 4: Data must be appropriate in scope

- The problem: Watson for Oncology was heavily marketed in traditionally under-served regions like Thailand, India, and South Korea
- It would give unsafe or irrelevant recommendations because important demographics and treatment practices were different
  - Certain drugs may not be available
  - A different patient population will react differently to different treatments
- Watson for Oncology was discontinued in 2023

Rule 4: Data must be appropriate in scope

- The lesson: the scope of your data should match the scope of your AI system's intended usage
- If Watson had only been used for MSKCC patients, it would have likely been seen as a success
  - But expansion to new patient populations required new data
- A model trained on local data only is not inherently bad!
  - But the scope of the system's application must remain local

So good data is...

Large enough for your domain/problem

Free from additional confounding variables

Without human biases

Appropriate to the scope of your usage

Good Data Example

- Google developed an AI-powered diabetic retinopathy detection model
- It was trained on data from underserved countries like Thailand and India with the hope of providing care where healthcare resources are more scarce
- The model achieves 95% accuracy, slightly higher than retina specialists
- Over 600,000 screenings have already been performed with this system
  - –Millions more are planned over the next decade

https://blog.google/company-news/inside-google/around-the-globe/google-asia/arda-diabetic-retinopathy-india-thailand/

What made this dataset good?

It was large- initial training data consisted of over 128,000 images

It was high quality- standardized labeling determined by expert teams to ensure accurate ground truth

It was appropriate to its scope- the models have only been applied in countries that data was used from for training

What does AI-ready data actually look like?

- Data can take many different forms depending on the domain and intended goal
- Three main types:
  - Structured
  - Unstructured
  - Semi-structured

Structured Data

- Typically takes the form of rows and columns in a spreadsheet or relational database
- Quantitative rather than qualitative

Structured Data Use Cases

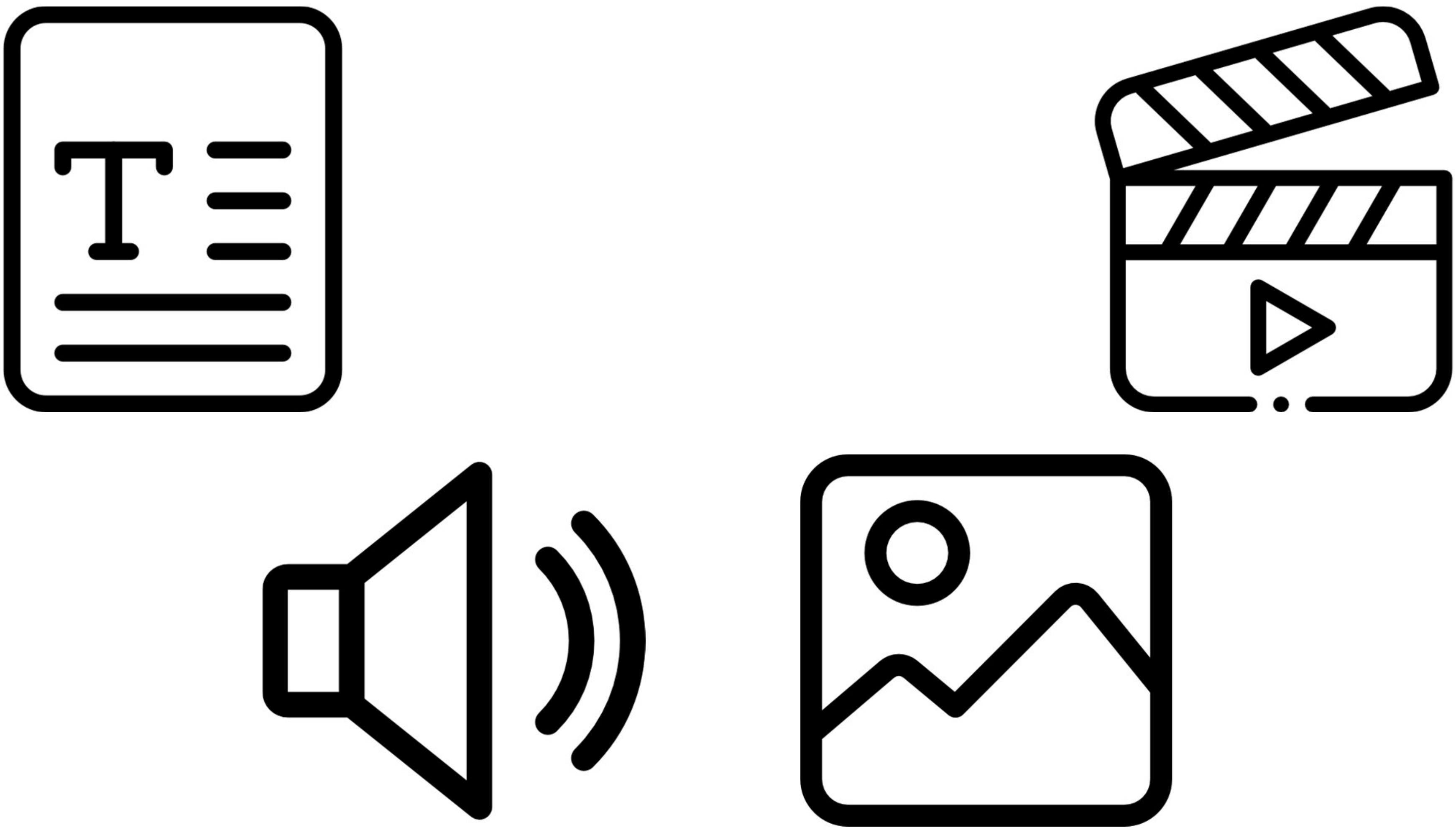
Type of Application	Type of Data	Example
Forecasting	Time series data- measurements gathered at regular intervals in a specific order	A hospital wants to forecast how many patients will come to the ER on a given day to determine staffing and transfer decisions
Classification	Tabular, cross-sectional data- snapshot of entities at a single point in time	An educator wants to predict which students will pass/fail their final exam, based on prior performances, so assistance can be given
Regression	Similar to classification, but target is a number rather than a category	Ride-sharing apps use regression algorithms to determine dynamic pricing based on a variety of factors
Anomaly Detection	Highly granular data generated by actions or events, where a small proportion of rows/events need to be flagged	Banks analyze credit card purchases to determine when transactions are likely to be fraud

Structured Data Pros/Cons

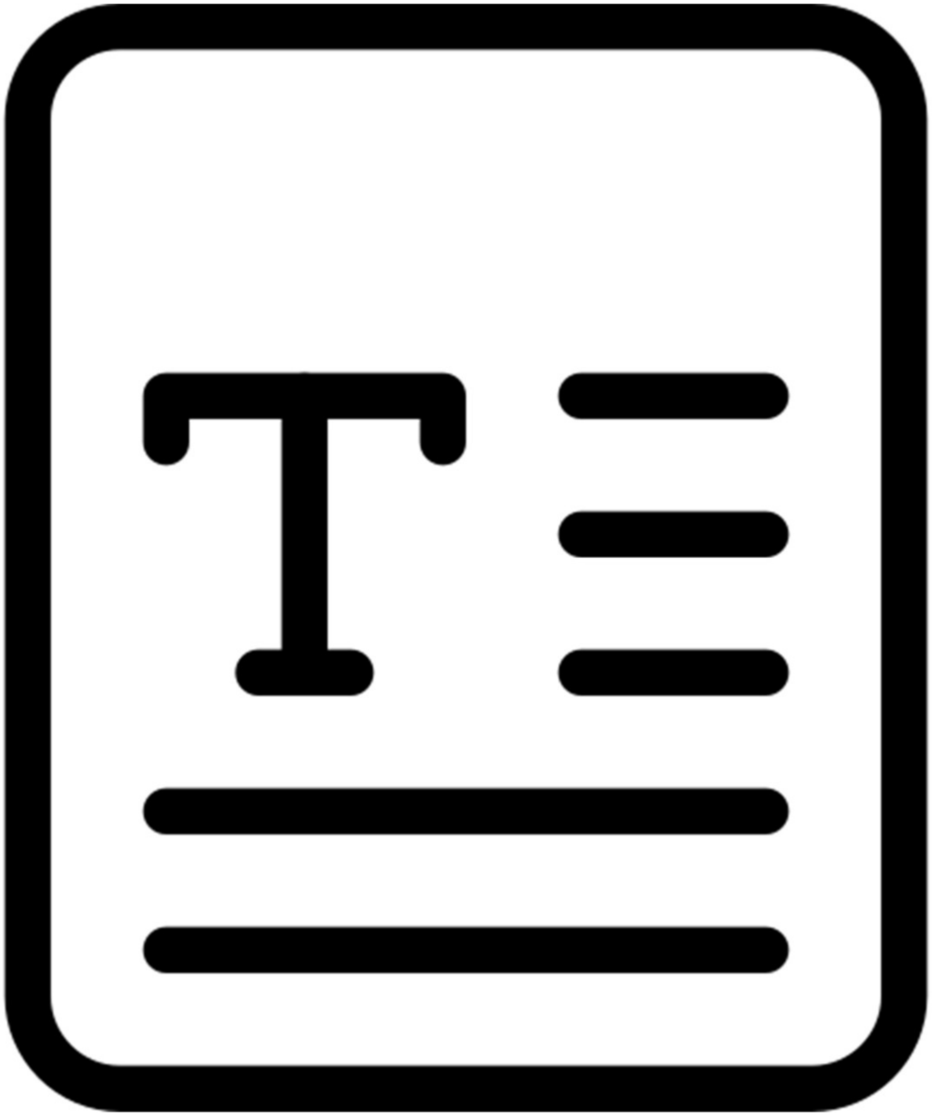
Advantage	Disadvantage
Easy to use due to rigid structure	Much of the world's real data is not gathered in a structured way
Easy to view/interpret as a human	Lots of pre-processing can be required to format data correctly and consistently
Applied to a huge variety of use cases and types of problems	Can fail due to small imperfections, like missing data
Many machine learning techniques are built around structured data	Doesn't work well with most kinds of data aside from numbers/booleans

Unstructured Data

- Data that cannot be stored in a traditional database structure

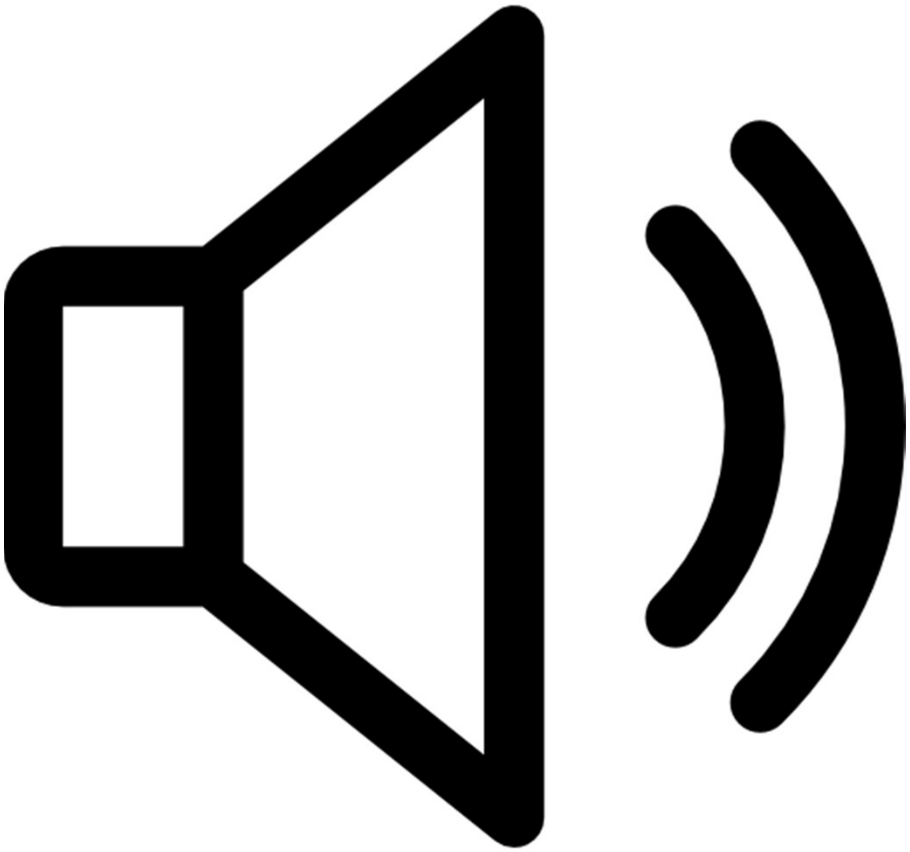


Unstructured Data



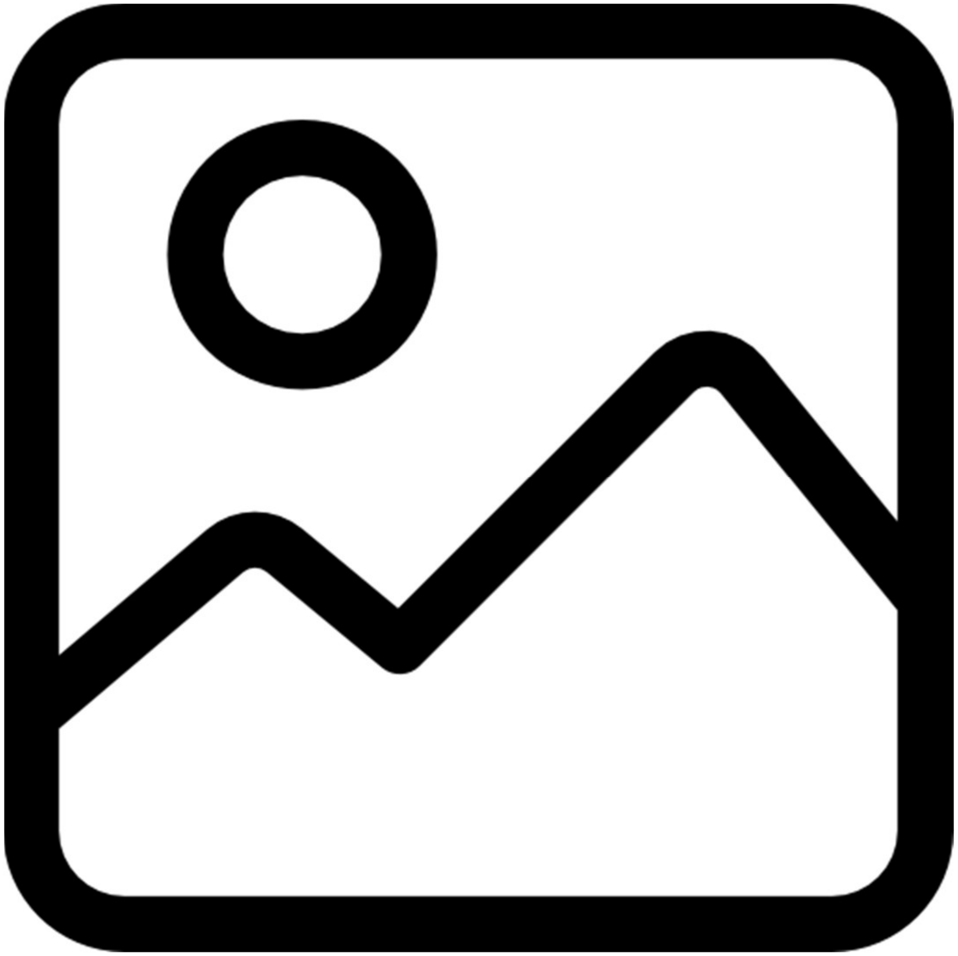
- Text data
- Interpretable to humans but viewed by machines as sequence of tokens (words)
- Words/phrases can be embedded into numbers
- Example: a hotel chain analyzes thousands of user reviews to flag complaints without a human reading them

Unstructured Data



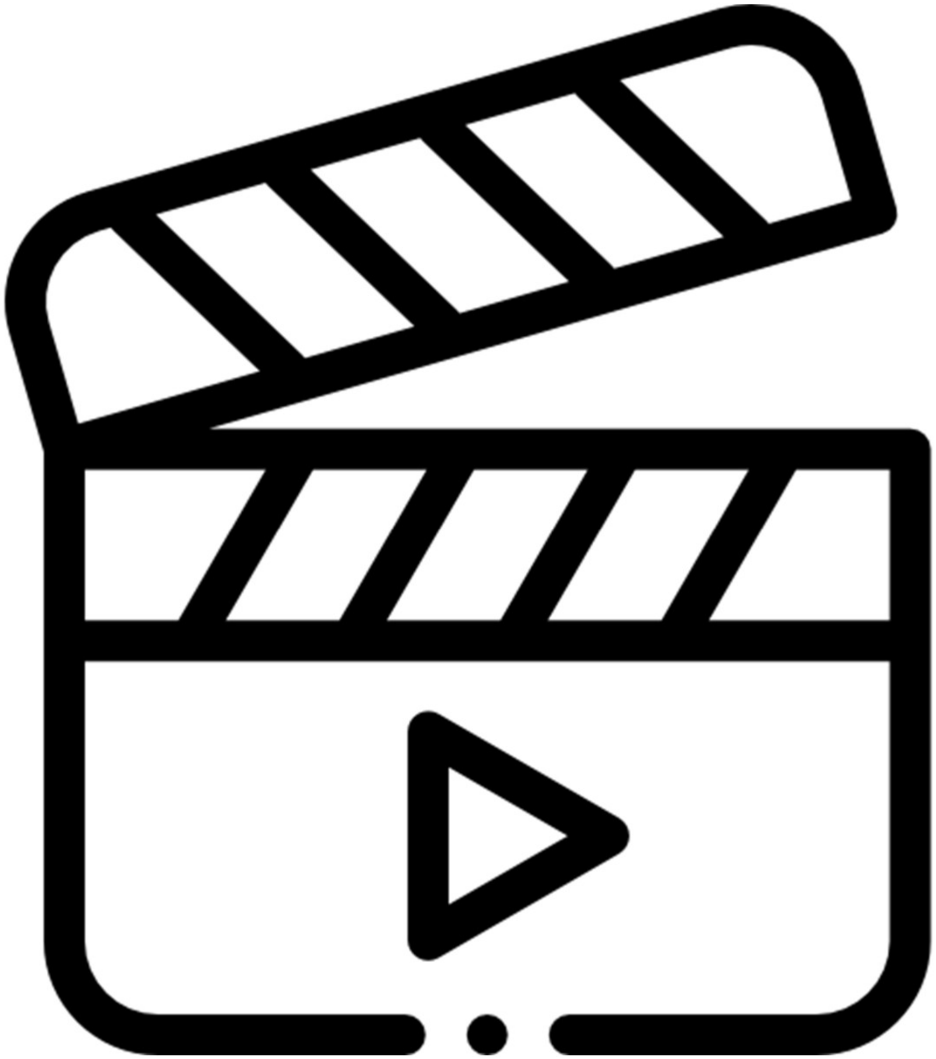
- Audio data
- Continuous signal representing air pressure changes
- Measured with attributes like frequency and amplitude
- Example: software that automatically transcribes meetings by listening to dialogue

Unstructured Data



- Image data
- To a computer, just a massive grid of numbers representing colors at each pixel
- The goal may be to classify an entire image, or to segment specific parts of it
- Example: X-ray images can be used to identify potential tumors

Unstructured Data



- Video data
- Each frame of the video can be analyzed the same as an image
- The combination of these frames, and changes over time, can be used to identify motion
- Example: video monitoring of laboratory rats can be used to identify seizures

Unstructured Data

Pros:

Very common and easy to obtain

Covers a huge range of data types and domains

Understandable and clear to humans

Cons:

Some limited use cases due to lack of structure

Burdensome to manually review/analyze

Frequently very computationally expensive to use

Semi-Structured Data

- Doesn't follow a rigid table format, but it does follow a looser structure that can still be analyzed and parsed
- Can take on a variety of forms

Semi-Structured Data

<div class="content">

<h1> Websites are built using this HTML format </h1>

<p> It's flexible and allows for a lot of variation, but there are still key structured elements, such as nesting and tags </p>

<h2> Example: </h2>

<p> An AI that scrapes the web for ticket prices needs to be able to parse elements from browser HTML code </p>

</div>

Semi-Structured Data

{

"title": "JavaScript Object Notation (JSON) is a very common semi-structured data format",

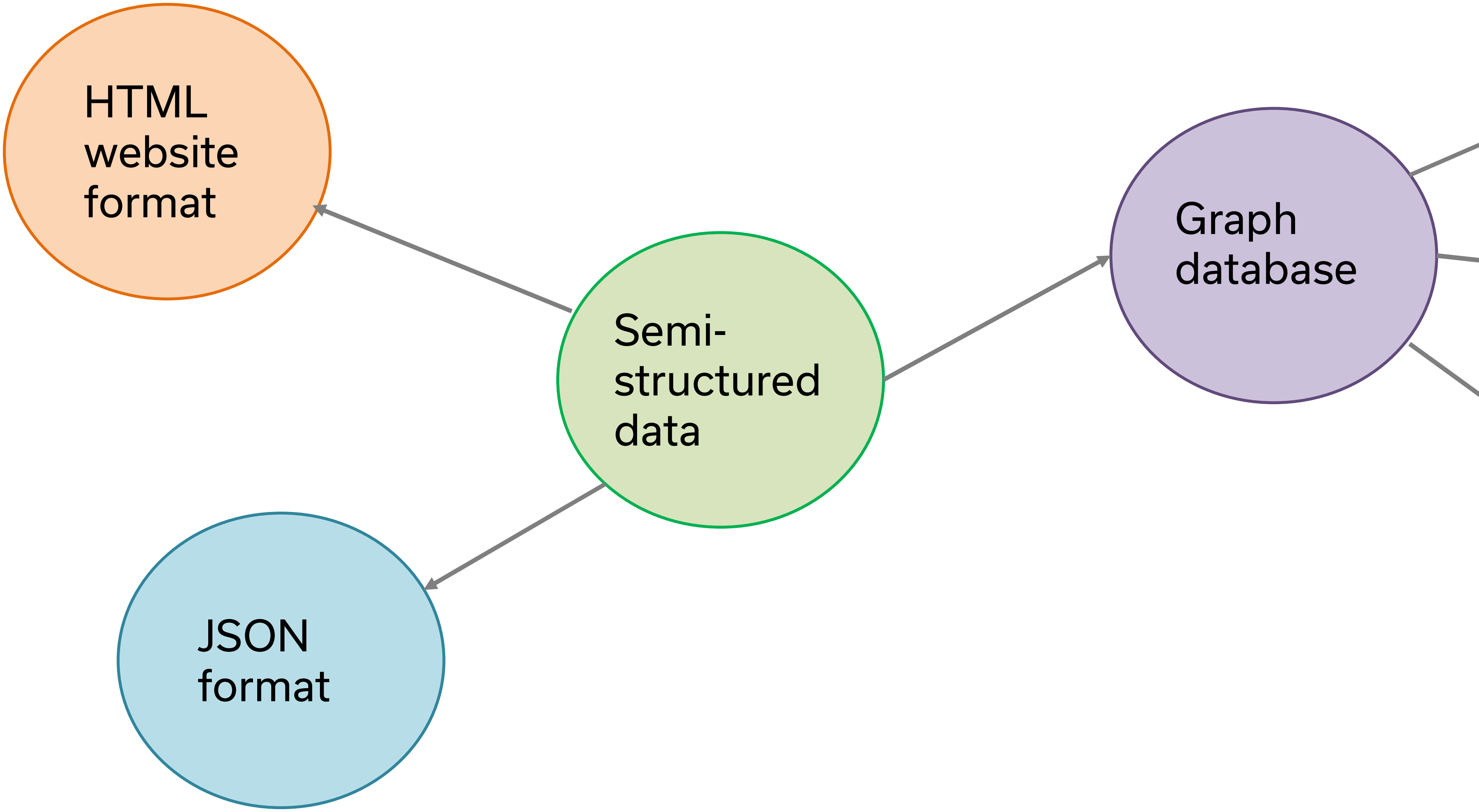
"definition": "Data is organized in groups of 'keys' and 'values'",

"sub-definition": "Values may follow no consistent structure, but they can always be indexed using the keys, which are reliable",

"example": "Smart devices, such as thermostats, frequently send data in JSON format, so an AI system can use this to intelligently manage heating in a building"

}

Semi-Structured Data



Semi-Structured Data

Graph database

- **Definition:** Made up of entities (nodes) and lines connecting them (edges)
- **Explanation:** There can be multiple node types as well as edge types
- **Example:** Intelligent search engines represent all their knowledge as large graphs, so a query will pull the other nodes related to it

Semi-Structured Data

- Pros:
  - –Easier to produce than fully structured data
  - –Rigid enough to still be effectively analyzed
  - –Very common for certain use cases
- Cons:
  - –Can still require pre-processing and data cleaning
  - –Not always interpretable for humans
  - –May require custom ML solutions

How is data stored?

- Lots of different database management systems (DBMS)

Structured	Unstructured	Semi-structured
MySQL	amazon S3	MongoDB
ORACLE	Microsoft Azure Blob Storage	neo4j
PostgreSQL	Google Cloud	redis

NAIRR Data Resources

- NAIRR provides multiple resources for storing data:

Amazon Web Services (AWS)	• Cloud platform for data storage, computing, and model training/inference
DataBricks	• Cloud-based data lake for reporting and analytics
Google Cloud	• Access to datasets and computing
Microsoft Azure	• Data storage and model services

Data Concerns

Bias and fairness

Quality (accurate labels, complete records)

Privacy and security

Data Concerns

Privacy and security

- Data anonymization is important in many fields
- Multiple techniques:
  - –Data masking
  - –Suppression to reduce details/group values
  - –Replacing identifiers with codes
  - –Synthetic data generation
    - NAIRR Resource: Lexset Seahaven

Conclusion

- AI is as diverse as the data required to create it
- Good data needs both quantity and quality
- Watch for bias, confounding factors, scale
- Tailor your use case to your available data
- Data can take many different forms and machine learning techniques exist for just about all of them
  - Think through how your data can be best represented

Digital Transformation, AI, and Cybersecurity:

History, Challenges, Technologies, and Actions

Adel S. Elmaghraby

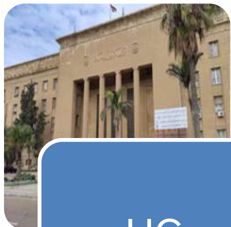
adel@louisville.edu

Jefferson Community & Technical College

Louisville, Kentucky

January 23rd, 2026

My Background



UG Alexandria

Grad Diplomas Cairo & INP

Egypt Military Service

Graduate UW – Madison

IEEE
IEEE Computer Society
Golden Core Member

Charter Member 1996

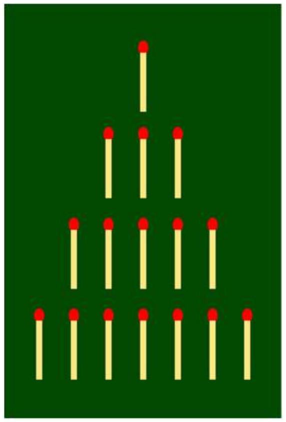
Winnia Professor Computer Science and Engineering	Director of Industry Research JB Speed School of Engineering	Director of Research and Innovation UofL Digital Transformation Center

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

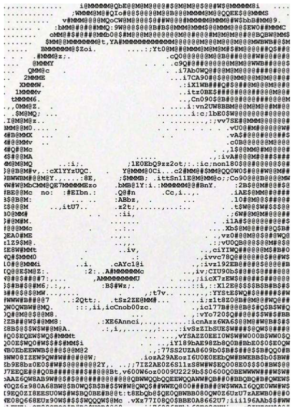
Alexandria University 1968

Transistor
Radio

IBM 1620



Matchsticks
Game



Monalisa

What got my attention ?

Music, Games, and Art

I made my Choice in 1970s and Enjoying it

I love being a computer scientists for the constant intellectual challenge, the ability to create new things, and the real-world impact of our innovations.

It is a unique field that blends logic and creativity, appealing to those with a strong desire to solve complex problems and learn continuously.

Why Digital Transformation and AI Now

Digital Transformation (DT) leverages technology to optimize business processes and delivering them digitally to constituents to create a rich new ecosystem

With data explosion, Artificial Intelligence (AI) techniques are becoming the tools of choice to transform them into knowledge and actionable investments.

Digital Transformation: A Critical Framework

Processes have traditionally been lengthy and designed for humans to handle. With digitization, new needs have evolved, and new approaches and flows need to be considered. Digital tools to help redesign processes exist and can be leveraged with appropriate use of AI.

The quick reliance on digital data has also created its own challenges related to security and privacy. This has accelerated the needs for cybersecurity policies, training, tools, and legislation.

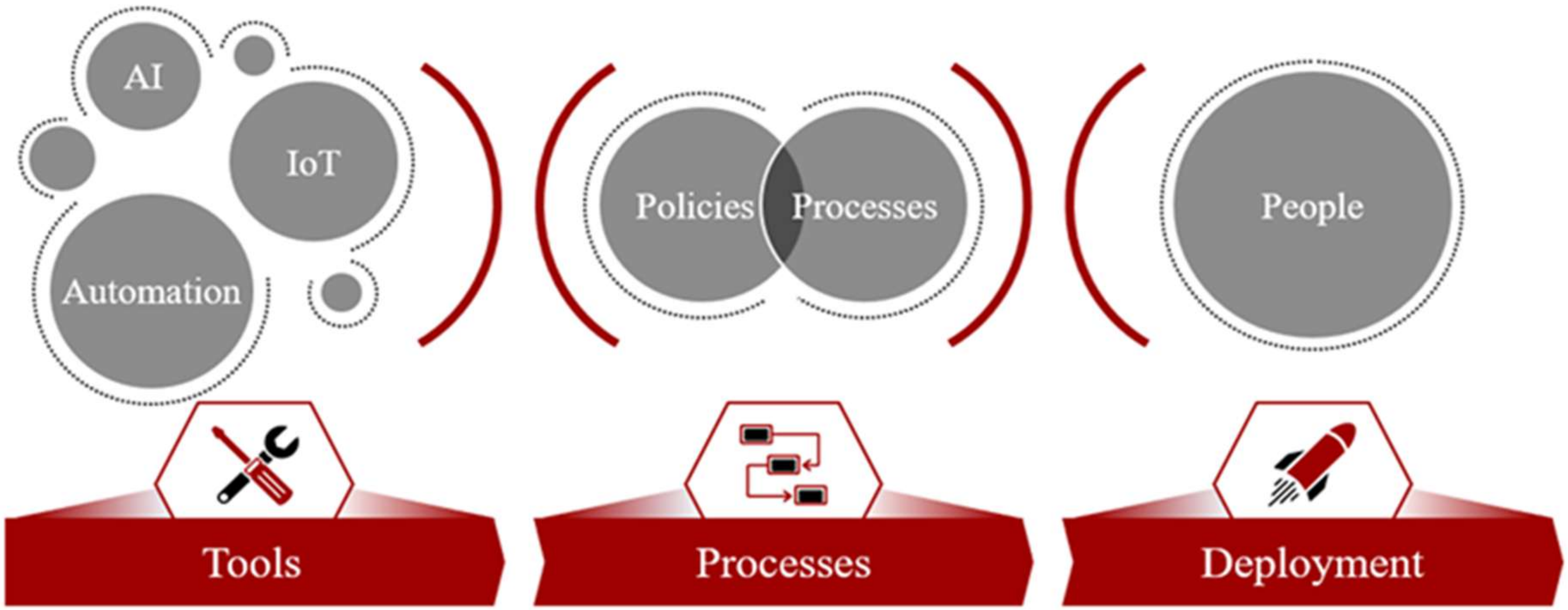
Building meaningful, significant, and secure data tools and applications that become the critical infrastructure for future business and government driven growth.

Digital Transformation: To Do or Not to Do

"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency."- Bill Gates

"You cannot simply automate manual processes … you need to create a new architecture for process flow." – Adel Elmaghraby

What Needs to Change



The Birth of AI

The Dartmouth Conference of 1956 was organized by Marvin Minsky, John McCarthy and two senior scientists: Claude Shannon and Nathan Rochester of IBM.

The proposal for the conference included this assertion: "every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it".

At the conference Newell and Simon debuted the "Logic Theorist" and McCarthy persuaded the attendees to accept "Artificial Intelligence" as the name of the field.

The term "Artificial Intelligence" was chosen by McCarthy to avoid associations with cybernetics and connections with the influential cyberneticists.

Wikipedia

How Do Machines Learn?

Algorithmic:
- An algorithm is implemented with all the envisioned combination of inputs.

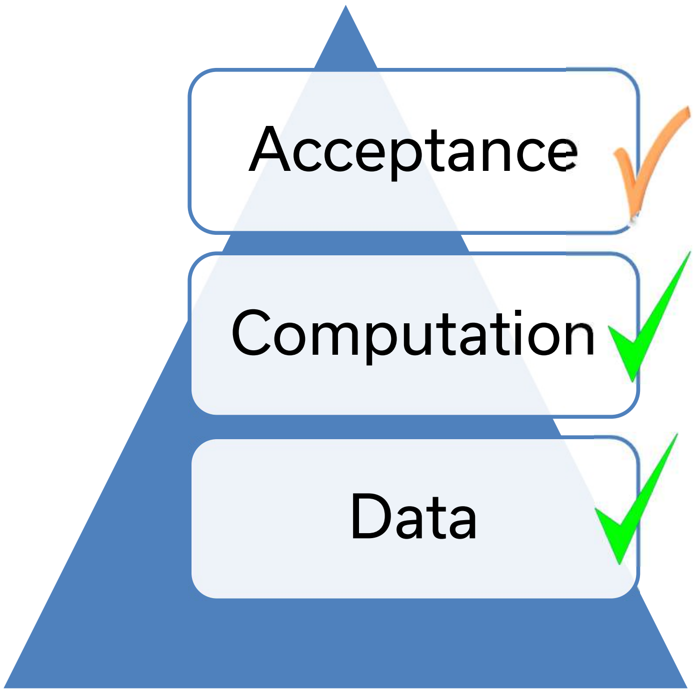
Knowledge Based Systems:
- Numerical AI including Bayesian
- Symbolic AI using languages such as LISP
- Expert Systems

Logic Programming:
- The implemented inference engines and languages such as PROLOG, Incorporation of Fuzzy Logic

Learning from Data:
- Supervised learning, Unsupervised learning, and Reinforcement learning

LLMs and Agentic AI:
- Generative AI using statistical inference and specialized learning from other agents.

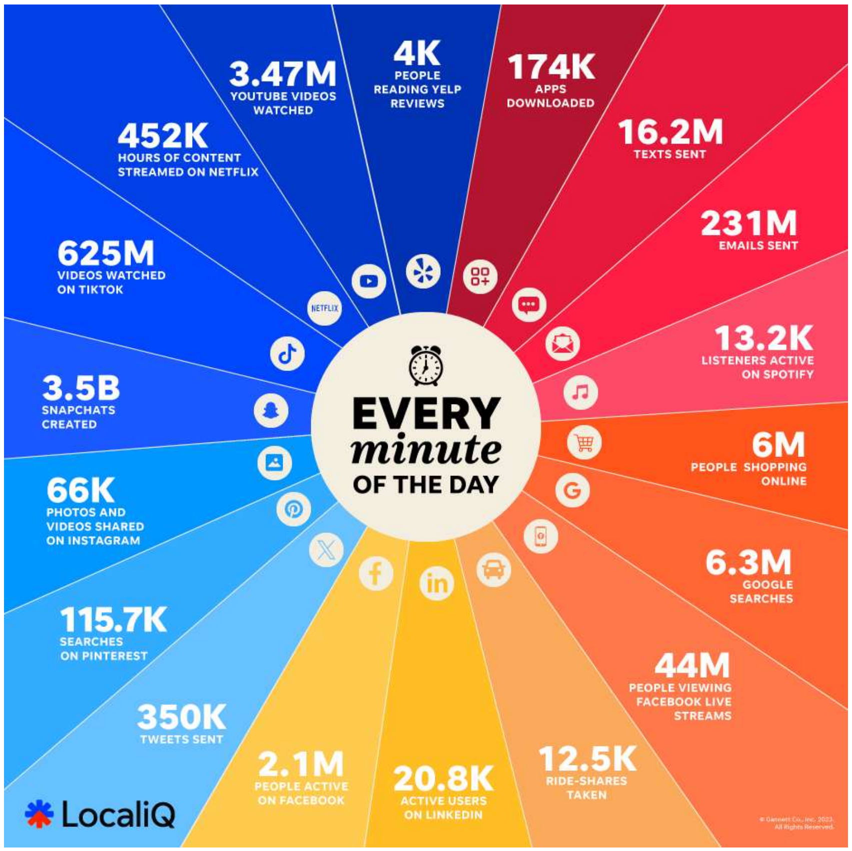
Why Artificial Intelligence Now?



Do you have all three?

Can you handle all three?

Data is Available

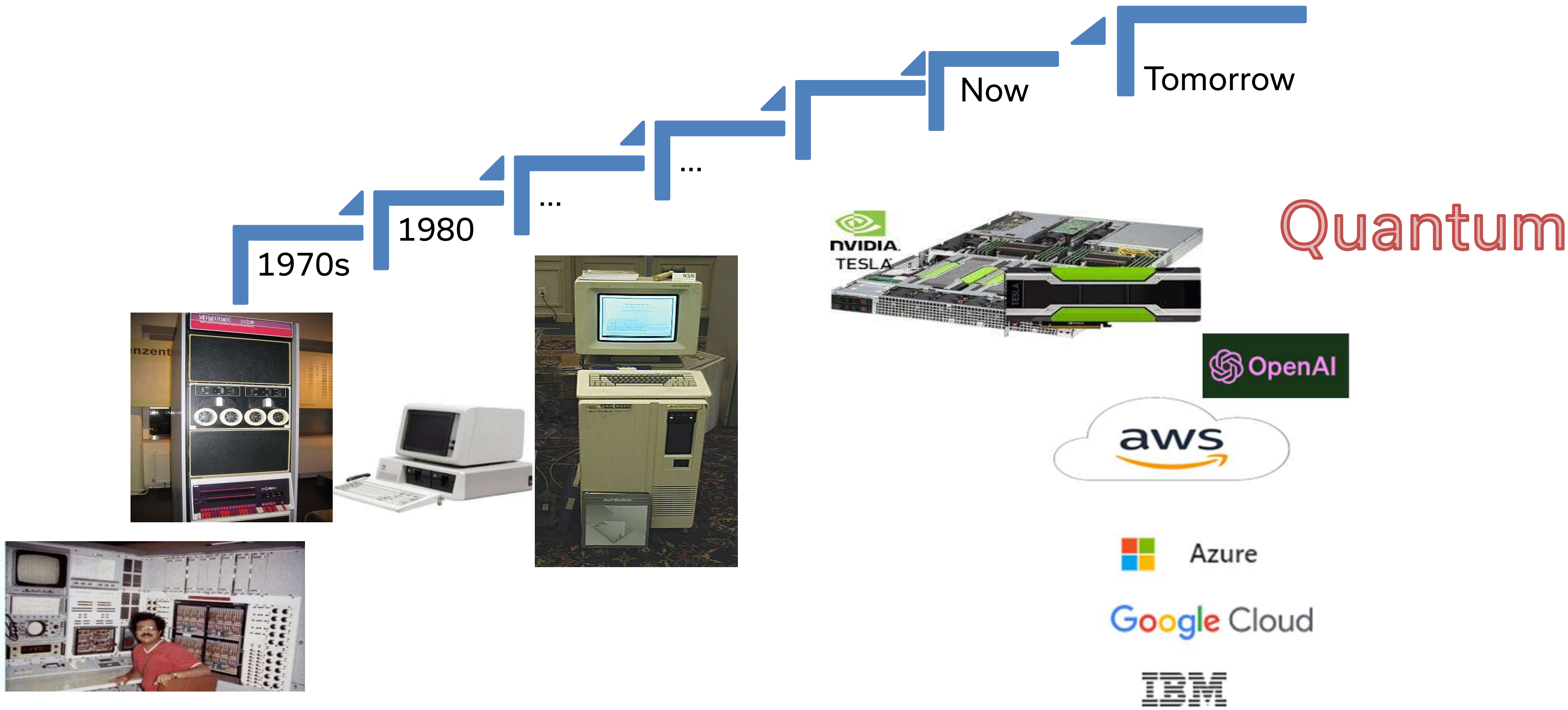


LocaliQ

What Happens in an Internet Minute: 90+ Fascinating Online Stats [Updated for 2024!]

Susie Marino
December 4, 2023 | Data & Metrics

Computing is Available



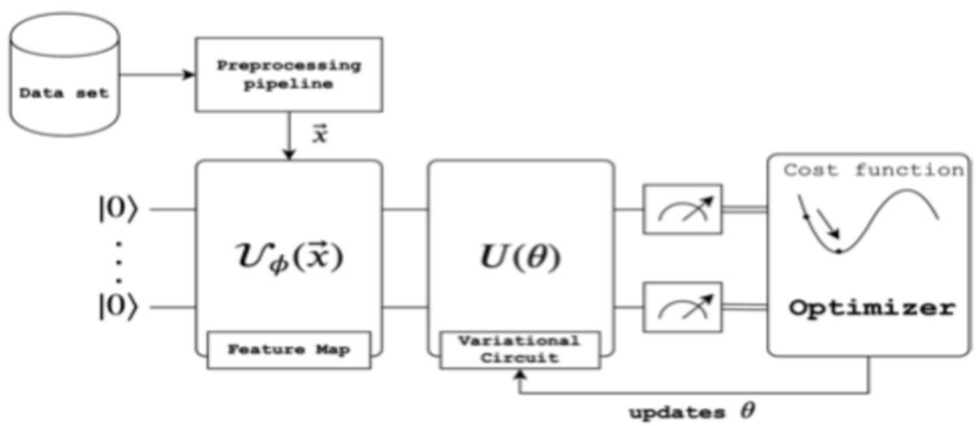
Can Quantum help?

SwissQuantumHub

Home > News > Dementia Prediction Using Variational Quantum Classification

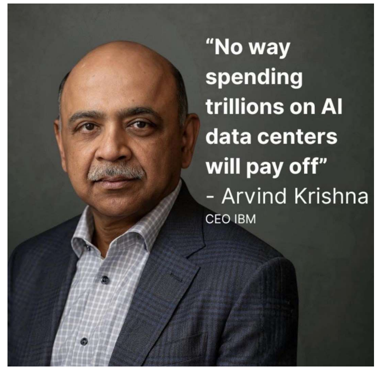
Dementia prediction using Variational Quantum Classification

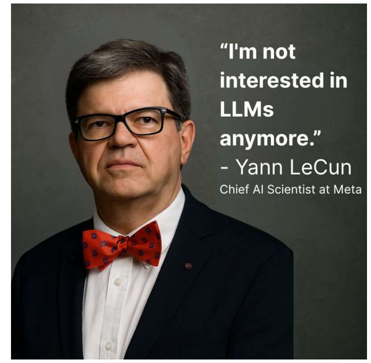
September 21, 2020 / wp_swiss

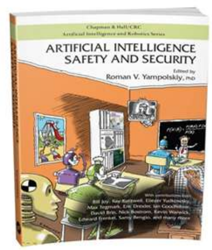


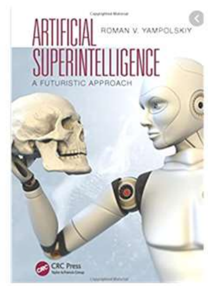
Dementia Prediction Applying Variational Quantum Classifier, Daniel Sierra-Sosa, Juan Arcila-Moreno, Begonya Garcia-Zapirain, Cristian Castillo-Olea, , and Adel Elmaghraby, arXiv preprint arXiv:2007.08653

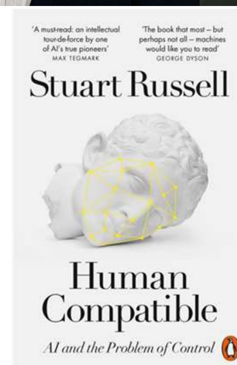
AI Acceptability











CNET Your guide to a better future

Tech > Computing Follow +

Apple Co-Founder Warns AI Will Make It Harder for You to Spot Scams

Steve Wozniak tells the BBC that "bad actors" will abuse the technology.

euronews.next

OpenAI CEO Sam Altman has said "potentially scary" uses for AI are on the horizon - but some experts say we are already in a "dystopic present".

Potential Human Cost

Displacement

Up to 30% of US jobs could be automated by 2030. The "Iceberg Index" shows massive exposure for entry-level white-collar roles, effectively "kicking away the ladder" for new professionals.

Deskilling

Over-reliance on AI for writing, coding, and analysis may lead to cognitive atrophy. When the machine does the thinking, we lose the ability to critique the output.

Loss of Agency

Algorithmic management (being fired by an app) and content curation reduce human autonomy, guiding our choices in subtle, invisible ways.

AI Future

The Path Forward

- Regulation: Enforceable standards for transparency and safety.
- Audits: Third-party verification of algorithms for bias.
- Human-in-the-Loop: Ensuring human judgment remains the final arbiter in critical decisions.

Why AI and Cybersecurity

Artificial Intelligence

- Increases Productivity
- Boosts Industry Growth
- More AI Jobs

Cybersecurity Threats

- Reduces Productivity
- Consumes resources
- Disrupts business

Adversarial AI

WIRED

Technology | Science | Culture | Gear | Business | Politic

Artificial Intelligence

AI cyberattacks will be almost impossible for humans to stop

ComputerWeekly.com

IT Management ▼ | Industry Sectors ▼ | Technology Topics ▼ | Search Computer Weekly

AI a threat to cyber security, warns report

Artificial intelligence is being incorporated into a range of cyber security products, but the technology may also introduce new threats, a report warns

Warwick Ashford
Security Editor
21 Feb 2018 10:45

Artificial intelligence (AI) poses a range of threats to cyber, physical and political security, according to a report by 26 UK and US experts

CSO FROM IDG

Home > Cyber Attacks

IDG CONTRIBUTOR NETWORK Want to Join?

NEXT-GEN AI

By Deepak Dutt, Contributor, CSO | JAN 10, 2018 7:41 AM PT

Opinions expressed by ICN authors are their own.

OPINION

2018: the year of the AI-powered cyberattack

2018 will not only be a worse year for data breaches, but the year when we start to see more and more cyberattacks powered by artificial intelligence, which will make prevention more difficult.

Information Age

Diversity

News | Data & Insight | Sectors | Topics | The City & Wall Street | Careers | Regions

Topics
AI & Machine Learning

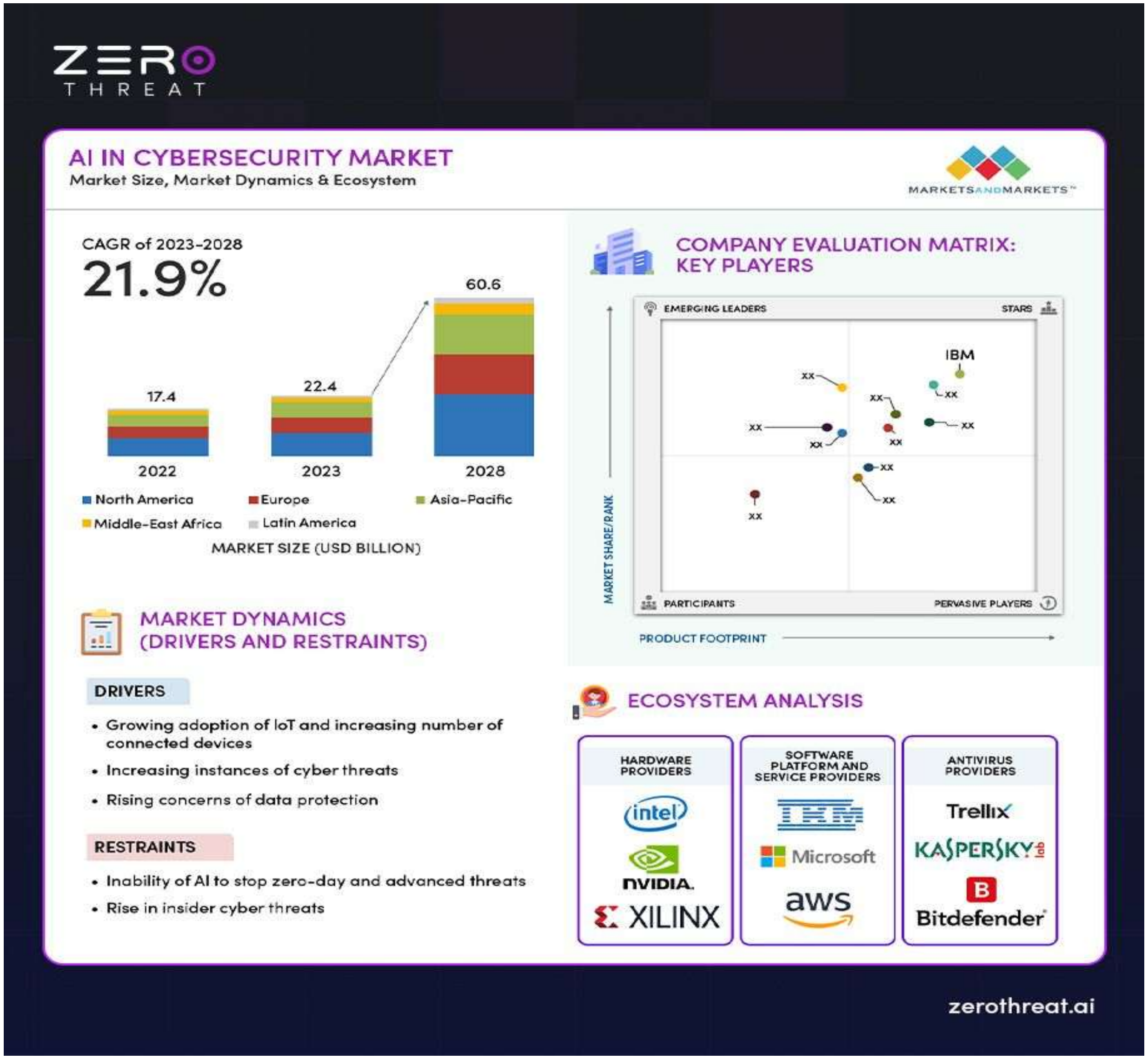
Nick Ismail
28 February 2018

Artificial intelligence technologies could boost capabilities of hackers

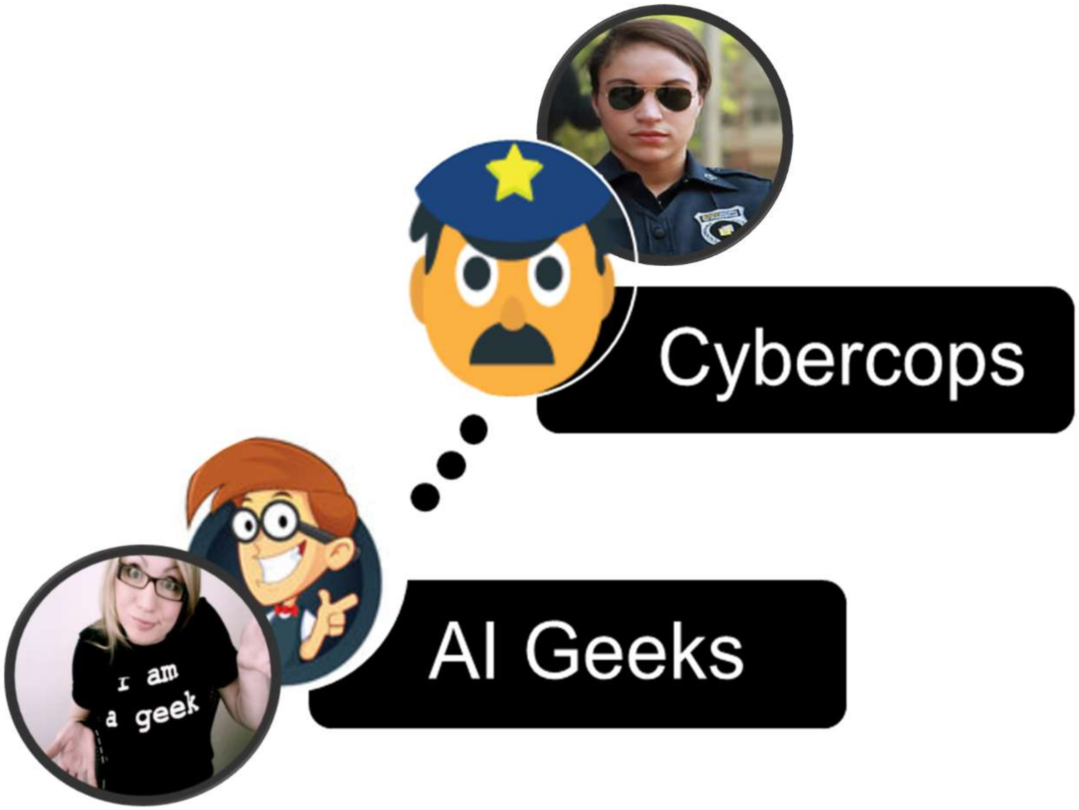
For many firms, the advent of artificial intelligence will lead to groundbreaking possibilities in a host of fields. However, cyber threat actors can also use this technology to cause havoc

Emerging AI Security Business

Market Projections



Bringing AI and Cybersecurity Together



UofL Digital Transformation Center

CREATING NEW PATHWAYS FOR PROGRESS, POSSIBILITY, IDEAS AND IMAGINATION

The University of Louisville Digital Transformation Center transforms the access, awareness, design and use of technology to embrace digital agility and enhance learning and discovery in the commonwealth.

UofL Digital Transformation Center Leadership Team

Sharon Kerrick
Assistant Vice President
Digital Transformation Center

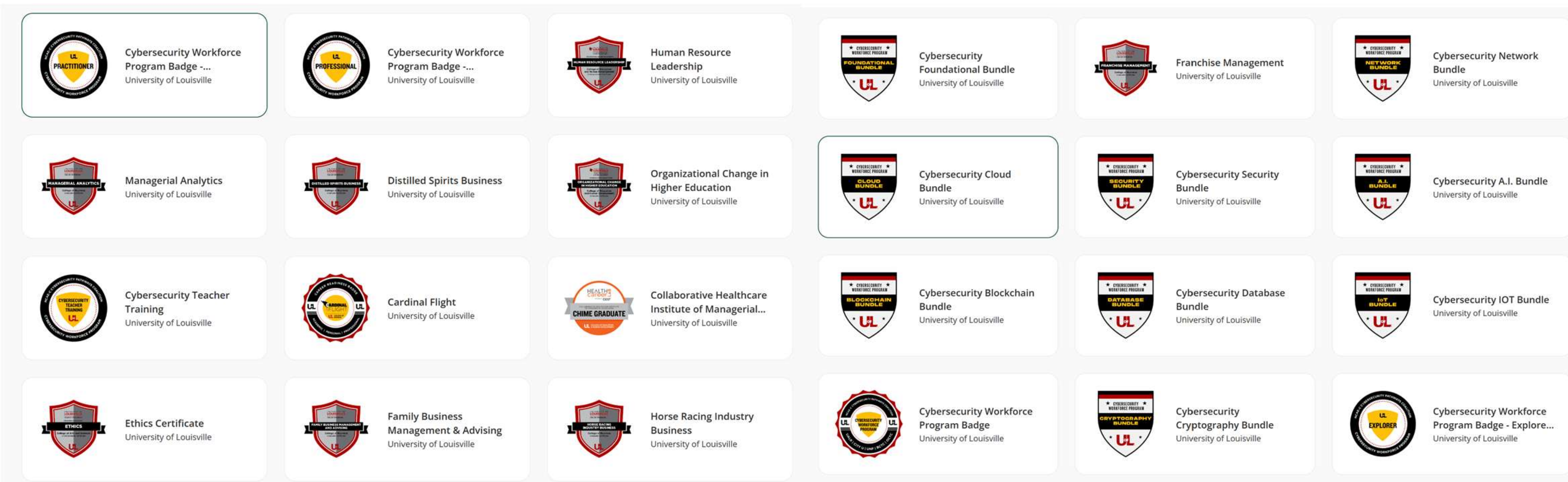
Dr. Andrew Wright
(Faculty Appointee)
Consulting Director

Dr. Adel Elmaghraby
(Faculty Appointee)
Director of Research & Innovation

Robert J. Kubash Jr
Executive Director

Digital Transformation University of Louisville

The University of Louisville Digital Transformation Center transforms the access, awareness, design and use of technology to embrace digital agility and enhance learning and discovery in the commonwealth.



CYBERSECURITY WORKFORCE CERTIFICATE PROGRAM

DIGITAL TRANSFORMATION CENTER

Supported via NCAE-C-003-2020 Grant

Explorer (Badges = AI Fundamentals & Enterprise Design Thinking)

- IT Basics
- Network Foundations
- Coding
- DB Management
- Privacy/Legal Foundations and Ethics
- Security Principles & Foundations
- Cryptography
- Artificial Intelligence

Practitioner (Badges = Cloud Security, Cisco, Blockchain)

- Cloud Foundations
- Network Security
- Information Security
- Cyber Threat Hunting
- Forensics
- Cognitive Computing
- Data Mining
- Blockchain

Professional (Badges = Azure IoT, RPA and Power Automate & Threat Modeling)

- DB Security
- Cloud Security
- IoT
- Post Quantum Cryptography
- Risk Analysis
- Robotic Process Automation Analysis
- Healthcare Capstone
- Logistics Capstone

Healthcare or Logistics Cybersecurity (Certificate) 6 months/8 week courses/online instructor led— Cohorts of 35-40
First group launched Fall 2021

Enhancements:

- Logistics
- Train the Trainer – inquire NOW
- Cybersecurity Analyst
- Cybersecurity Technical Specialist

PATHS to:

*Associate
*Bachelors
*Graduate Degrees
And/or Certificates

Curricula developed as core foundational with tracks in Healthcare industry and Logistics (Labs, Datasets etc.)

Tech Industry Badges earned throughout (IBM, Microsoft, Google etc.)

Train the Trainer and Open Source modules available- please inquire

Success Coaches assigned to each student

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

CyberHIVE: AI innovation Studio

UofL began initial discussions with Kindred in 2014, with full collaboration taking shape in 2016. HIVE facility located on UofL campus for easy student access. Opened November 2017. The modern works space was designed to foster collaboration and innovative thinking.

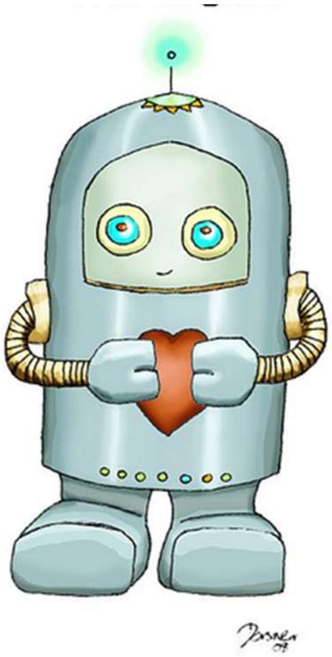
Hive is envisioned as the go to catalyst for bringing innovative AI and Cybersecurity research to serve the community at large. We bring our experience in Healthcare and Cybersecurity to our industry and community partners.

Questions

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

Contact Information

Thank you



adel@louisville.edu

1601 S. Floyd St., Louisville
KY, 40208

hivehub.org

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

AI, the Workforce, and Higher Education:

Implications and Opportunities for Kentucky

Christopher Ledford, PhD

Jefferson Community & Technical College
Louisville, Kentucky

January 23rd, 2026

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

Agenda

- **AI and the Workforce: Implications for Kentucky**
    - Early signals of broad disruption
    - How can national patterns inform our work in Kentucky?
    - What if Kentucky isn't prepared?
    - How do we get prepared? AI can actually help!

- **AI and Postsecondary Education: Student Success Opportunities**
    - Sustaining success in Kentucky
    - Informing the public

AI and the Workforce: Implications for Kentucky

NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

University of Kentucky

Early Signals of Broad Disruption

AI performs ~16% of U.S. labor tasks today

- AI writes over **a billion lines of code per day**
- AI reads over **150,000 words per second**

Widespread layoffs and hiring freezes

- Major companies are laying off **thousands in IT management and HR**
- **Hundreds of thousands of engineers** were laid off this year with nationwide hiring freezes
- Some data corporations estimate **50% of their customer service and sales work** in 2025 was done by AI

National Insights -> Kentucky

By 2030, thirty percent of the workforce will be unmeasurable

Manufacturing:

There will be a **50%+ reduction in production jobs** by 2030

Customer Service:

Over **60% of customer service** transactions are automated

Transportation and Logistics:

Autonomous vehicles logged **71 million miles** last year

NAIRR Pilot | Kentucky Community & Technical College System | CPE Higher Education Matters | University of Kentucky

What if Kentucky isn't prepared?

Core industries are at risk

- **Transport/Logistics and Manufacturing** account for nearly **¾ million jobs.**
  - Experts estimate that KY could face a **10% reduction in median income** and a **double-digit spike in unemployment**
  - A **quarter million Kentucky jobs** and **$11 billion of revenue** at imminent risk
- Counties along the **I-64 and I-75 corridors** are at severe risk

States lack visibility into AI-driven labor shifts

- For most, it seems like trying to study and understand water while holding it in your hand

How do we get prepared?

State and local gov't must investigate current impact

- **Create AI labor indices** to quantify and understand the problem – specifically measuring **AI job loss** – at a local, granular level.
- **Measure and understand industries at risk** in Kentucky's workforce
  - Map **skills and upskill potential** to other industries
- **Risk Modeling**
  - Test, simulate, predict, and retest – **AI can actually help us prepare!**

How do we get prepared?

Postsecondary education must adapt for future impact

- Are there opportunities to **reprioritize academic programs?**
- Advocacy for and adaptability of **state support around critical programs less threatened by AI**
  - **Healthcare**: HWIF
  - **Skilled Trades:** Kentucky Work Ready Scholarship (KWRS)
  - **Education:** Teacher Program Scholarships
  - **Early Postsecondary Ops:** Dual Credit and WRDC Scholarships

AI and Postsecondary Education: Student Success Opportunities

NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

University of Kentucky

How do we get prepared?

Sustaining student success improvement

- How can we better understand **how key factors (affordability, transitions, etc.) impact student success?**

Increasing public access to PSE data

- How can we **help the public and lawmakers quickly answer simple but important data questions** about higher education?

Student Success

Example: Using ML and AI to target financial aid

- We modeled how out-of-pocket costs (and at what thresholds) impact second-year retention

- Now we're using ML and AI tools to **turn inferential research into predictive power**

- **Providing information back to institutions** to help target institutional aid and resources

Student Success

Implementation on a grand scale?

- **How can we integrate AI tools with the best longitudinal postsecondary data system in the United States?**
  - Estimate massive **exploratory inferential models**: What could we be investigating that's not our radar?
  - **Predict, simulate, and constantly inform** student success interventions
  - How can we do this in a controlled server with **minimal security risks?**

Informing the Public

How can we use AI to increase public awareness?

- **Example:** How have retention rates at Kentucky public institutions changed over the last decade?
- Create interactive AI chat to **rapidly answer simple, important questions**
- We're investigating creating secure servers with aggregated data tables, **creating zero security risk opportunities**

References

- Chopra 2025. “Project Iceberg: Workforce Security for Kentucky in AI Age.”
- Kentucky Cabinet for Economic Development 2026. “Top Industries.”
- Kentucky Center for Statistics (KYSTATS) 2026. “Employment and Wages by Industry.”
- Kentucky Workforce Innovation Board 2026. “Kentucky’s Top Five In-Demand Sectors.”

Lunch

12:00PM – 12:45PM

Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

Machine Learning

Aaron Mullen, MS

Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

What is Machine Learning?

- Getting computers to perform a task without being explicitly programmed by humans
- The computer learns from data in its own way

Traditional Learning

Data + Rules = Answers

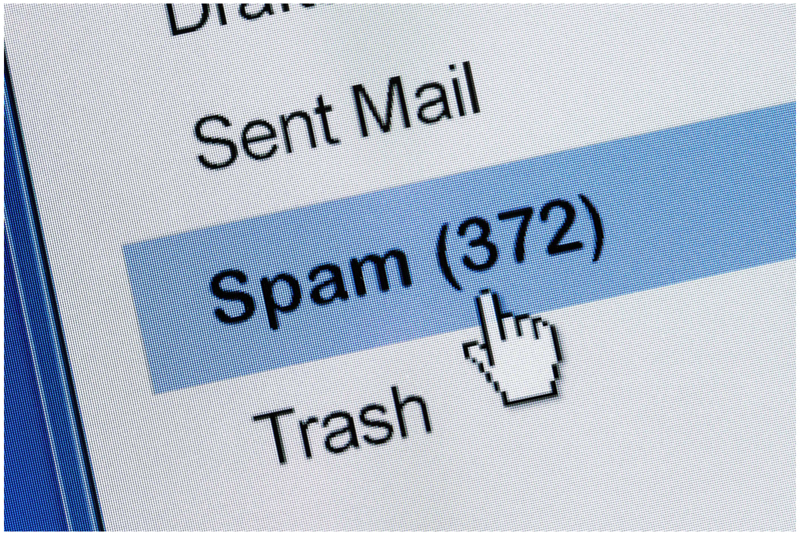
Machine Learning

Data + Answers = Rules

Machine Learning Example 1

Machine Learning Example 1

- Email providers use classification algorithms to analyze subject line, sender information, and specific phrases in emails
- Models can be trained with many examples of emails, labeled “Spam” or “Safe” so the machine can learn the difference
- It learns that certain phrases (“Congratulations! You won…”), senders, times, or images are associated with spam emails more frequently



Machine Learning Example 1

Why Machine Learning?

- The first spam filters were rule based and checked for specific keywords:
  - IF email contains "Click Here" THEN mark as Spam
- Scammers could change spelling/punctuation to avoid these filters
  - "Cl1ck Here" or "C!ick He re"
- ML checks for intent and structure rather than specific keywords

Machine Learning Example 2

Machine Learning Example 2

- ML model can learn geometry of a face to unlock your phone
- Rule based systems would be inadequate here
  - –Machine learning ensures the system works even if your face is at a different angle, or you're wearing glasses

Categories of Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

- Model learns from data that is already labeled with correct answers
  - Spam detection is an example: emails used to train models are already labeled "Spam" or "Safe" so the model can learn the difference
- Multiple types:
  - Classification (malignant vs benign tumor)
  - Regression/forecasting (how many sales made in the next quarter)
  - Segmentation (which part of this image contains a dog)
- But how does the machine actually "learn" from these examples?

How does a machine learn?

- Most models iterate through epochs, learning from the data at each step
- Dataset is split into different groups:

Training set	• ~70% of your data • Used to train the model
Validation set	• ~15% of your data • Used in training to ensure the model is on the right track
Test set	• ~15% of your data • Completely unseen during training and used at the end to determine applicability of the model

Mop vs. Dog

Training Set

Validation Set

Test Set

NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

University of Kentucky

Mop vs. Dog

Training Set

Epoch 1

Model

Mop vs. Dog

Training Set

Epoch 1

Training data is fed into the model, which tries to determine which features of the data distinguish best between the classes

Model

Features:

- Dog
  - Has nose/tongue
- Mop:
  - Has handle
  - Is indoors

Mop vs. Dog

Validation Set

Epoch 1

Now the model will use the features that it learned to classify the validation set

Model

Features:

- Dog
  - Has nose/tongue
- Mop:
  - Has handle
  - Is indoors

Mop vs. Dog

Validation Set

Dog: 60% confident
- Has nose
- Is outdoors
- No tongue?

Epoch 1

Now the m
features that it learned to
classify the validation set

Model

Features:
- Dog
  - Has nose/tongue
- Mop:
  - Has handle
  - Is indoors

Mop vs. Dog

Validation Set

Dog: 52% confident
- No nose/tongue
- Handle different color
- Outdoors!

Epoch 1

Now the model will use the features that it learned to classify the validation set

Model

Featu
- Dog
  - H
- Mo
  - Has handle
  - Is indoors

Mop vs. Dog

Validation Set

This is why varied and extensive training data is important!

Epoch 1

Now the model will use the features that it learned to classify the validation set

Model

Features:

- Dog
  - Has nose/tongue
- Mop:
  - Has handle
  - Is indoors

Mop vs. Dog

Training Set

Epoch 2

The model now re-examines the entire training data to improve its validation accuracy

Model

Features:

- Dog
  - Has nose/tongue
- Mop:
  - Has handle
  - Is indoors

Mop vs. Dog

Training Set

Epoch 2

The model now re-examines the entire training data to improve its validation accuracy

Model

Features:

- Dog
  - Has nose
- Mop:
  - Has handle
  - Less "oval" shape

Mop vs. Dog

Validation Set

Epoch 2

Now the model validates again and hopefully improves in performance and confidence

Model

Features:
- Dog
  - Has nose
- Mop:
  - Has handle
  - Less "oval" shape

Mop vs. Dog

Validation Set

This iterative process continues repeatedly until a set number of epochs is reached, or the model stops improving

Epoch 2

Now the model validates again and hopefully improves in performance and confidence

Model

Features:
- Dog
  - Has nose
- Mop:
  - Has handle
  - Less "oval" shape

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

Mop vs. Dog

Test Set

Epoch 10

Now the model makes predictions on the test set, which can provide a metric on overall model performance

Model

Features:

- Dog
  - Has nose
- Mop:
  - Has handle
  - Less "oval" shape

Mop vs. Dog

Test Set

Epoch 10

Dog

Mop

Now the model makes predictions on the test set, which can provide a metric on overall model performance

Model

Features:

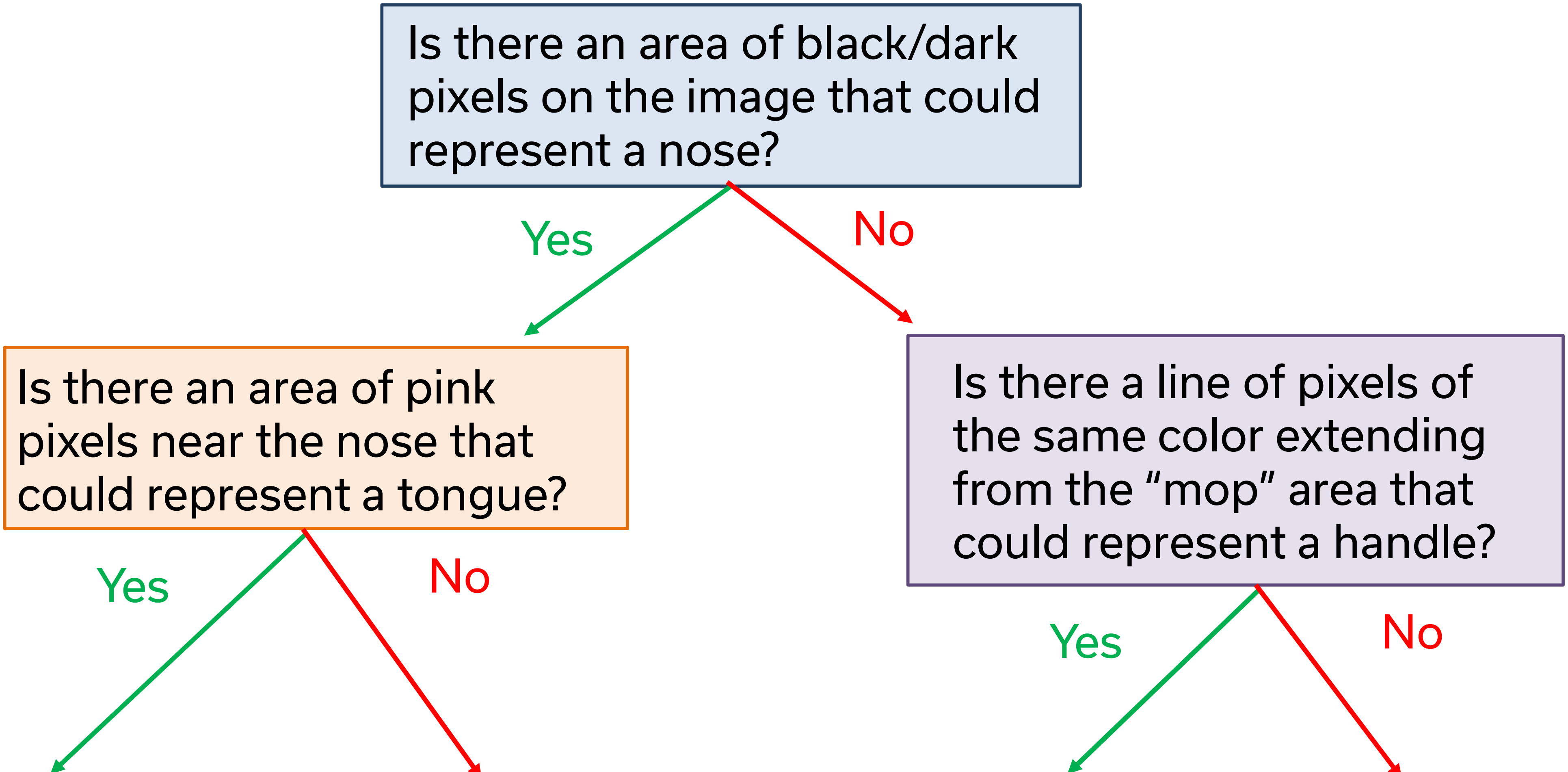
- Dog
  - Has nose
- Mop:
  - Has handle
  - Less "oval" shape

How does a model learn features?

- During the training step, how does the model determine what features it should use to distinguish classes or make predictions?
- It depends on the type of machine learning model!

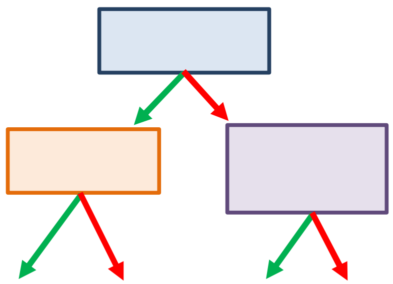
Decision Tree

- The model essentially builds a flowchart that maximizes accuracy when followed for classification



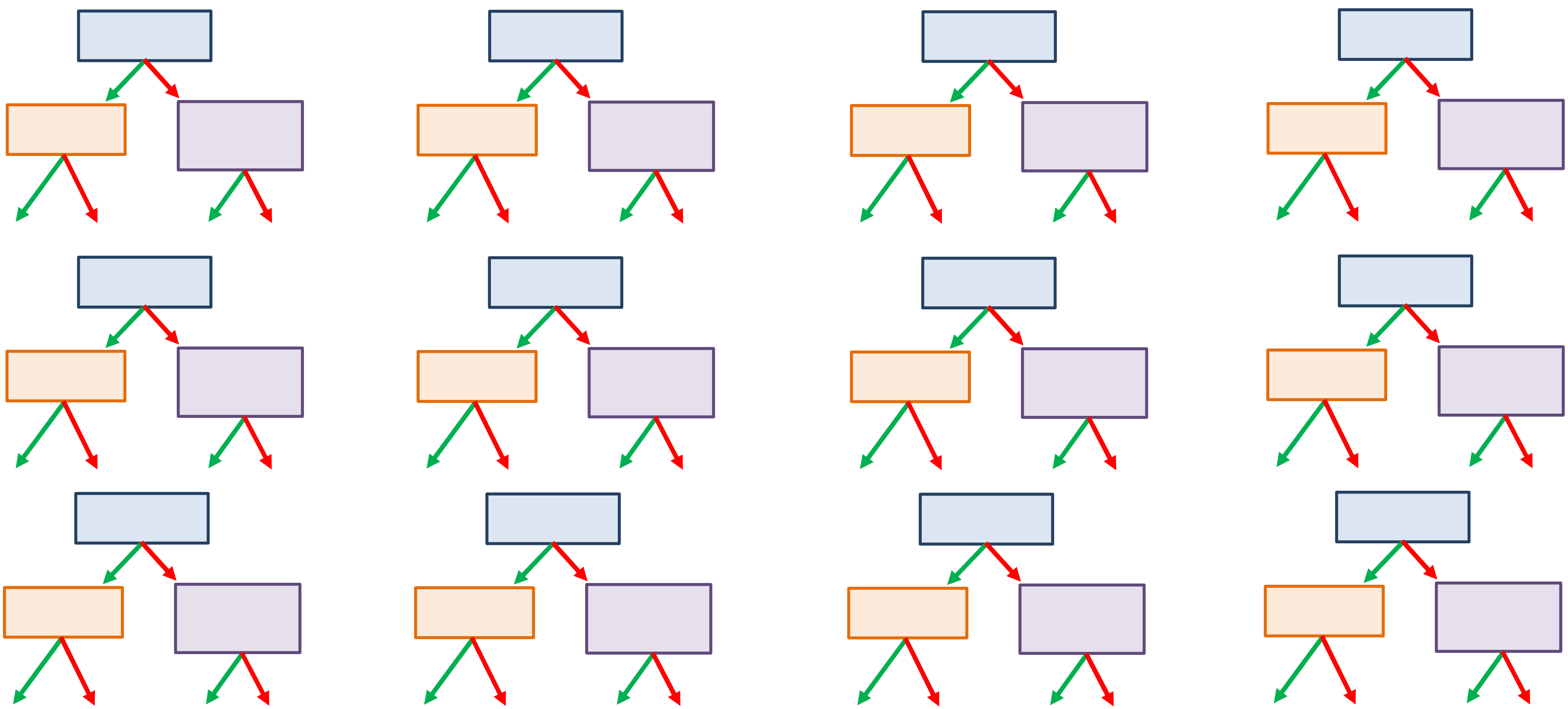
Decision Tree

- The model essentially builds a flowchart that maximizes accuracy when followed for classification

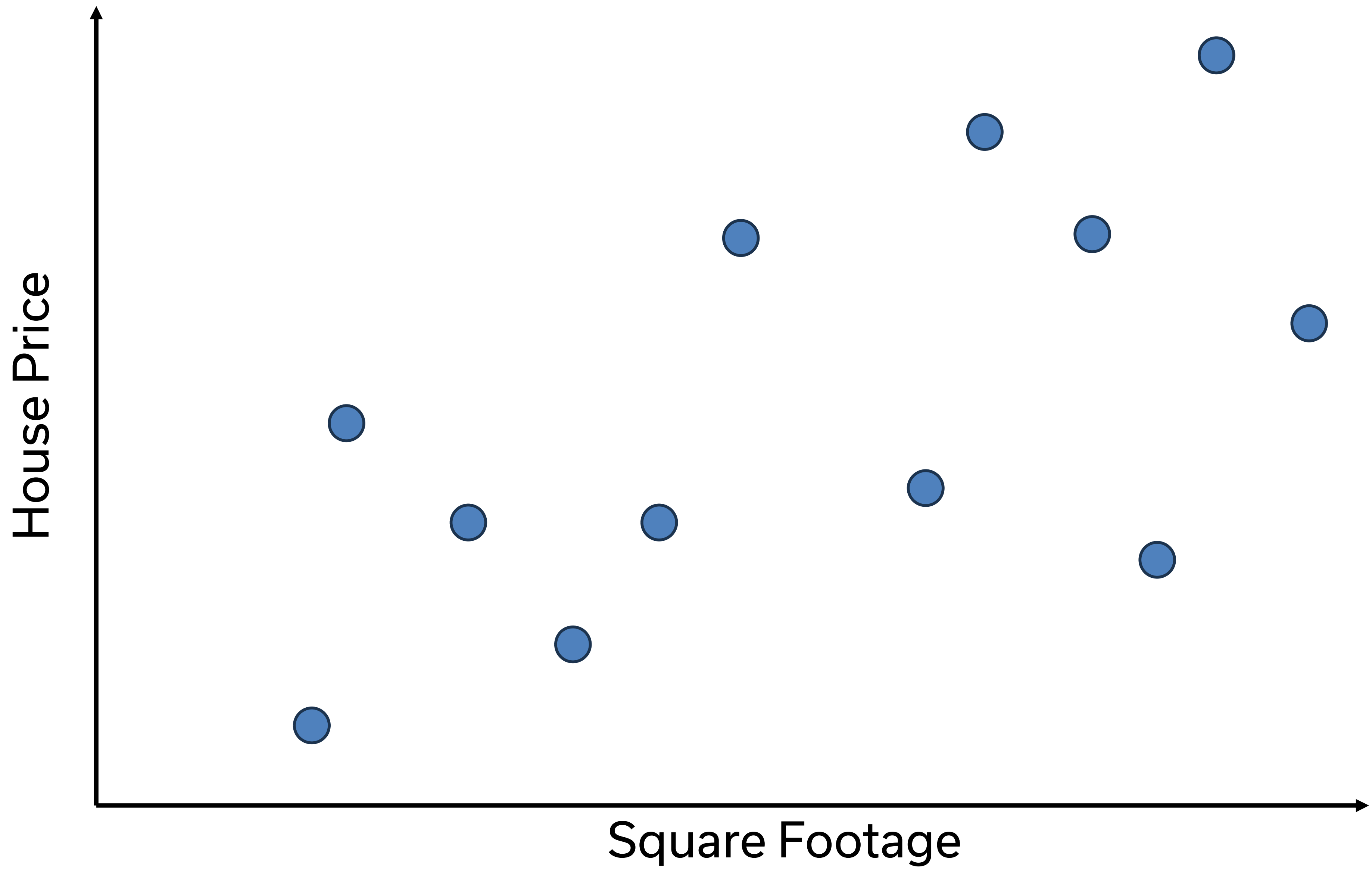


Random Forest

- Building many decision trees based off different subsets of data and features improves the model significantly

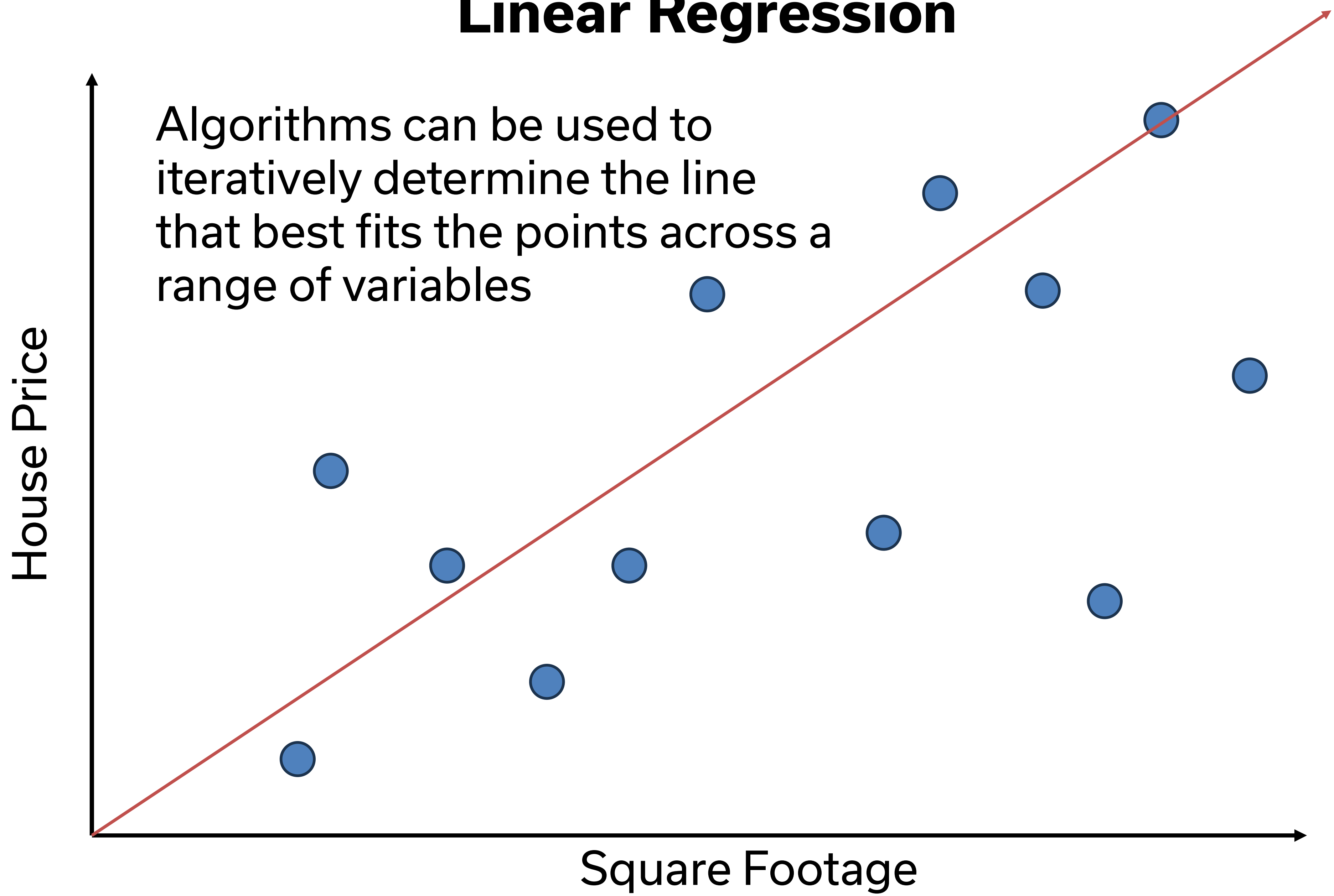


Linear Regression

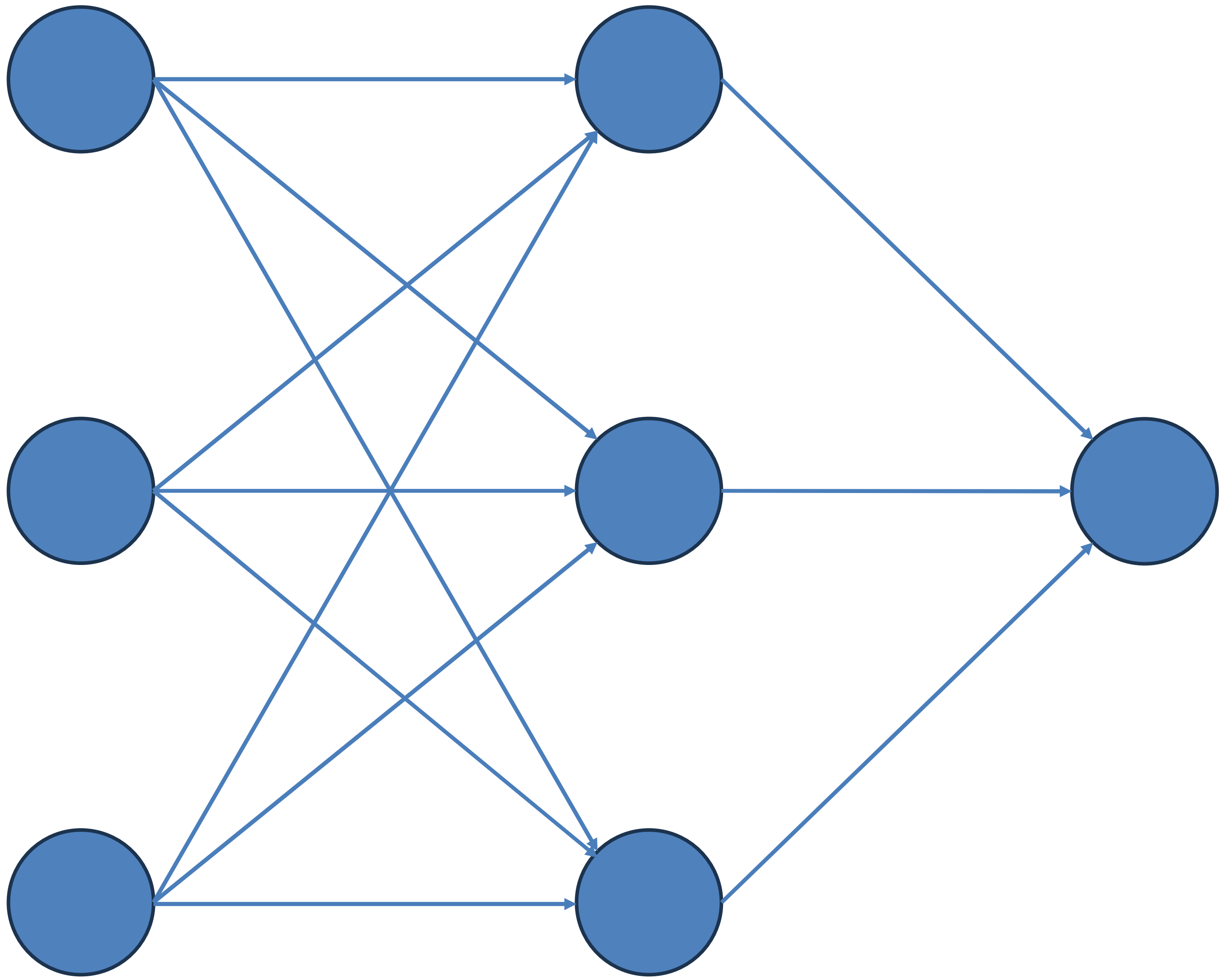


Linear Regression

Algorithms can be used to iteratively determine the line that best fits the points across a range of variables

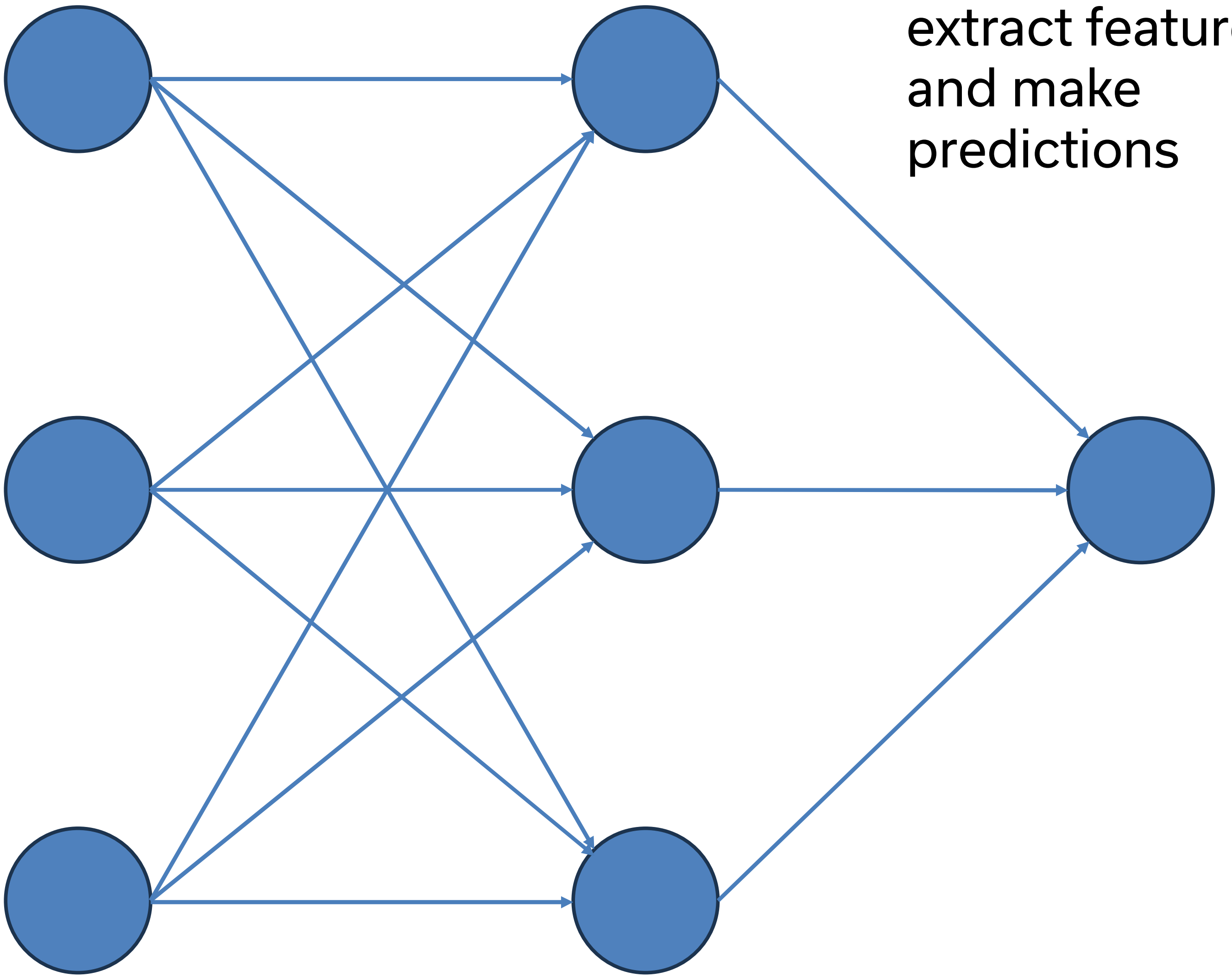


Neural Network



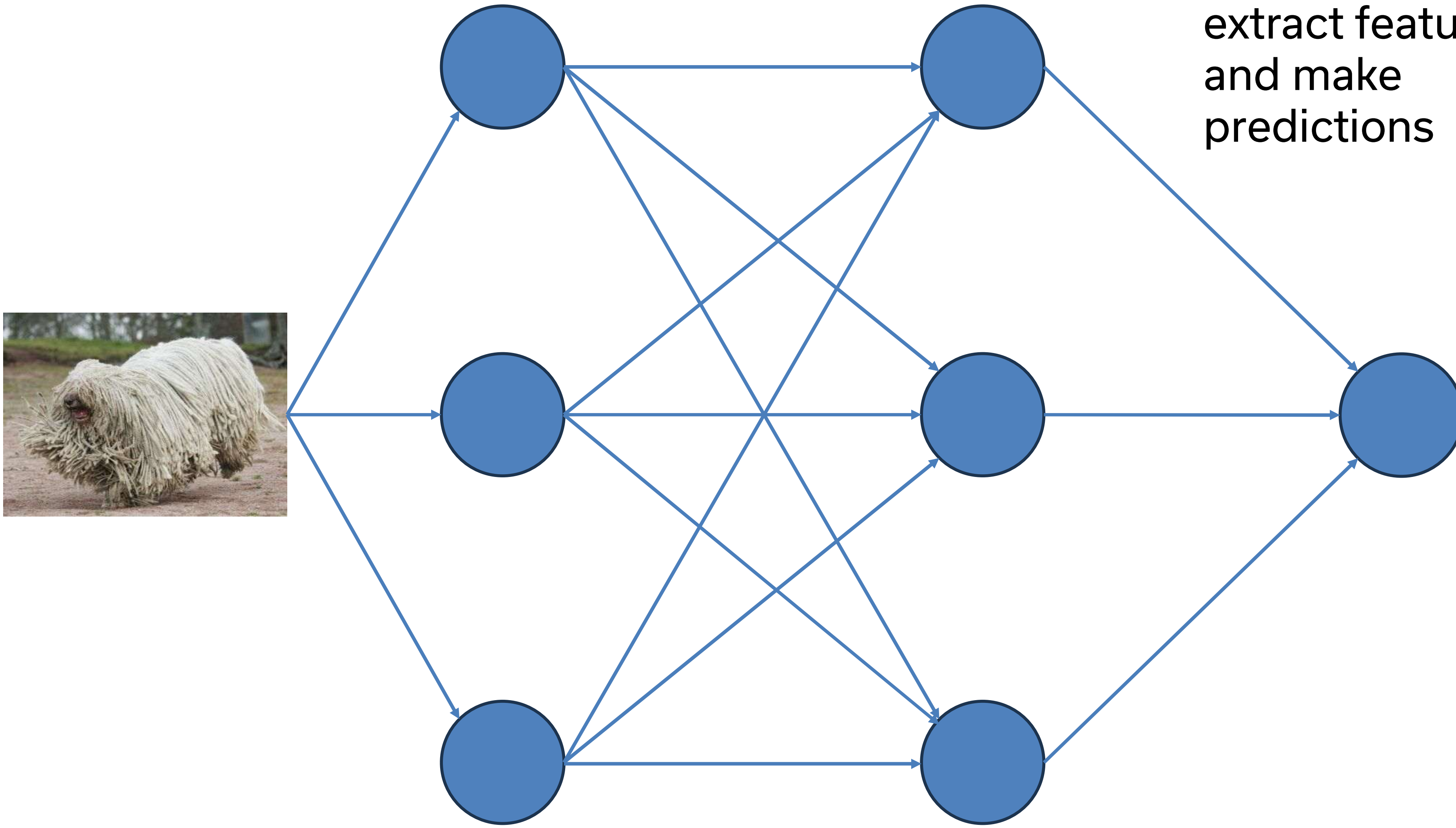
Neural Network

Each node of the network is iteratively tuned to extract features and make predictions



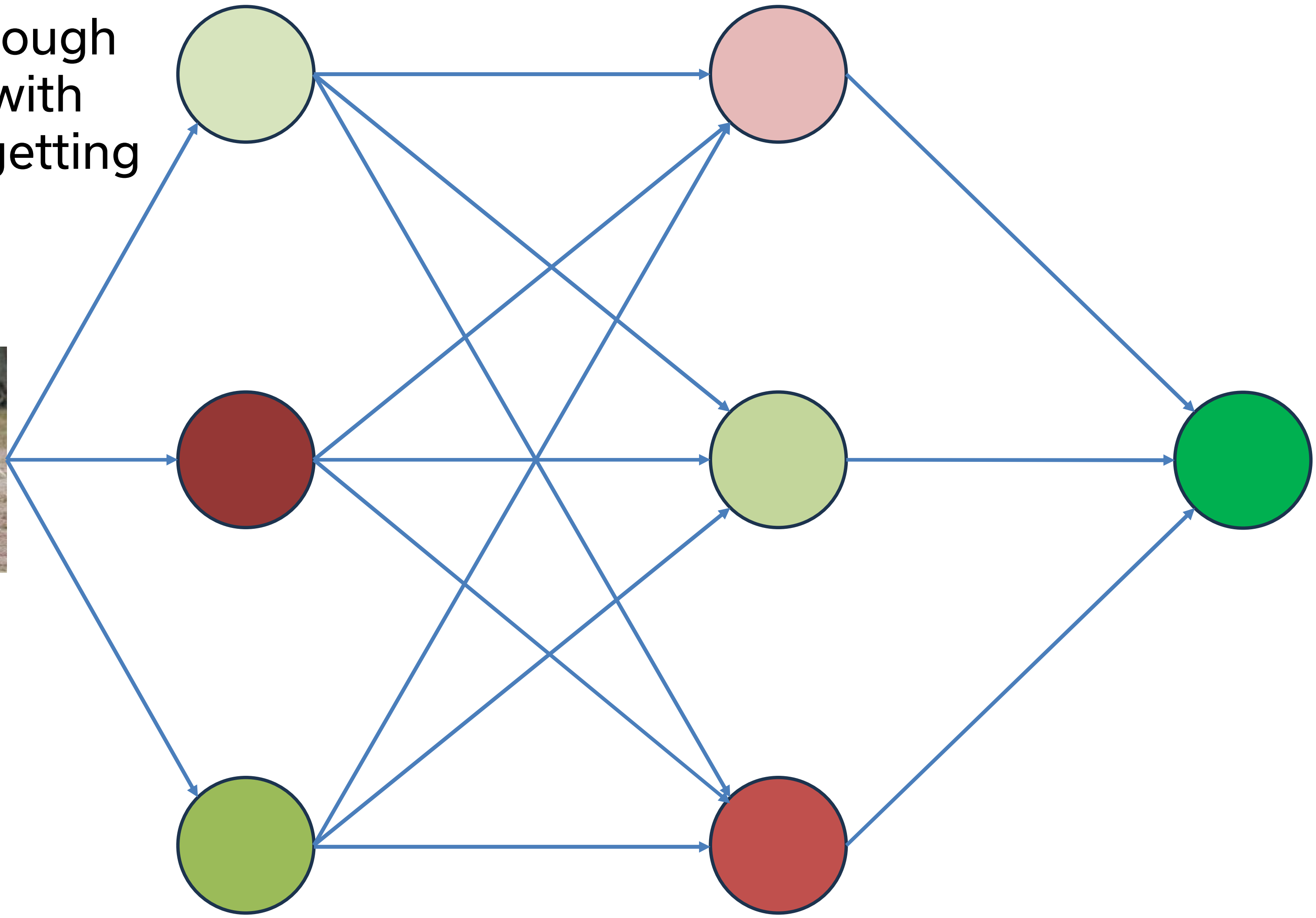
Neural Network

Each node of the network is iteratively tuned to extract features and make predictions



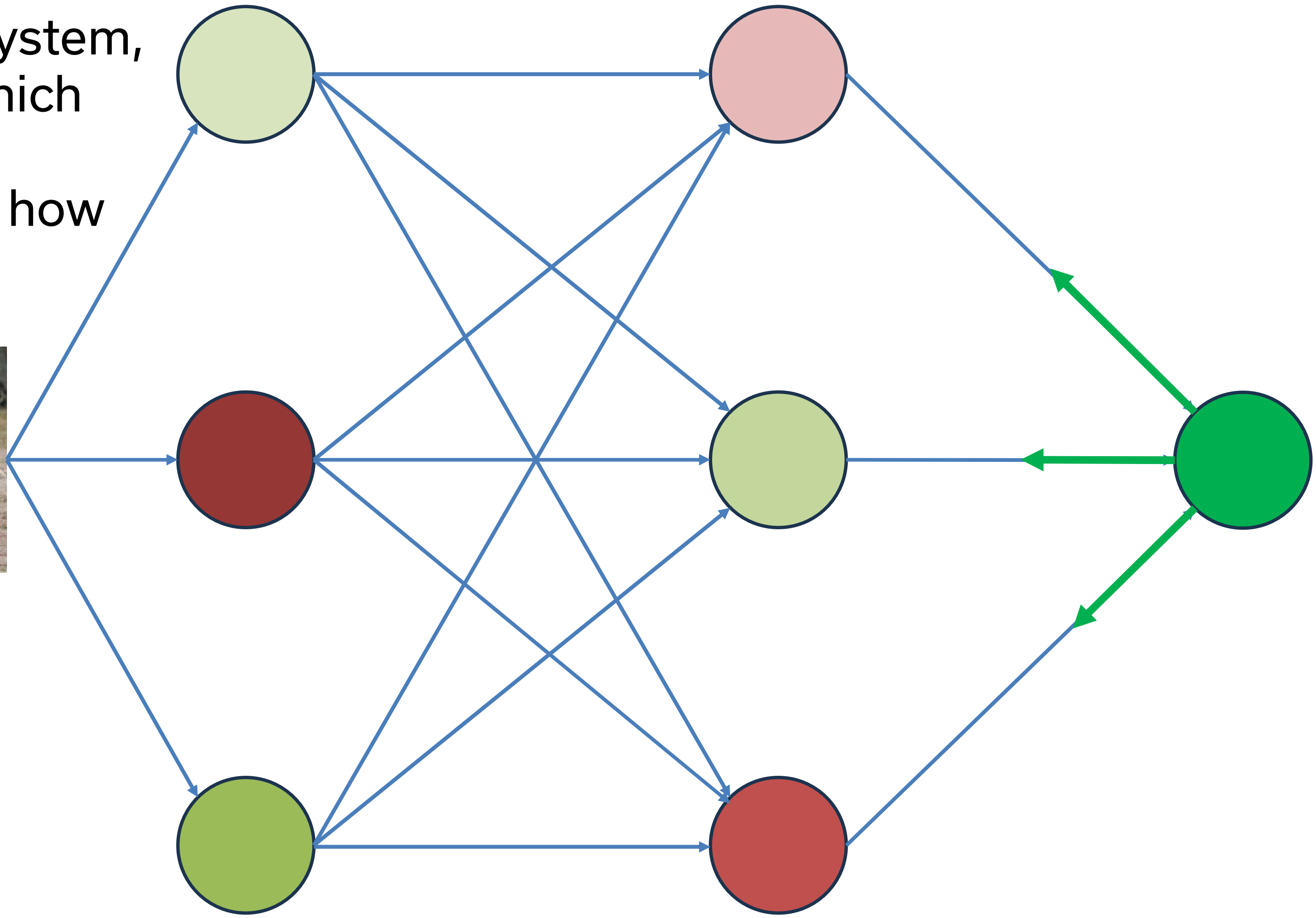
Neural Network

The features are turned into numbers which propagate through the network, with some nodes getting activated and others not

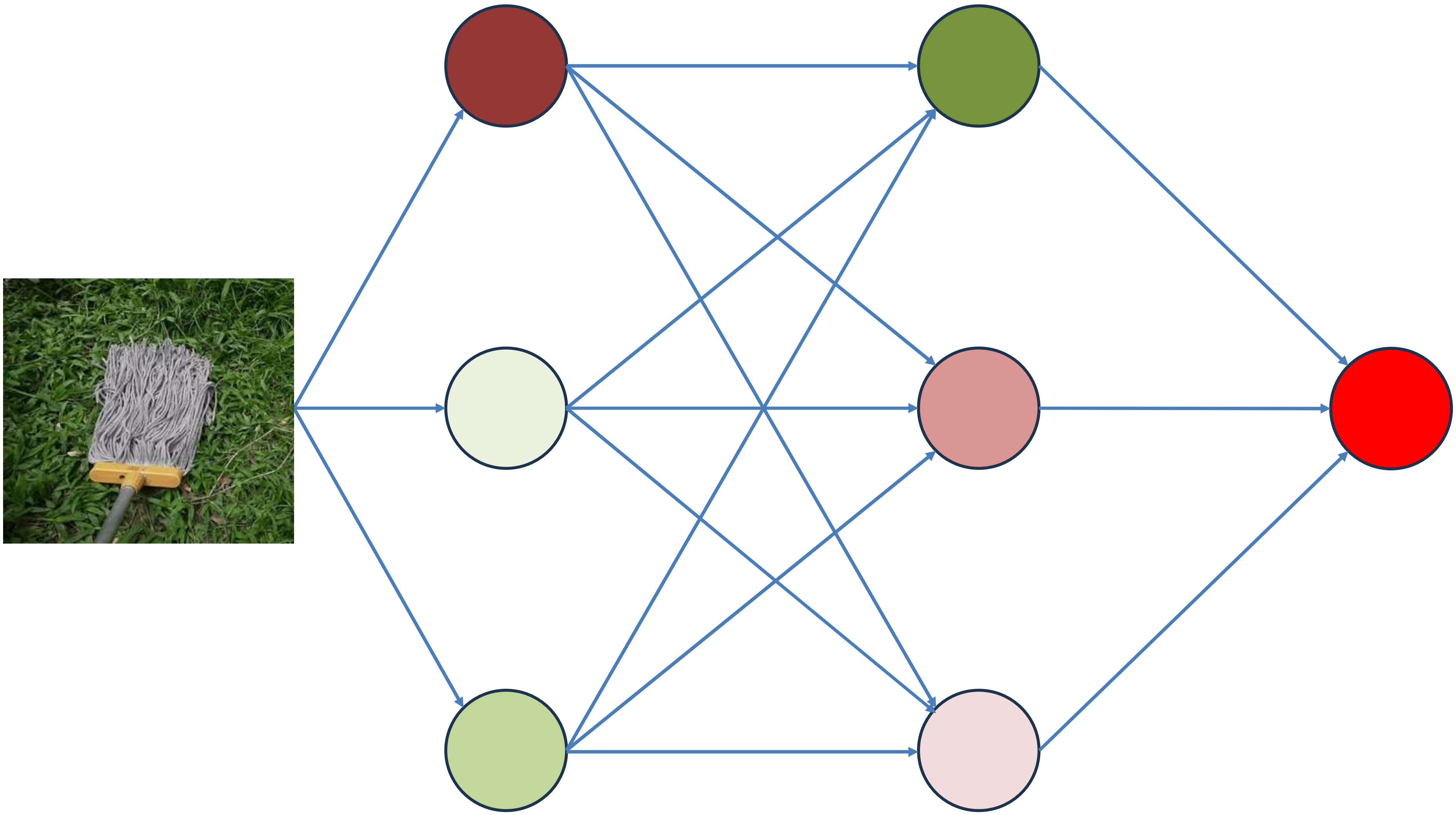


Neural Network

If the prediction at the end is right, this will propagate back through the system, reinforcing which nodes were activated and how

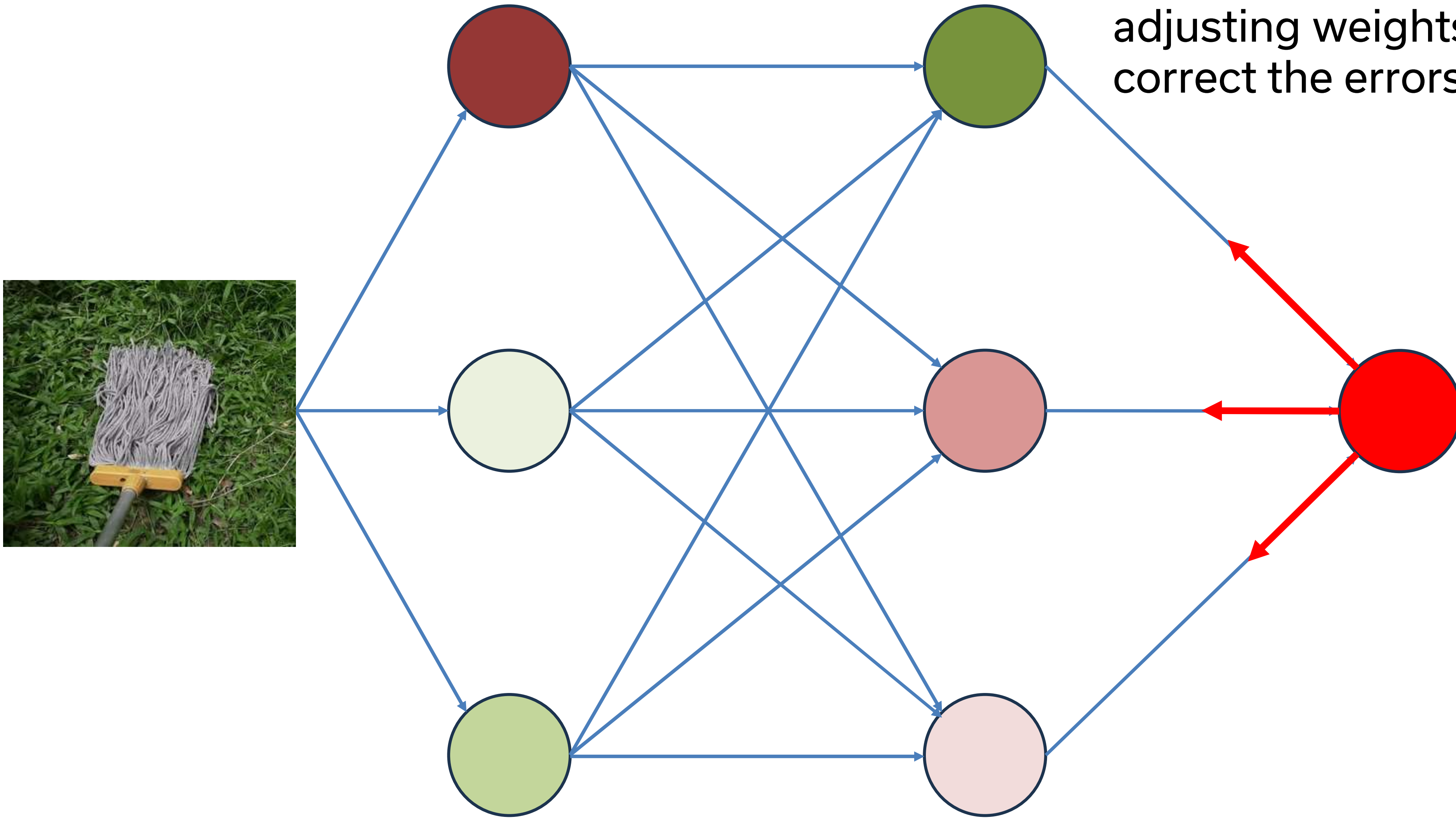


Neural Network



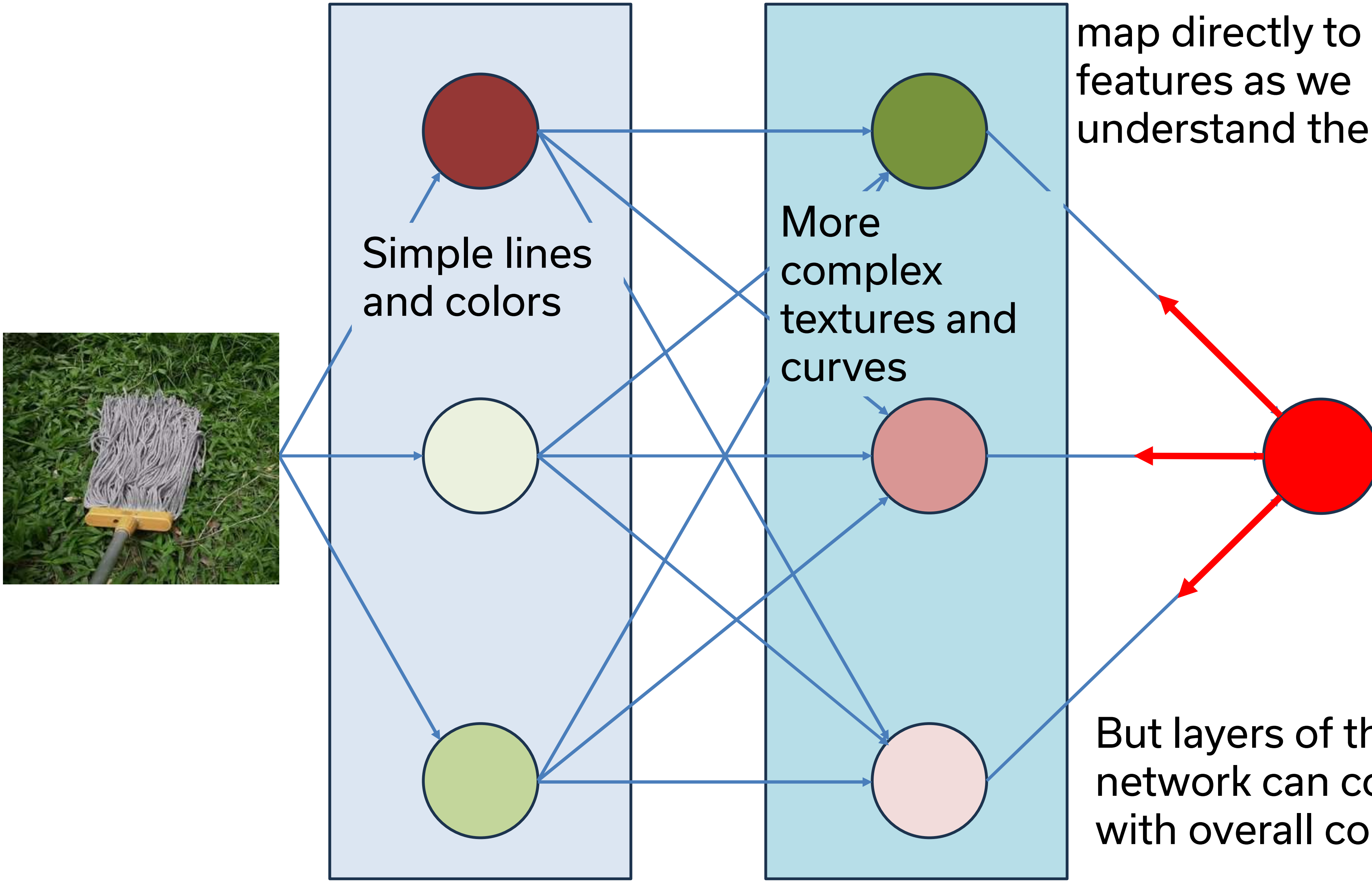
Neural Network

If the prediction is wrong, the signal is still back-propagated , adjusting weights to correct the errors



Neural Network

Individual nodes rarely map directly to features as we understand them

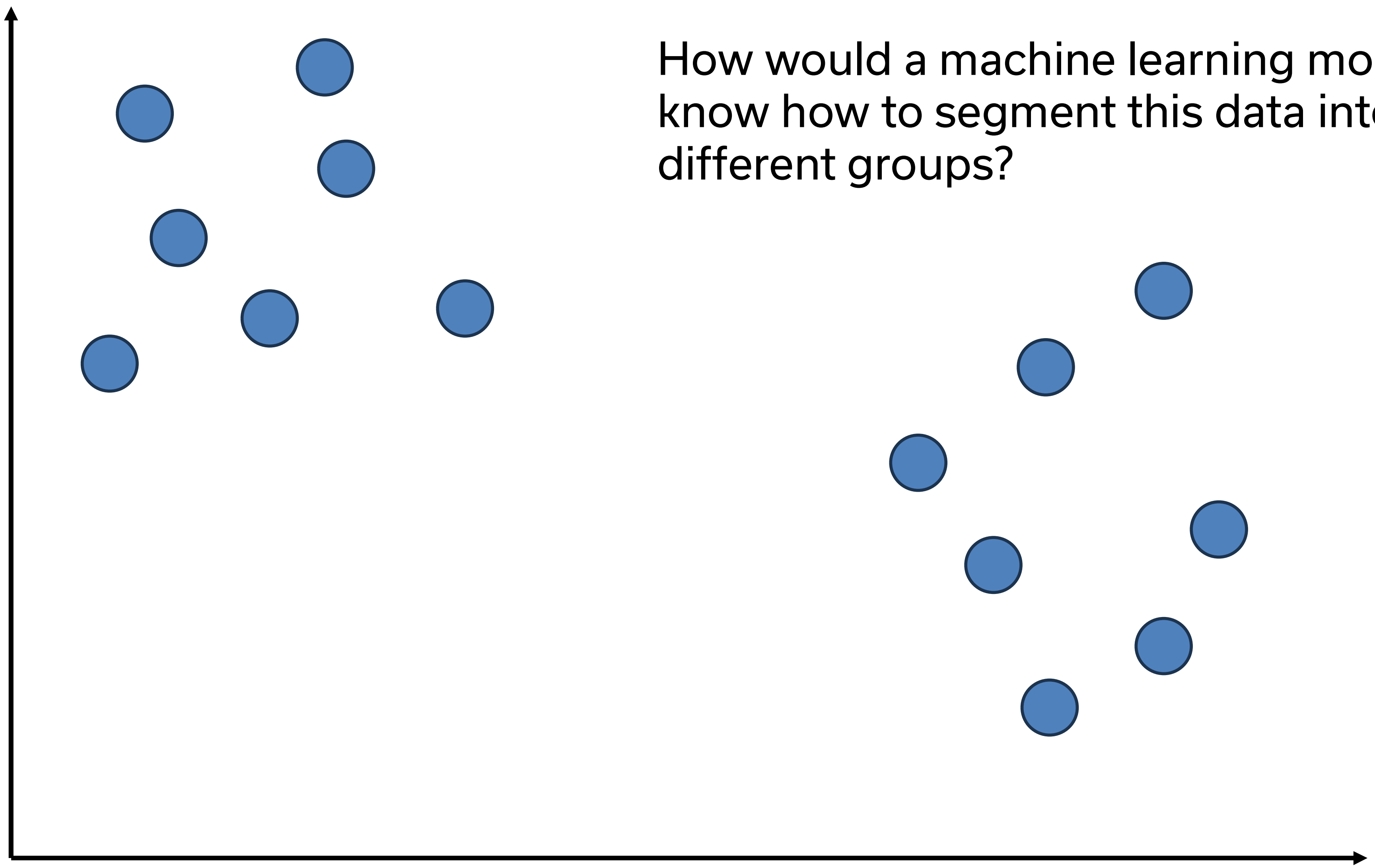


But layers of the network can correlate with overall concepts

Unsupervised Learning

- The model is given unlabeled data and must find patterns or structures itself
- Can be used to identify previously unknown data trends
  - –Clustering: segmenting customers into different marketing groups
  - –Association: people with condition X also tend to have condition Y

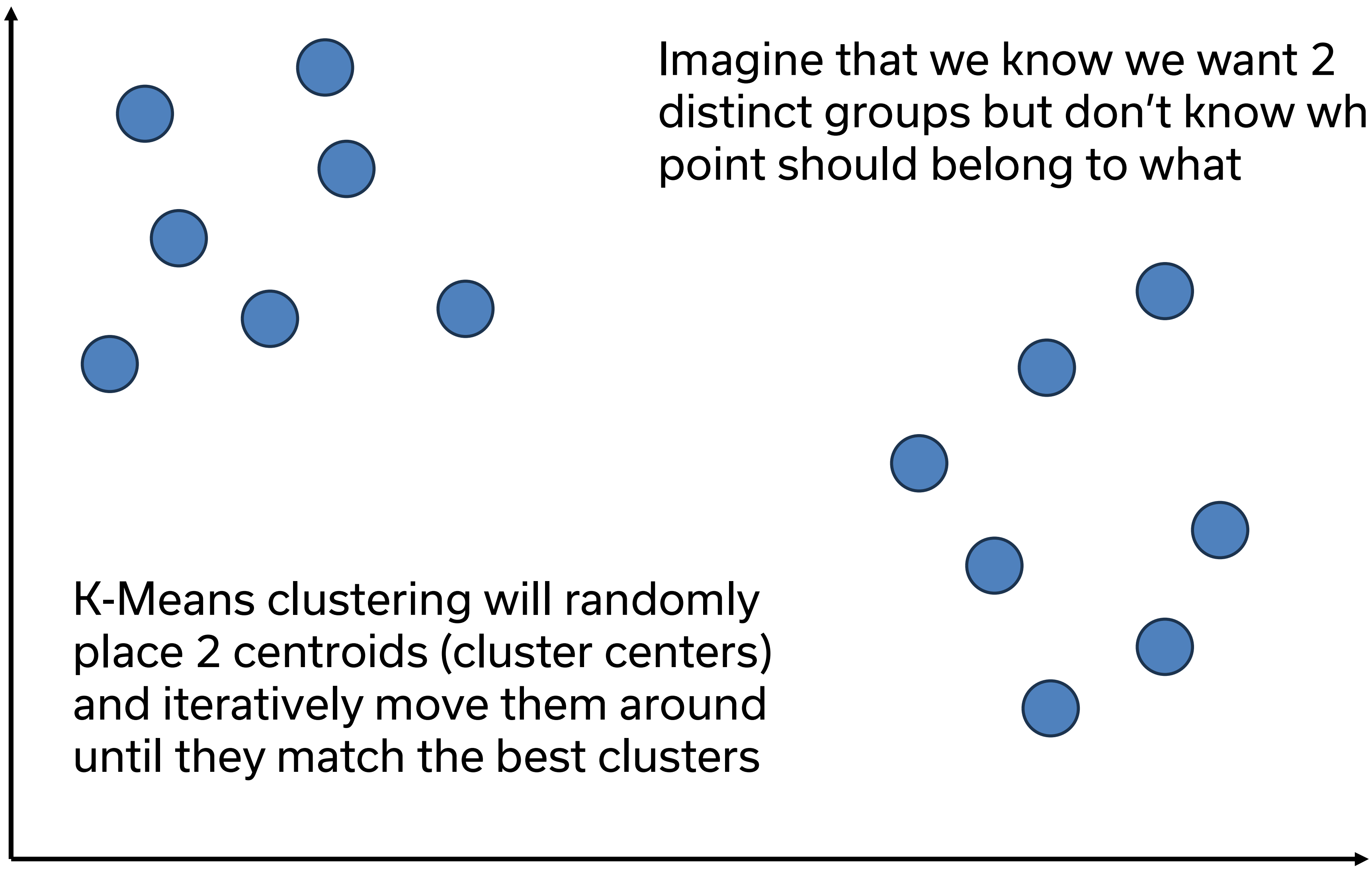
K-Means Clustering



How would a machine learning model know how to segment this data into 2 different groups?

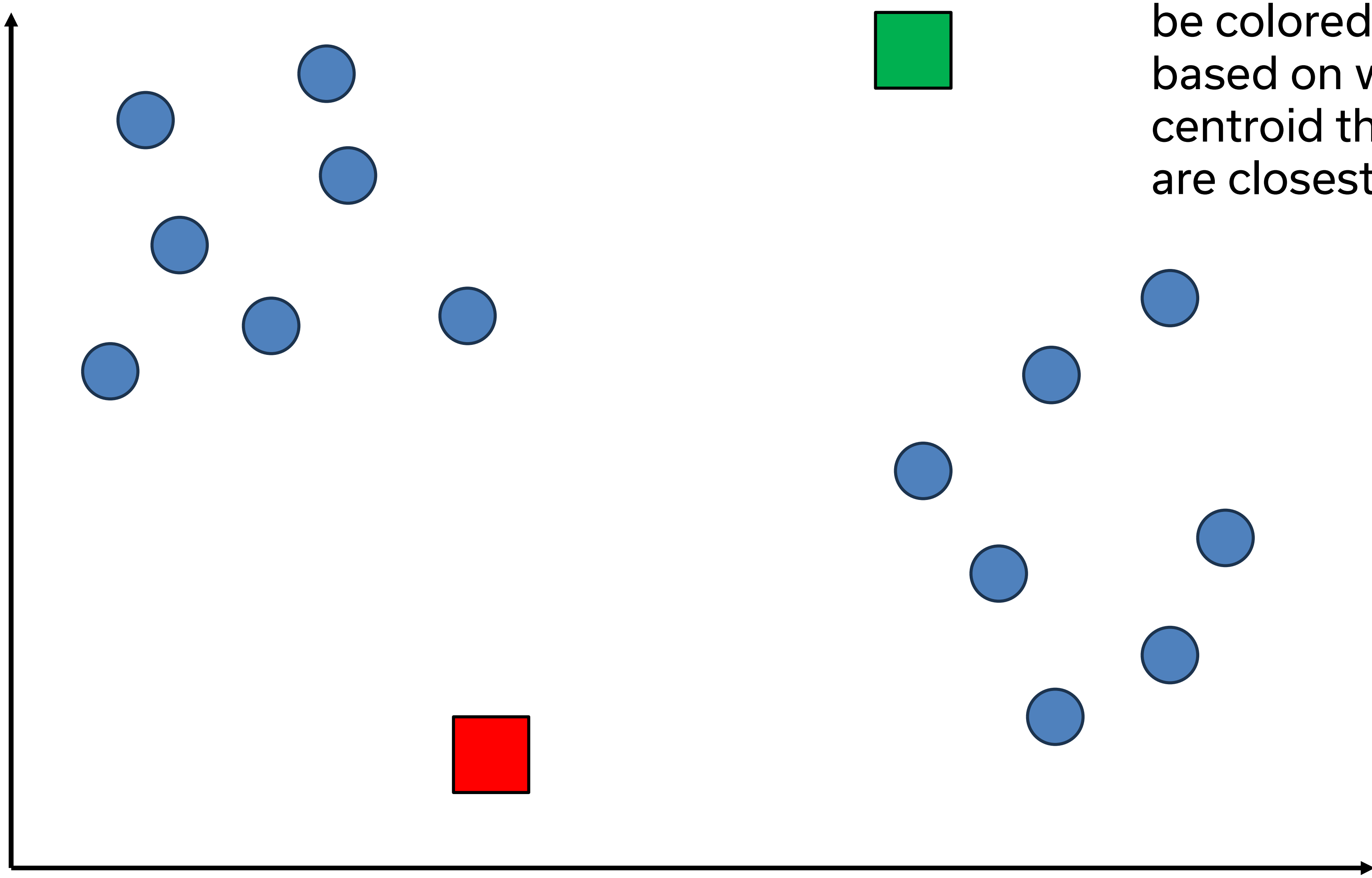
K-Means Clustering

Imagine that we know we want 2 distinct groups but don't know which point should belong to what



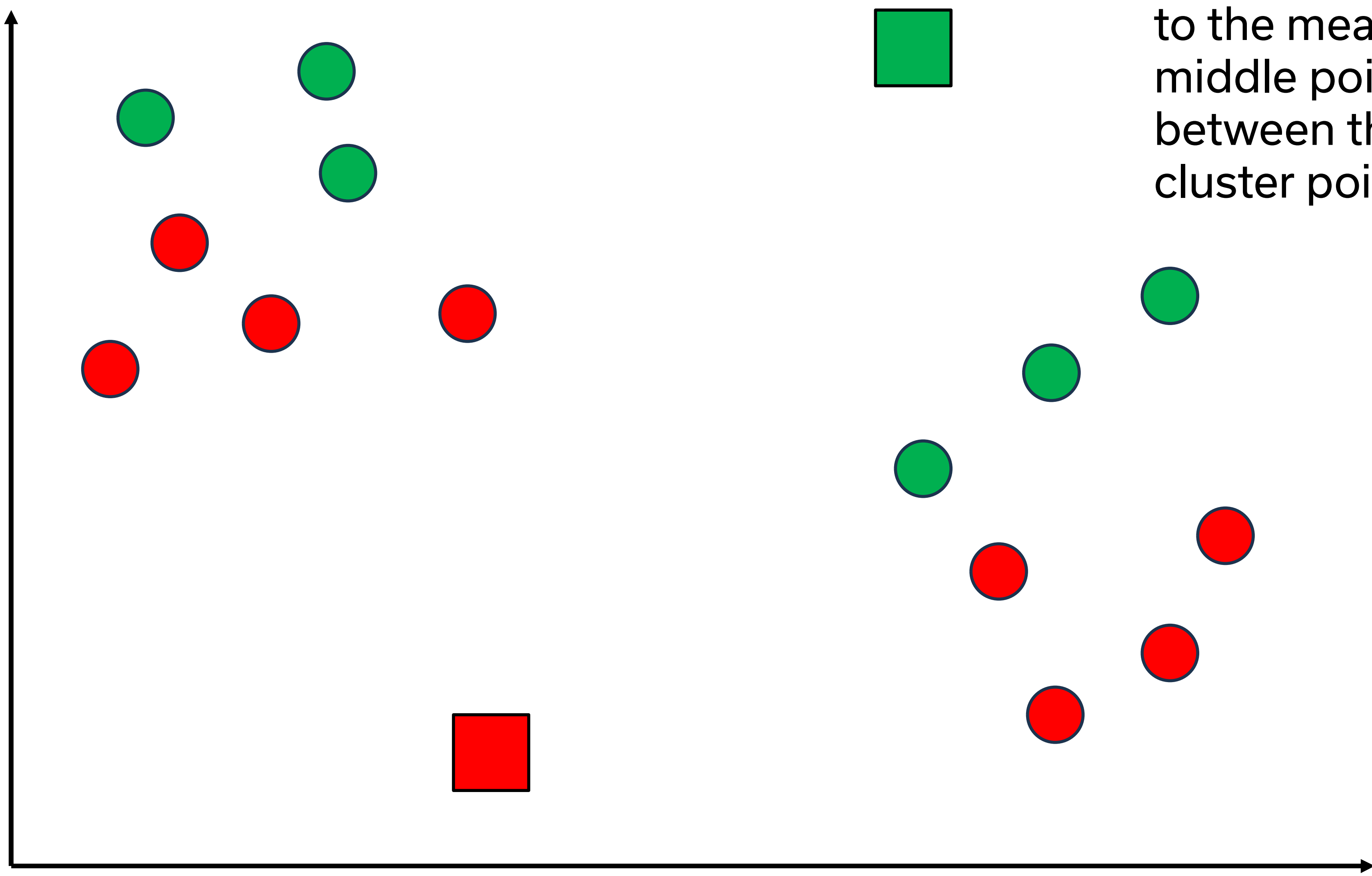
K-Means clustering will randomly place 2 centroids (cluster centers) and iteratively move them around until they match the best clusters

K-Means Clustering



All points will be colored based on what centroid they are closest to

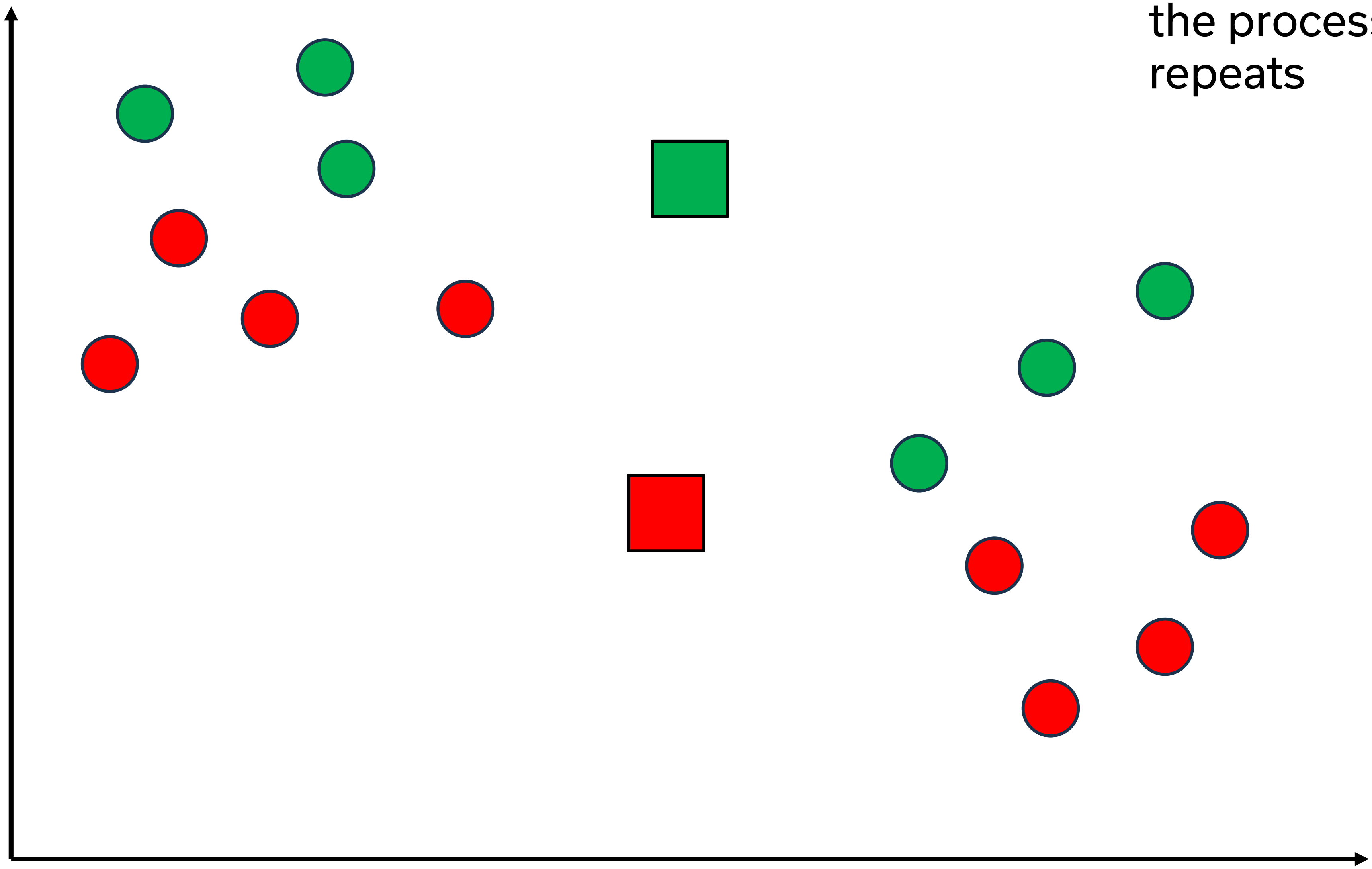
K-Means Clustering



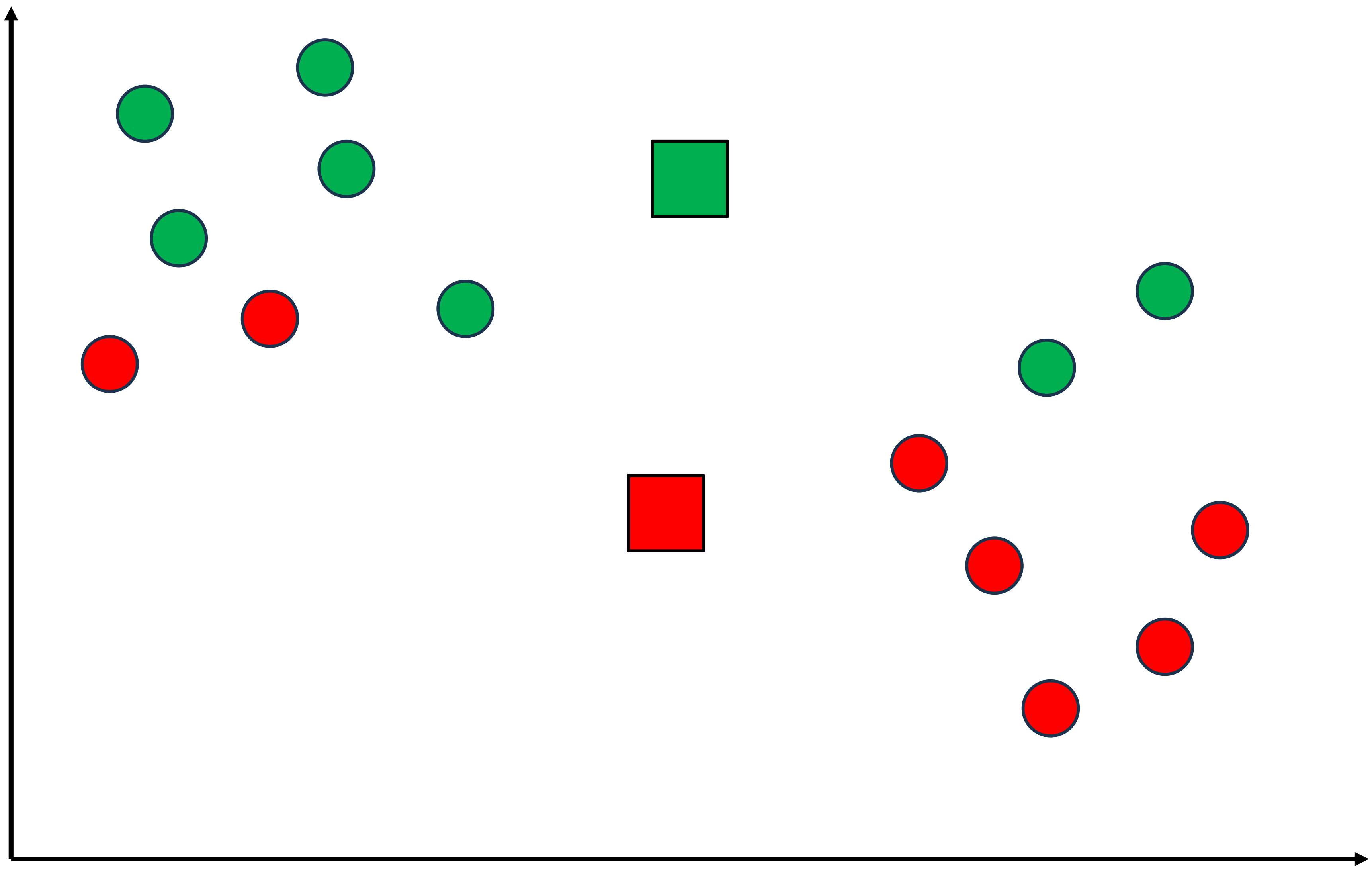
Now the centroids move to the mean, or middle point, between their cluster points

K-Means Clustering

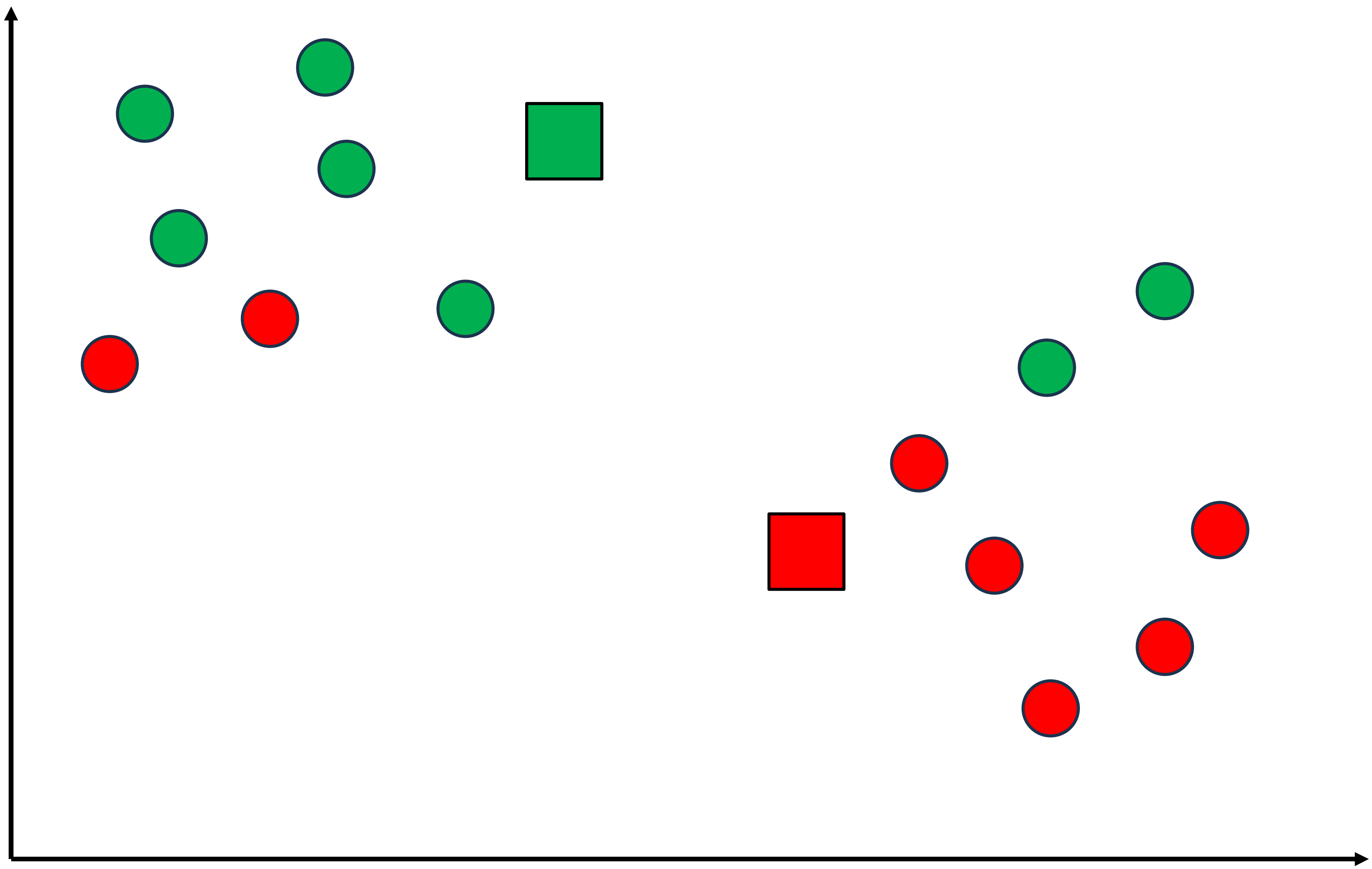
The points are recolored and the process repeats



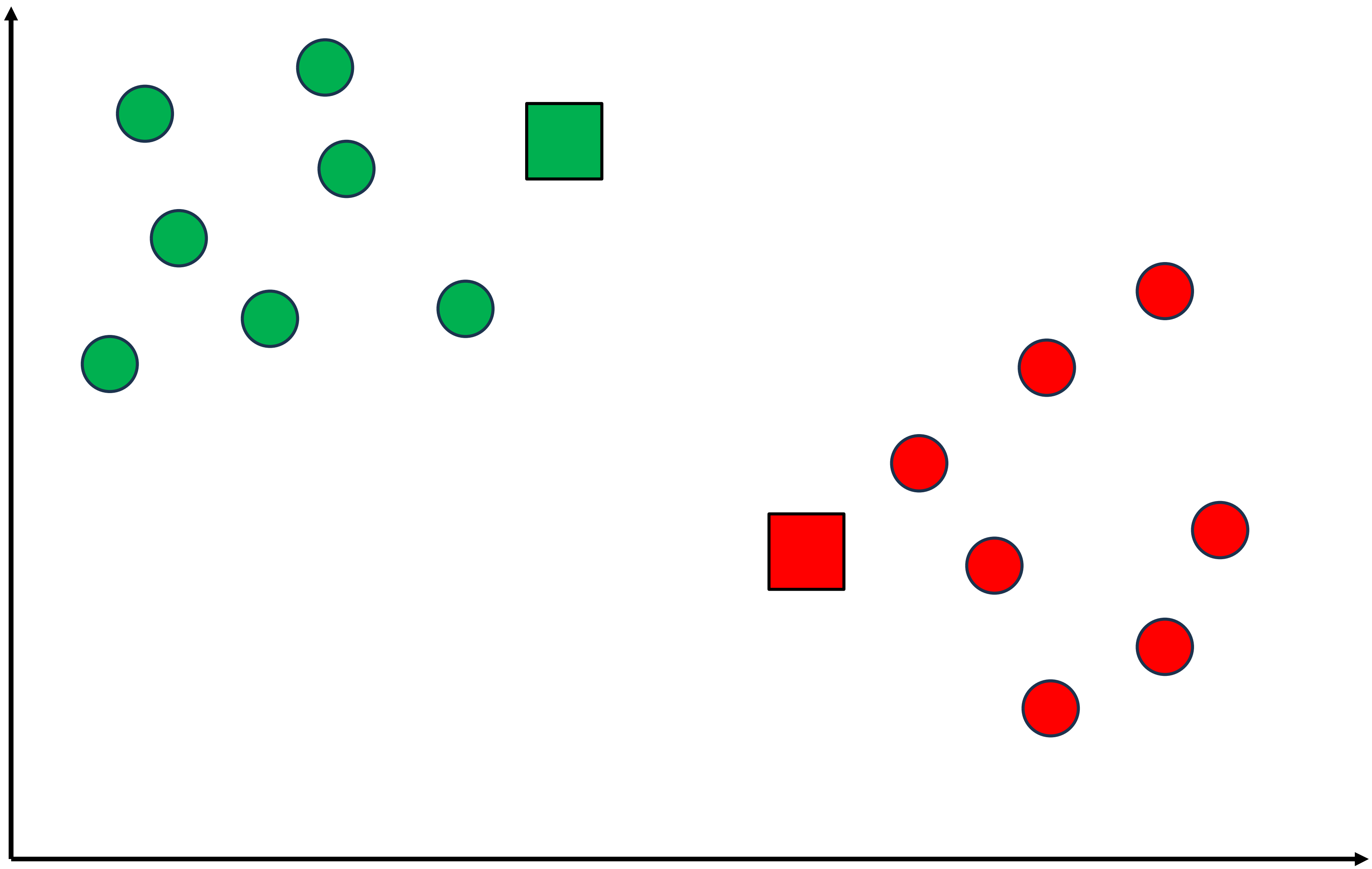
K-Means Clustering



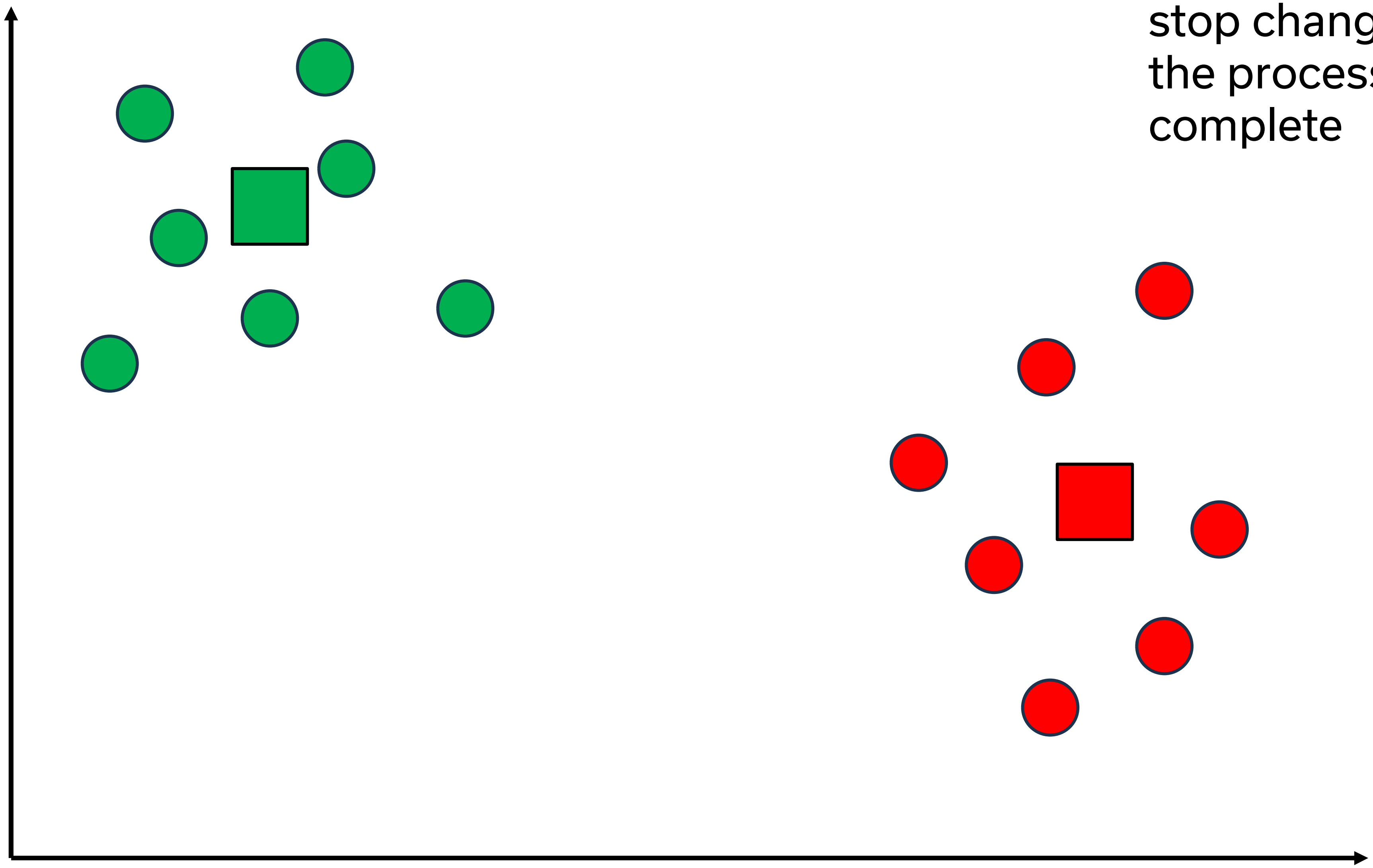
K-Means Clustering



K-Means Clustering



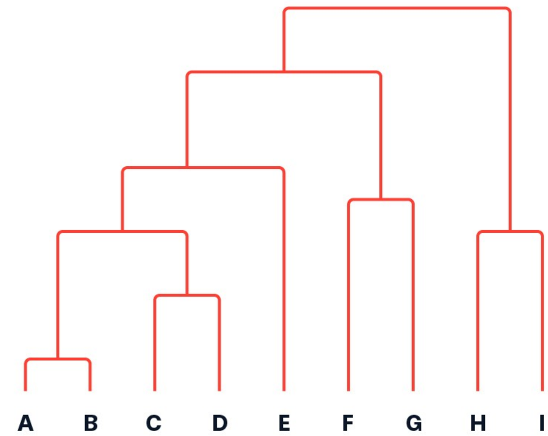
K-Means Clustering

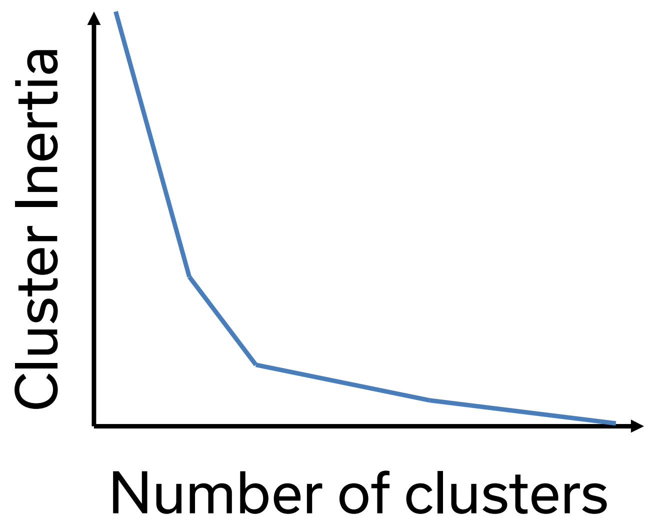


Once our colors and centroids stop changing, the process is complete

Unsupervised Learning

- What if you don't know how many clusters you should have?
  - –Build a hierarchical cluster by pairing individual points, and then pairing those pairs
  - –Define "distances" between points that could indicate groupings, and the density of points will be analyzed
  - –Test multiple values of k for traditional clustering and use visual analysis or calculate clustering scores to determine the optimal number of groups





Self-supervised Learning

- Modern AI models can sometimes turn an unsupervised learning problem into a supervised learning problem
- The model generates labels that it then tries to predict, rather than requiring external labels passed into it
- So what are the labels?
  - Part of the input data that is "held out" from the model

Self-supervised Learning

Input Sentence:

The chef [redacted] a delicious meal.

Now the model must determine the missing part of the sentence (the label)

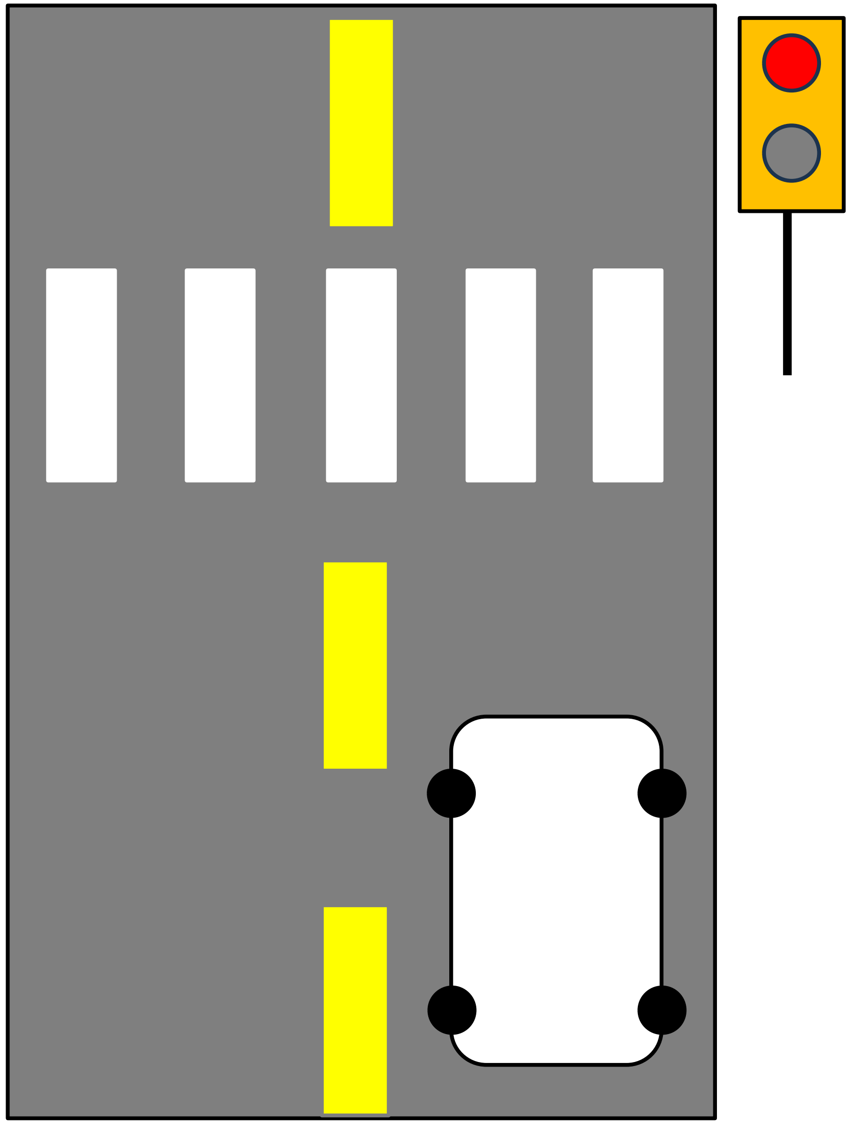
Reinforcement Learning

- An “agent” learns by performing actions and receiving rewards or punishments for those actions
- This is how AI that “makes decisions” learns
  - –Chess bot
  - –Robotic system learning to walk
  - –Self-driving cars
- The agent can repeatedly take actions in a simulation to learn which ones are favored in certain situations

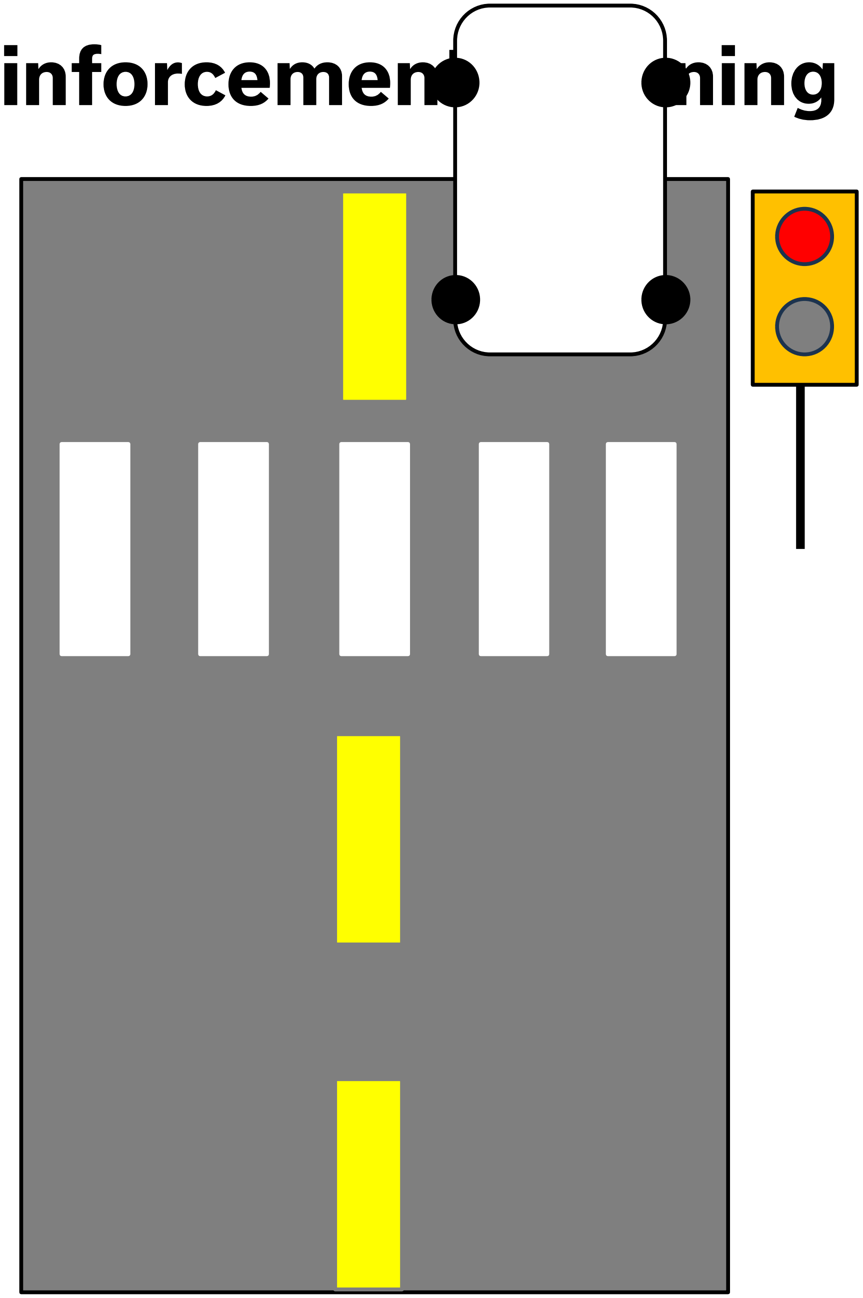
Reinforcement Learning

What is the right action to take in this situation?

In a simulation, an agent will try everything it can to see what gets rewarded and what gets punished



Reinforcemen ning



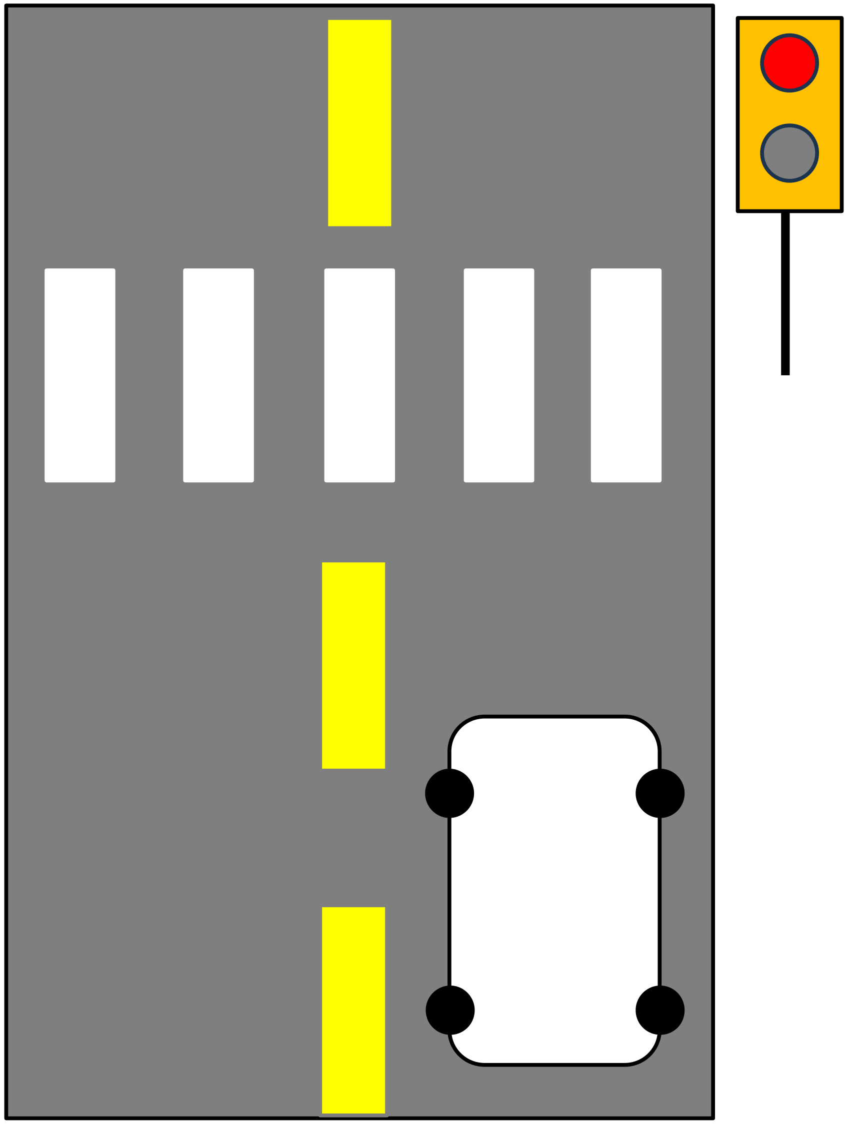
What is the right action to take in this situation?

WRONG
-50 points

When an action is taken that has been defined as undesirable, an appropriate penalty to the machine's reward function will be taken

Reinforcement Learning

What is the right action to take in this situation?

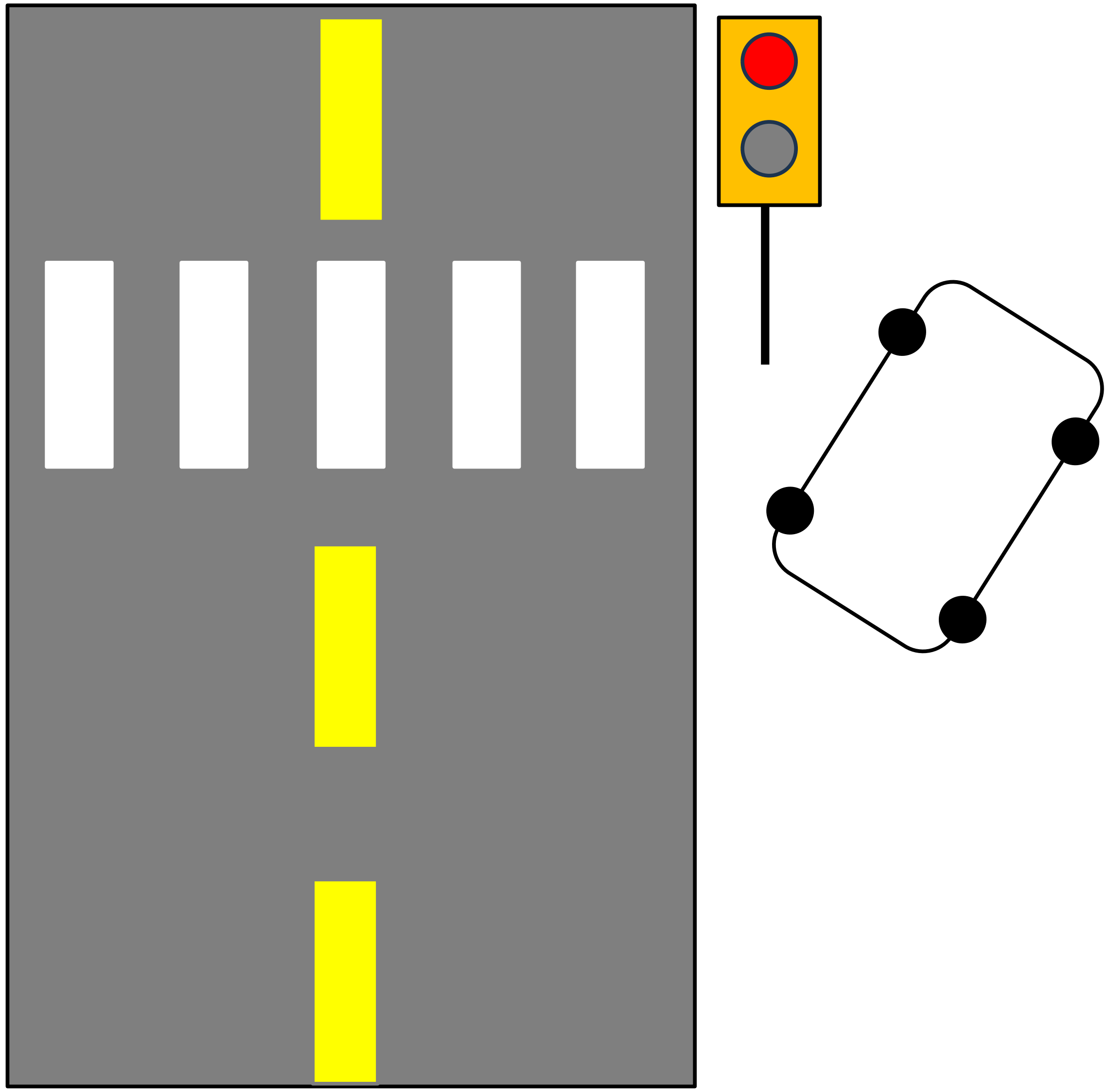


Reinforcement Learning

What is the right action to take in this situation?

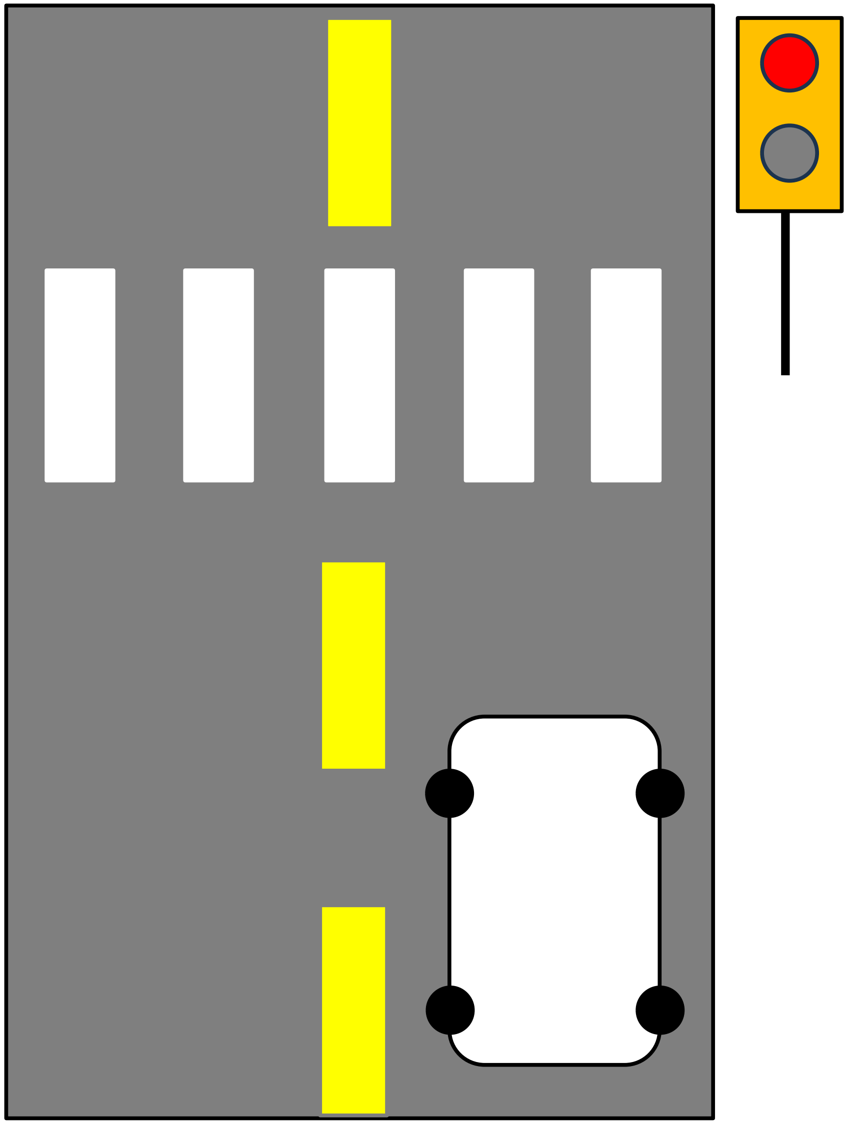
WRONG
-100 points

Different actions will have different penalties to emphasize to the machine what to do and what not to do



Reinforcement Learning

What is the right action to take in this situation?

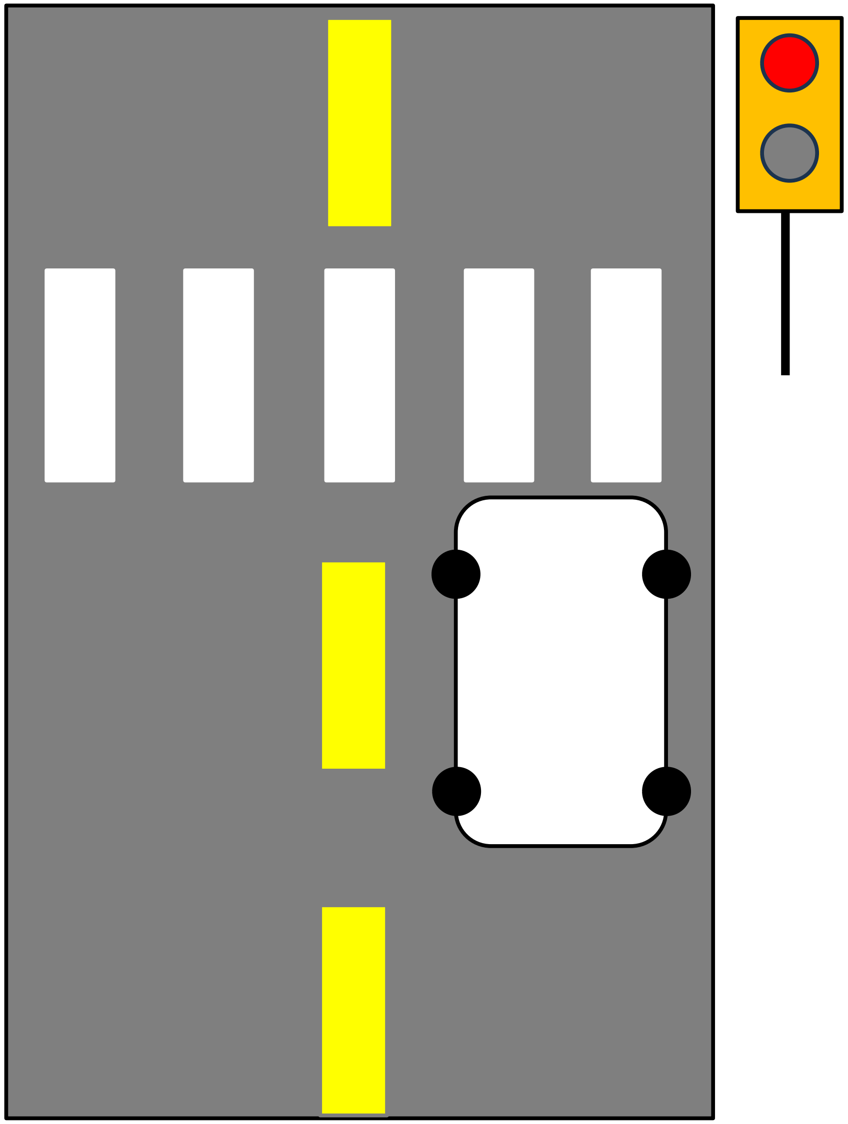


Reinforcement Learning

What is the right action to take in this situation?

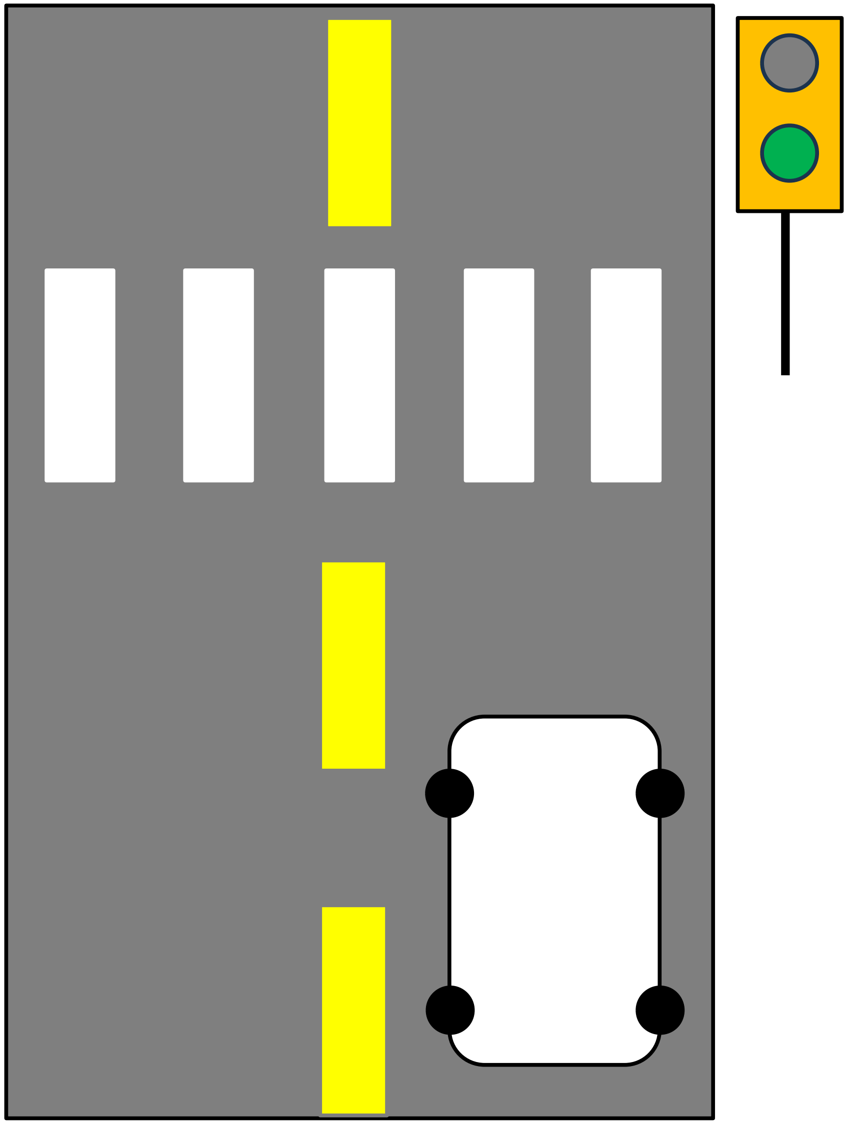
CORRECT
+100 points

When the machine does something desirable, it is rewarded



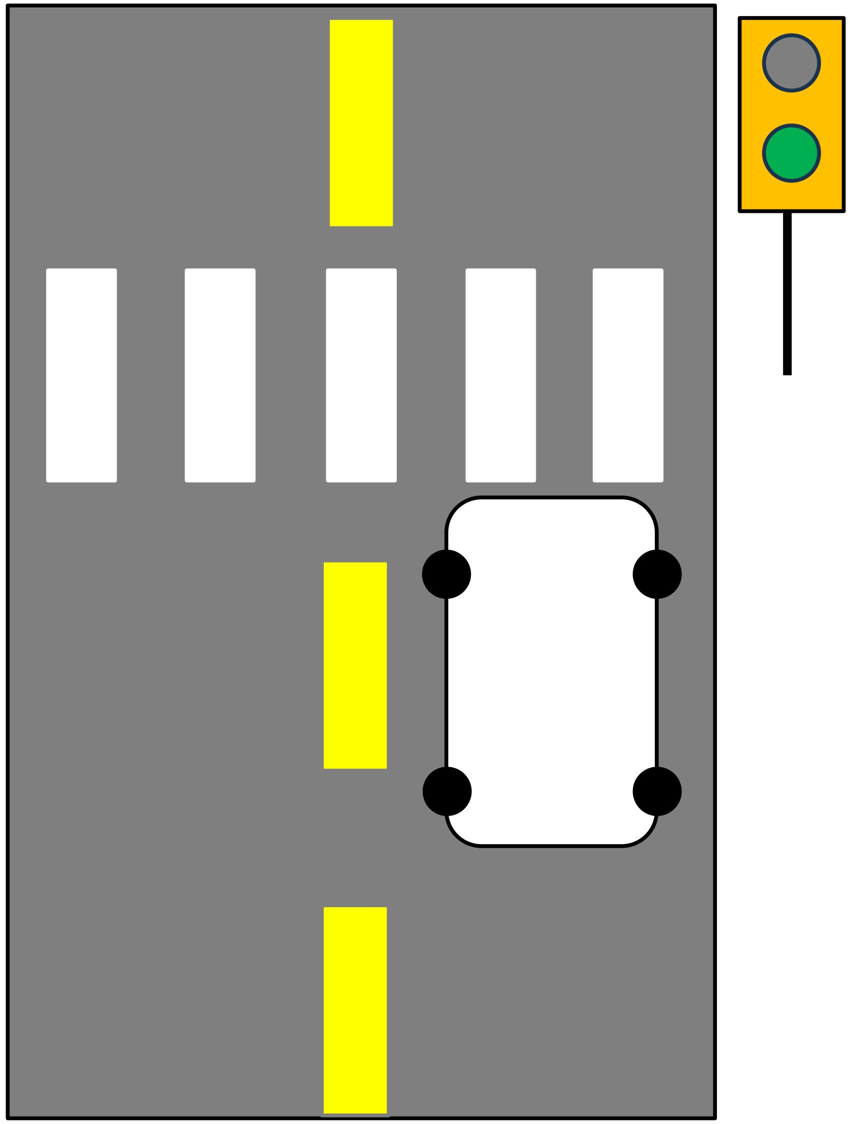
Reinforcement Learning

It will learn to take in information from its environment, as the correct action to take depends on the context



Reinforcement Learning

It will learn to take in information from its environment, as the correct action to take depends on the context

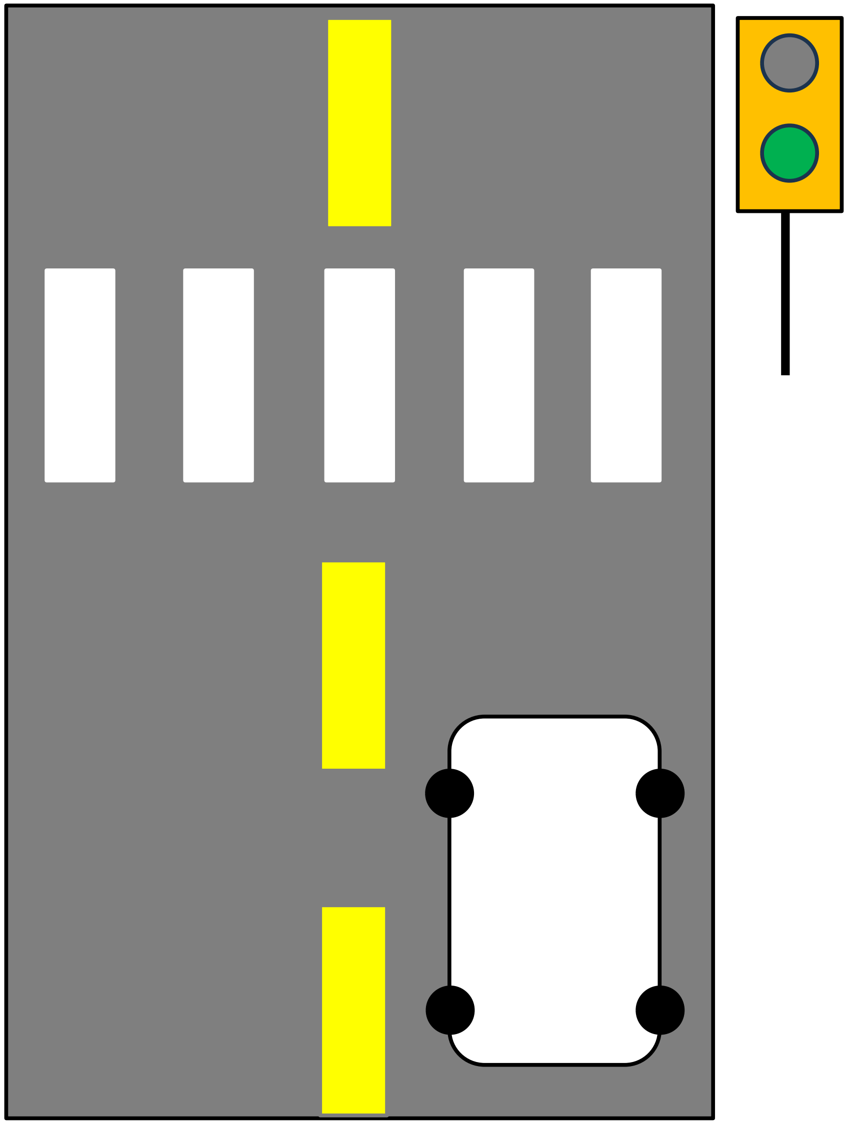


WRONG
-20 points

Now that the stoplight has changed, the action it previously took correctly is now incorrect

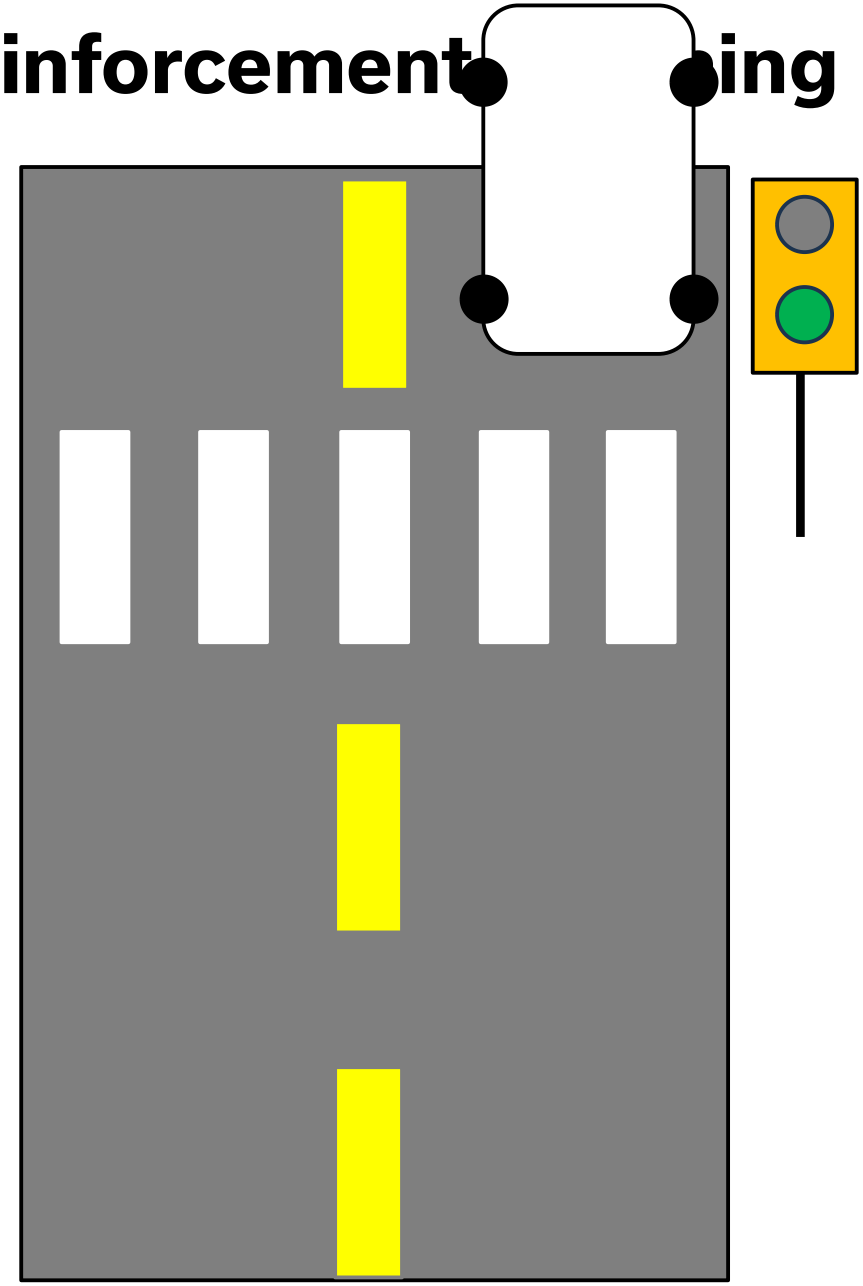
Reinforcement Learning

It will learn to take in information from its environment, as the correct action to take depends on the context



Reinforcement ing

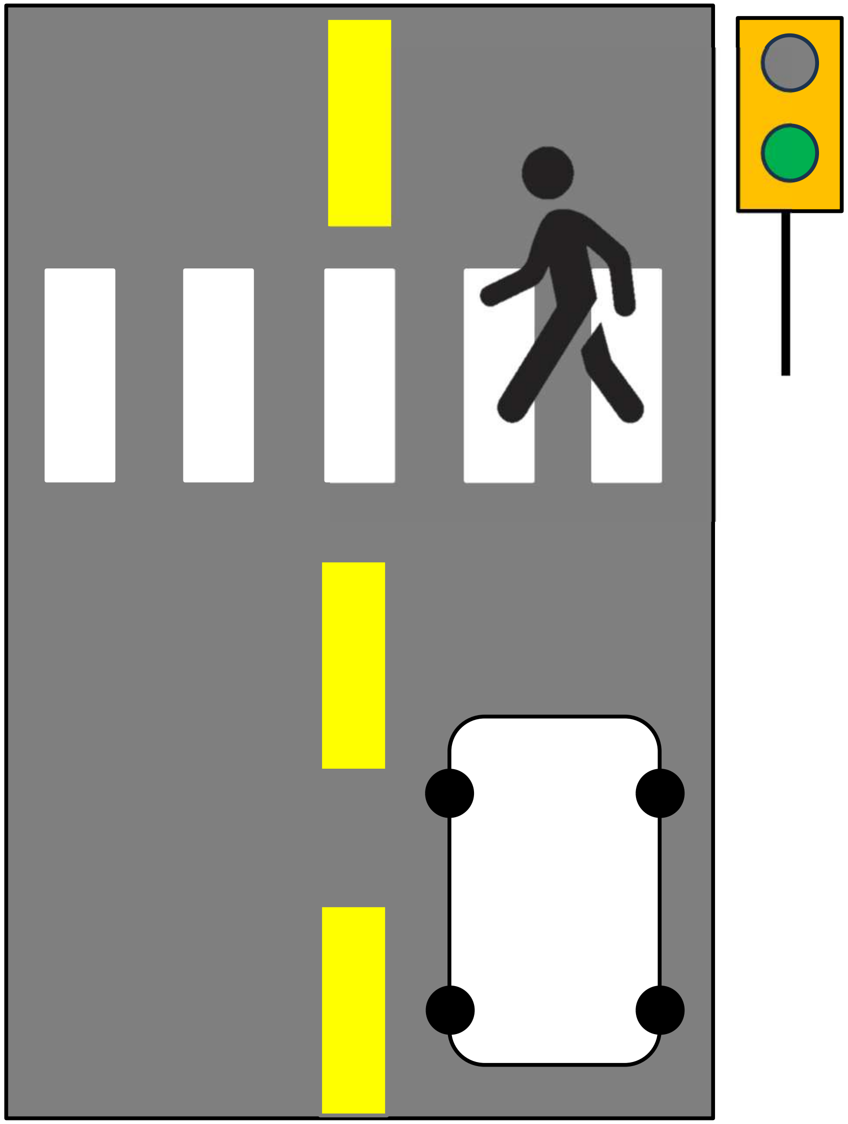
It will learn to take in information from its environment, as the correct action to take depends on the context



CORRECT
+50 points

Reinforcement Learning

The rewards and punishments for each action will determine how it should weigh each option

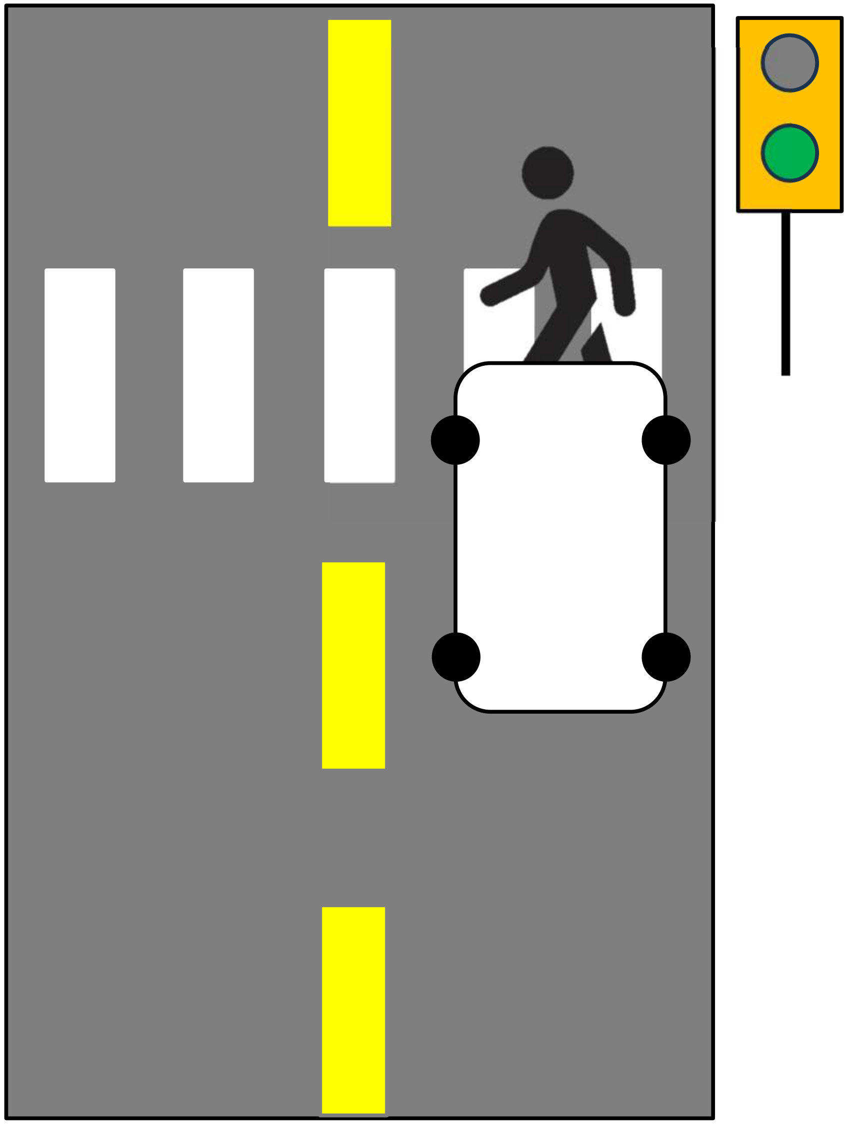


Reinforcement Learning

The rewards and punishments for each action will determine how it should weigh each option

WRONG
-1000 points

The punishment for hitting a pedestrian is greater than the reward for obeying the green light



Q-Learning

- A reinforcement learning algorithm that builds a large table of rewards for every situation it encounters, learning the "quality" of taking each action

Stoplight?	Pedestrian?	Action	Reward
Red	No	Drive	-50
Red	No	Stop	+100
Green	No	Drive	+100
Green	No	Stop	-20
Green	Yes	Drive	-1000
Green	Yes	Stop	+100

Gaps in Reinforcement Learning

- If we have to pre-define every action and situation's rewards, then we're no better off than rule-based learning
- Several ways to address this:
  - –Define only a reward for the final goal (stop the car) and automatically ripple smaller rewards back for the actions required to reach that goal (apply the brakes, steer forward, etc.)
  - –Or the opposite: define rewards for small steps and let them be added up to create rewards for larger goals
    - Rather than rewarding the car for driving 1000 feet (+100), reward it for each foot it drives (+0.1) and penalize it for each second it remains still (-0.1) or moves backwards (-0.5)
  - –ML can adapt to new situations by analyzing similarity to previous situations
    - It can "predict" the reward for an action by analyzing previous actions it has taken so it can act accordingly

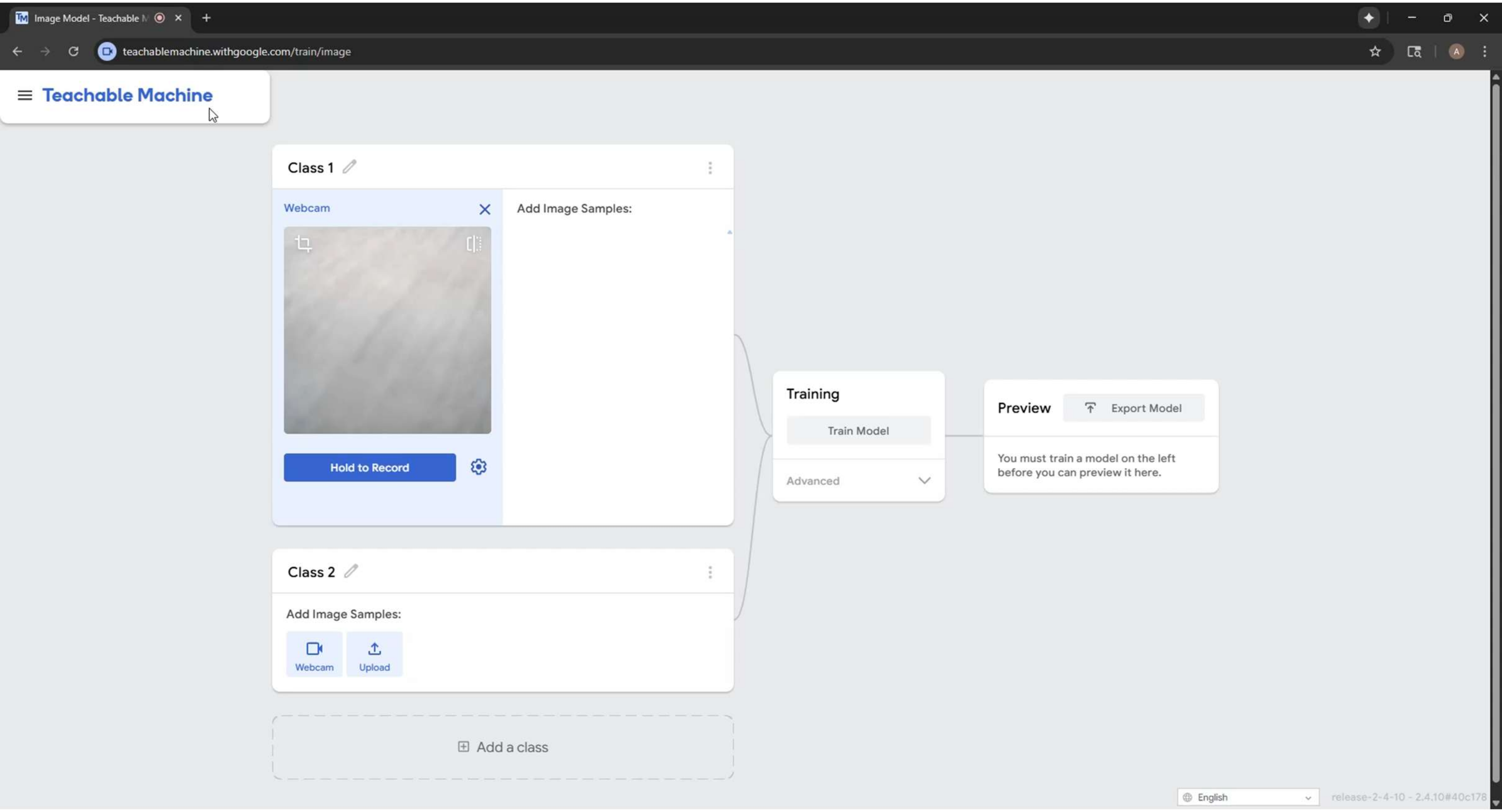
NAIRR Resources

- NAIRR provides access to many platforms that offer computing power for machine learning workflows
  - GPU Access: Indiana Jetstream 2, NCSA Delta, NVIDIA DGX, Purdue Anvil, SDSC Expanse, TACC Frontera
- Weights & Biases MLOps: tracking experiments and visualizing model performance
- Hugging Face: hosting and sharing of ML models and applications

Conclusion

- Machine learning covers a wide range of models, systems, and applications
- At its core, it’s about having a machine learn the rules of the situation instead of teaching it the rules
- With sufficient high-quality data, machine learning can match human expertise in a variety of fields

Demo Video



NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

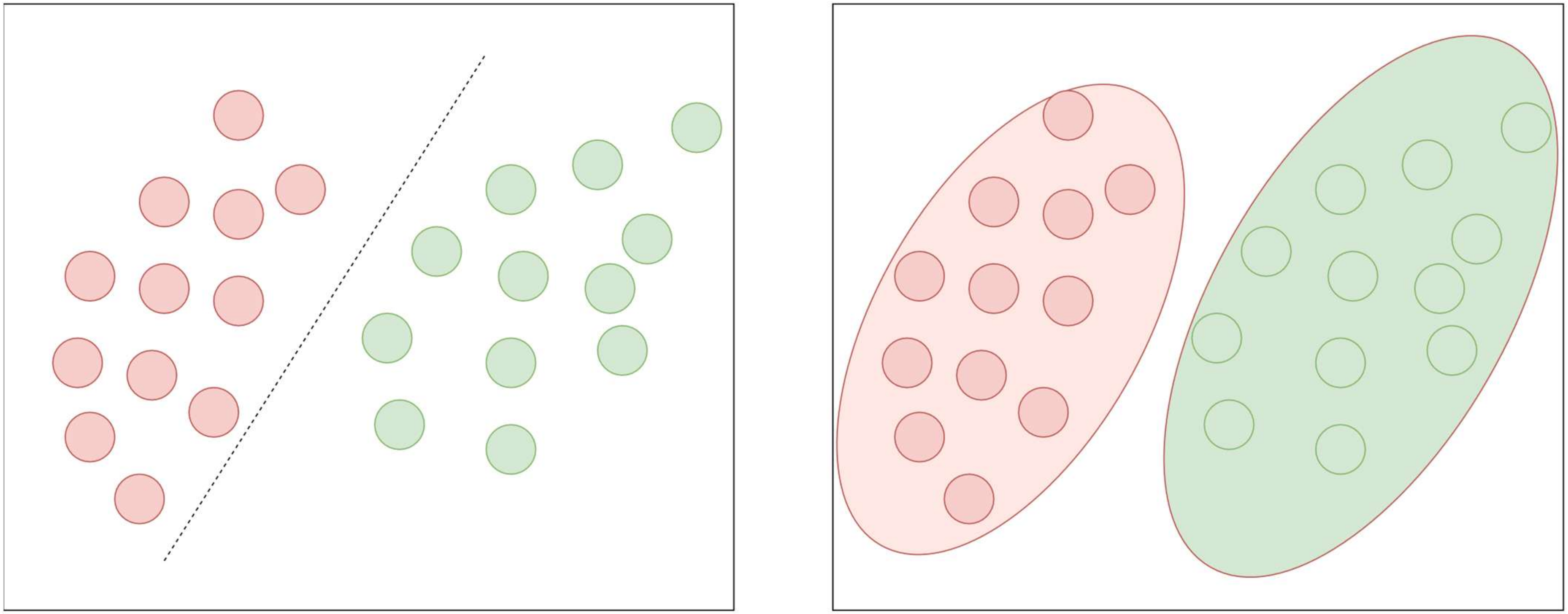
Generative AI & Vibe Coding

Vaiden Logan, BS

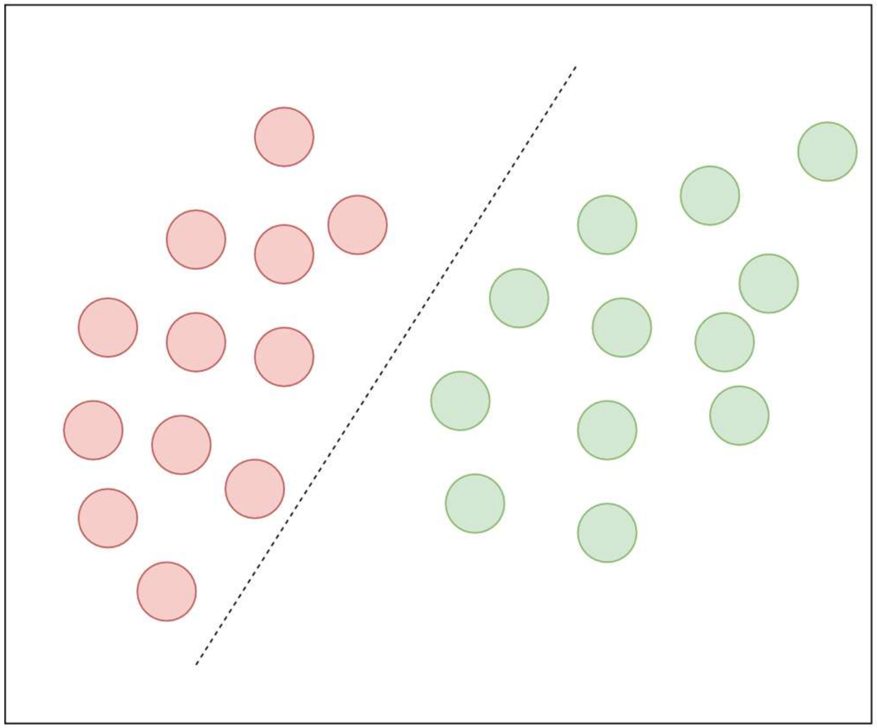
Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

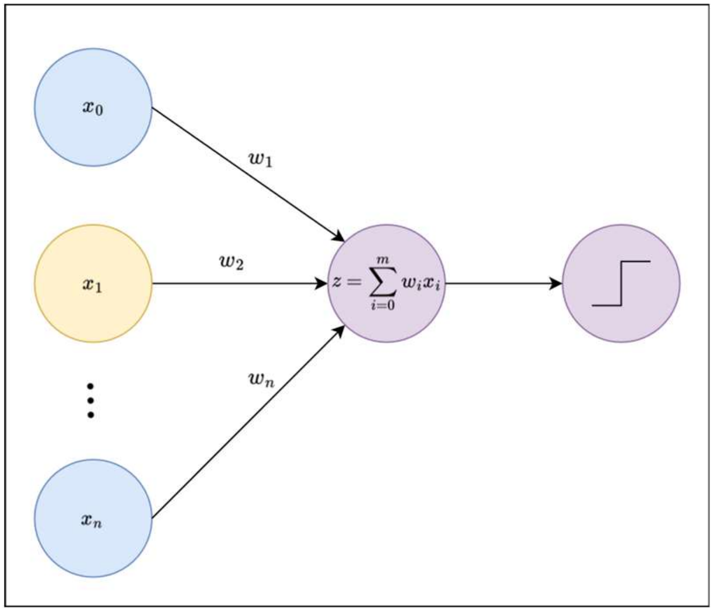
NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

Classifiers to Generative Models

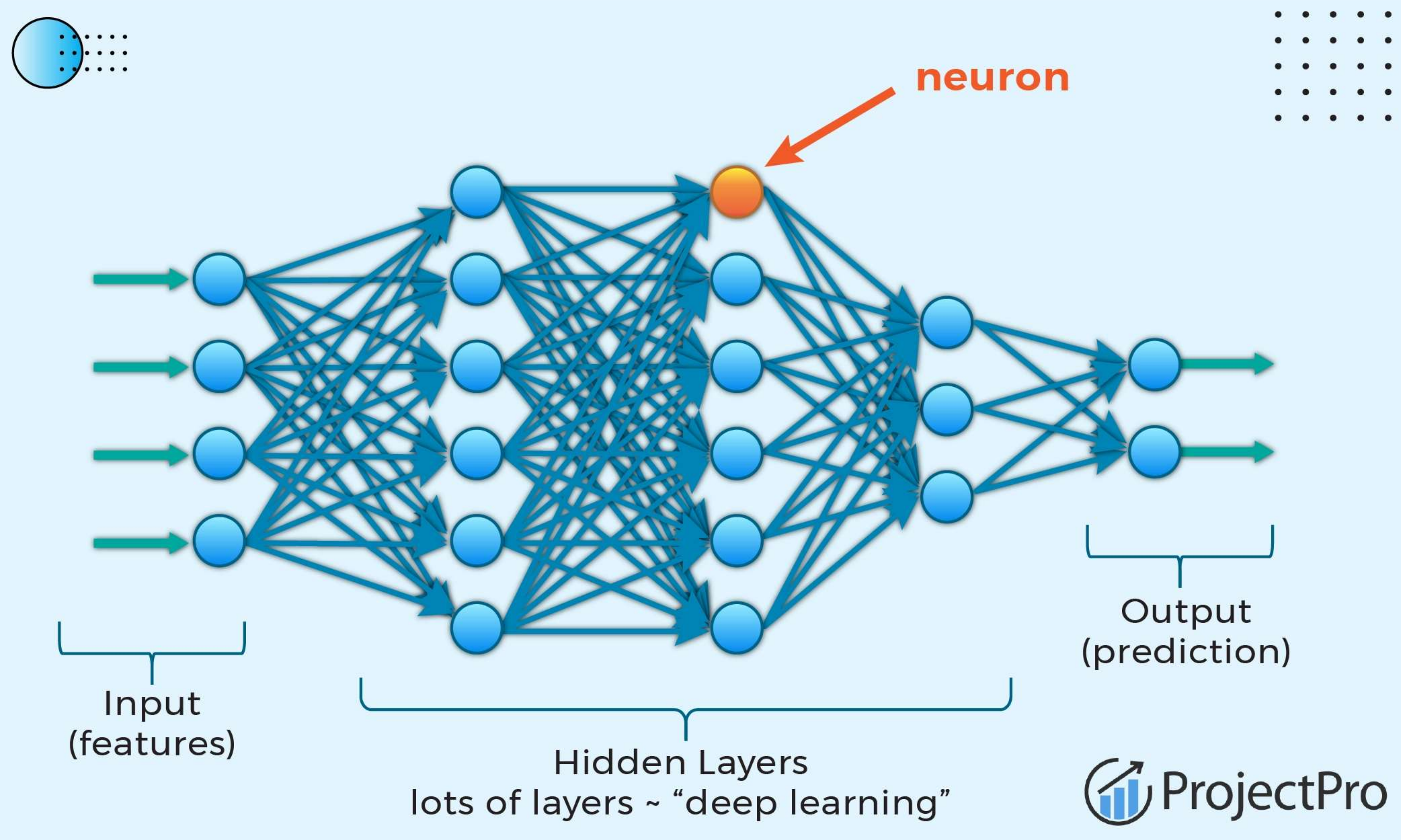


Linear Classifiers and The Perceptron





Deep Learning and CNNs



Classifier as a Teacher

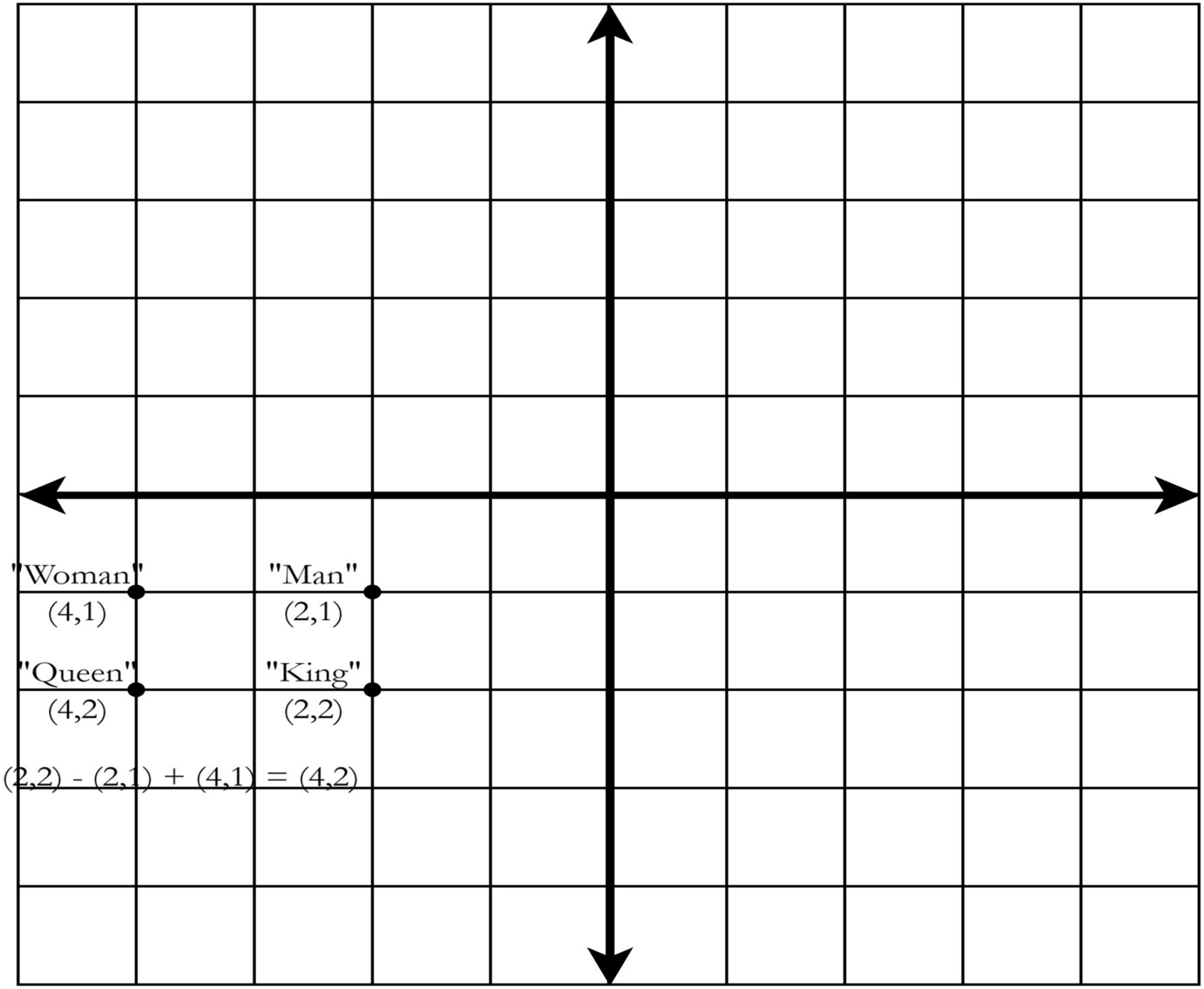
- A deep classifier extracts high-level abstract features
  - Edges, textures, shapes, semantic relationships, etc.
- A classifier "encodes" a complex input into a small, meaningful, vector to make a prediction
- Can this process be inverted
- A perfect classifier understands the "shape" of the data, and what separates a valid input from an invalid one

Autoencoders and Latent Space

- Latent space is where embeddings “live”
- An embedding is a single point in latent space

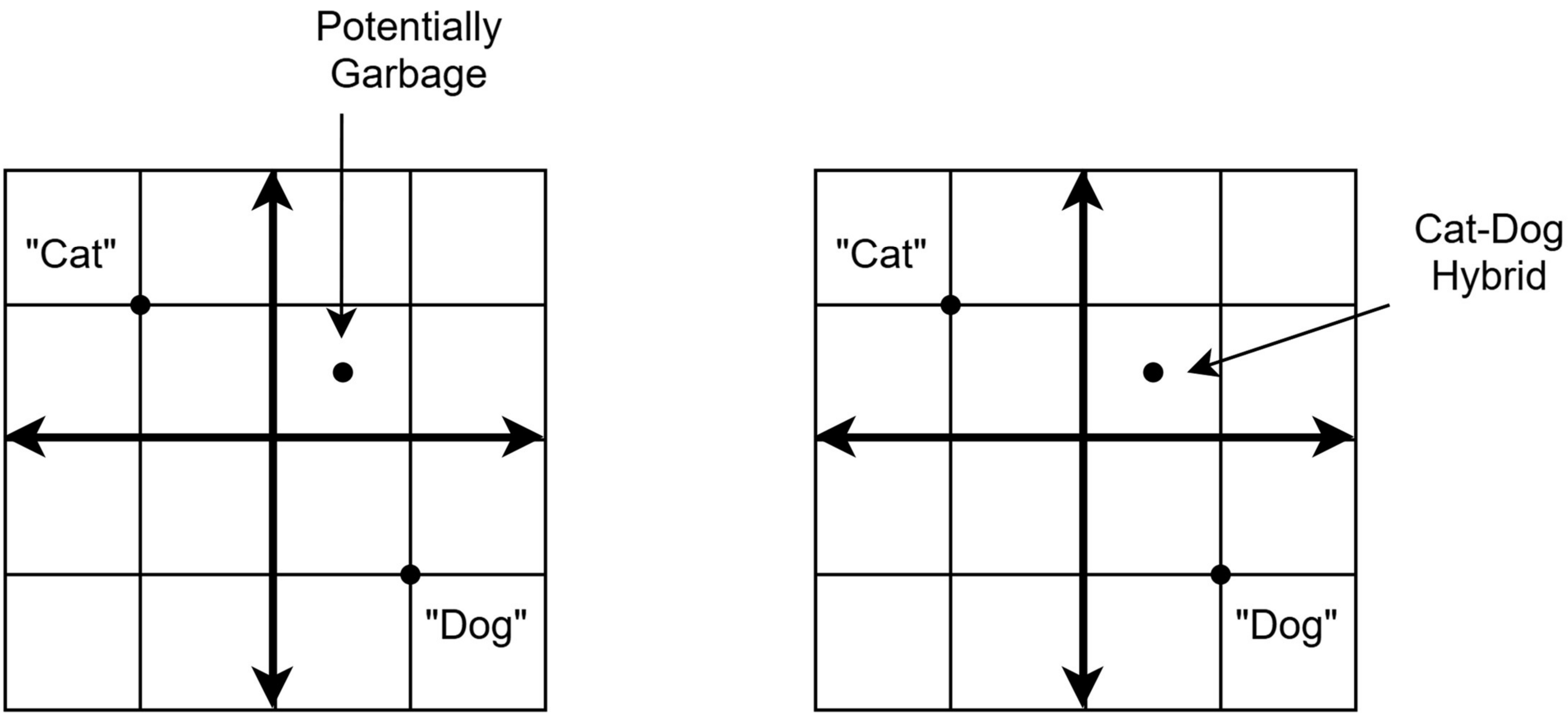
- Famous Example:

$$Vector(\text{"King"}) - Vector(\text{"Man"}) + Vector(\text{"Woman"}) = Vector(\text{"Queen"})$$


"Woman"
(4,1)
"Man"
(2,1)
"Queen"
(4,2)
"King"
(2,2)
(2,2) - (2,1) + (4,1) = (4,2)

Autoencoders and Latent Space

- A Variational Autoencoder (VAE) turns this “compression” into mapping



- Decoders then convert this to the original data format

VAEs vs Encoders

Feature	Standard Encoder	VAE
Logic	Discrete Mapping $(X \rightarrow Z)$	Probabilistic mapping $(X \rightarrow \mu, \sigma)$
Latent Space	Disjointed/Sparse (has "dead zones")	Continuous/Dense
Goal	Perfect reconstruction	Meaningful Generation

Applications for Biology or Physics research:

- **Compression:** Use the VAE to compress sensor feeds so they can be efficiently stored
- **Mapping:** Use the latent space to see if Experimental Group embeddings are shifting away from Control Group embeddings
- **Generation:** Use the decoder to generate "simulated" experimental results to test your analysis pipeline before actual data arrives

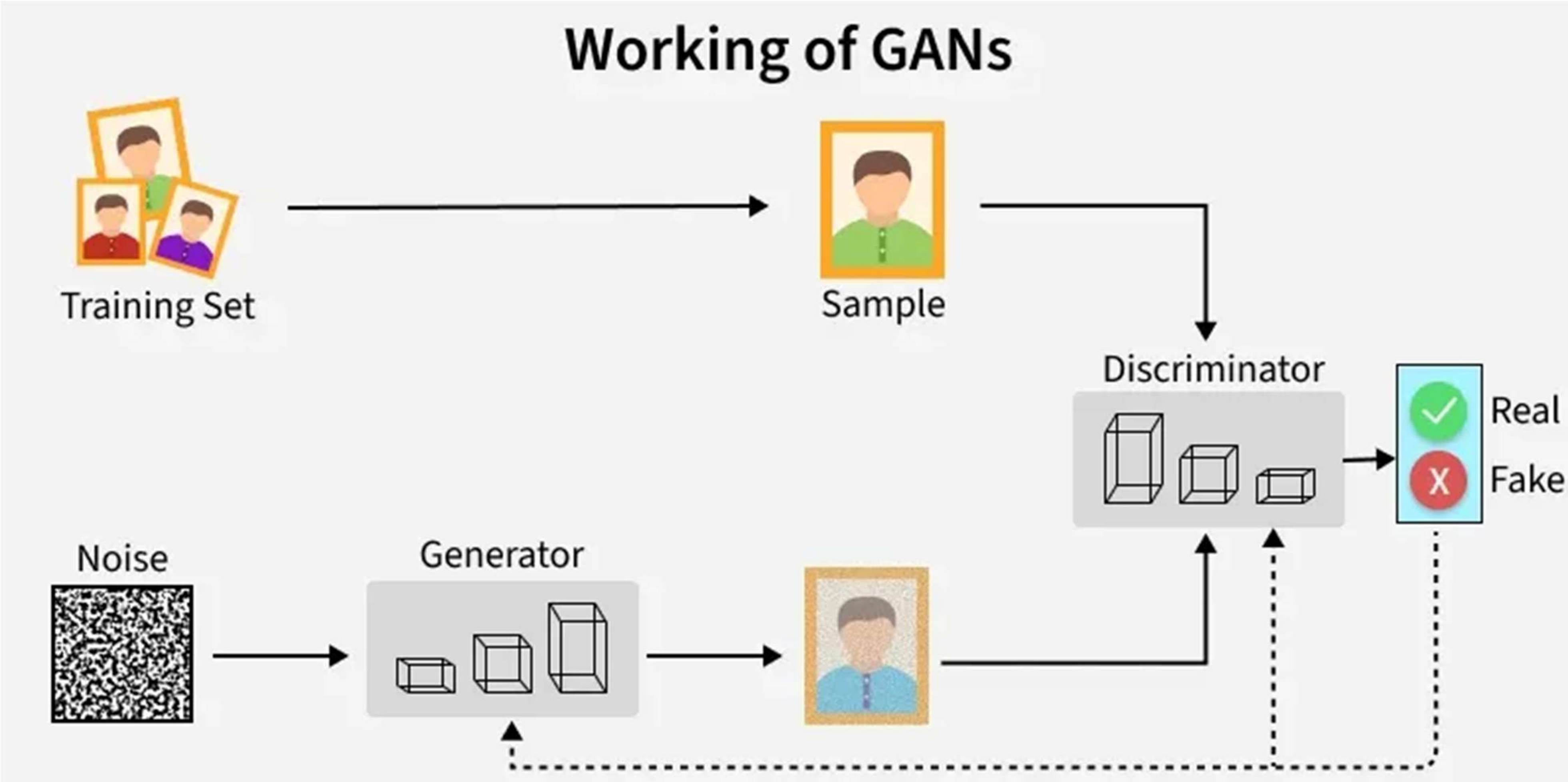
Generative Adversarial Networks (GANs)

- Two parts:
  - The Generator: Transforms a vector of random noise into a piece of data that looks real
  - The Discriminator: A "Classifier" that outputs a probability: 1 (Real) or 0 (Fake)

- Pretty simple equation:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

- Discriminator Goal: Maximize the probability of assigning the correct label to both real and fake examples
- Generator Goal: Minimize the probability that the Discriminator catches its fakes



- This is a compute-heavy and unstable technique
- Accessing HPC clusters like Summit or Delta through the NAIRR Pilot allows users to run these adversarial loops across multiple GPUs
  - This reduces training time from weeks to hours

Transformers

Issues with RNNs:

- Processed data in loops
- Very forgetful
- The model would “forget” the subject of a sentence by the time it reached the object

Self-Attention:

- RNN loop replaced with a matric operation
- Every token (word/pixel/data point) is compared against every other token simultaneously
- Every token now has a “direct pointer” to every other token

LLM Scaling Laws

Transformer models follow a Power Law relationship with three variables:

- Compute (C): Total floating-point operations (FLOPs) used during training
- Dataset Size (D): The number of tokens (words/code/data) the model sees
- Parameters (N): The number of weights/connections in the neural network

$$L(X) \propto X^{-\alpha}$$

Diffusion Models

Forward Diffusion:

- Gaussian noise is added over T steps until an image becomes static

Reverse Diffusion:

- A neural network is trained to predict how much noise is added at each step
- Starting with a block of static, noise is removed until a high-fidelity image emerges

Vibe Coding

- Creating code with generative AI
- Coined by Andrej Karpathy in 2025, it describes a workflow where you "fully give in to the vibes" and forget the underlying code exists.

Vibe Coding

- The Use Case: "Disposable" research scripts, quick data visualizations, and front-end dashboards for demos.
- The Workflow: "Describe → Generate → Execute → Provide Feedback → Repeat."
- "The "6-Month Wall":
  - Pros: 70% faster prototyping; allows domain experts (biologists/physicists) to build tools without deep CS roots.
  - Cons: Compounding technical debt; security vulnerabilities; "Spaghetti code" that is hard to maintain without the AI's help.

The Pre-training Pipeline

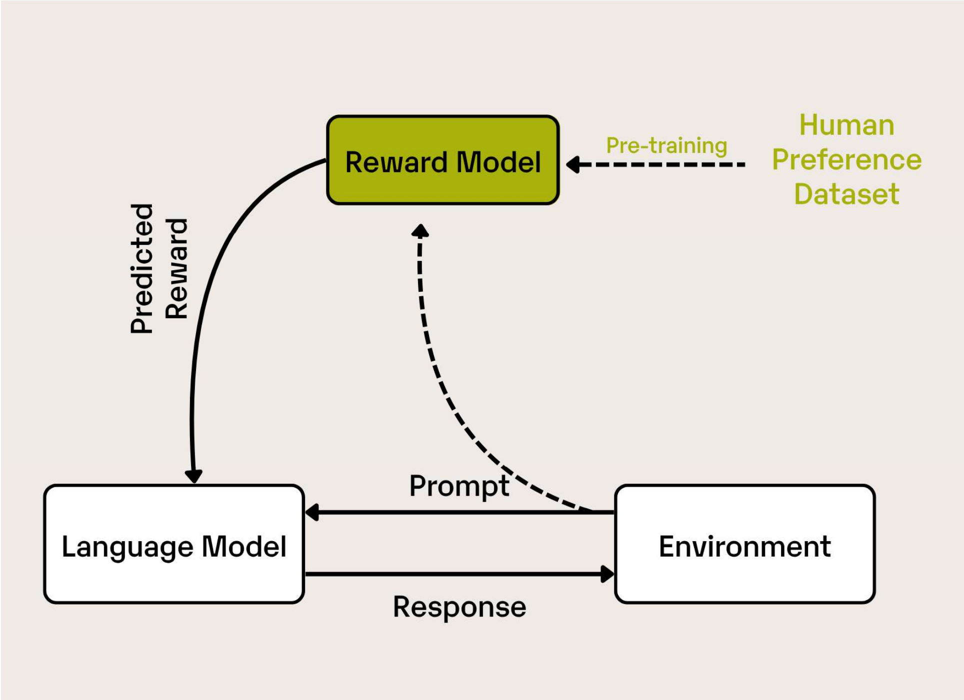
- Data Sourcing (The Corpus): Trillions of tokens from public GitHub repositories, StackOverflow threads, and technical documentation.
- Cleaning & Deduplication: Aggressive filtering to remove "boilerplate" (like massive node_modules accidentally checked in), low-stars repositories, and exact duplicates.
- Objectives:
  - Understanding the syntax of a language (grammar, brackets, indentation, etc.)
  - Fill-In-The-Middle: Model is given the start and end of a file and must guess the middle (Allows a model to edit existing code rather than just appending to the end)

stackoverflow

GitHub

Alignment & RLVR

- Supervised Fine-Tuning (SFT): Training on high-quality "Instruction-Solution" pairs (e.g., "Write a BFS algorithm in Rust").
- RLHF (Human Feedback): Humans rank multiple code solutions from "Best Practice" to "Buggy."
- The Breakthrough: RLVR (Reinforcement Learning with Verifiable Rewards):
  - Unlike natural language, code can be objectively verified.
  - The model generates a solution → The system runs it through a Compiler and a Unit Test suite.
  - Reward: If it compiles and passes the tests, the model is "rewarded" (weights are reinforced). If it fails or has a runtime error, it's "punished."



The Upside

- The Workflow: Natural Language → Logic → Artifact.
- The Advantage: Eliminating the "Syntax Tax." Developers spend more time on system design and user experience rather than debugging semicolons or library imports.
- Prototyping Velocity: faster task completion in controlled lab settings.

The Downside

- Technical Debt: AI-generated code is often "Spaghetti Code" that works today but is structurally fragile.
- Security Gaps: Studies in late 2025 showed that up to 45% of AI-generated code contains classic vulnerabilities (OWASP Top 10), like missing input validation or hardcoded secrets.
- The Maintenance Wall: If you didn't write it, you can't easily "rewire" it when it breaks.

Feature	Pros	Cons
Speed	Near-instant prototypes	Hard-to-find logic bugs
Effort	Low cognitive load	High Maintenance Debt
Accessibility	Anyone can build	Security/Compliance risks
Logic	“It just works”	“Why does it work?”

Foundational Models, LLMs, & Computer Vision

Mitchell Klusty, BS
Vaiden Logan, BS

Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

Applying ML to Language

Applying ML to Language

Applications of NLP

- Search Engines
- Voice Assistant
- Email Spam Filter
- Auto-complete/Grammar Check
- Literature Review
- Clinical Note Processing
- Translation
- Code Generation
- Summarization
- Question Answering
- And More

How Computers "Read"

- Strings: Data type of text
- Words are made up of characters
- To a computer, just seen as a set of bits

Hello world!

Human Readable Text String

How Computers "Read"

- Strings: Data type of text
- Words are made up of characters
- To a computer, just seen as a set of bits

Hello world! — Human Readable Text String

\x48\x65\x6C\x6C\x6F\x20\x57\x6F\x72\x6C\x64\x21 — UTF-8 Encoding

How Computers “Read”

- Strings: Data type of text
- Words are made up of characters
- To a computer, just seen as a set of bits

Hello world!	Human Readable Text String
\x48\x65\x6C\x6C\x6F\x20\x57\x6F\x72\x6C\x64\x21	UTF-8 Encoding
01001000 01100101 01101100 01101100 01101111 00100000 01110111 01101111 01110010 01101100 01100100 00100001	Binary representation

Early Approaches: Text as Statistics

- Instead of meaning, look for counts or patterns

Bag of Words: Find word importance by frequency within inputs

"Large language models are transforming research by enabling rapid analysis of text, automation of documentation, and intelligent search across large collections of data. These models help researchers summarize papers, generate hypotheses, and explore complex datasets more efficiently."

word	count
of	3
large	2
and	2
models	2
transforming	1
research	1
language	1

Early Approaches: Text as Statistics

Word Cloud

Highlights most common words, meant to show meaning of text

Early Approaches: Text as Statistics

Problems:

- "Stop-words": of, and, the, etc.
- Order of words ignored
- Doesn't tell us much
- Meaning of text not captured very well

Early Approaches: Text as Statistics

N-grams: sequences of *n* consecutive words

- Captures some context and patterns

This is a bird
This is a cat
This is a dog

n = 2

Early Approaches: Text as Statistics

N-grams: sequences of *n* consecutive words

- Captures some context and patterns

This is a bird
This is a cat
This is a dog

n = 2

Early Approaches: Text as Statistics

N-grams: sequences of *n* consecutive words

- Captures some context and patterns

This is a bird
This is a cat
This is a dog

n = 2

Early Approaches: Text as Statistics

N-grams: sequences of *n* consecutive words
- Captures some context and patterns

This is a bird
This is a cat
This is a dog

n = 2

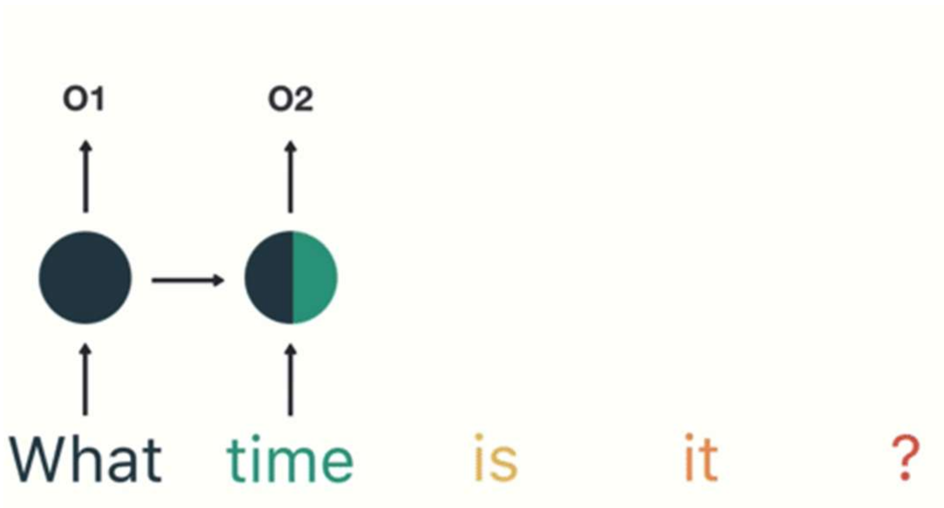
Predictive Text
- If we see "This", predict next word is "is"
- What if we see "a"?

Contextual Understanding
- "read a book" vs "book a flight"
- "This is good" vs "This is not good"

More advanced, but still very limited

Recurrent Neural Networks (RNNs)

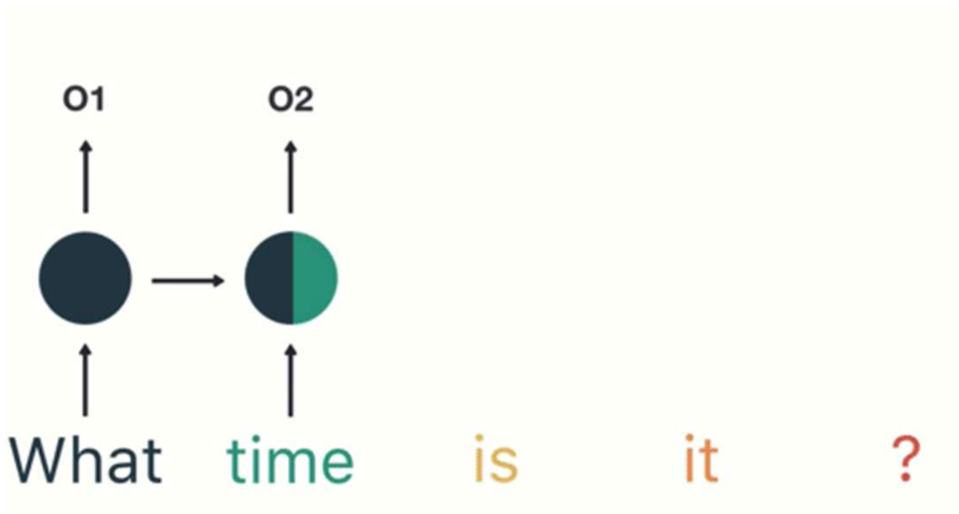
- ML technique to understand information in a sequence
- Processes data in order, start to finish, one step at a time
- Keeps a “memory” of what it’s already seen
- Each step dependent on what came before



Recurrent Neural Networks (RNNs)

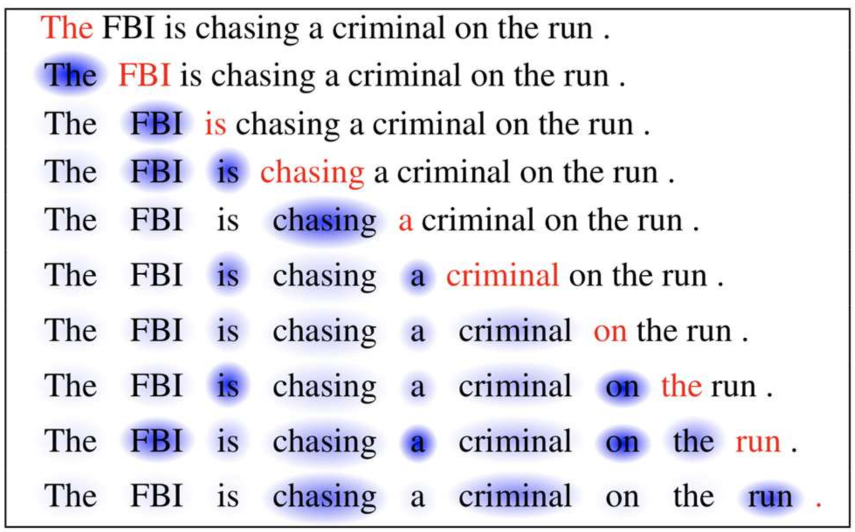
Limitations

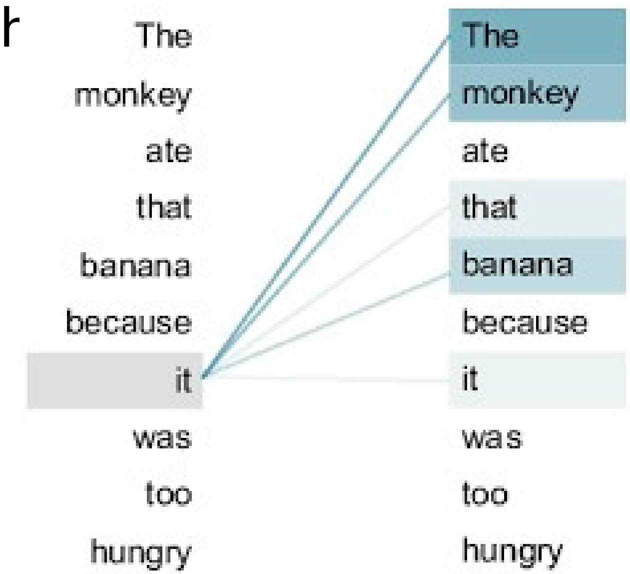
- “Memory” fades over long sequences
- Slow and hard to scale



Attention Is All You Need

- 2017 paper introducing the concept of attention and transformer models
- Through seeing vast amounts of examples, model begins to understand relationships in text
- Model looks at entire sequence at once
- Each word “attends” to each other word
  - Stronger attention indicates stronger relationsh

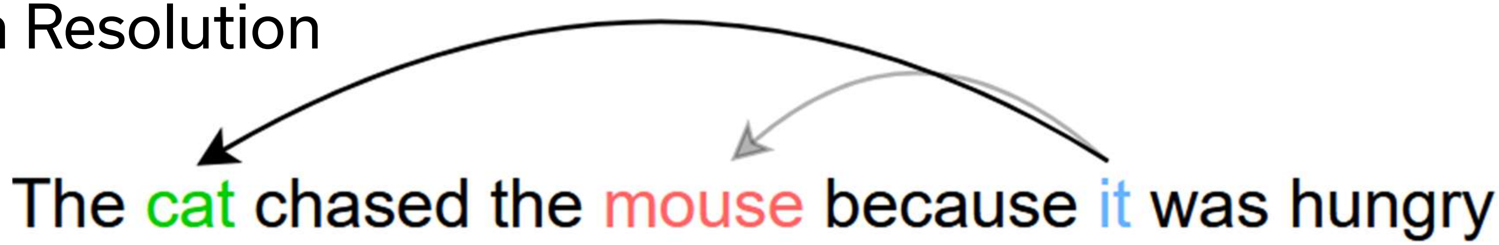




Attention Is All You Need

- 2017 paper introducing the concept of attention and transformer models
- Through seeing vast amounts of examples, model begins to understand relationships in text
- Model looks at entire sequence at once
- Each word “attends” to each other word
  - Stronger attention indicates stronger relationship

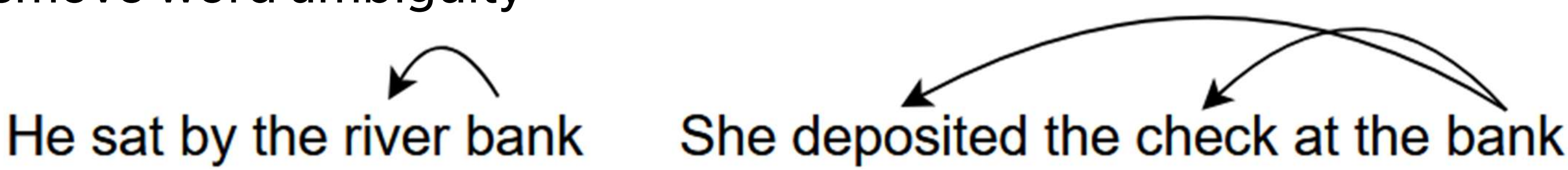
Pronoun Resolution



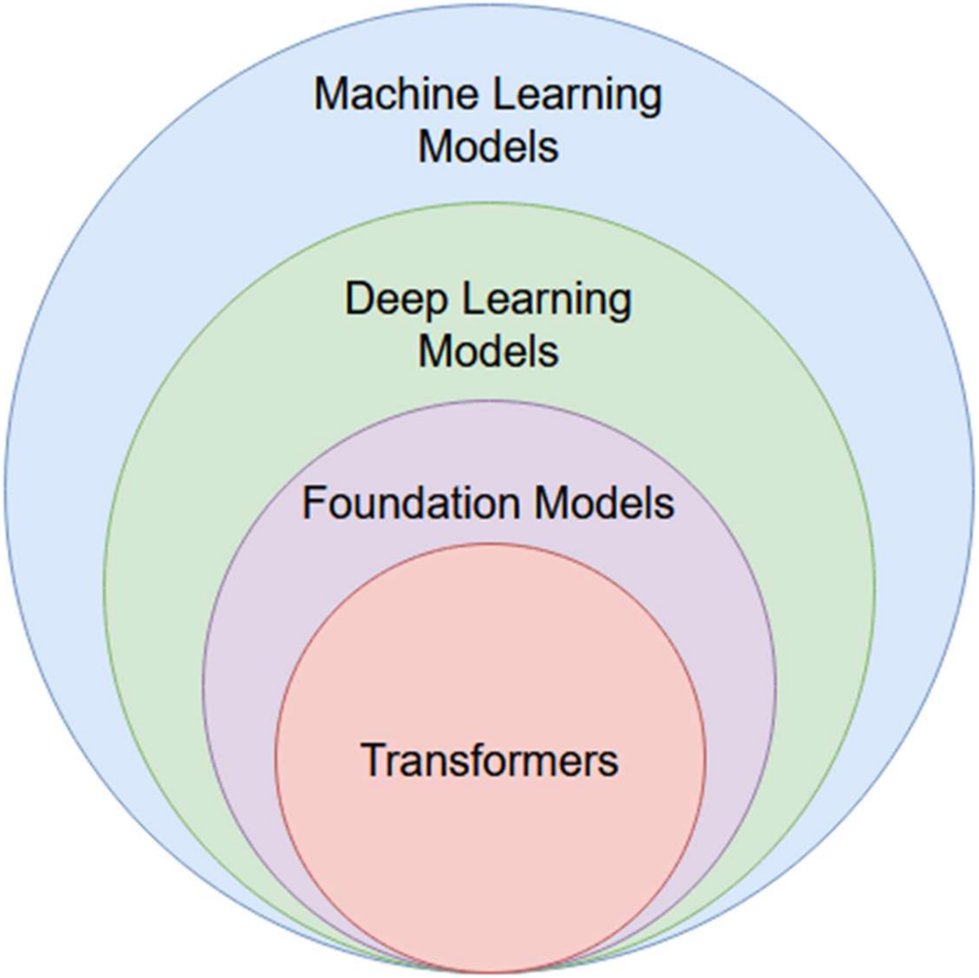
Attention Is All You Need

- 2017 paper introducing the concept of attention and transformer models
- Through seeing vast amounts of examples, model begins to understand relationships in text
- Model looks at entire sequence at once
- Each word "attends" to each other word
  - Stronger attention indicates stronger relationship

Remove word ambiguity



Transformers as Foundation Models

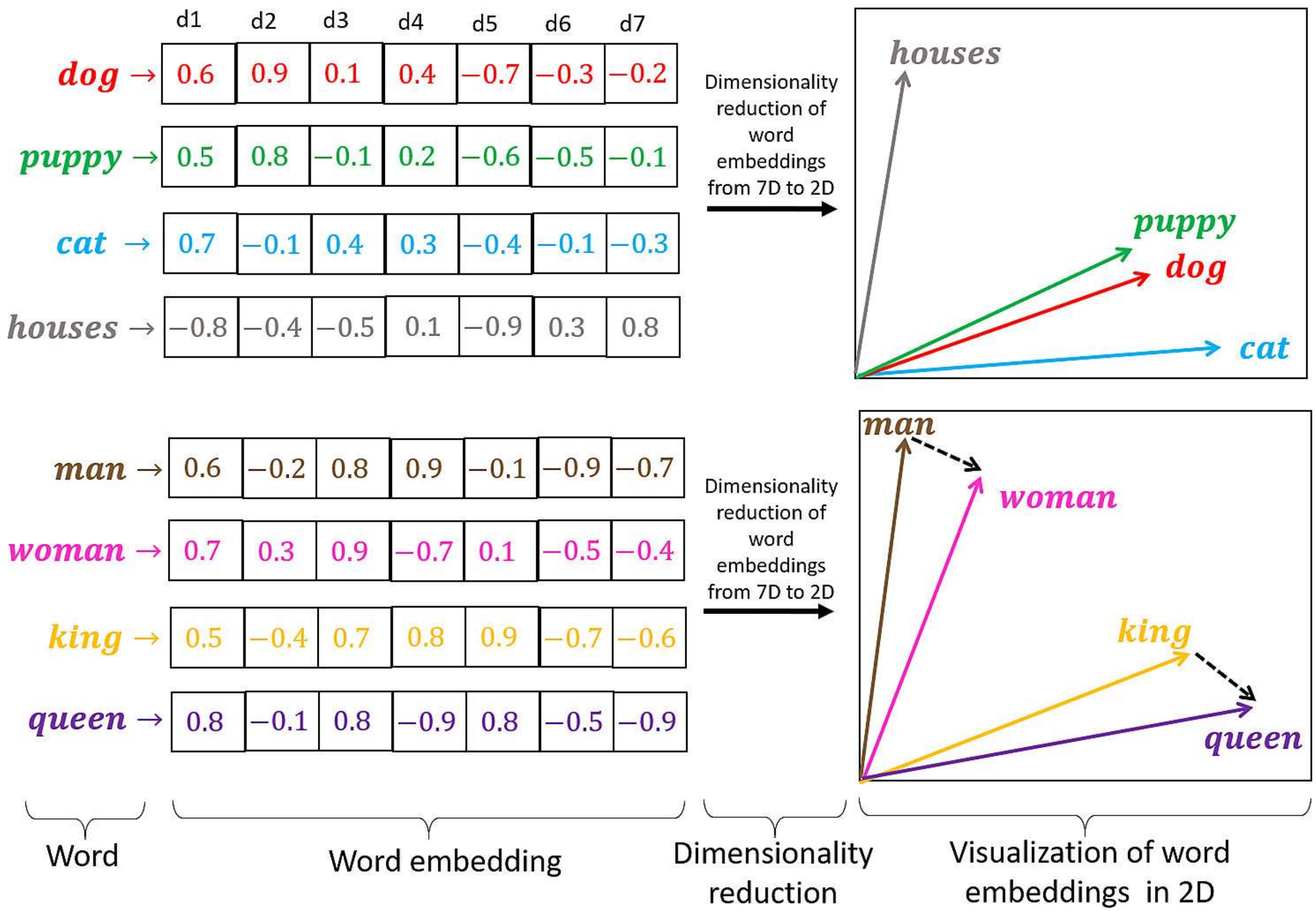


What is a Foundation Model?

- Originally, AI systems were narrow-purposed and task-specific
- Foundation models (FMs) are general-purpose
  - Trained once, applied to many tasks
- Most FMs are transformer-based
  - Self-supervised training
    - As it sees more data,
    - Learn broader features of the data rather than labels
    - Create embeddings of data

What are Embeddings?

- Embeddings are a high-dimensional vector (set of numbers) that represent the features of input data based on FM's knowledge



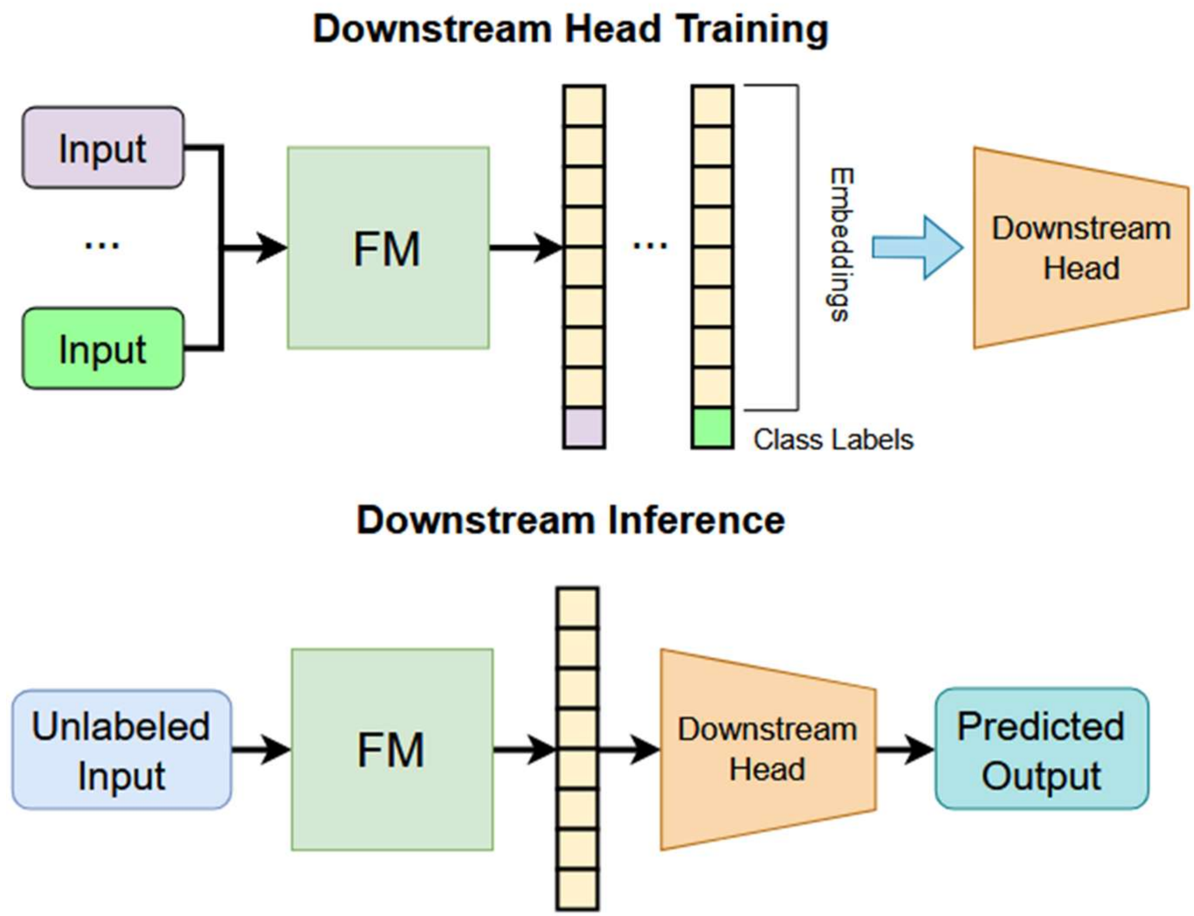
Applying Foundation Models

- Task-specific heads
  - Transfer Learning from the FM to heads for downstream tasks
  - Allows for model versatility

Embeddings used as features in traditional ML model architectures

Tasks include:

- Classification of images or text
- Image Segmentation
- Linear Regression
- Next token prediction



Training a Large Language Model

1. Collect and Preprocess Data
2. Tokenization
3. Pre-Train the Transformer
4. Fine-tune

Training a Large Language Model –
Collect and Preprocess Data

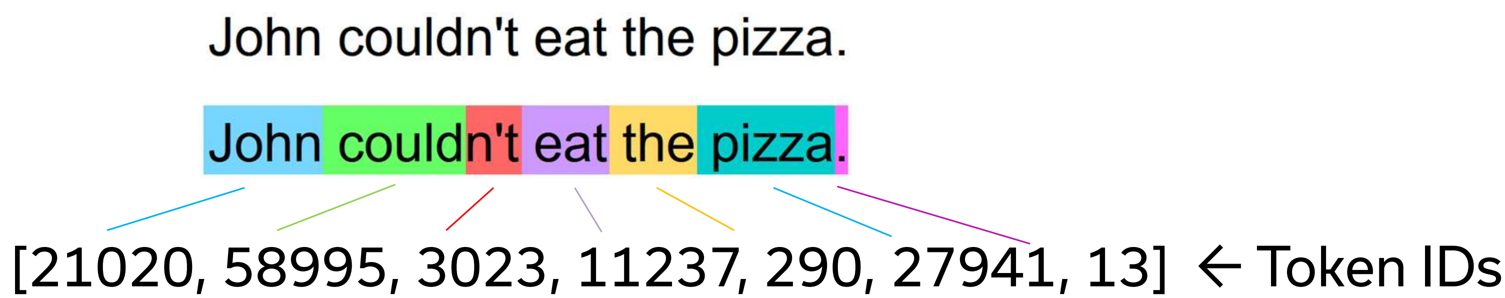
- Scrape sources from the web
- Existing datasets
  - HuggingFace
- Personalized data

Datasets

Training a Large Language Model – *Tokenization*

What is a token?

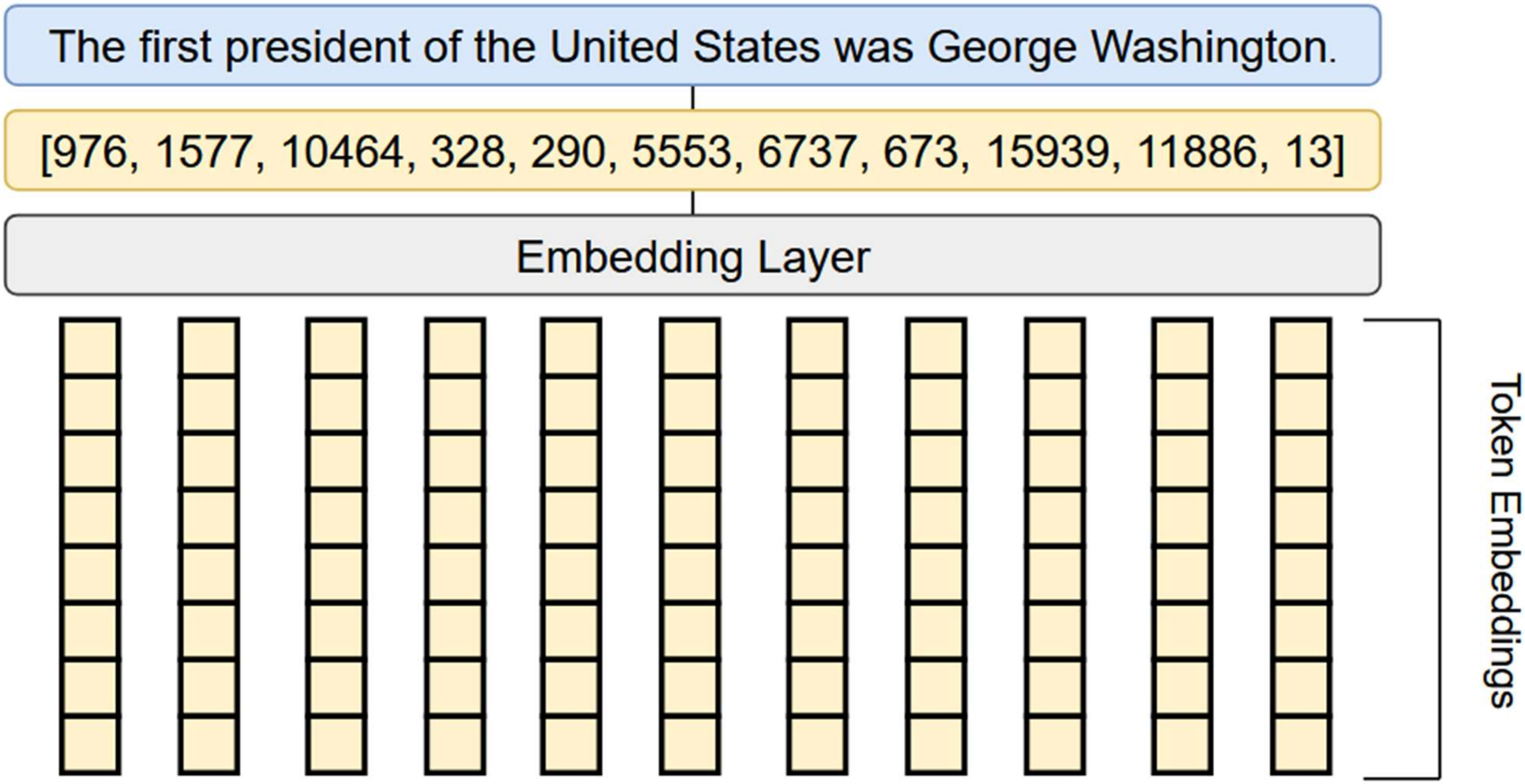
- Word/word-like unit
- Units of the input sequence for an LLM



- Common sequences within the training dataset
  - Often turns out to be words, but not always
  - Punctuation, contractions, etc.
- The set of all tokens is the model’s vocabulary
- Each token associated with an ID that maps to its embedding
- This ID is what the model sees

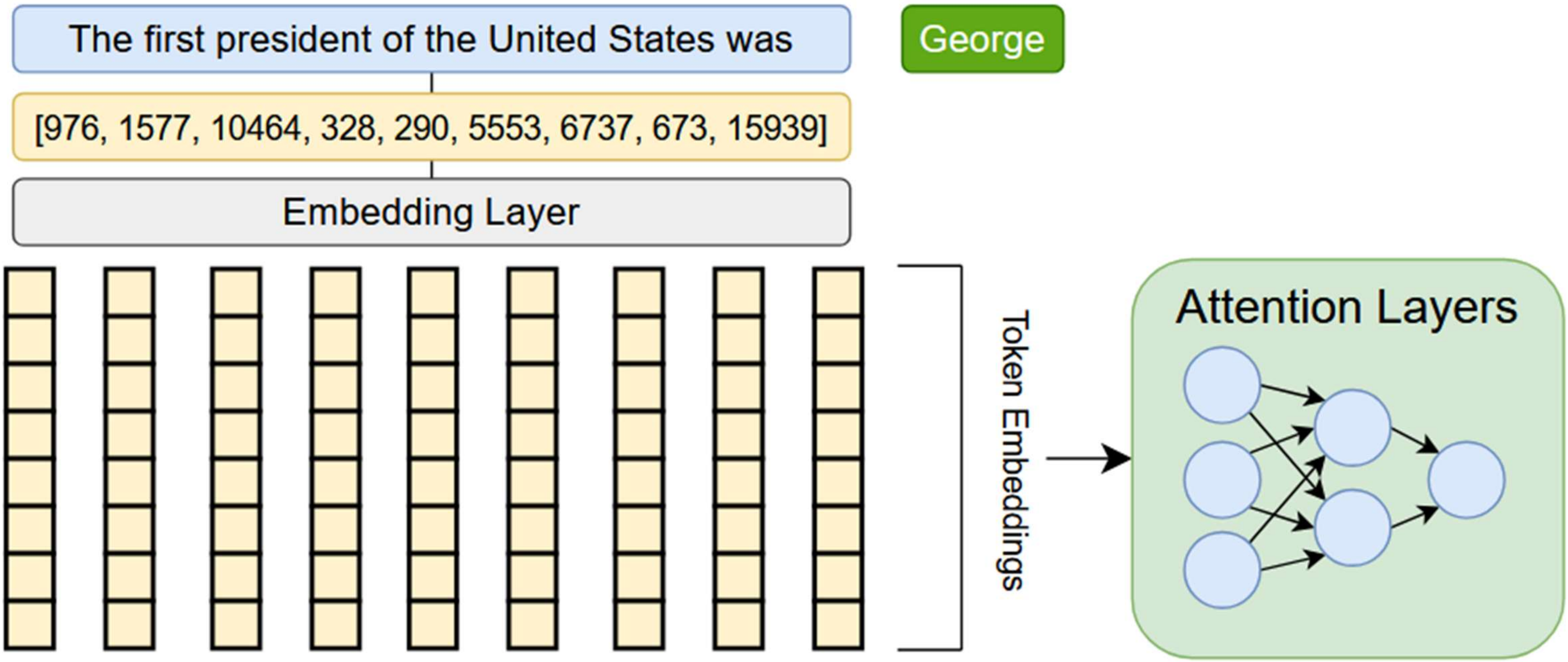
Training a Large Language Model –
Pre-Training the Transformer

- Sequence has existing embeddings for each token



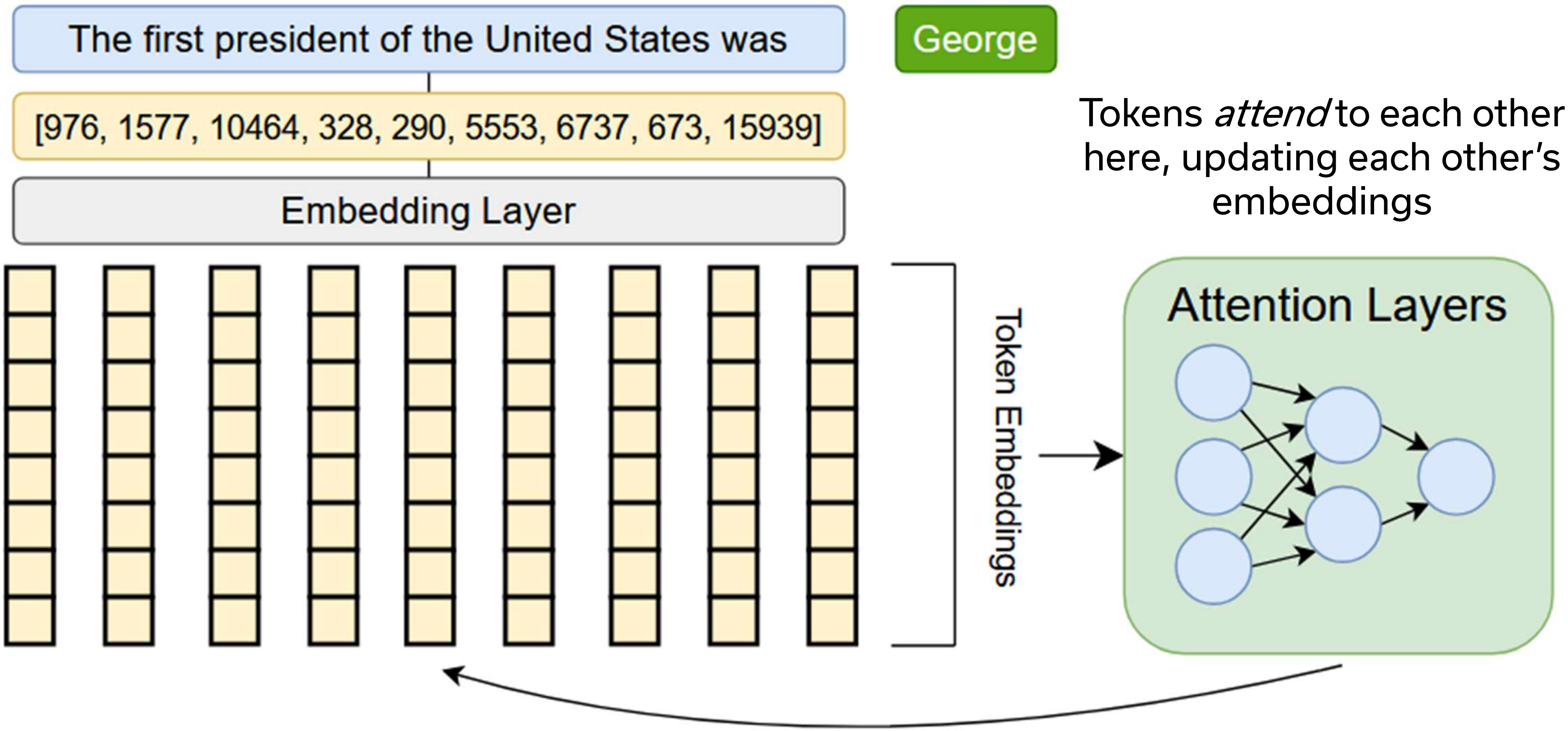
Training a Large Language Model –
Pre-Training the Transformer

- Provide an unfinished sequence
- Model must accurately predict next token



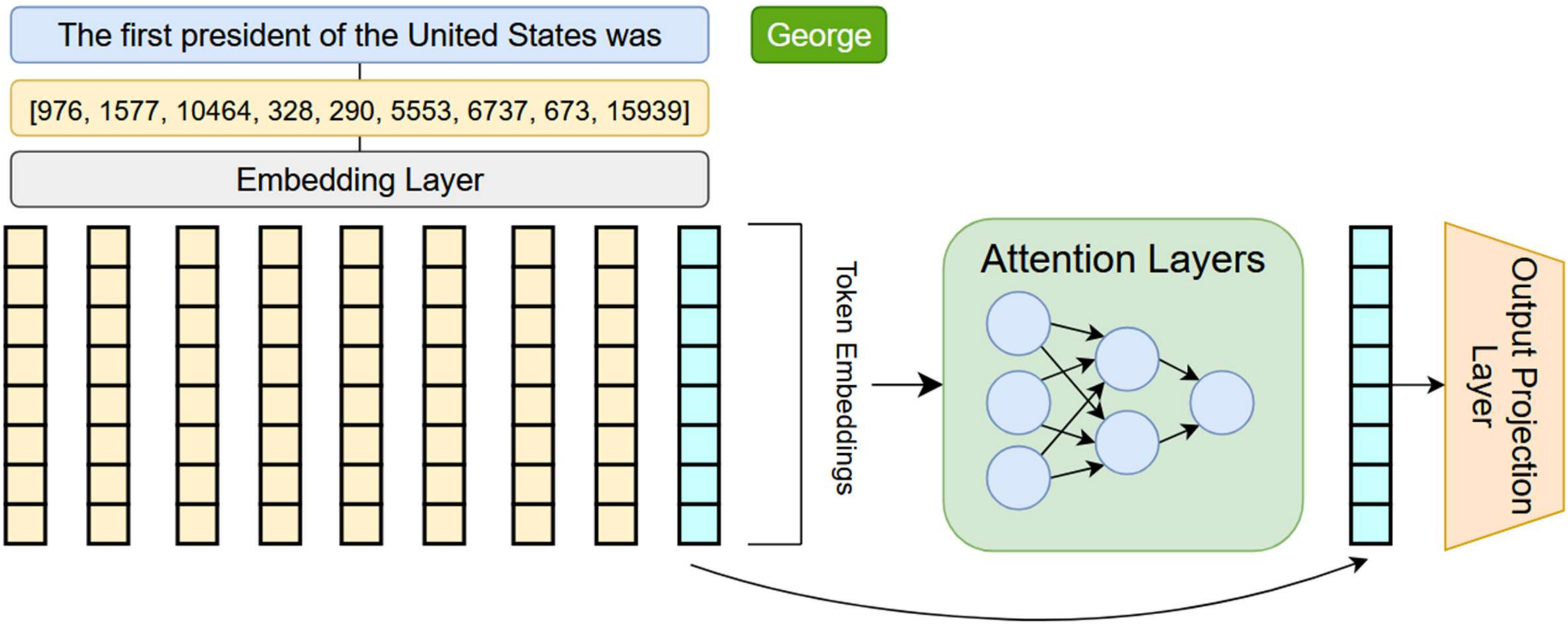
Training a Large Language Model –
Pre-Training the Transformer

- Provide an unfinished sequence
- Model must accurately predict next token



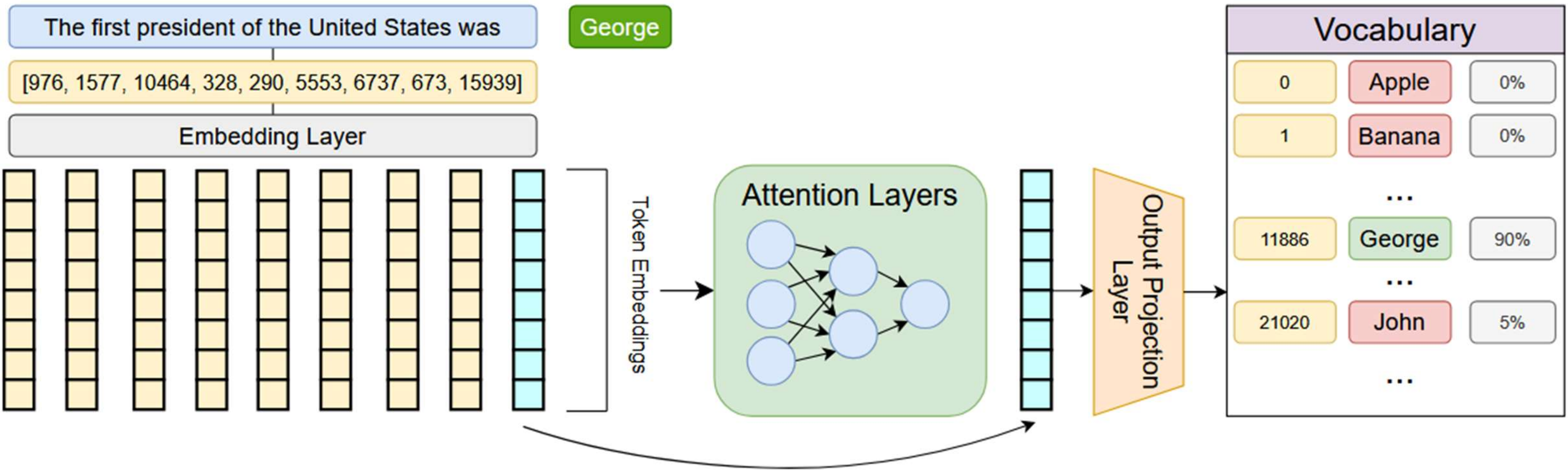
Training a Large Language Model – *Pre-Training the Transformer*

- Attended embedding of the last token passed to the output projection layer
  - Also called Language Modeling (LM) head



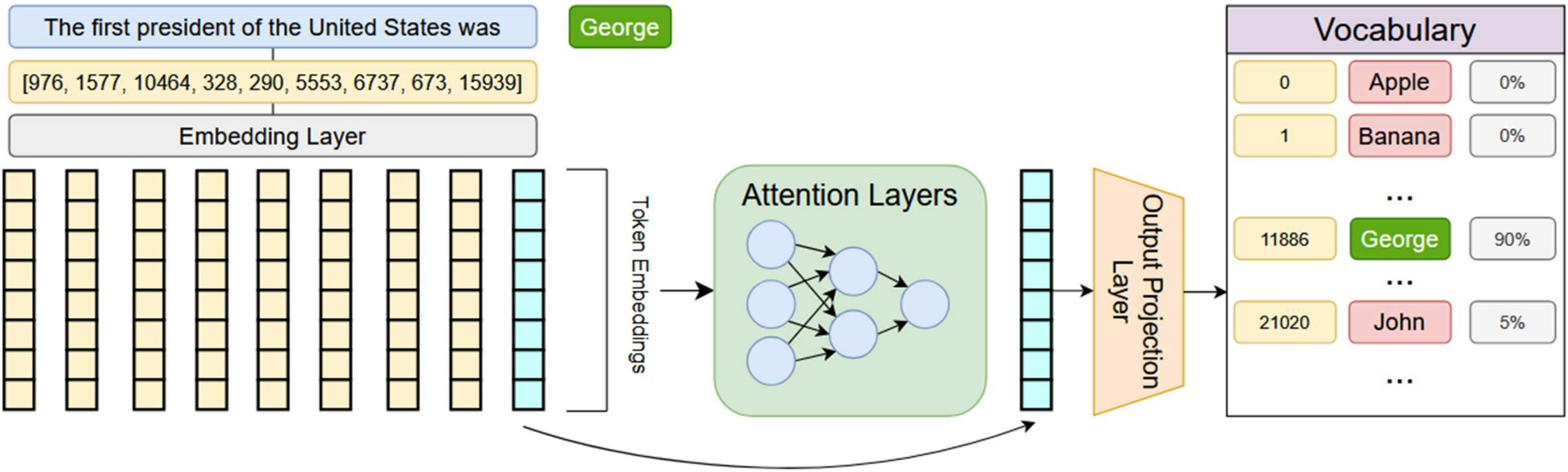
Training a Large Language Model –
Pre-Training the Transformer

- Creates a probability distribution for entire vocabulary
- Highest probability token is selected



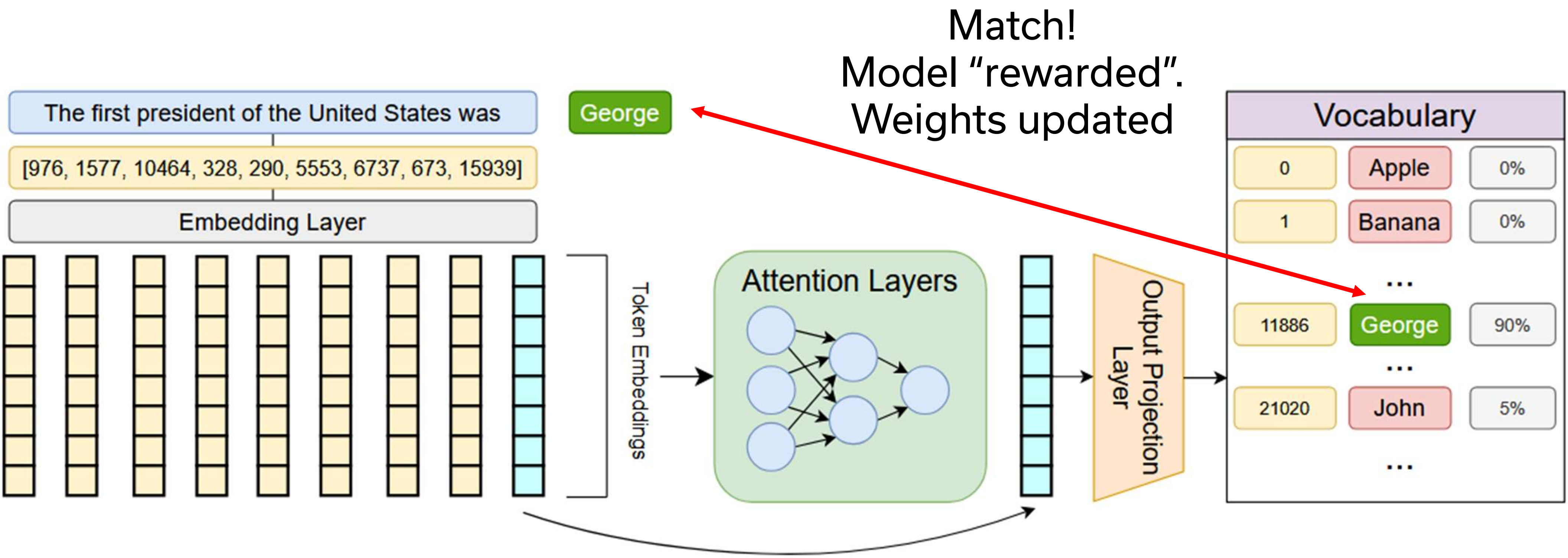
Training a Large Language Model –
Pre-Training the Transformer

- Creates a probability distribution for entire vocabulary
- Highest probability token is selected



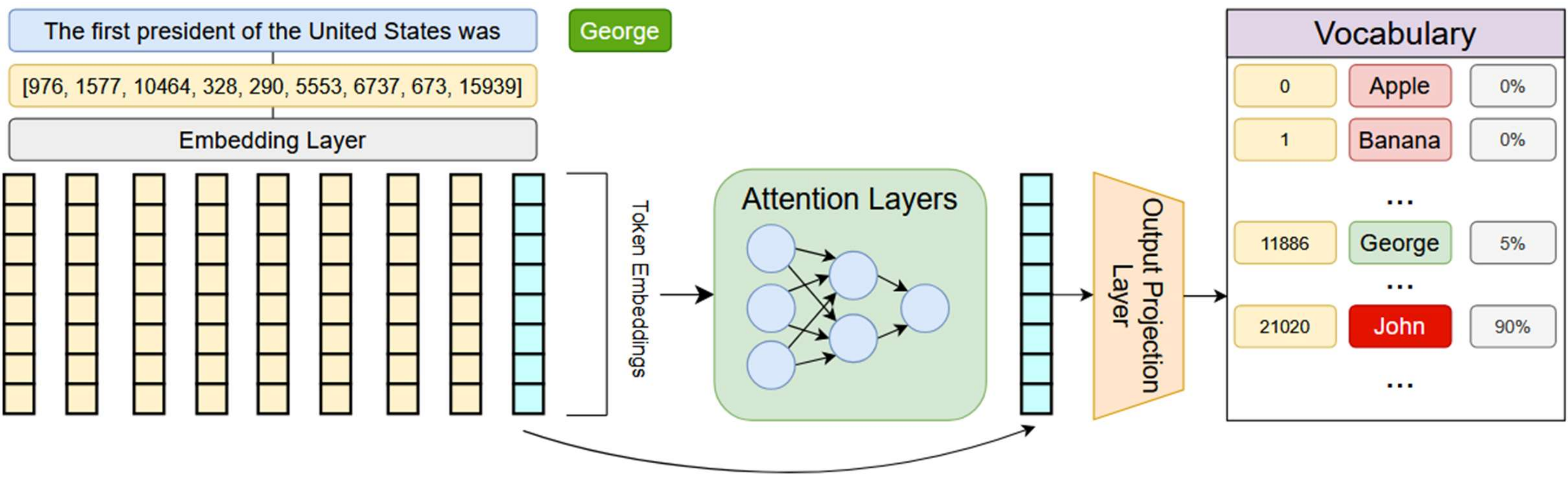
Training a Large Language Model –
Pre-Training the Transformer

- Creates a probability distribution for entire vocabulary
- Highest probability token is selected



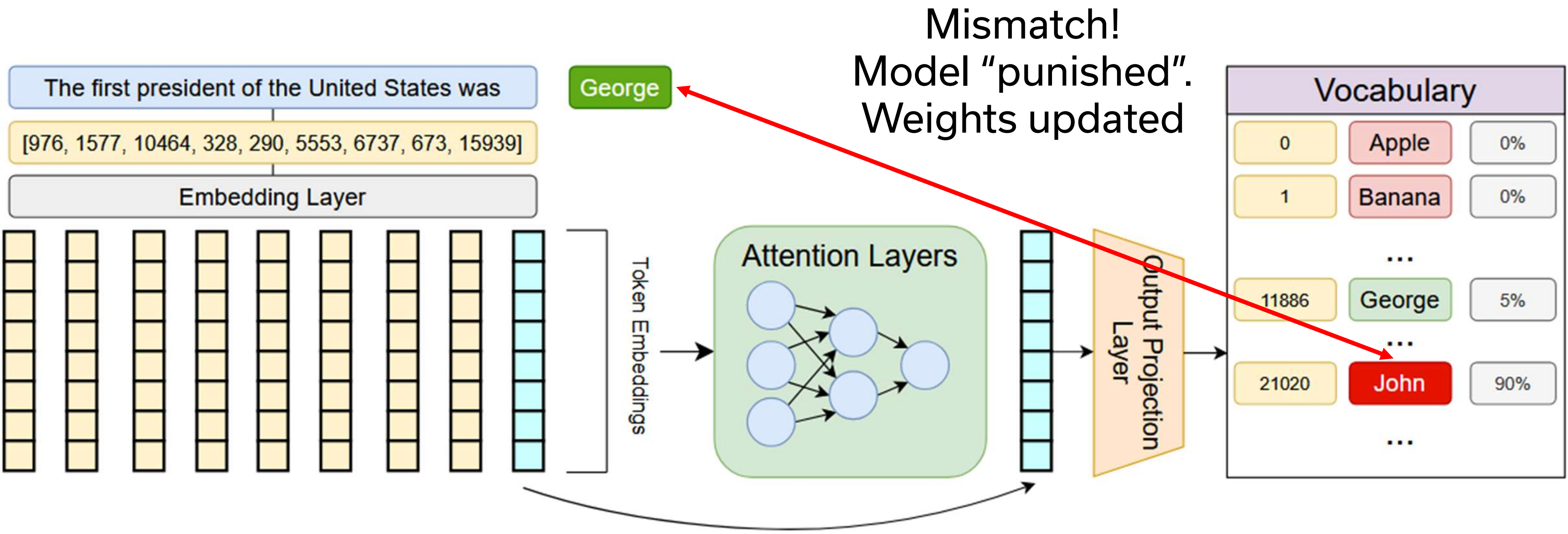
Training a Large Language Model –
Pre-Training the Transformer

- Creates a probability distribution for entire vocabulary
- Highest probability token is selected



Training a Large Language Model –
Pre-Training the Transformer

- Creates a probability distribution for entire vocabulary
- Highest probability token is selected

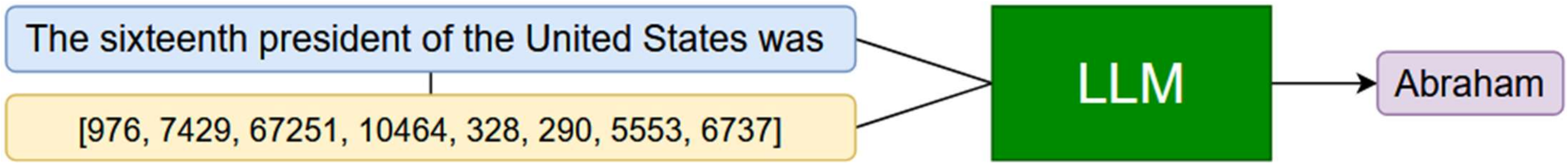


Training a Large Language Model –
What do we have now?

- For an input sequence, the model can now produce the next token until we tell it to stop!
- Generative
  - For any given input, mimic training data to complete sequence
- After enough training on accurate data, model gains “knowledge”

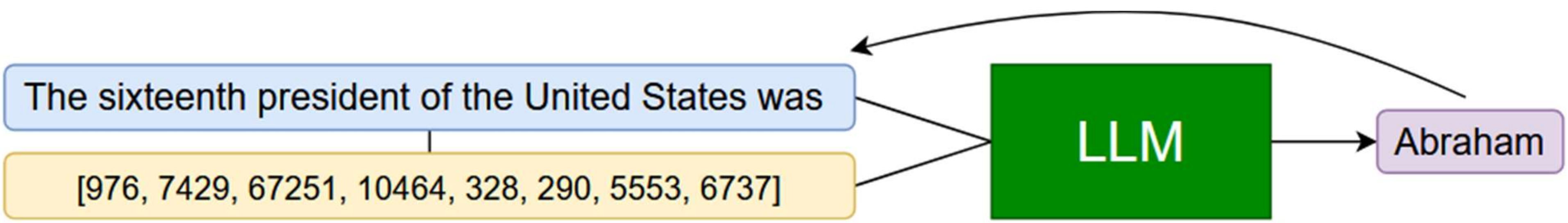
Training a Large Language Model – *What do we have now?*

- For an input sequence, the model can now produce the next token until we tell it to stop!
- Generative
  - For any given input, mimic training data to complete sequence
- After enough training on accurate data, model gains "knowledge"



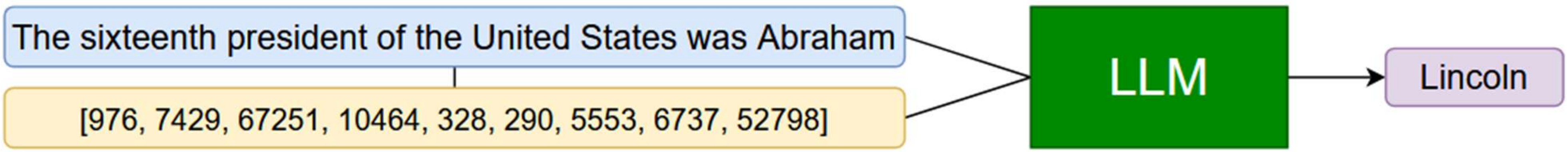
Training a Large Language Model –
What do we have now?

- For an input sequence, the model can now produce the next token until we tell it to stop!
- Generative
  - For any given input, mimic training data to complete sequence
- After enough training on accurate data, model gains "knowledge"



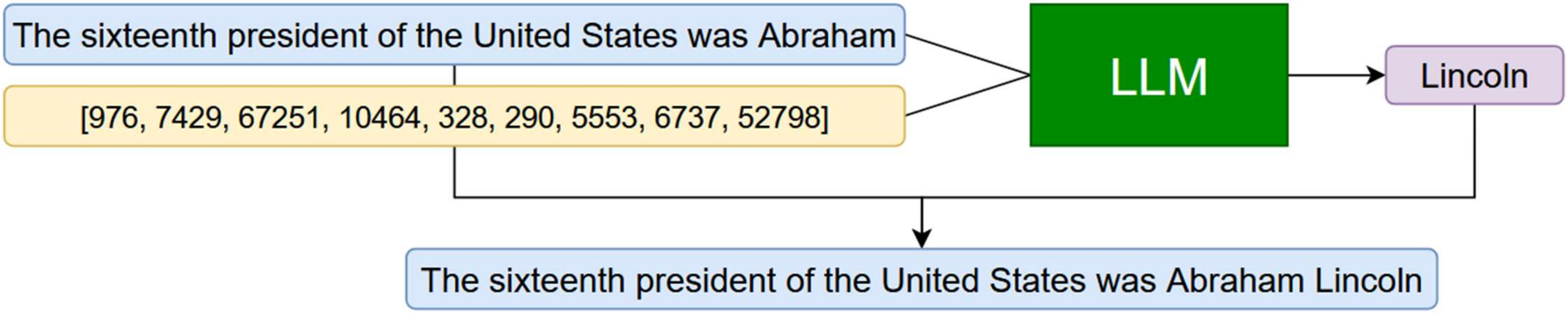
Training a Large Language Model –
What do we have now?

- For an input sequence, the model can now produce the next token until we tell it to stop!
- Generative
  - For any given input, mimic training data to complete sequence
- After enough training on accurate data, model gains "knowledge"



Training a Large Language Model –
What do we have now?

- For an input sequence, the model can now produce the next token until we tell it to stop!
- Generative
  - For any given input, mimic training data to complete sequence
- After enough training on accurate data, model gains “knowledge”



Training a Large Language Model –
What do we have now?

Problem:

- Only performs sequence “completions”
- No conversational capacity
- Can’t serve as a chatbot/assistant

Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

- Following pre-training
- Fine-tuning: further training on a smaller, specialized dataset
- Instruction Fine-tuning: Fine-tune on dataset with “instruction” format
  - Turns of user input followed by model responses

Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

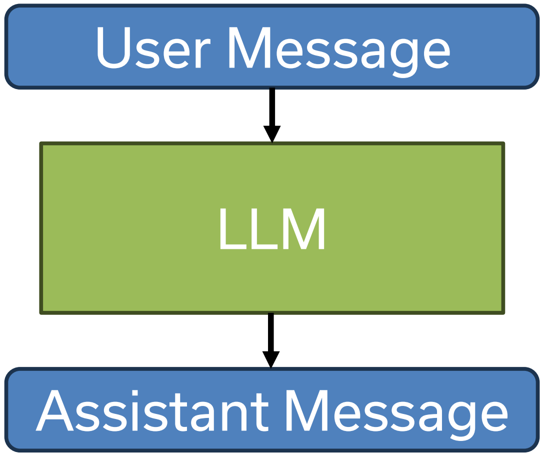
- Following pre-training
- Fine-tuning: further training on a smaller, specialized dataset
- Instruction Fine-tuning: Fine-tune on dataset with “instruction” format
  - Turns of user input followed by model responses

User Message

Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

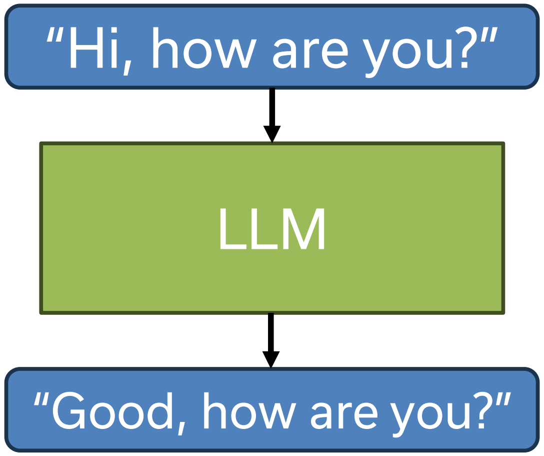
- Following pre-training
- Fine-tune on dataset with "instruction" format
  - Turns of user input followed by model responses
- Fine-tuning: further training on a smaller, specialized dataset



Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

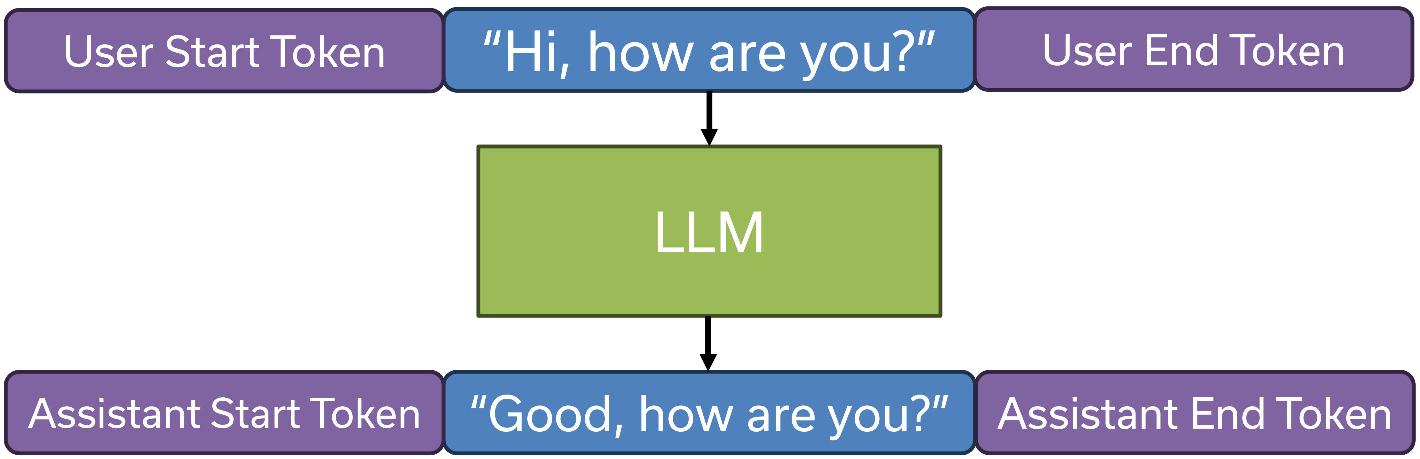
- Following pre-training
- Fine-tune on dataset with "instruction" format
    - Turns of user input followed by model responses
- Fine-tuning: further training on a smaller, specialized dataset



Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

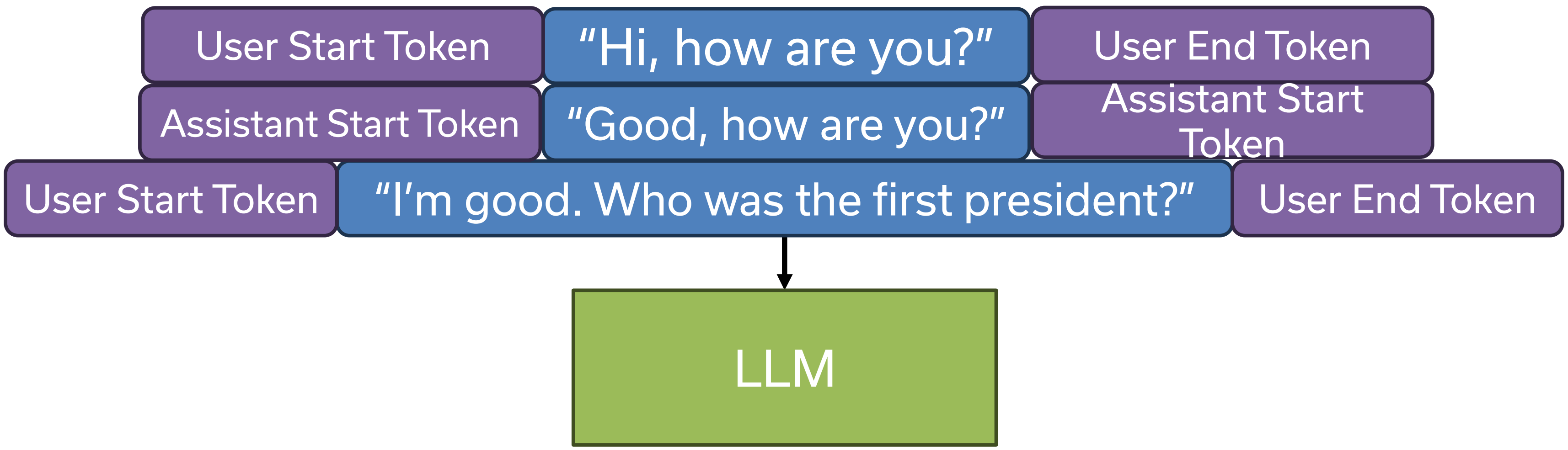
- Following pre-training
- Fine-tune on dataset with "instruction" format
  - Turns of user input followed by model responses
- Fine-tuning: further training on a smaller, specialized dataset



Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

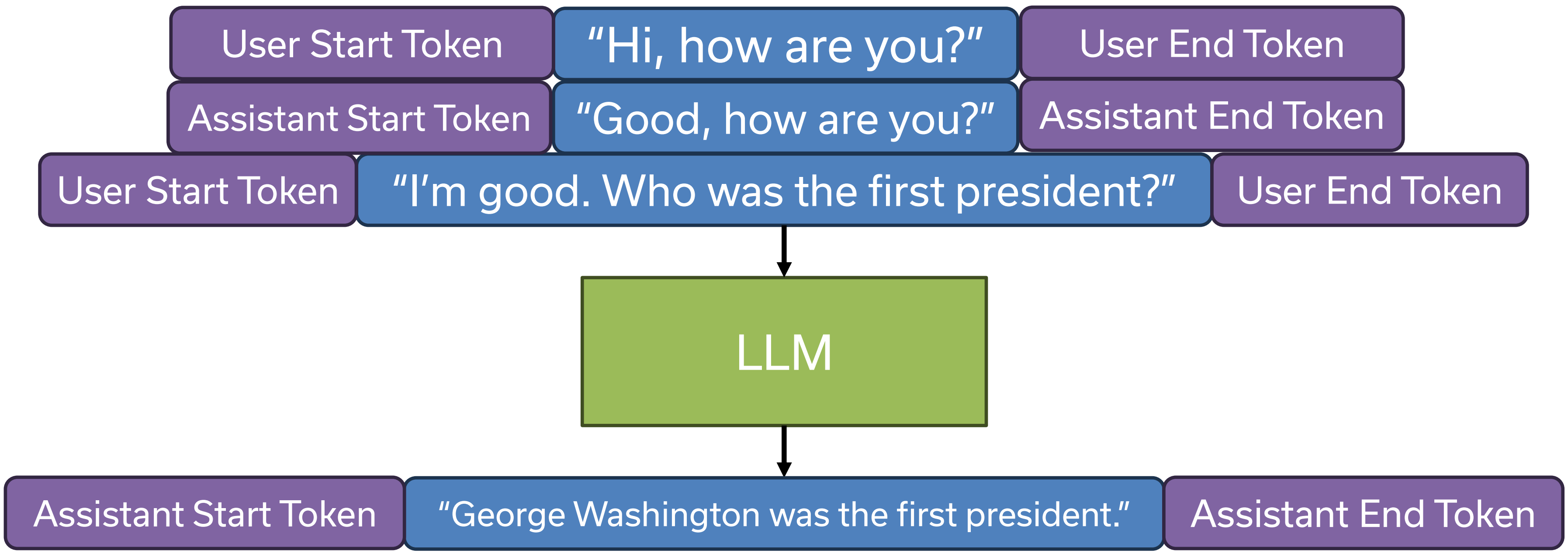
- Following pre-training
- Fine-tune on dataset with "instruction" format
  - Turns of user input followed by model responses



Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

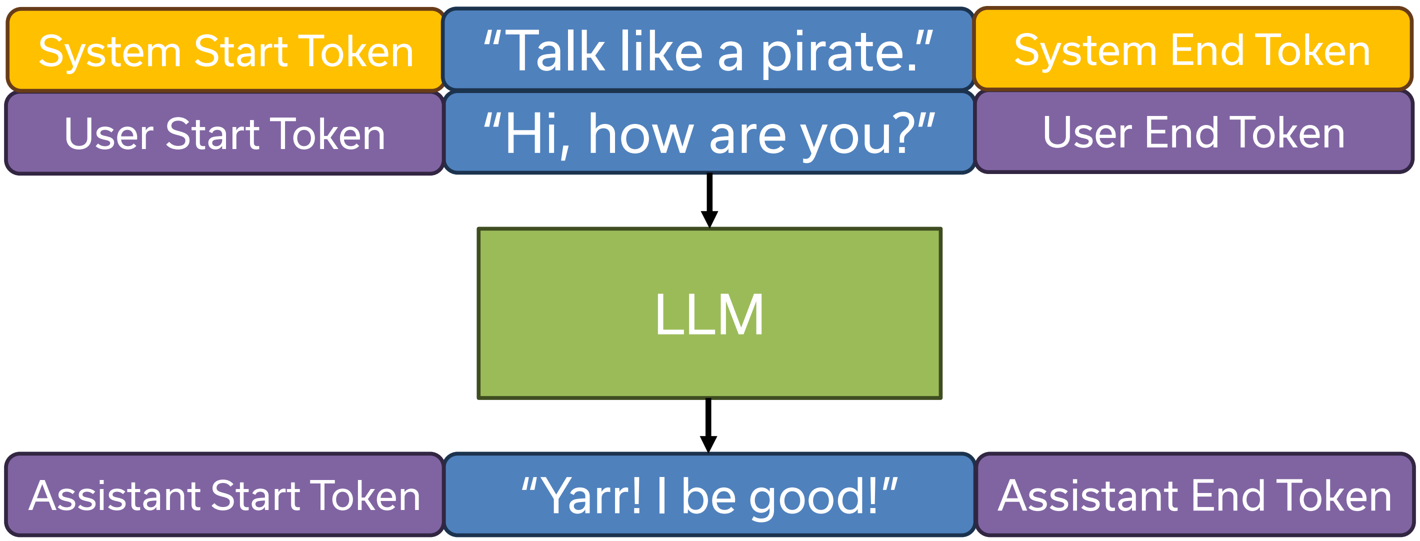
- Following pre-training
- Fine-tune on dataset with "instruction" format
  - Turns of user input followed by model responses



Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

- Following pre-training
- Fine-tune on dataset with “instruction” format
  - Turns of user input followed by model responses



Training a Large Language Model – *Fine-Tuning*

Instruction Fine-Tuning

- Following pre-training
- Fine-tune on dataset with “instruction” format
  - Turns of user input followed by model responses

- Create a dataset with messages formatted like this
- LLM response set as answers to questions, responses to instructions

Training a Large Language Model – *Fine-Tuning*

stackoverflow

Delete an element from a dictionary

Asked 14 years, 8 months ago Modified 9 months ago Viewed 2.9m times

2245

1. How do I delete an item from a dictionary in Python?
2. **Without** modifying the original dictionary, how do I obtain **another** dictionary with the item removed?

See also How can I remove a key from a Python dictionary? for the specific issue of removing an item (by key) that may not already be present.

python dictionary del

← Question

Training a Large Language Model – *Fine-Tuning*

stackoverflow

Delete an element from a dictionary

Asked 14 years, 8 months ago Modified 9 months ago Viewed 2.9m times

2245

1. How do I delete an item from a dictionary in Python?
2. **Without** modifying the original dictionary, how do I obtain **another** dictionary with the item removed?

← Question

See also How can I remove a key from a Python dictionary? for the specific issue of removing an item (by key) that may not already be present.

python dictionary del

Response →

2571

The `del` statement removes an element:

```
del d[key]
```

Note that this mutates the existing dictionary, so the contents of the dictionary changes for anybody else who has a reference to the same instance. To return a *new* dictionary, make a copy of the dictionary:

```
def removekey(d, key):
    r = dict(d)
    del r[key]
    return r
```

The `dict()` constructor makes a *shallow copy*. To make a deep copy, see the `copy` module.

Training a Large Language Model – *Fine-Tuning*

Input Sequence

User Start Token

1. How do I delete an item from a dictionary in Python?
2. Without modifying the original dictionary, how do I obtain another dictionary with the item removed?

User End Token

Expected Output Sequence

Assistant Start Token

The del statement removes an element:
del d[key]

Note that this mutates the existing dictionary, so the contents of the dictionary changes for anybody else who has a reference to the same instance. To return a new dictionary, make a copy of the dictionary:

```
def removekey(d, key):
    r = dict(d)
    del r[key]
    return r
```

The dict() constructor makes a shallow copy. To make a deep copy, see the copy module.

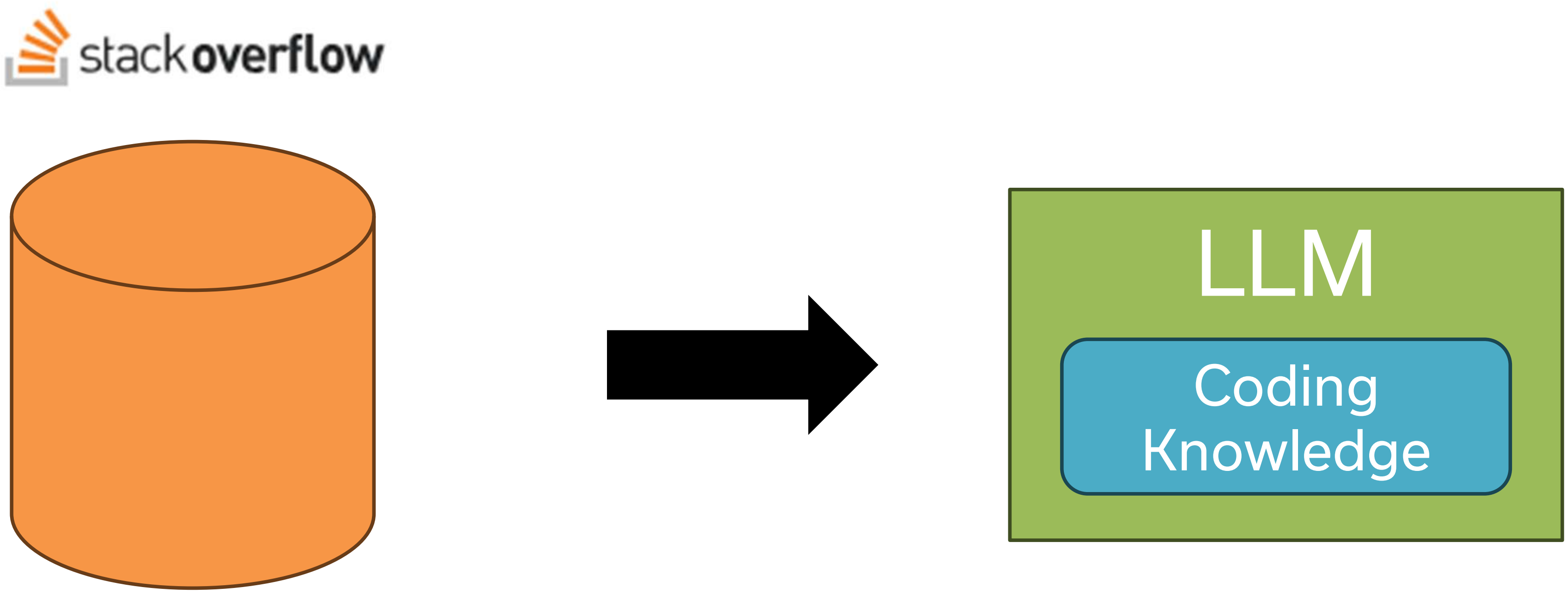
Assistant End Token

Training a Large Language Model

- After training at scale, you get modern LLMs

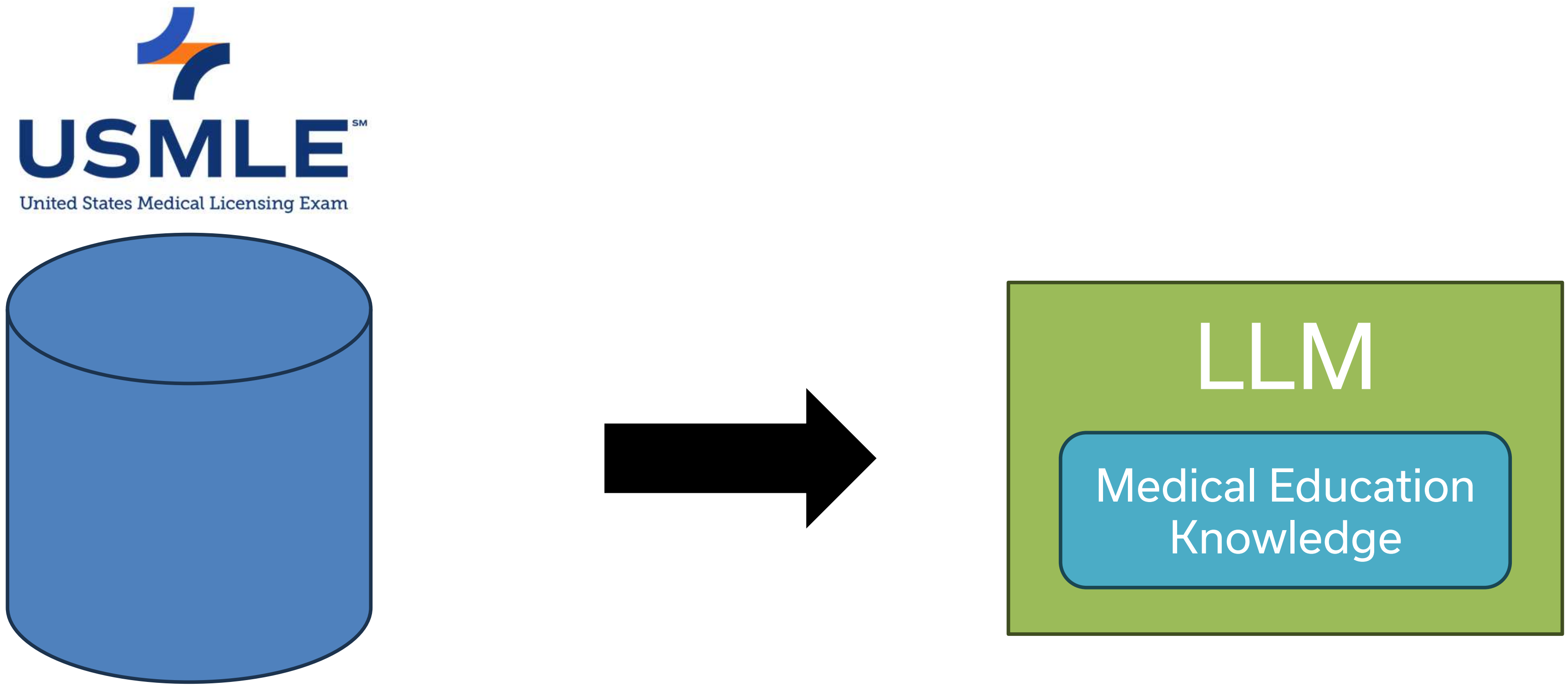
Training a Large Language Model

- After training at scale, you get modern LLMs



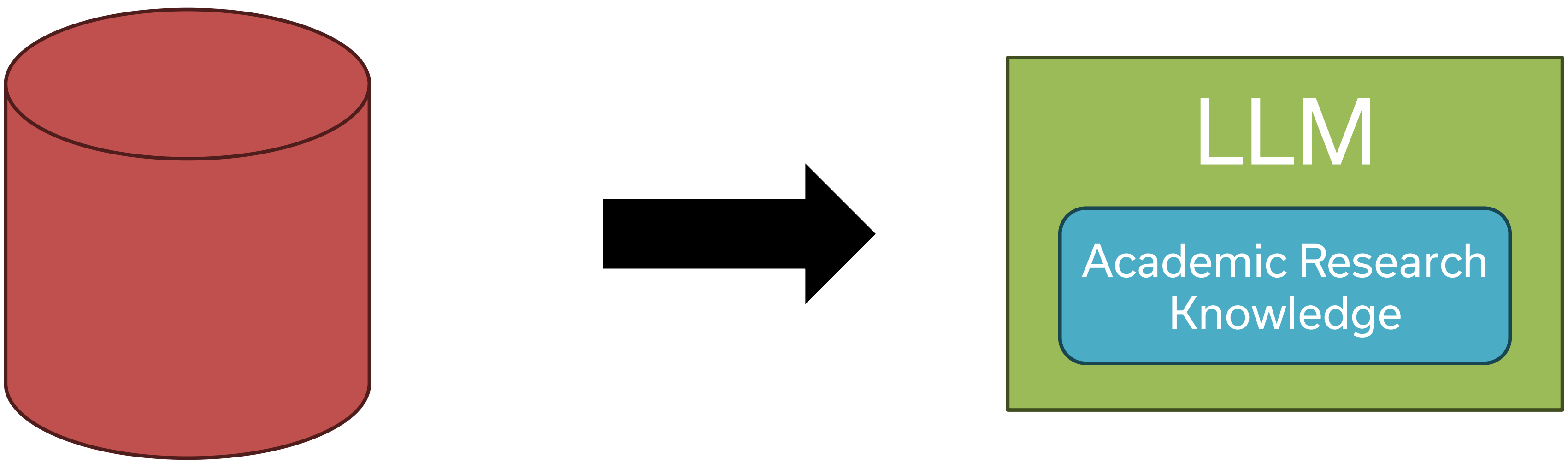
Training a Large Language Model

- After training at scale, you get modern LLMs



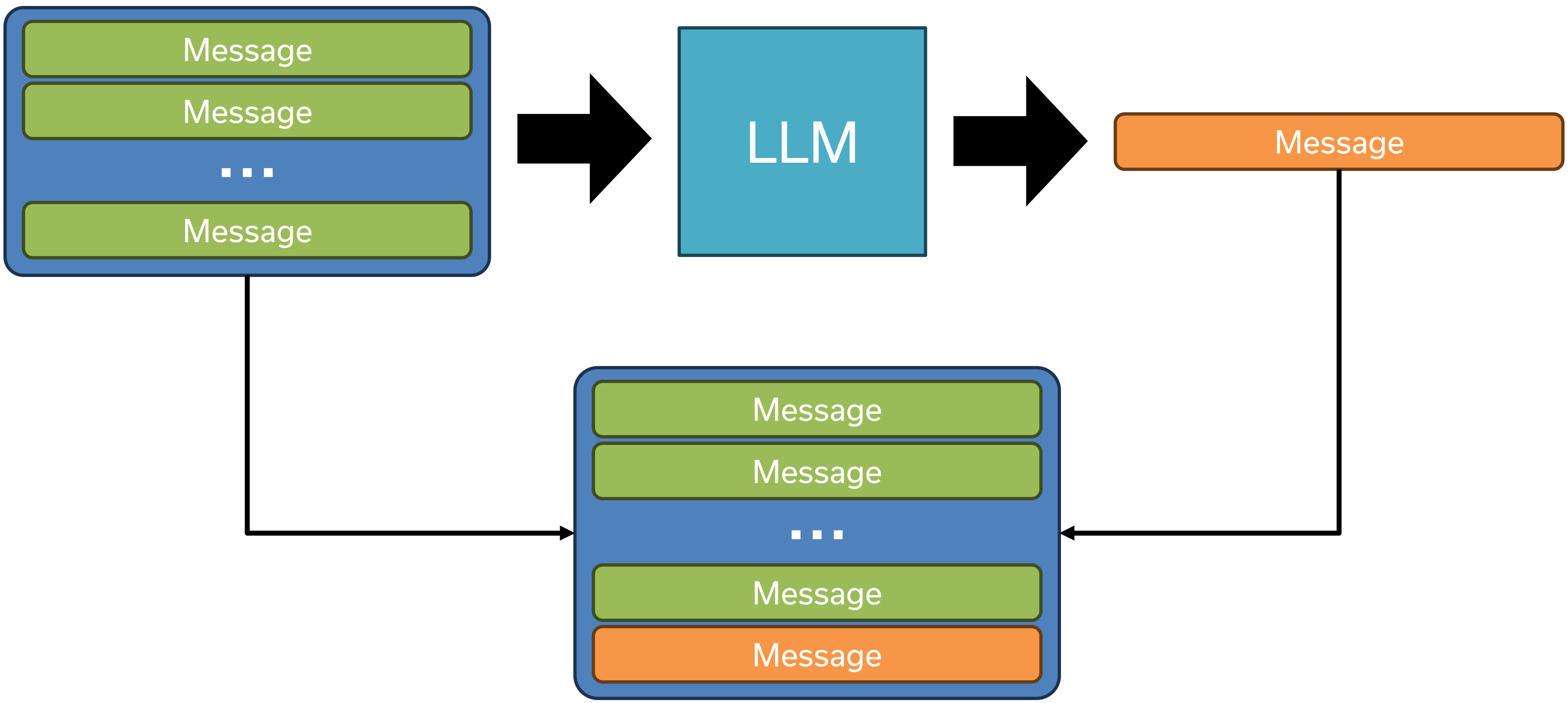
Training a Large Language Model

- After training at scale, you get modern LLMs

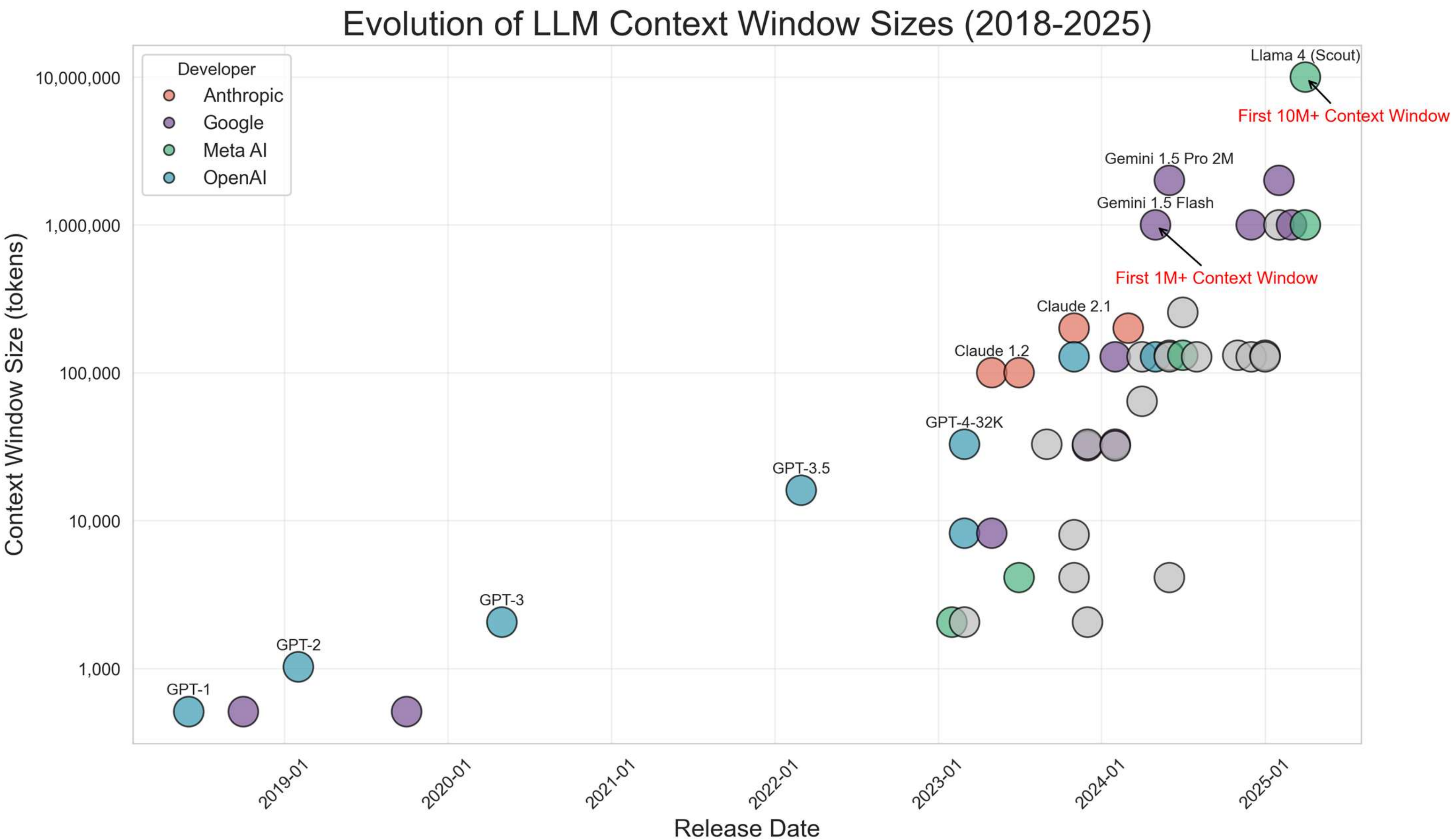


Context Length

- The maximum number of tokens a model can process at once
  - Total input and output tokens



Context Length



NAIRR Pilot · Kentucky Community & Technical College System · CPE Higher Education Matters · University of Kentucky

Training a Large Language Model

Understanding these principles makes LLMs less of a black box

What resources are required for this?

Training a Large Language Model

Requirements:

- Data
- Computational Power
- Time

Size	Use Case	Dataset Size	Computation	Time
Small	Basic chat, classification	Modest	Single GPU	Minutes-Hours
Medium	Reasoning, summarization	Large	Multiple GPUs	Hours-Days
Large	Translation, coding, agent	Massive	GPU cluster	Days
Frontier	General purpose AI	More massive	Data center	Months

Training a Large Language Model

- More parameters (weights) → Higher memory requirements
- More data → More training steps to learn
- More steps → More compute time and energy

Estimated time and cost to train ChatGPT 3.5

	1 NVIDIA V100 GPU	1024 NVIDIA A100 GPUs
Available Memory	32 GB	81,920 GB
Time	355 years	~34 days
Cost	$4.6-5 million	$2-4 million

Training a Large Language Model

Use and fine-tune existing LLMs rather than train from scratch

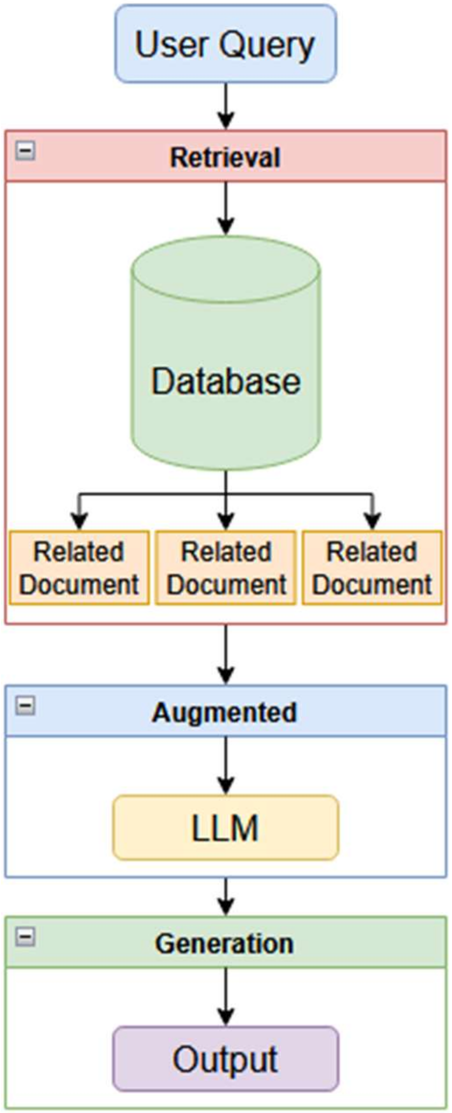
Lean on efforts from companies with resources to train these models

Training a Large Language Model

How do you customize models for your specific needs?

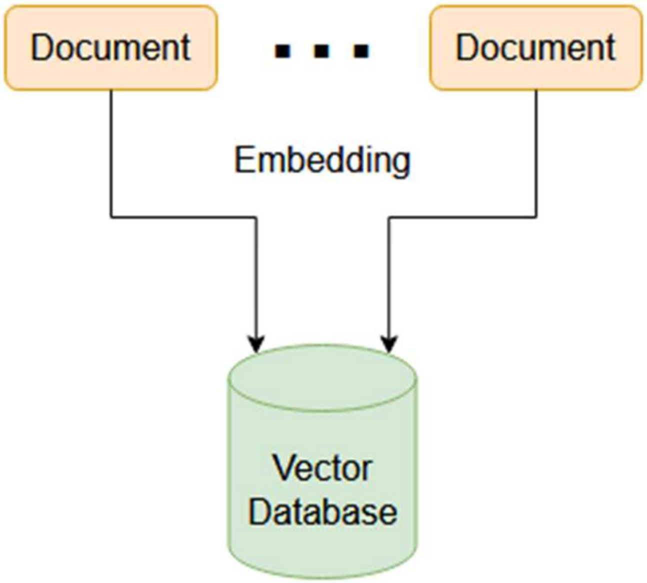
Retrieval-Augmented Generation (RAG)

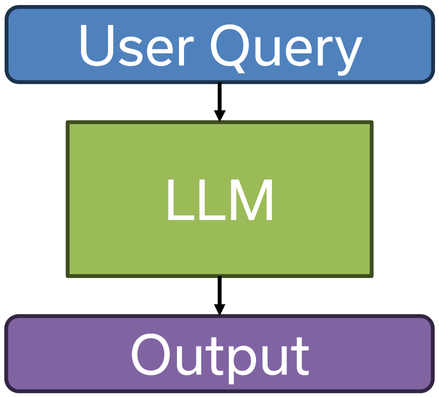
- LLMs often hallucinate
  - Provide inaccurate responses with high confidence
- RAG: Supplying specific information to LLM context
- Great for large document/data stores
  - All data would not fit within context window
  - Intelligently retrieve relevant data

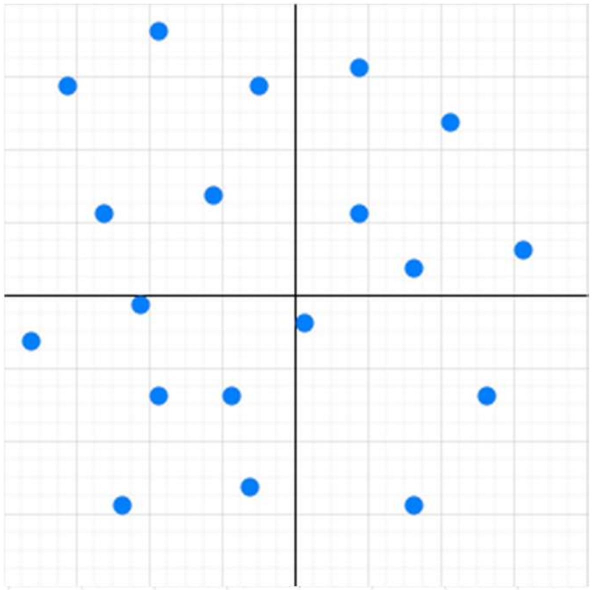


Vector Databases and Embedding Search

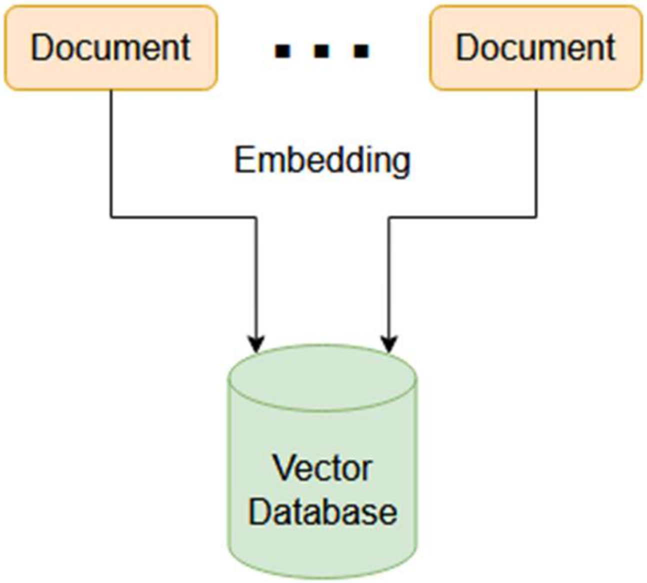
- Embed data with transformer model
- Store in embedding-data pair in vector database
- Similarity Searching
  - More like documents embed closer together



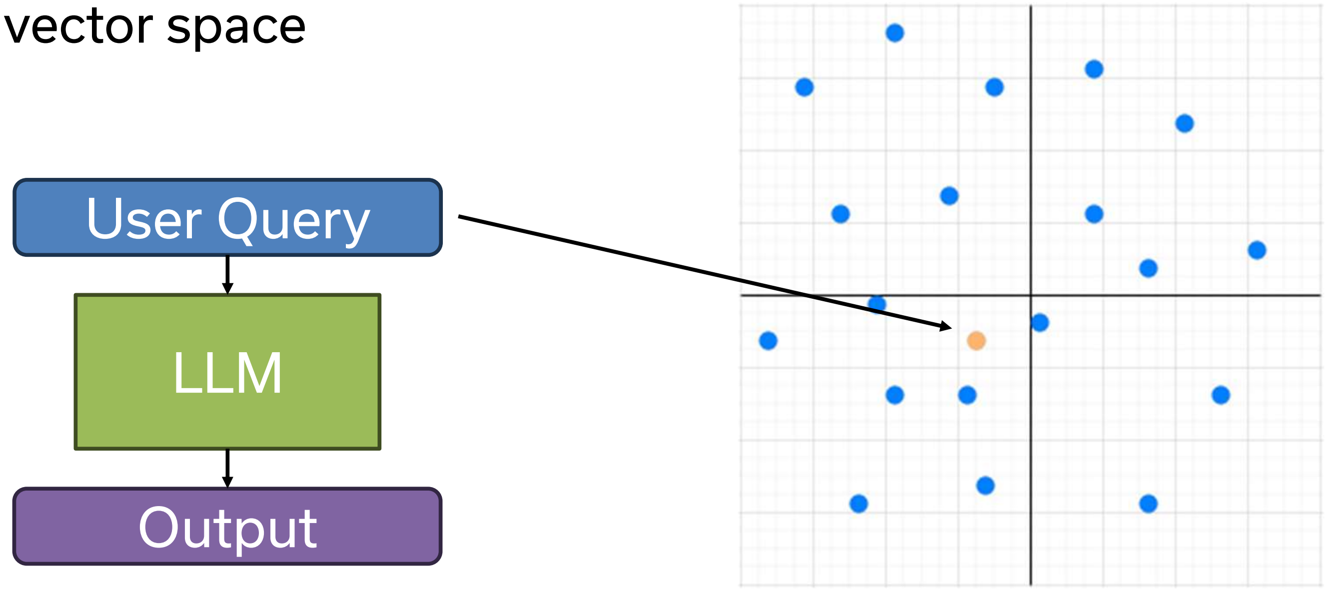




Vector Databases and Embedding Search

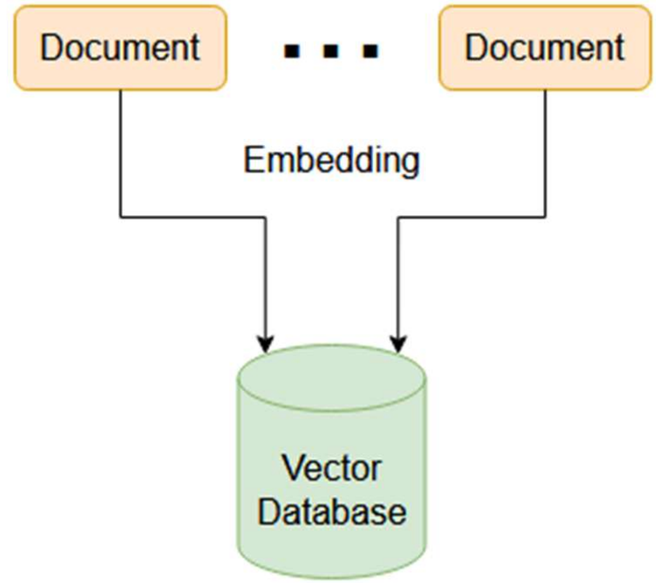


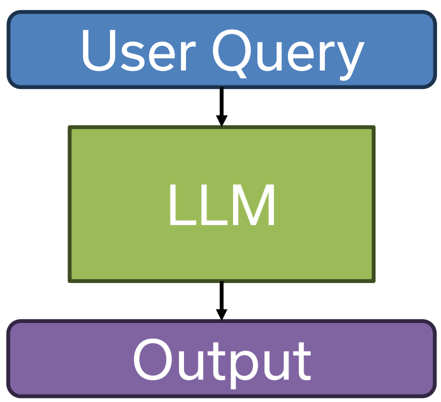
- Embed data with transformer model
- Store in embedding-data pair in vector database
- Similarity Searching
  - More like documents embed closer together
  - Convert query to strong search term
  - Embed search term in vector space

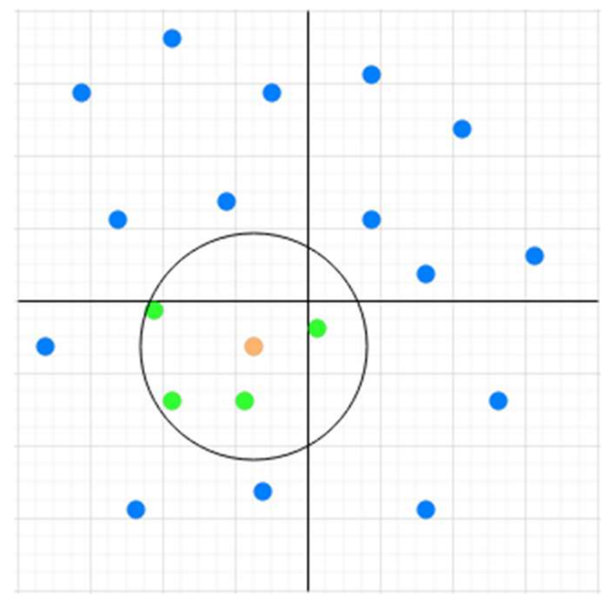


Vector Databases and Embedding Search

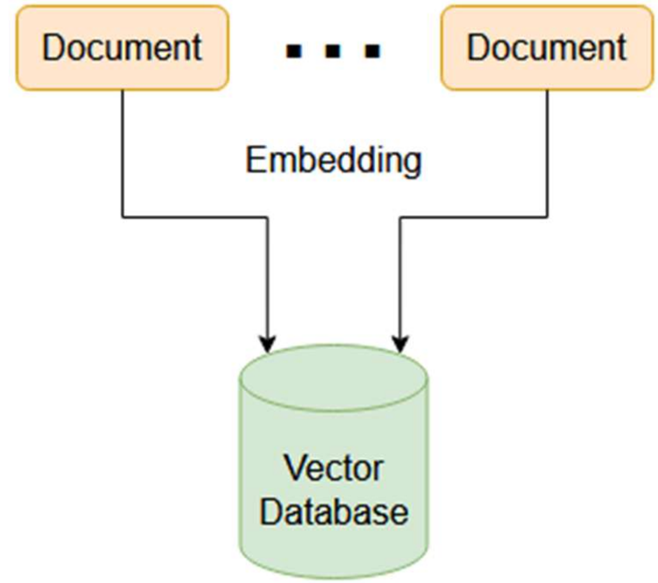
- Embed data with transformer model
- Store in embedding-data pair in vector database
- Similarity Searching
  - More like documents embed closer together
  - Convert query to strong search term
  - Embed search term in vector space
  - Find most similar docs



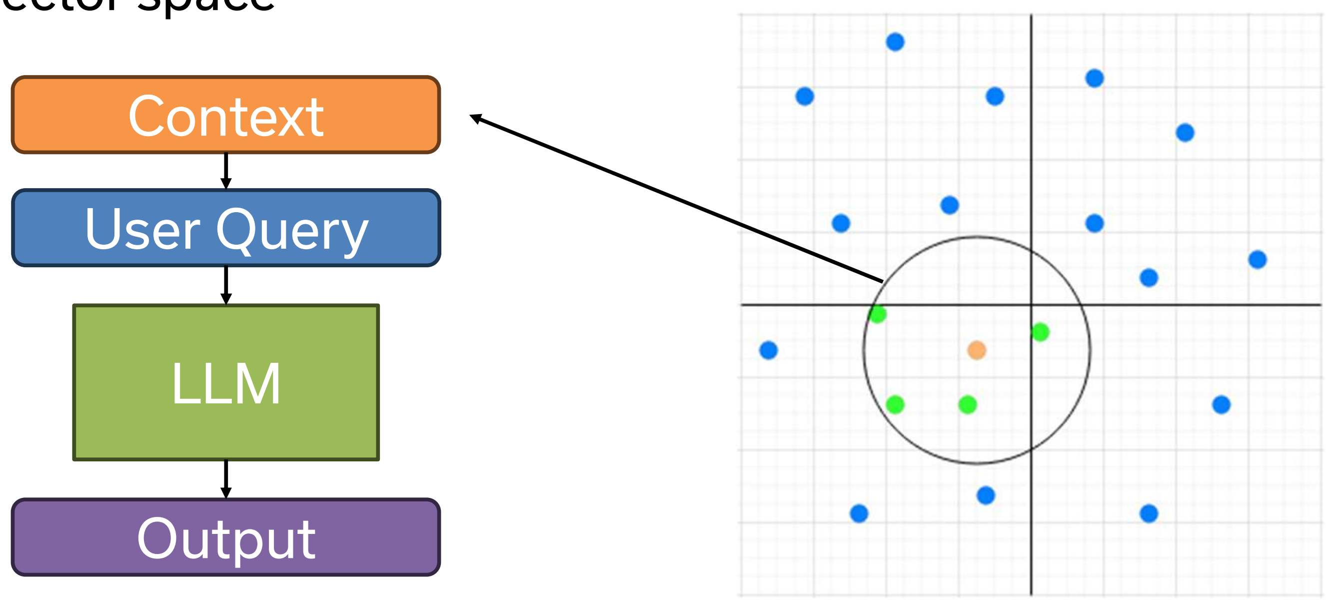




Vector Databases and Embedding Search



- Embed data with transformer model
- Store in embedding-data pair in vector database
- Similarity Searching
  - More like documents embed closer together
  - Convert query to strong search term
  - Embed search term in vector space
  - Find most similar docs
  - Add to context



Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities

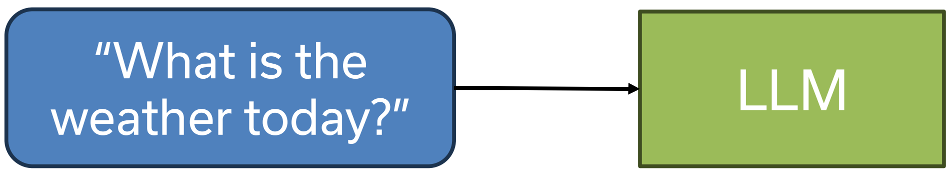
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities

This is a hallucination

Tool Calling and Agents

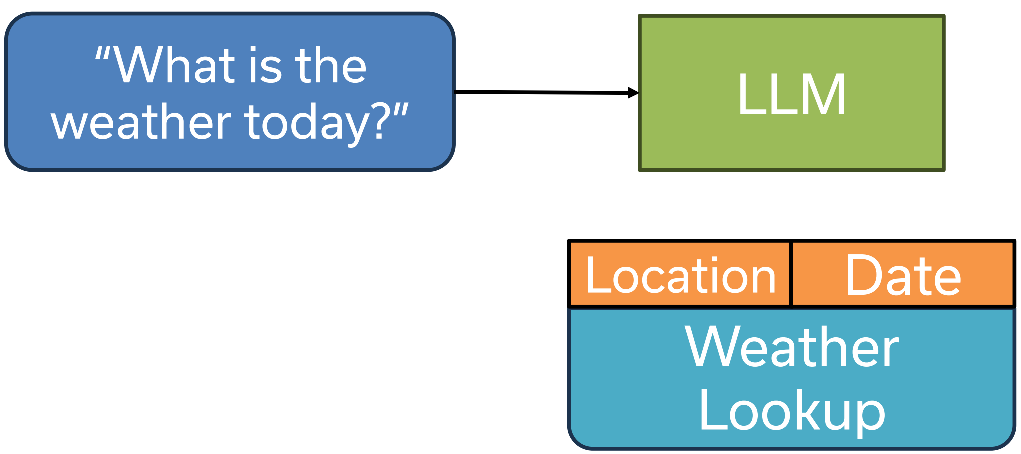
- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



Need:
- Location
- Date
- Current weather for that date and location

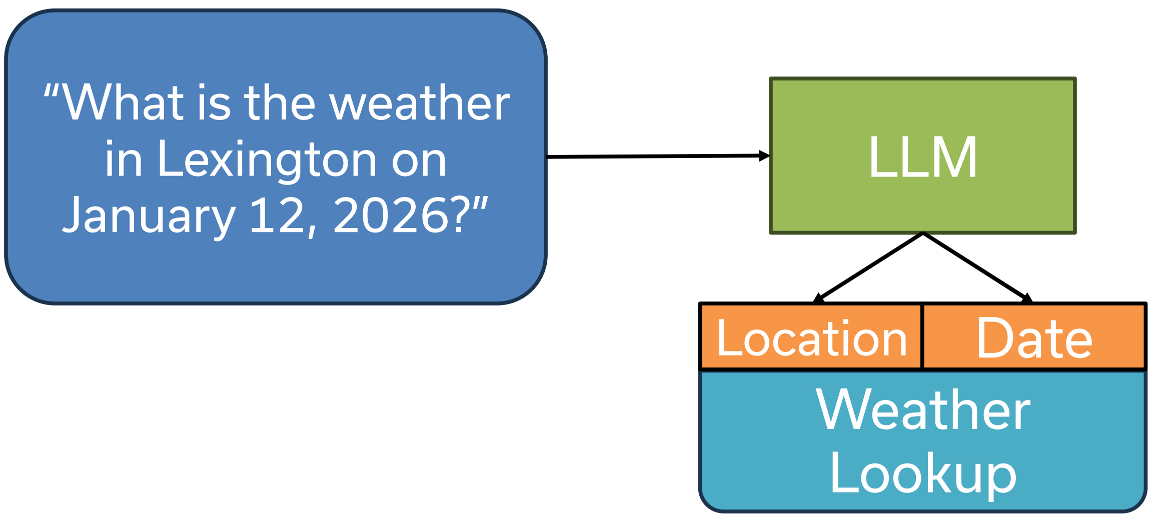
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



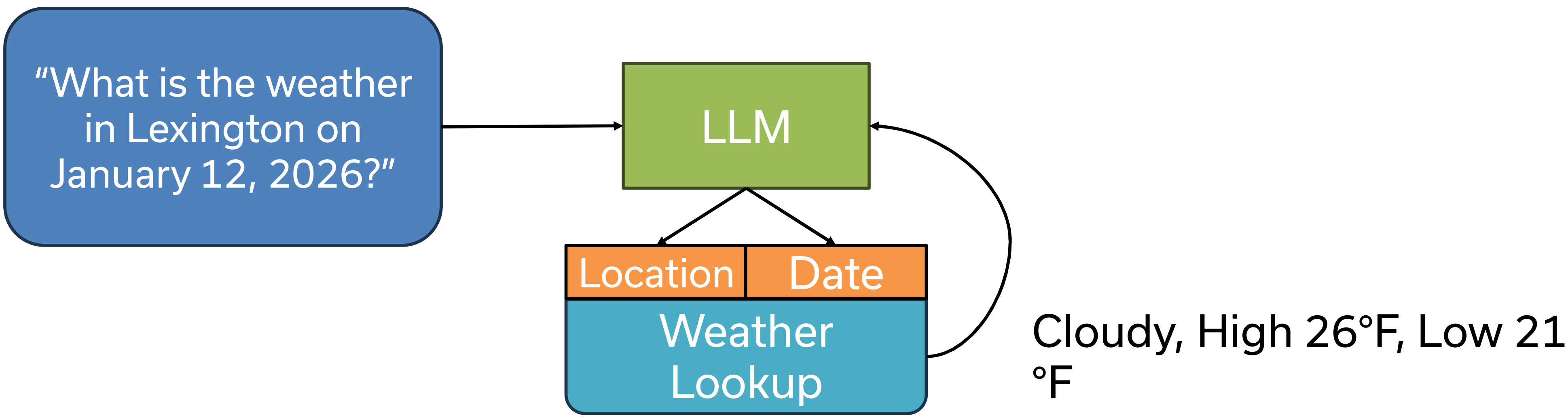
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



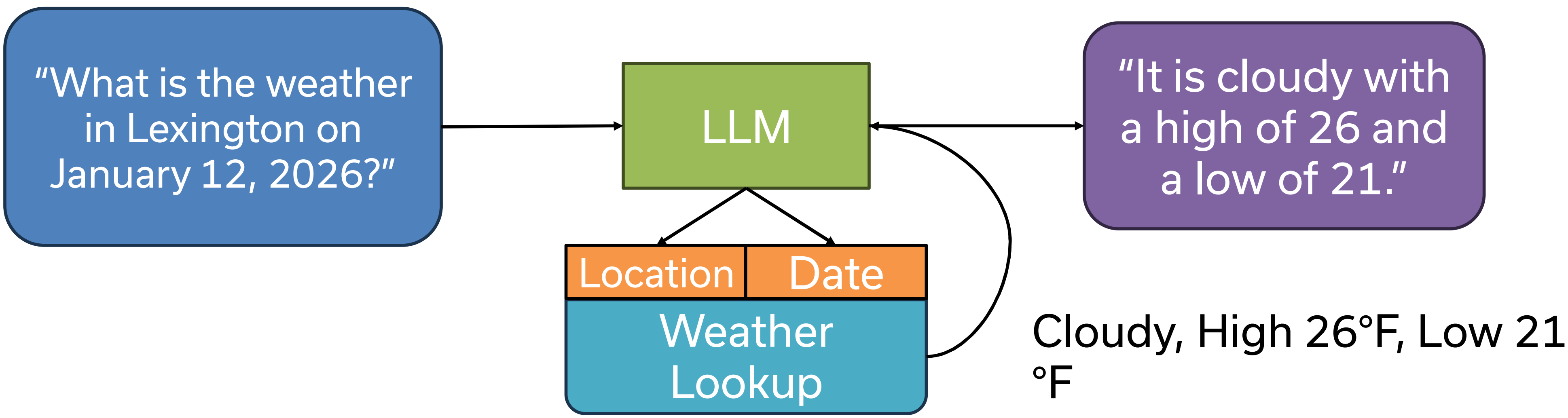
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



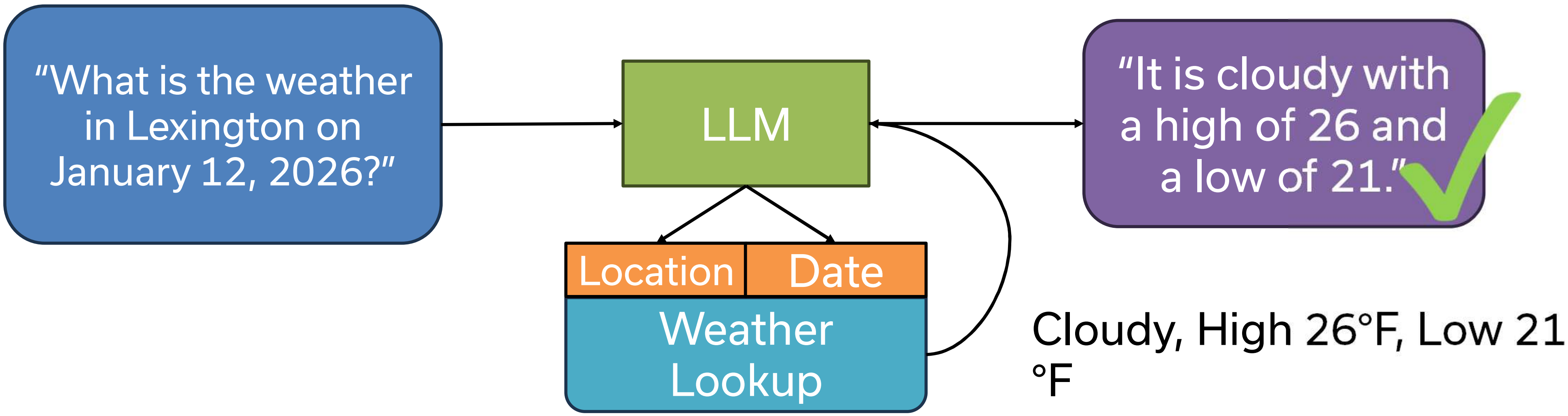
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



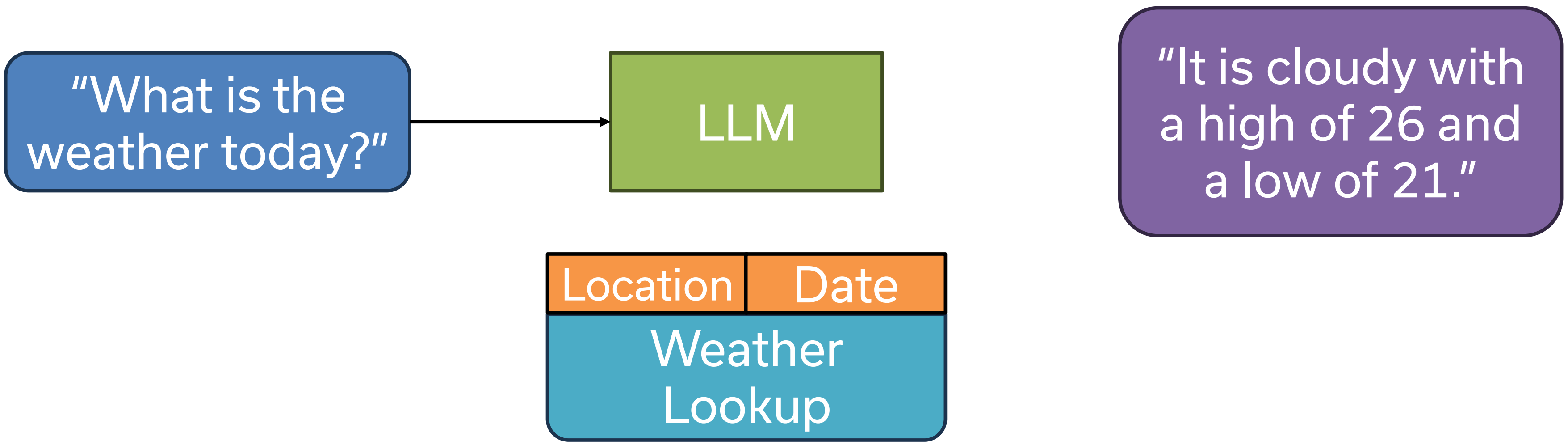
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



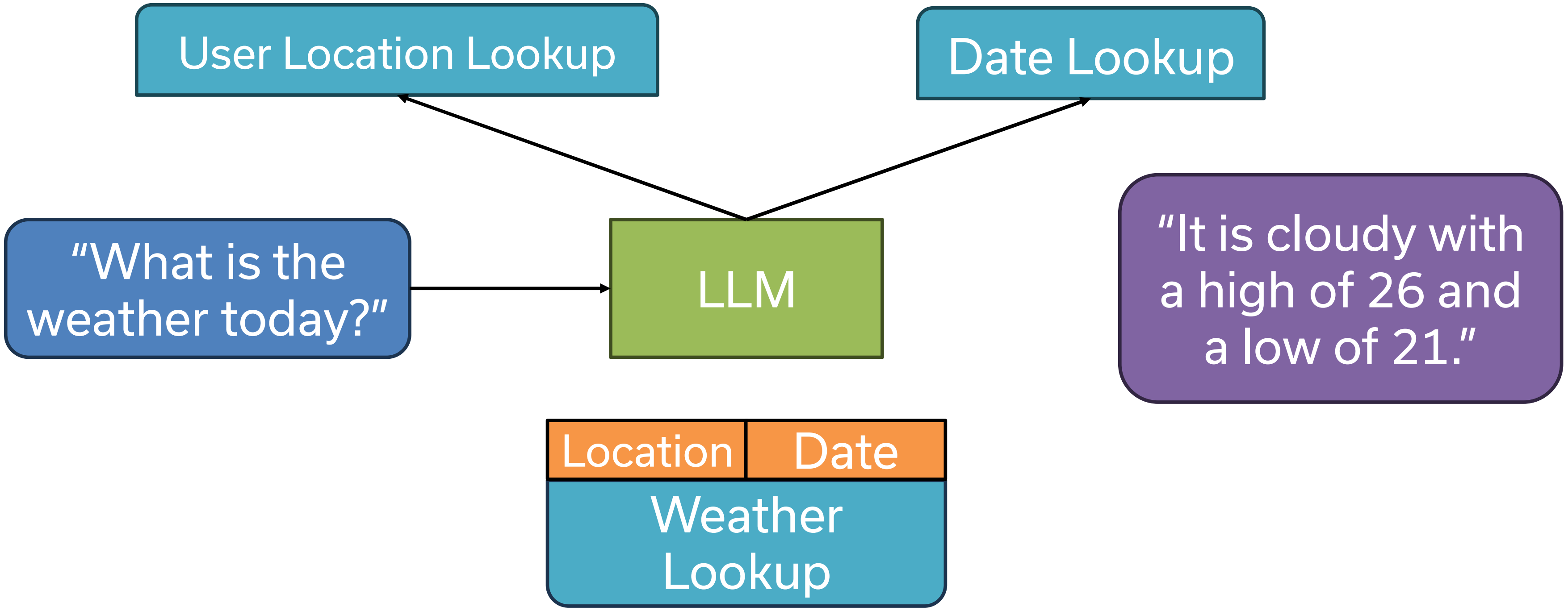
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



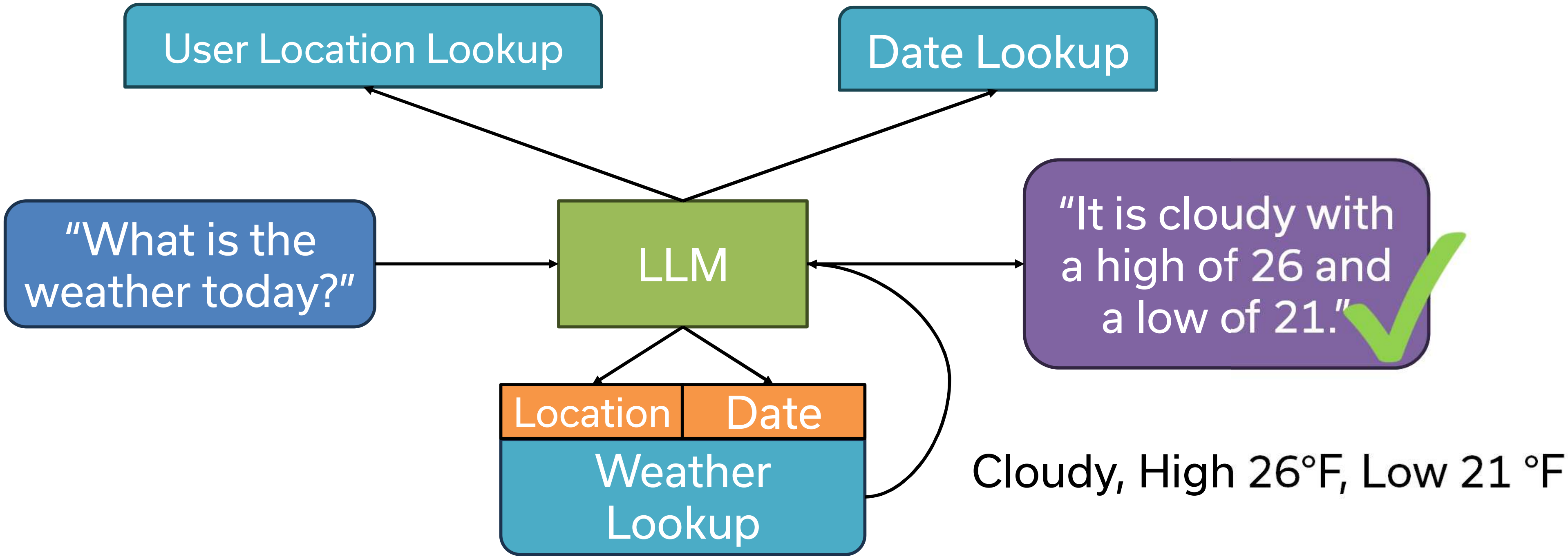
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



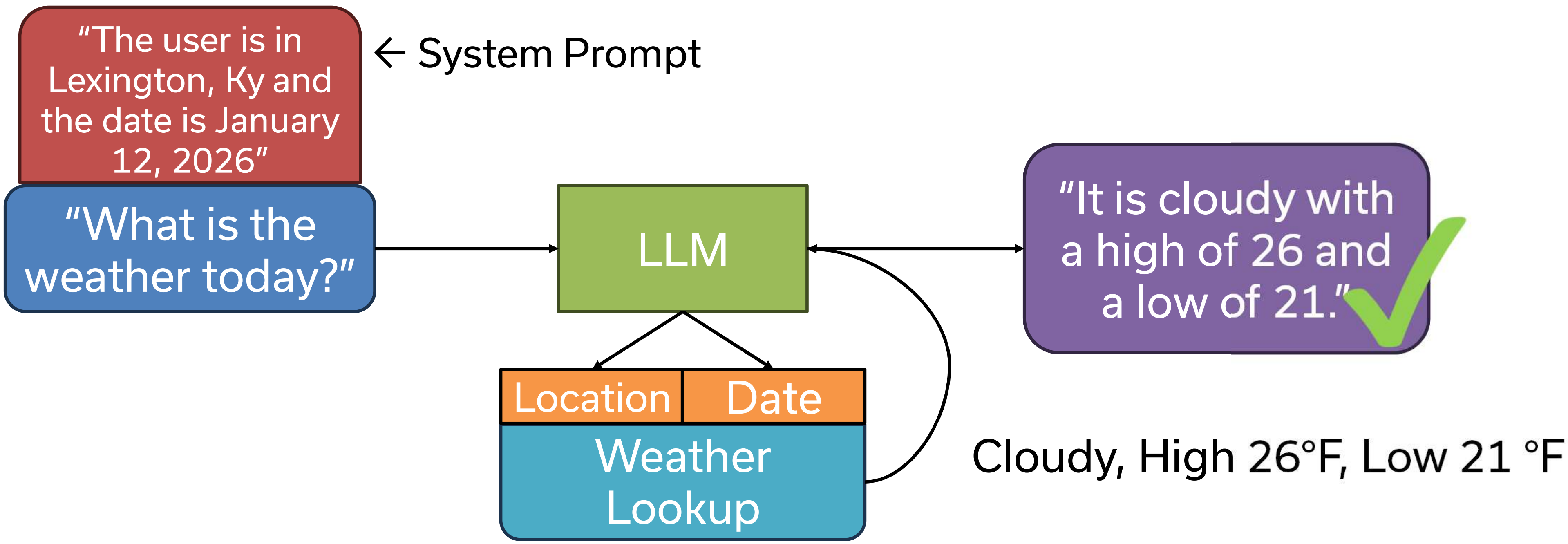
Tool Calling and Agents

- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



Tool Calling and Agents

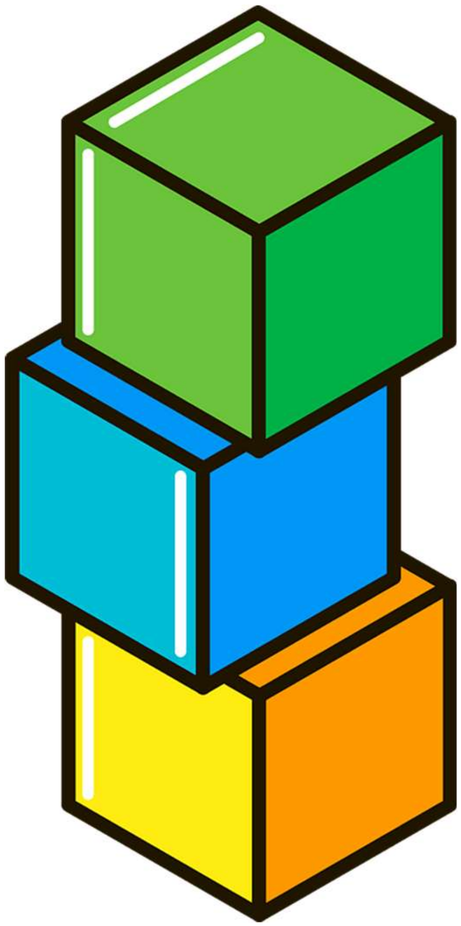
- Register external capabilities (tools) for LLM to call on
- LLMs are entirely text based
- From a prompt, can identify a tool that would aid in response and generate the parameters that tool needs
- Agents: An LLM assistant provided with specific tool capabilities



Training a Large Language Model

- Building block of language AI
- Further advancements expand on these fundamentals

- *How do you integrate tool calling and RAG to create your unique agent?*



Computer Vision Models

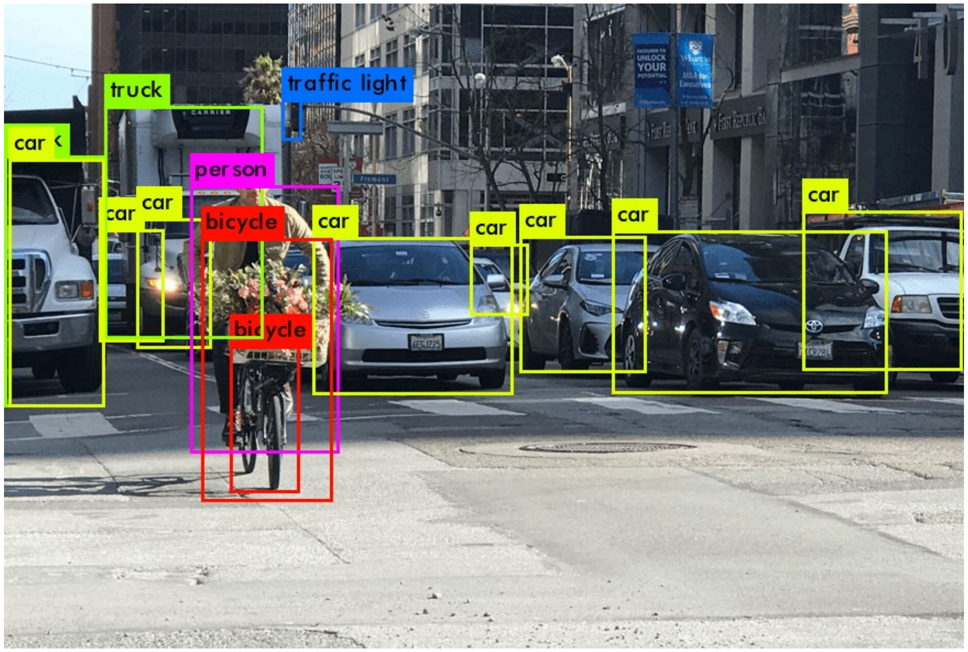
Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

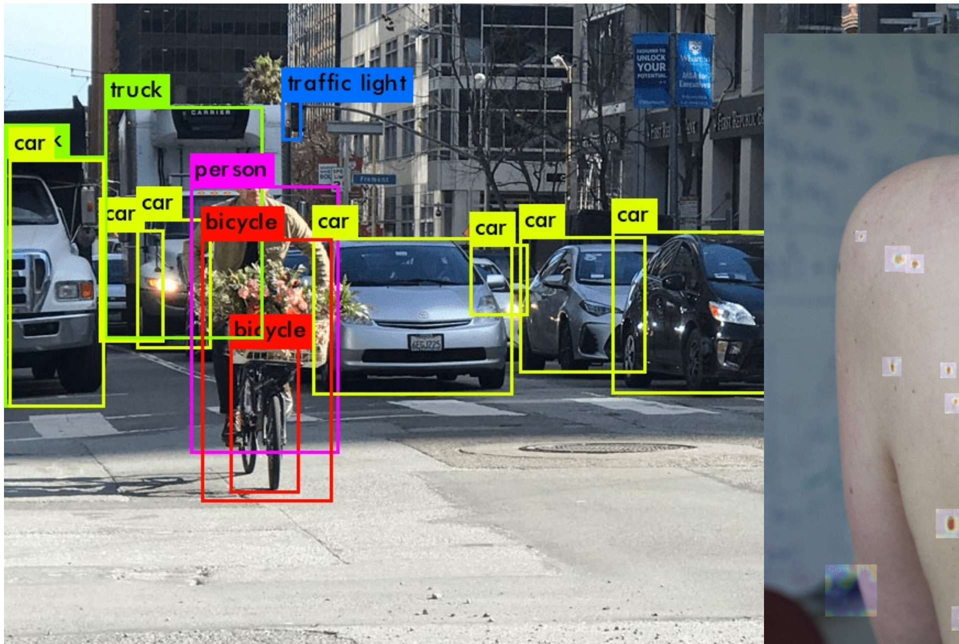
Computer Vision Models

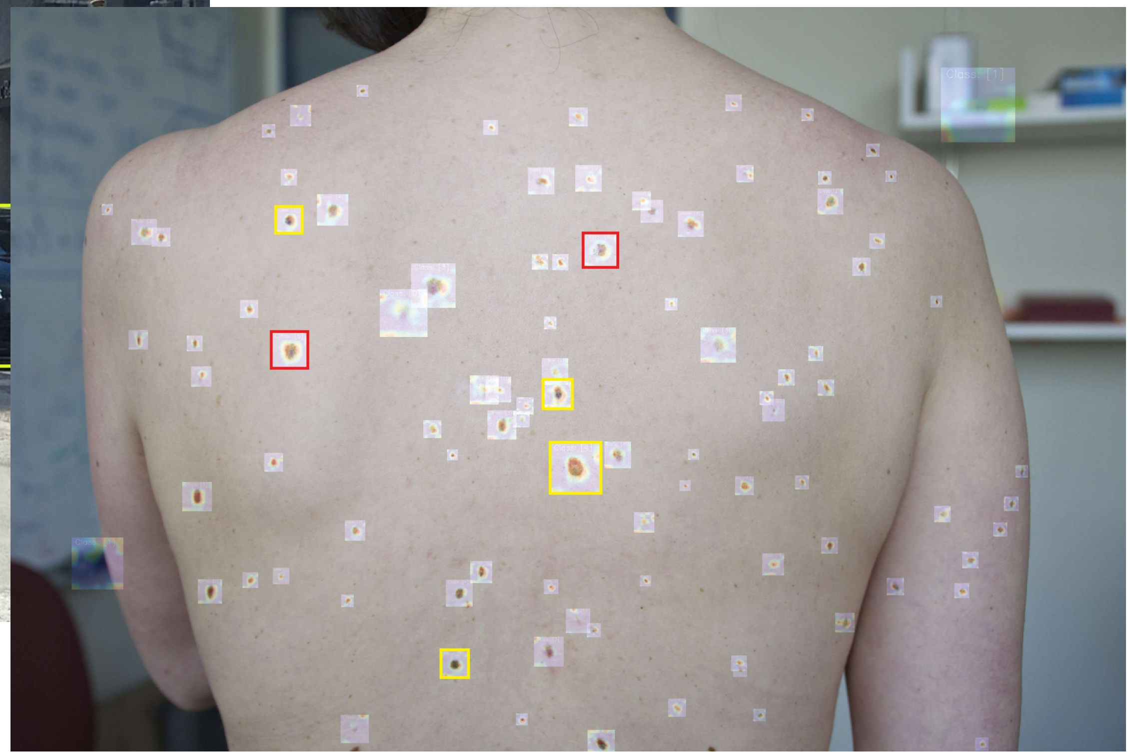
Where do we see vision models used?

Computer Vision Models

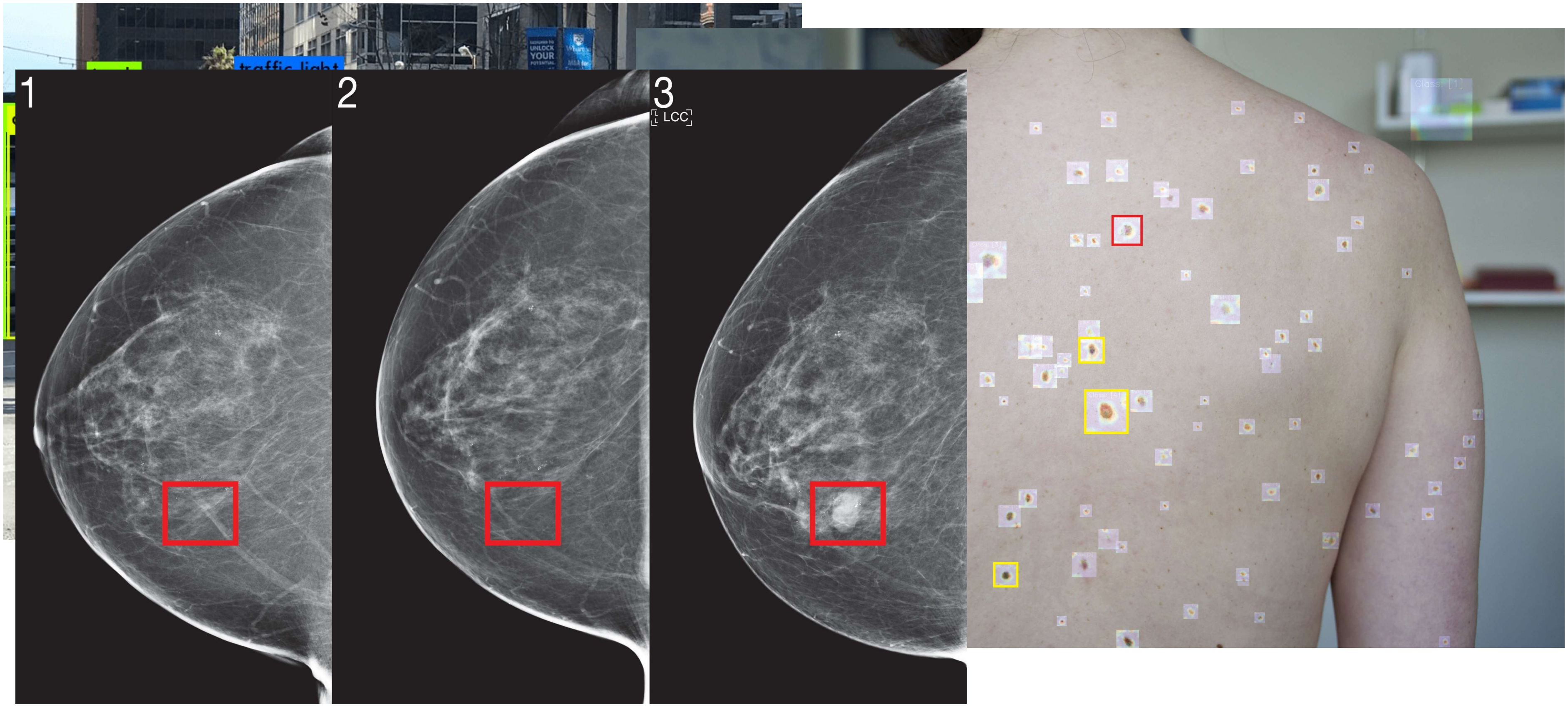


Computer Vision Models





Computer Vision Models



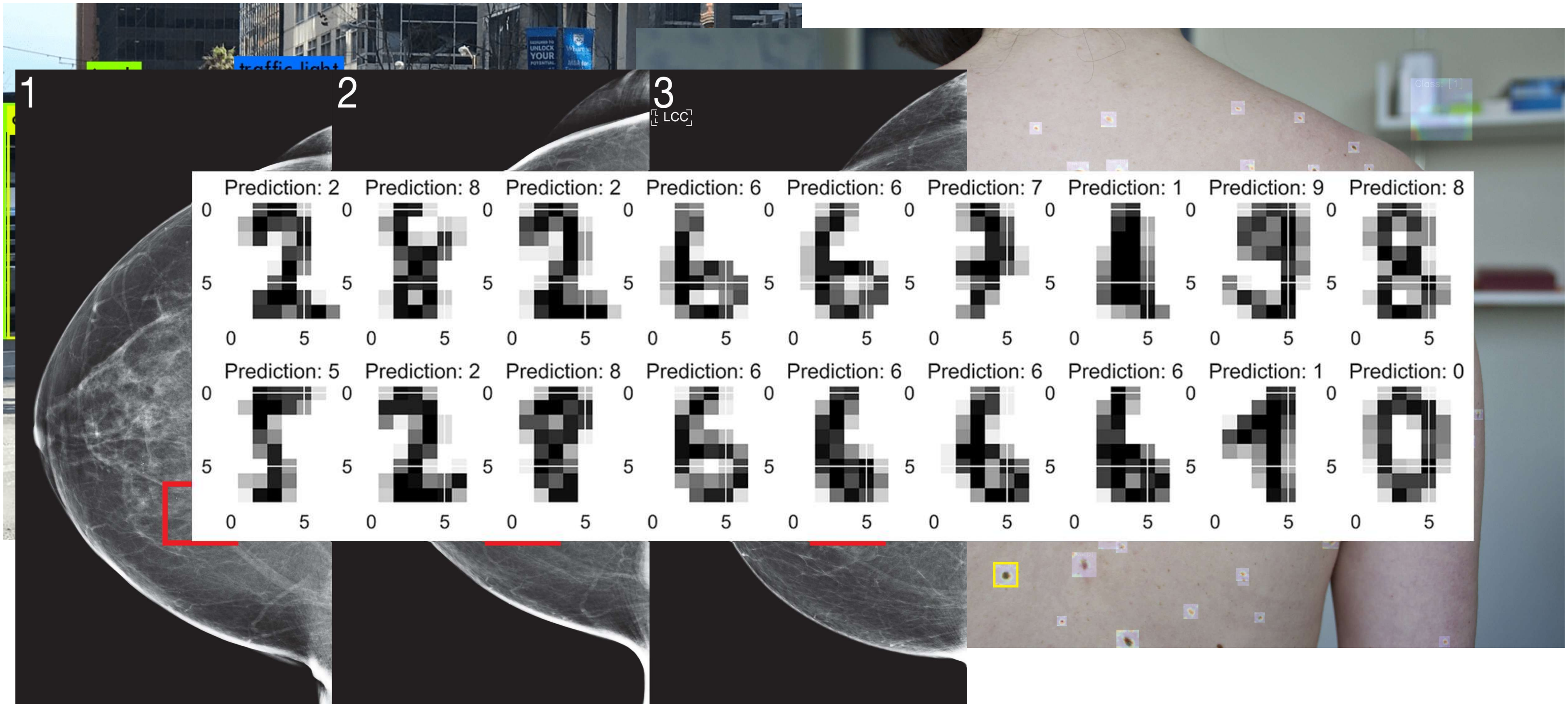
NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

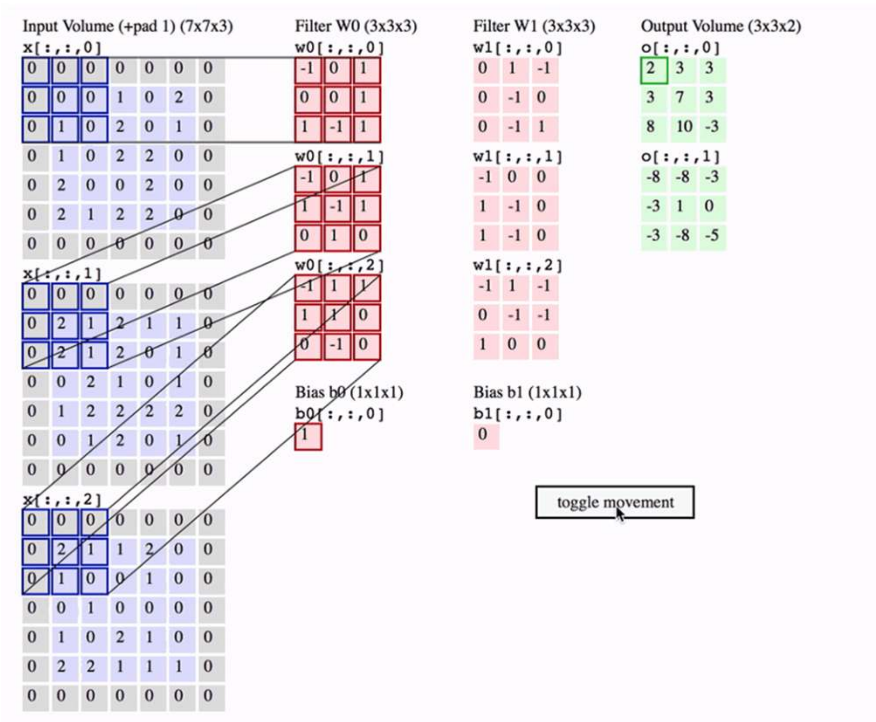
University of Kentucky

Computer Vision Models

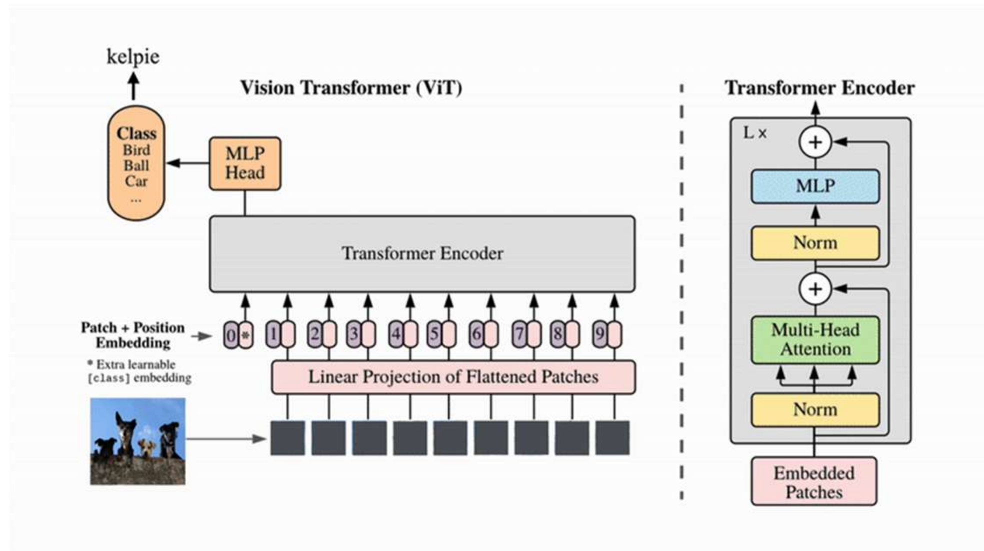


Two Main Architectures

Convolutional Neural Networks (CNNs)



Vision Transformers (ViTs)



Two Main Architectures

Convolutional Neural Networks (CNNs)

- Strengths:
  - Data Efficiency
  - Computational Efficiency
  - Excellent for Local Features
  - Strong Performance without massive-scale pre-training
- Weaknesses:
  - Limited Global Context
  - Rigid Architecture

Vision Transformers (ViTs)

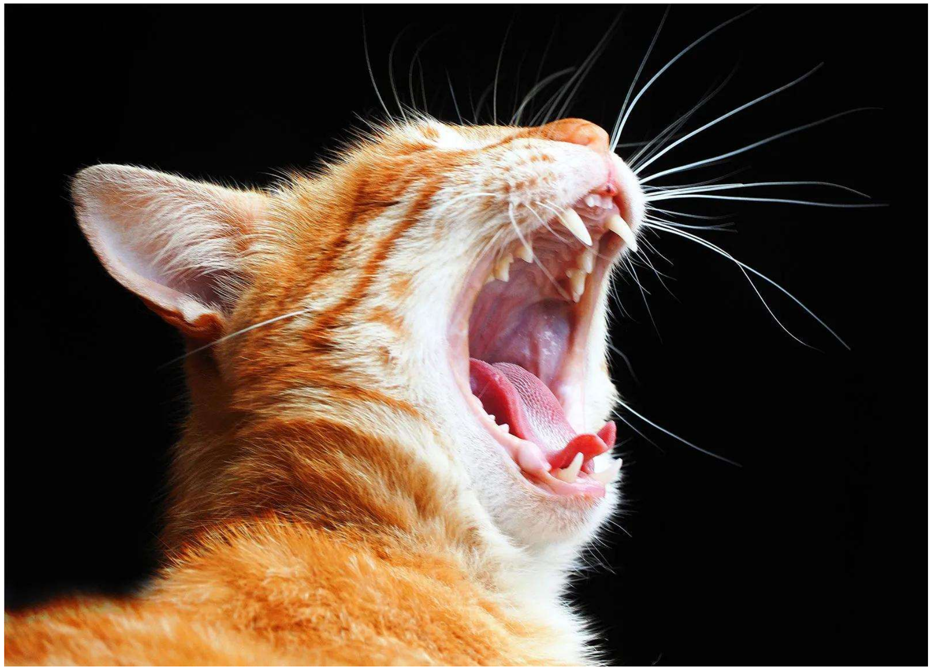
- Strengths:
  - Superior Global Context
  - Scalability
  - Flexibility
  - Ideal for Multimodality
- Weaknesses:
  - Data-Hungry
  - Computationally Expensive

Convolutional Neural Networks

- How can a program determine if this is a cat, or not a cat?

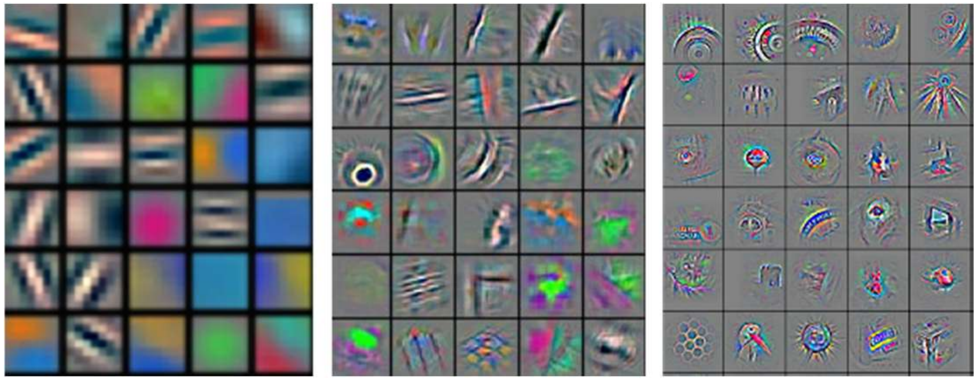
Cats can look like this...

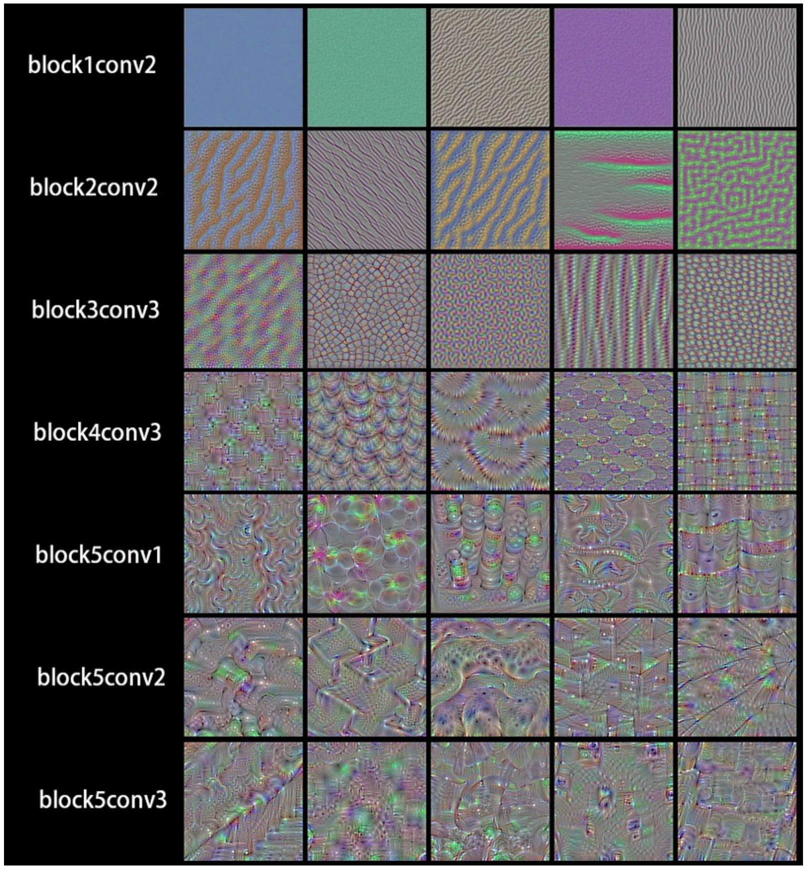
Cats can look like this... or like this!

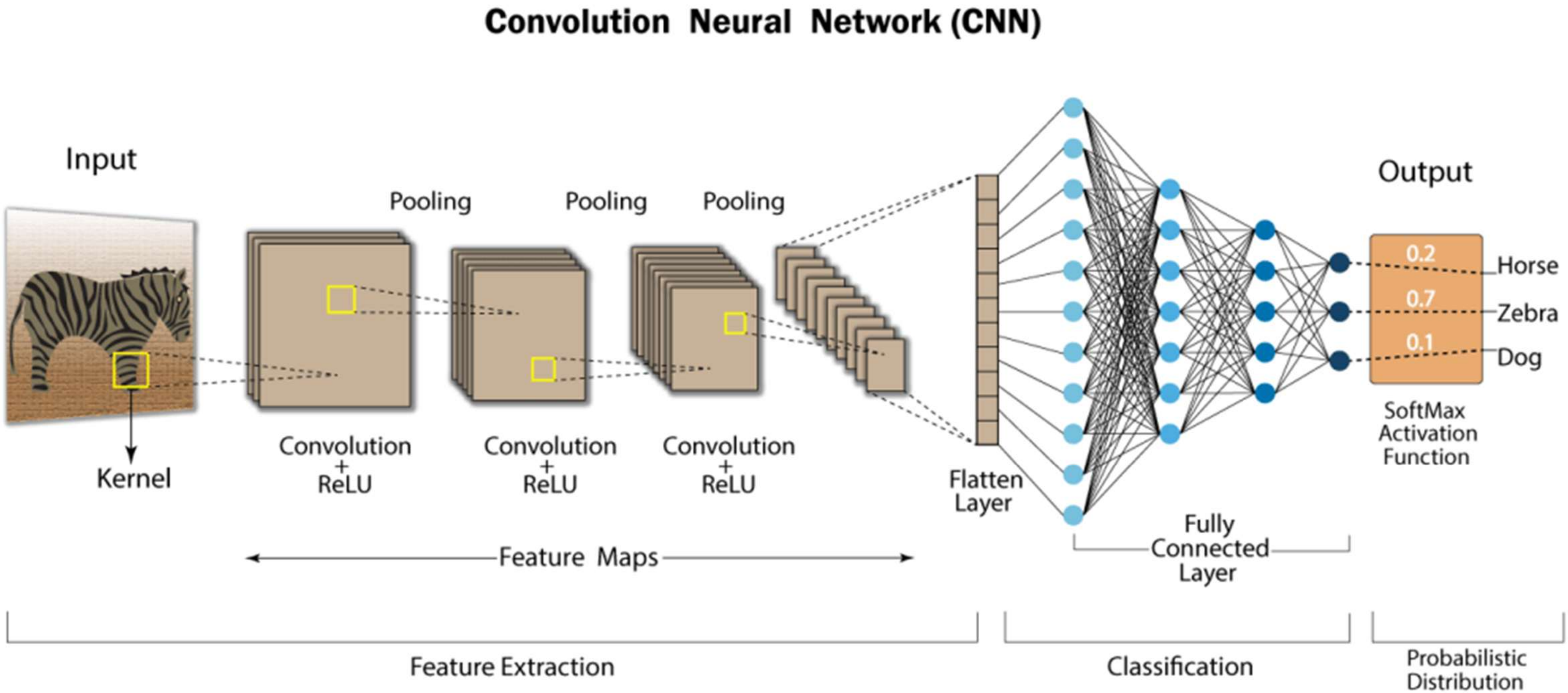


Convolutional Neural Networks (CNNs)

- Hierarchical feature extraction using filters (kernels)







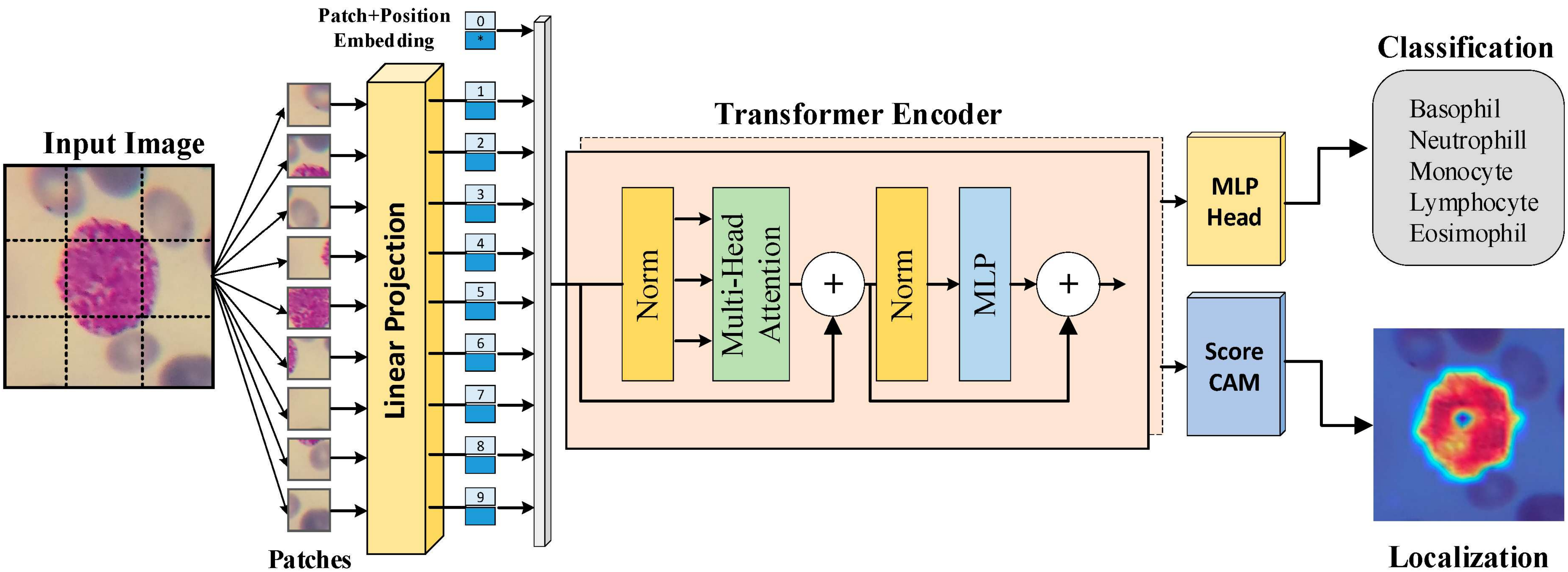
Convolution Neural Network (CNN)
Input
Pooling
Pooling
Pooling
Output
0.2
Horse
0.7
Zebra
0.1
Dog
Kernel
Convolution + ReLU
Convolution + ReLU
Convolution + ReLU
Flatten Layer
SoftMax Activation Function
Fully Connected Layer
Feature Maps
Feature Extraction
Classification
Probabilistic Distribution

NAIRR Pilot Resources

- NCSA Delta AI
  - Best For: Massive Scale Training
  - Why it helps: If a user wants to train a massive Vision Transformer (ViT) or a multimodal model (like CLIP or LLaVA) from scratch, standard clusters might be too slow. The Grace Hopper nodes here provide the massive bandwidth needed for huge image datasets.
- PSC Neocortex CS-2
  - Best For: Training from scratch
  - Why it helps: This system is architected specifically to eliminate bottlenecks in training large foundation models. It is a strong alternative for users whose models are too big for a single DGX node to handle efficiently.
- Lexset Seahaven:
  - Best For: Synthetic Data Generation
  - Why it helps: Vision models often fail because they lack diverse training data. This tool allows users to generate synthetic 3D environments and images to train models before testing them in the real world.

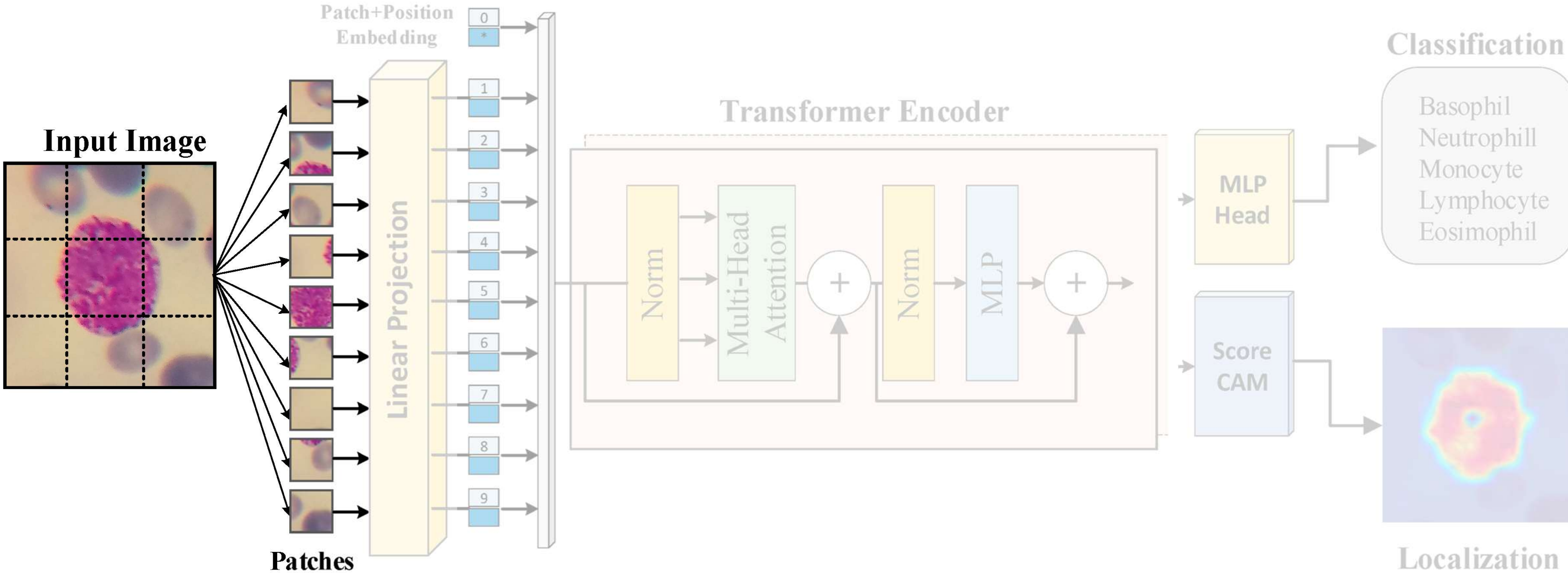
Vision Transformers (ViTs)

Unlike CNNs, which scan pixels with sliding windows, ViTs treat an image like a sentence of words



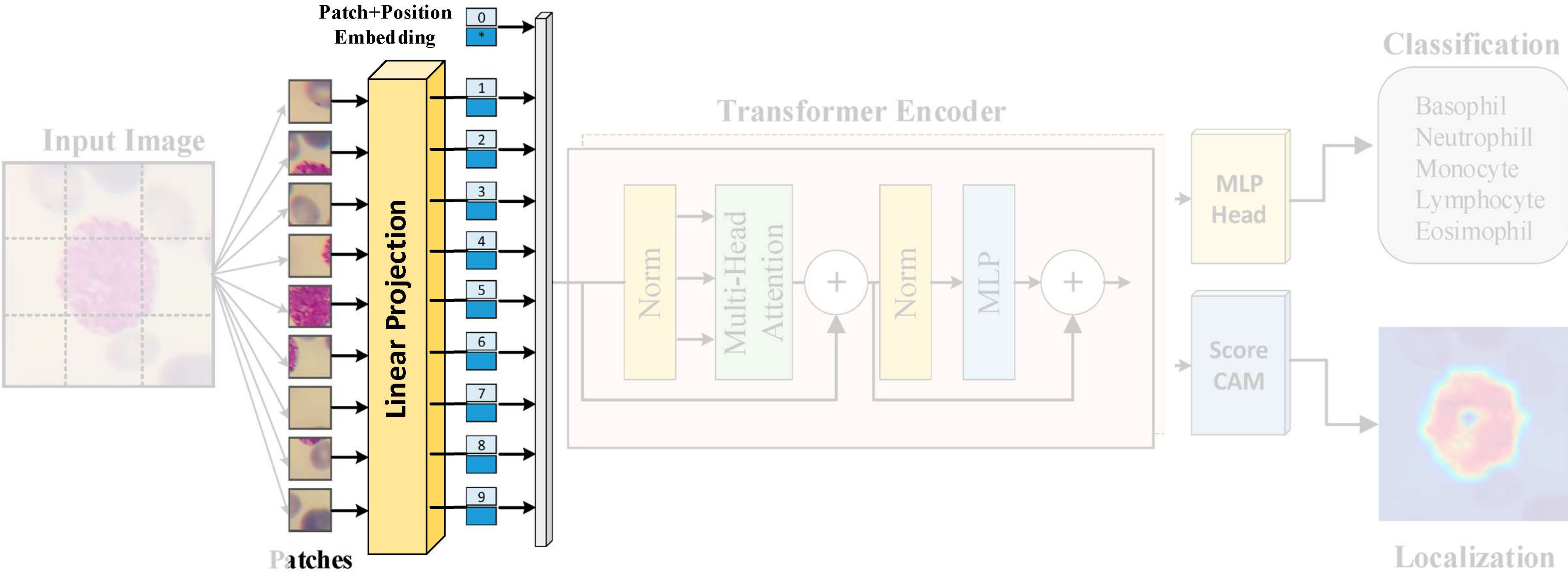
Vision Transformers (ViTs)

1. Input and Patching



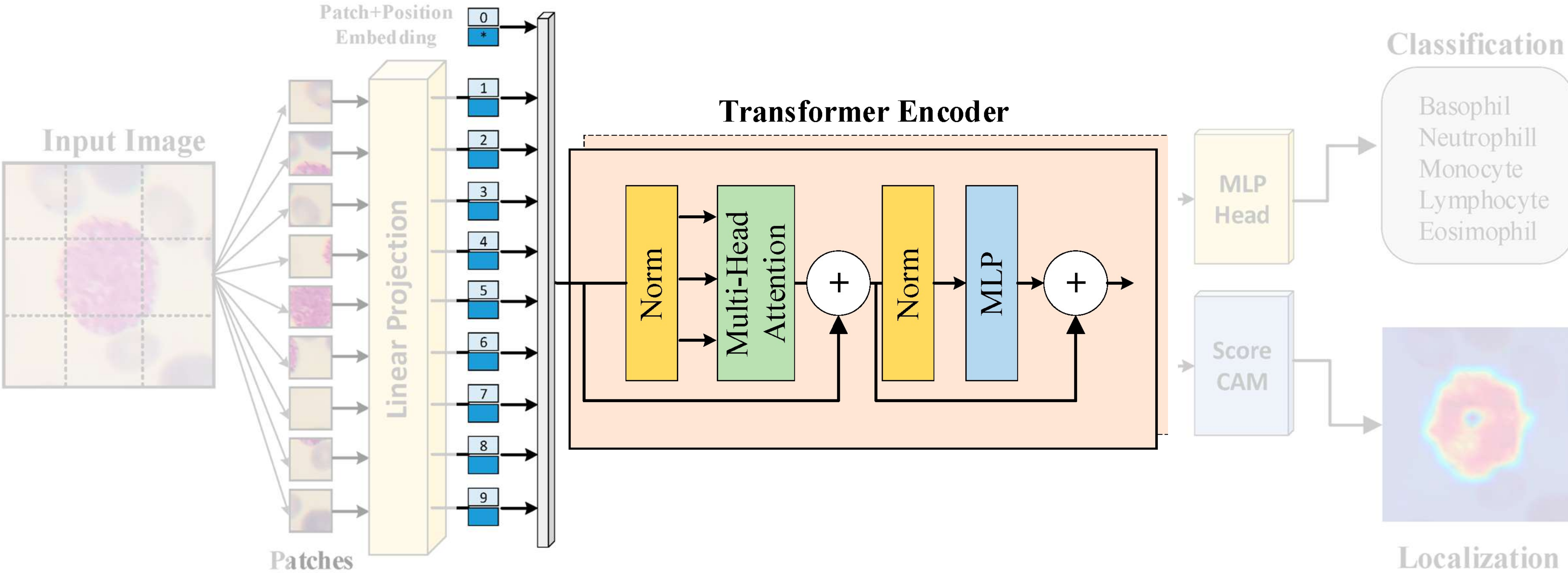
Vision Transformers (ViTs)

2. Linear Projection and Embeddings



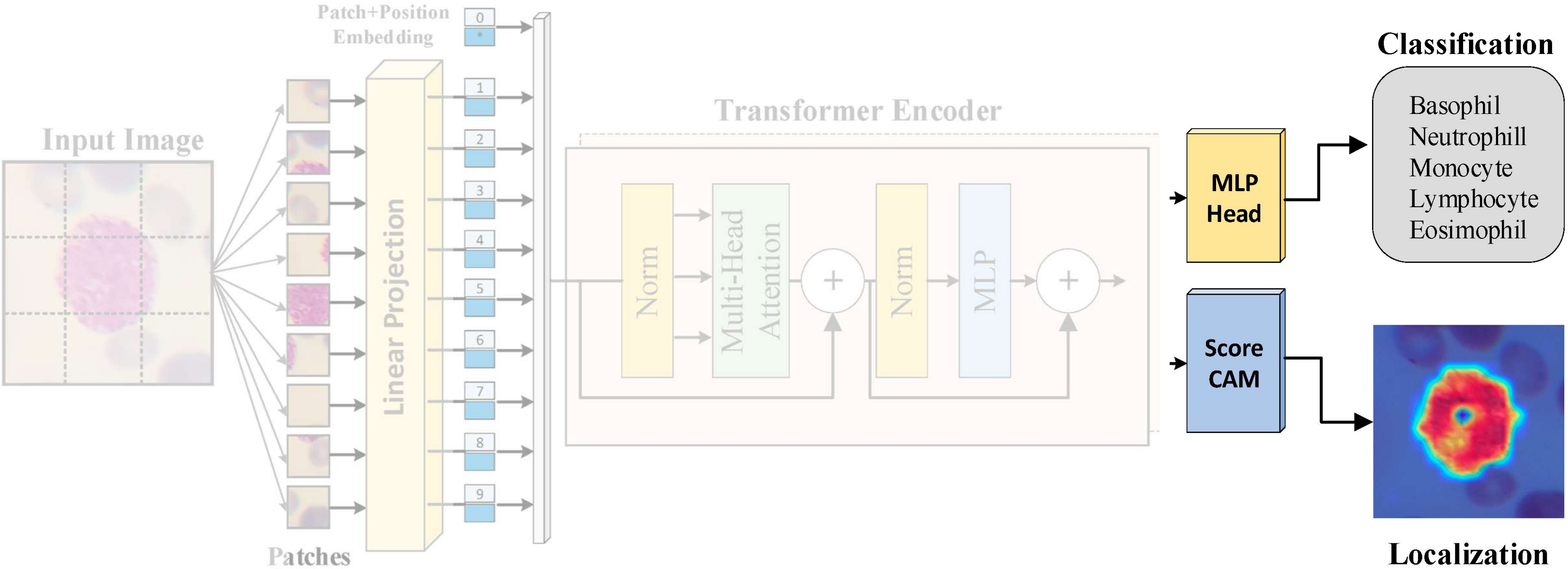
Vision Transformers (ViTs)

3. The Transformer Encoder



Vision Transformers (ViTs)

4. The Output



Vision Transformers (ViTs) vs CNNs

Summary:

1. Create patches from the input image
2. Create linear vectors (embeddings) from the patches
3. Perform Attention
4. Output results

NAIRR Pilot Resources

- Sage Edge Computing Platform
  - Best For: Edge Deployment and Real-Time Vision
  - Why it helps: Vision models are frequently deployed on cameras and robots, not just servers. Sage provides access to nodes with "Edge" GPUs (like NVIDIA Jetson or T4) across the US, allowing researchers to test how their models handle real-world sensor data and lower power constraints.

NAIRR Pilot Resources

- NCSA Delta AI
  - Best For: Massive Scale Training
  - Why it helps: If a user wants to train a massive Vision Transformer (ViT) or a multimodal model (like CLIP or LLaVA) from scratch, standard clusters might be too slow. The Grace Hopper nodes here provide the massive bandwidth needed for huge image datasets.
- PSC Neocortex CS-2
  - Best For: Training from scratch
  - Why it helps: This system is architected specifically to eliminate bottlenecks in training large foundation models. It is a strong alternative for users whose models are too big for a single DGX node to handle efficiently.

Coffee Break

Please join us in the Community Room (151)

Ask us questions & network with your peers!

Reconvene by 3:00 PM

NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

University of Kentucky

AI Pipelines

Mitchell Klusty, BS

Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

What are AI Pipelines and Why are They Important?

- Standardized processes for application management
- Integrating AI with your unique data and systems
- Deploying appropriate resources

- *Each project will have unique needs*
- What you need to do will be dependent on your project's goals and resources

- *Identify where AI is useful, and where it isn't*

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
   - What problem do you want to solve?
   - How does AI apply?
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
   - What data do you have to work with?
   - What needs to be done to make that data "AI-ready"?
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
   - What purpose should your model serve?
   - How can you apply your data to achieve that goal?
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
   - Connecting a model to larger architecture
   - How do you interface with the model?
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
   - How does the model perform?
   - How does the larger system perform alongside it?
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
   - Is data being properly protected?
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
   - How much load can you expect?
   - Where will you host?
   - What will it cost?
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
   - Continuous monitoring of security
   - Are protections holding post deployment?
9. Monitoring
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
   - Look for points of failure/areas for improvement
10. Iteration loops

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops
    - Address issues found in monitoring or security checks
    - How can you make the project better?

Full AI Project Lifecycle

1. Concept development
2. Data cleaning + exploration
3. Model prototyping
4. Pipeline buildout
5. Evaluation
6. Security and Privacy checks
7. Deployment
8. Security and Privacy checks again
9. Monitoring
10. Iteration loops

- This is standard for most software development projects, not just AI
- How are AI projects unique?

Consider:

Where does AI fit into my workflow?

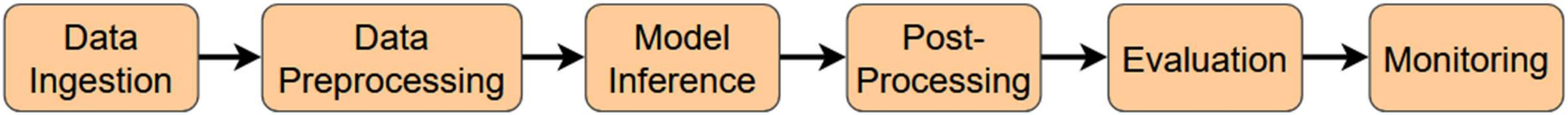
- How would introducing AI benefit individual features?

Not every problem needs an AI solution

- AI can be costly
- Is there another solution that will work just as well?

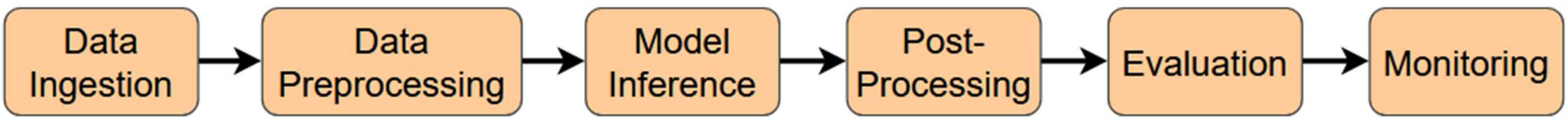
Components of an AI Workflow

- AI needs data
  - How to handling project-specific data?
- How to support model inference?
- How to evaluate the performance?
- Where to look for performance issues/improvements



Components of an AI Workflow

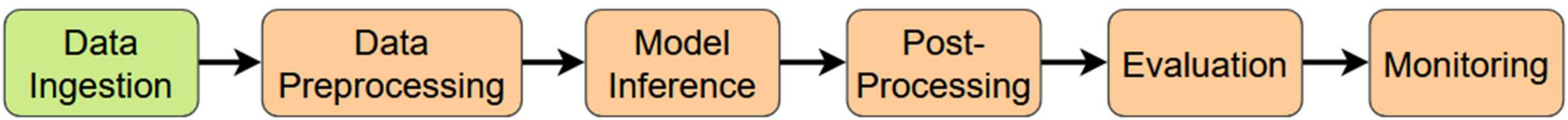
- AI needs data
  - How to handling project-specific data?
- How to support model inference?
- How to evaluate the performance?
- Where to look for performance issues/improvements



Every piece is dependent on what comes before, and depended on by what comes after

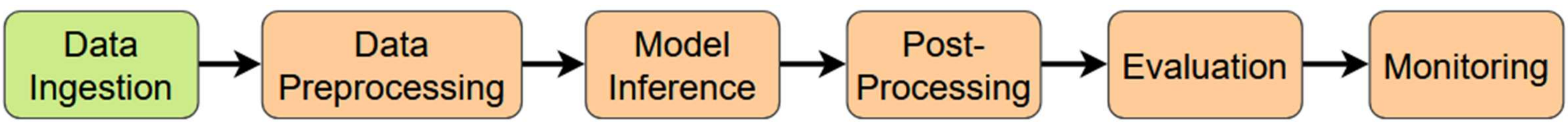
Components of an AI Workflow – Data Ingestion

- What data do you need?
- How will users supply data?
- How to collect data?
- Structure data to be adaptive and reusable
  - Exact goals/conditions may change



Components of an AI Workflow – Data Ingestion

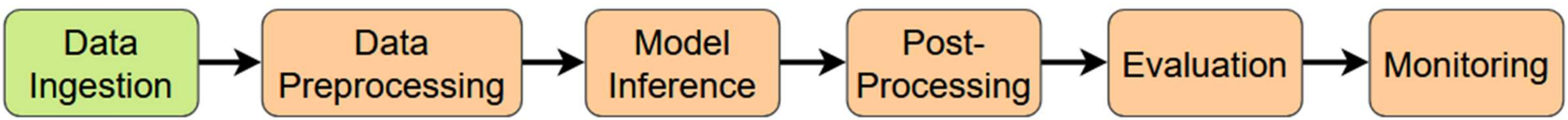
- Do you need to make a public facing interface?
  - Web Hosting
  - Application Programming Interface (API)
- How much data to ingest
  - Network loads
  - Storage considerations



Components of an AI Workflow – Data Ingestion

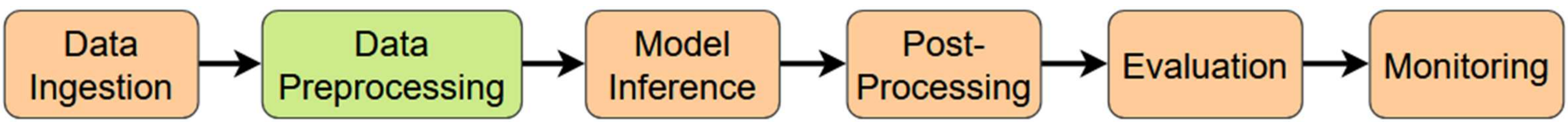
Common Oversights or Problems

- Not considering network constraints
- Not considering storage constraints



Components of an AI Workflow – Data Preprocessing

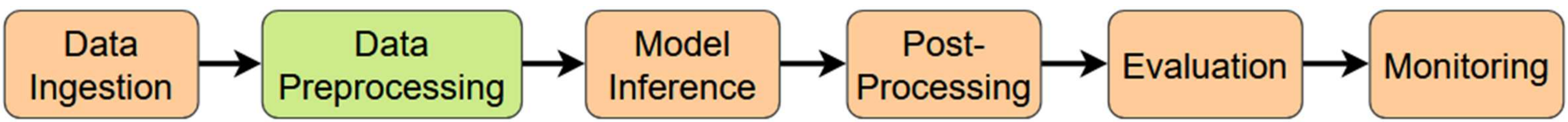
- Massive datasets exist that aren't "AI-ready"
  - Large sets of images, no labels/annotations
  - Large bodies of text
- How to prepare data for training?
- How to best perform inference on data?
  - Resources for inference can be expensive
  - How to make most efficient use of those resources when applying data?



Components of an AI Workflow – Data Preprocessing

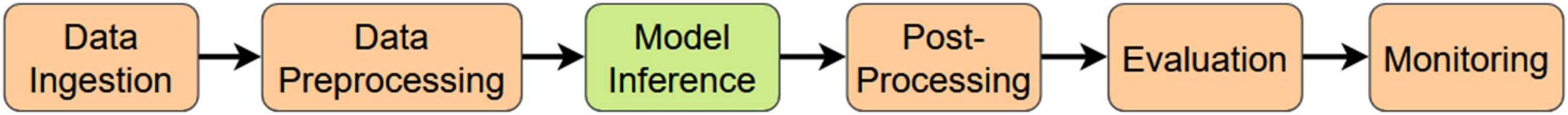
Common Oversights or Problems

- Not storing data efficiently
- Not making data easily accessible for next steps
- Garbage In, Garbage Out
  - Applies to both training and inference
  - Must ensure data is cleaned after intake



Components of an AI Workflow – Model Inference

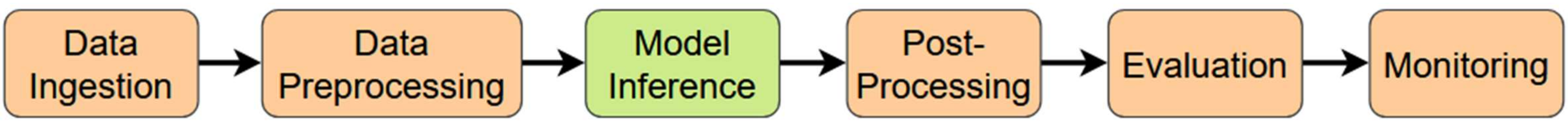
- What type of model are you running?
- What resources does that need?



Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

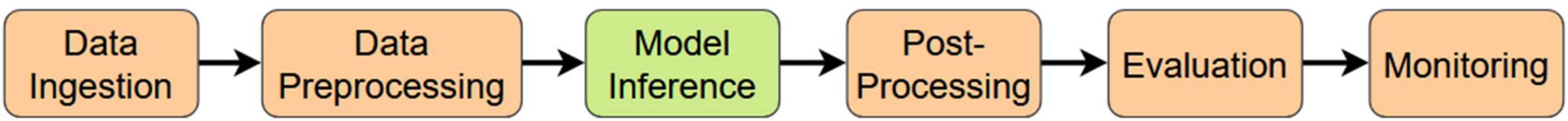
- LLMs require higher GPU memory to host
  - Dependent on size of model
    - Number of parameters
    - Precision/Quantization



Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- LLMs require higher GPU memory to host
  - Dependent on size of model
    - Number of parameters (internal weights)
      - Llama 3 8B, 70B, 405B
      - DeepSeek-V3 37B active, 671 total
      - Gemini 3 estimated 20B active, 1 trillion total
    - Precision/Quantization

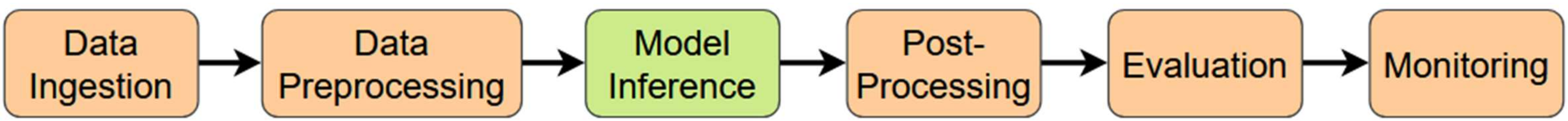


Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- LLMs require higher GPU memory to host
  - Dependent on size of model
    - Number of parameters
    - Precision/Quantization
      - How many bits in each parameter

	FP32	FP16	INT8	INT4
Bits per parameter	32	16	8	4
Bytes per parameter	4	2	1	0.5

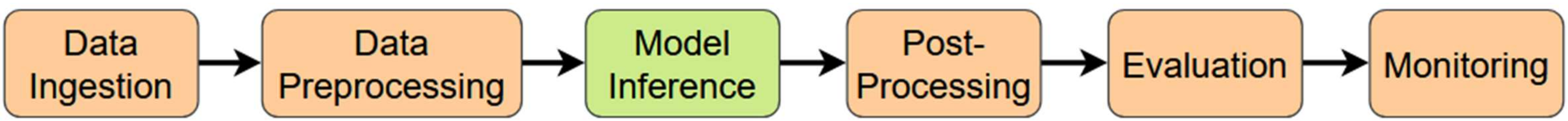


Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- LLMs require higher GPU memory to host
  - Dependent on size of model
    - Number of parameters
    - Precision/Quantization

$$Required\ VRAM\ (GB) \approx \frac{parameters * bytes\ per\ parameter}{1\ billion}$$
$$\approx \#\ billion\ parameters * bytes\ per\ parameter$$

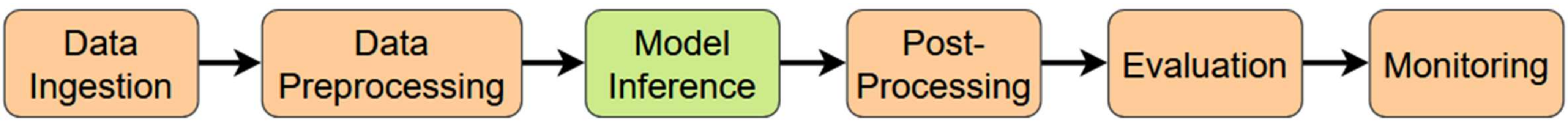


Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- LLMs require higher GPU memory to host
  - Dependent on size of model
    - Number of parameters
    - Precision/Quantization

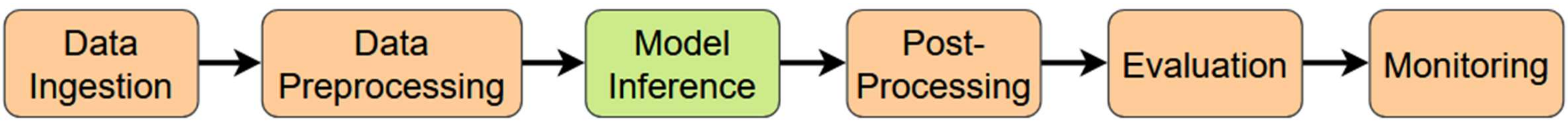
$$Required\ VRAM\ (GB) \approx \frac{parameters * bytes\ per\ parameter}{1\ billion}$$
$$\approx \#\ billion\ parameters * bytes\ per\ parameter$$
$$+\ overhead$$



Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- Hosting LLMs
  - 8B model at FP16: ~20GB VRAM
  - 400B model at FP16: ~850GB VRAM
  - 1T model at FP16: ~2TB VRAM



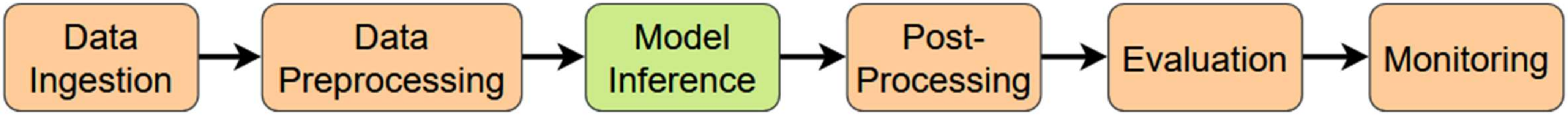
Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- Hosting LLMs
  - 8B model at FP16: ~20GB VRAM
  - 400B model at FP16: ~850GB VRAM
  - 1T model at FP16: ~2TB VRAM

GPU Type	VRAM (GB)	Cost ($)	Cloud Rental
A10	24	~2k	$0.50-2/hr
A100	40/80	10-17k	$1-4/hr
H100	80	27-30k	$2-7/hr
H200	141	31-32k	$4-10/hr

- NAIRR Resources can help get started



Components of an AI Workflow – Model Inference

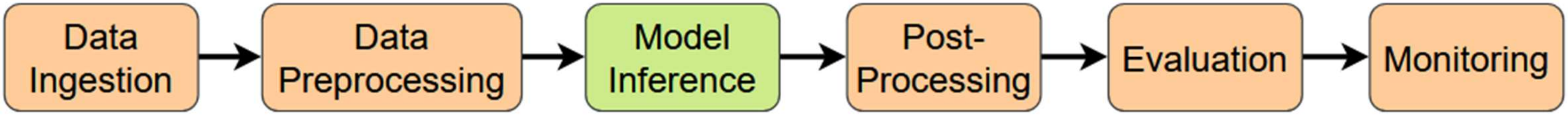
- What type of model are you running?
- What resources does that need?

- Hosting LLMs
  - 8B model at FP16: ~20GB VRAM
  - 400B model at FP16: ~850GB VRAM
  - 1T model at FP16: ~2TB VRAM

GPU Type	VRAM (GB)	Cost ($)	Cloud Rental
A10	24	~2k	$0.50-2/hr
A100	40/80	10-17k	$1-4/hr
H100	80	27-30k	$2-7/hr
H200	141	31-32k	$4-10/hr

- NAIRR Resources can help get started

1 GPU ≈ $1500/mo



Components of an AI Workflow – Model Inference

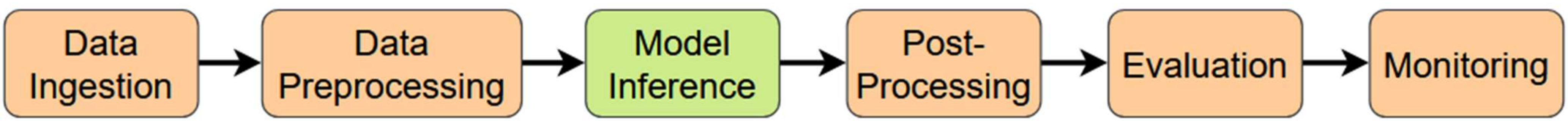
- What type of model are you running?
- What resources does that need?

- Hosting LLMs
  - 8B model at FP16: ~20GB VRAM
  - 400B model at FP16: ~850GB VRAM
  - 1T model at FP16: ~2TB VRAM

GPU Type	VRAM (GB)	Cost ($)	Cloud Rental
A10	24	~2k	$0.50-2/hr
A100	40/80	10-17k	$1-4/hr
H100	80	27-30k	$2-7/hr
H200	141	31-32k	$4-10/hr

- NAIRR Resources can help get started

8 GPU ≈ $24k/mo



Components of an AI Workflow – Model Inference

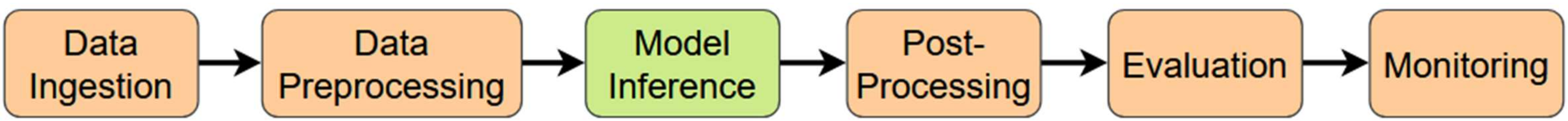
- What type of model are you running?
- What resources does that need?

- Hosting LLMs
  - 8B model at FP16: ~20GB VRAM
  - 400B model at FP16: ~850GB VRAM
  - 1T model at FP16: ~2TB VRAM

GPU Type	VRAM (GB)	Cost ($)	Cloud Rental
A10	24	~2k	$0.50-2/hr
A100	40/80	10-17k	$1-4/hr
H100	80	27-30k	$2-7/hr
H200	141	31-32k	$4-10/hr

- NAIRR Resources can help get started

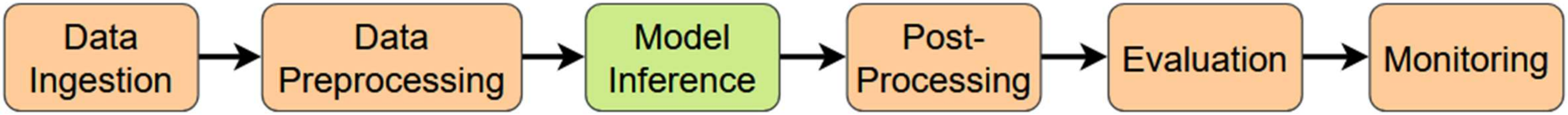
8 GPU ≈ $60k/mo



Components of an AI Workflow – Model Inference

- Model-as-a-Service (MaaS)
  - OpenAI API
  - Claude API
  - Gemini API

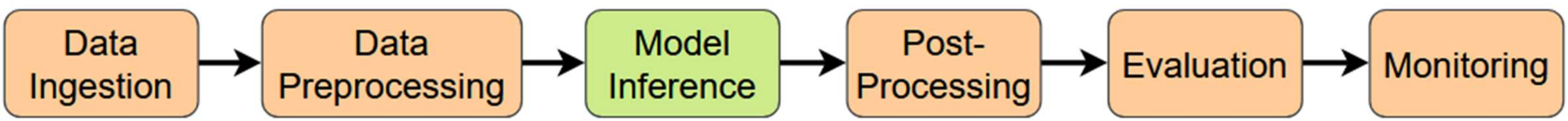
- Cost based on token usage
- Estimate token usage per month for your project
  - During evaluation step
  - NAIRR resources available



Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

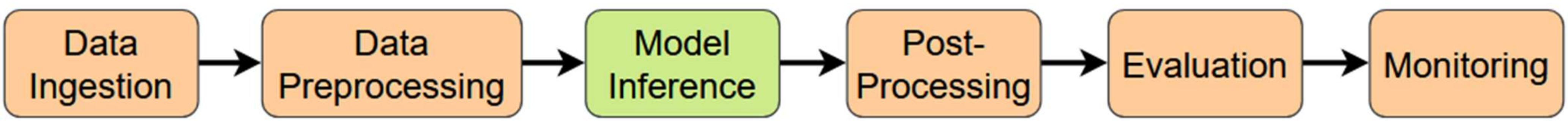
- Vision and traditional ML models more lightweight
  - Significantly less active memory requirements for inference
  - Many can run on only CPU



Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

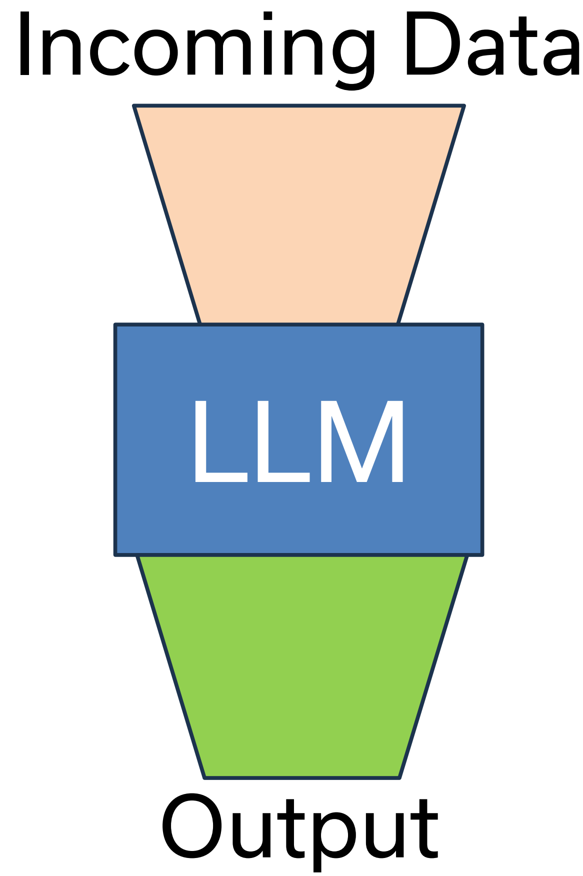
- How many users do you expect?
- Run multiple models in parallel?

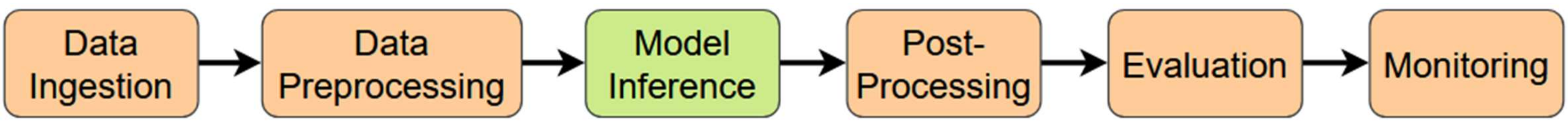


Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- How many users do you expect?
- Run multiple models in parallel?

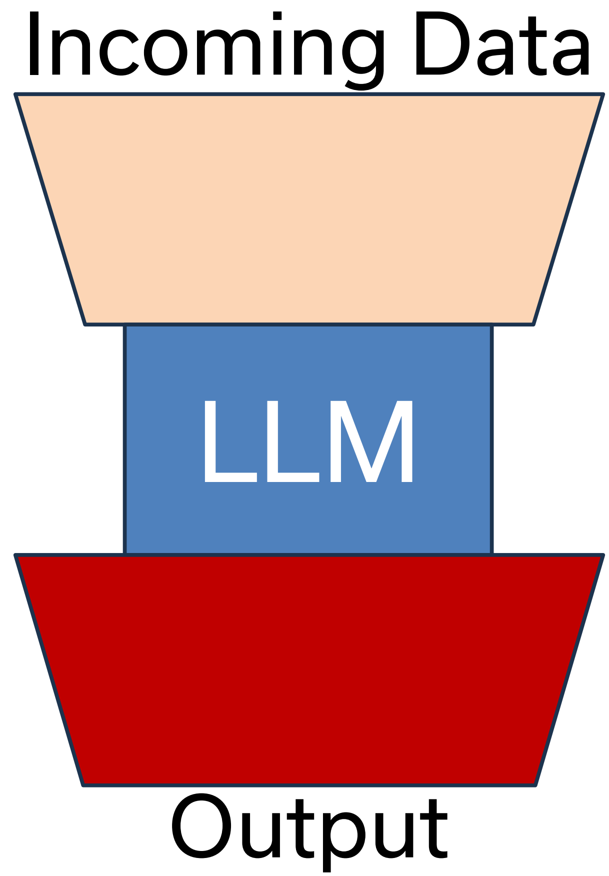


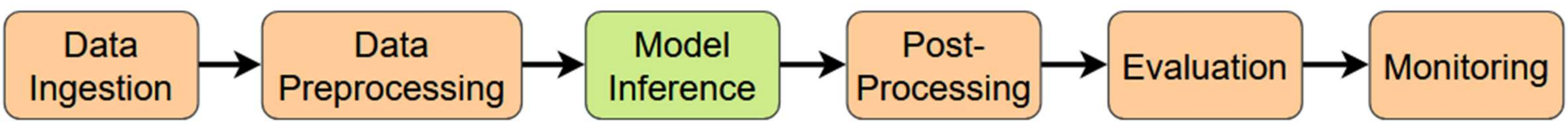


Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- How many users do you expect?
- Run multiple models in parallel?

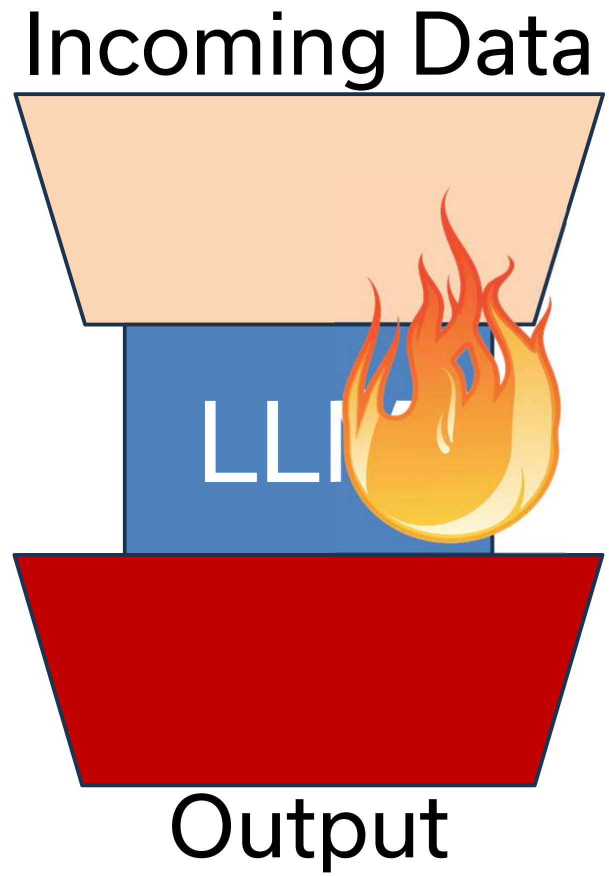


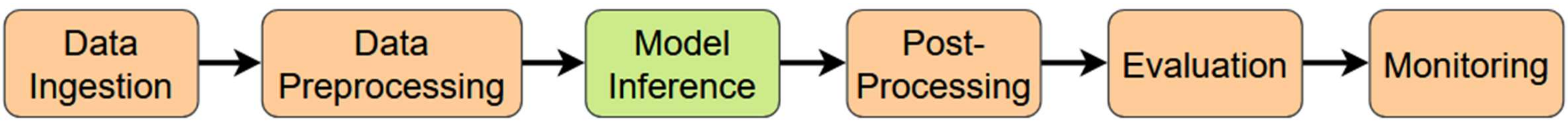


Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- How many users do you expect?
- Run multiple models in parallel?

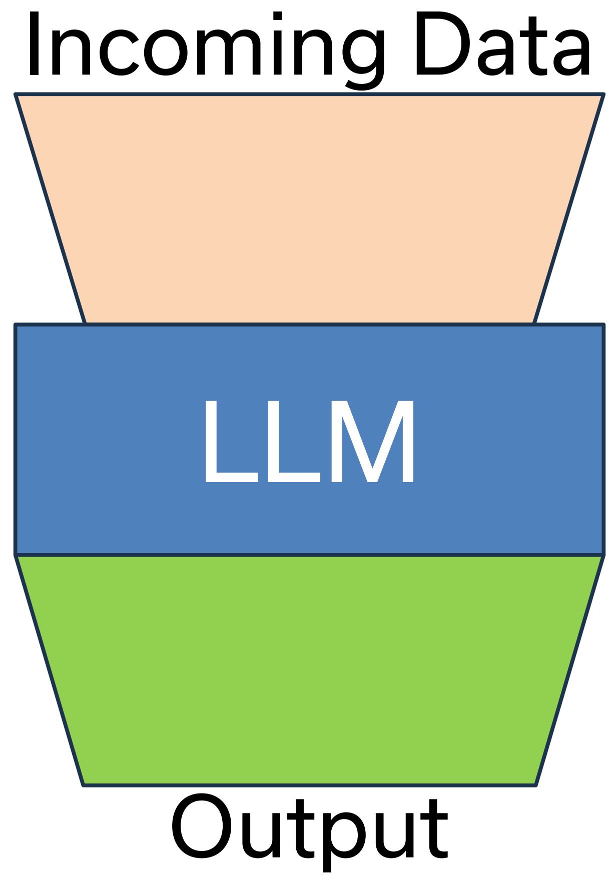


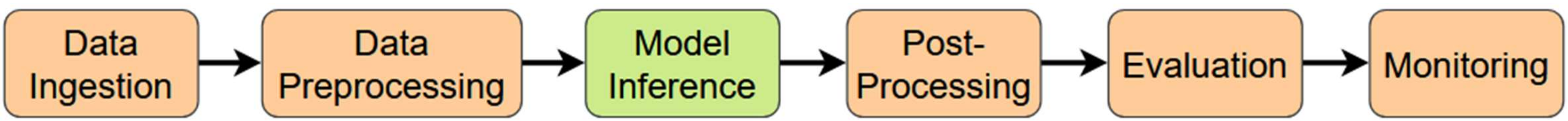


Components of an AI Workflow – Model Inference

- What type of model are you running?
- What resources does that need?

- How many users do you expect?
- Run multiple models in parallel?

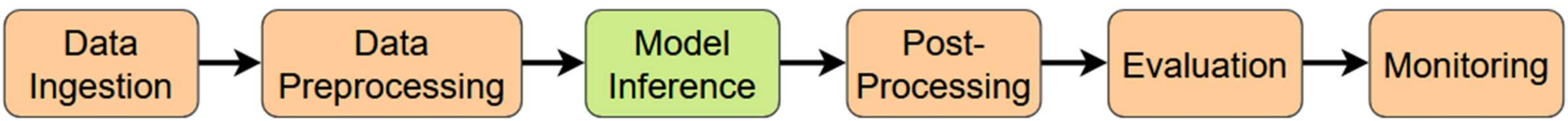




Components of an AI Workflow – Model Inference

Common Oversights or Problems

- Underestimating user load
- Not properly testing model
- Over-relying on the model
  - Especially important and difficult for LLMs
  - Inputs can be very broad, need measures for when model is wrong
  - Use RAG and other techniques

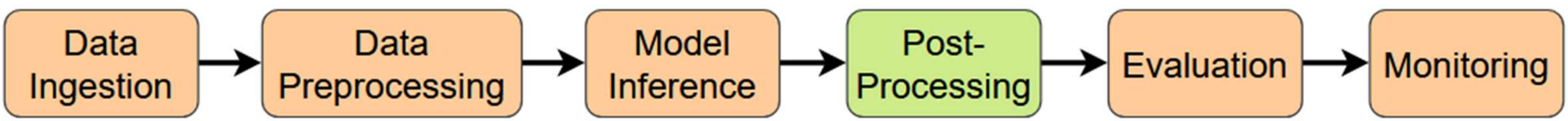


Components of an AI Workflow – Post-Processing

- How to interpret the output?
- How to present the output?

- If user-facing:
  - How to make output human-friendly?
  - How to report confidence?

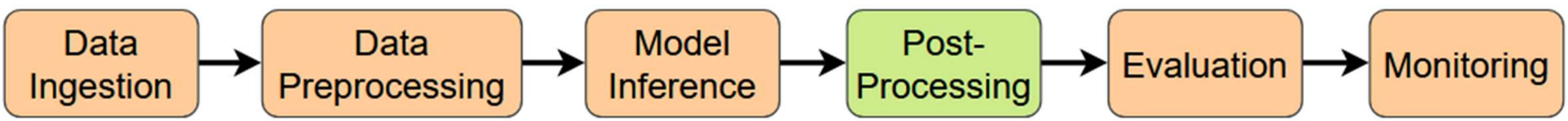
- How to limit network load with the output?



Components of an AI Workflow – Post-Processing

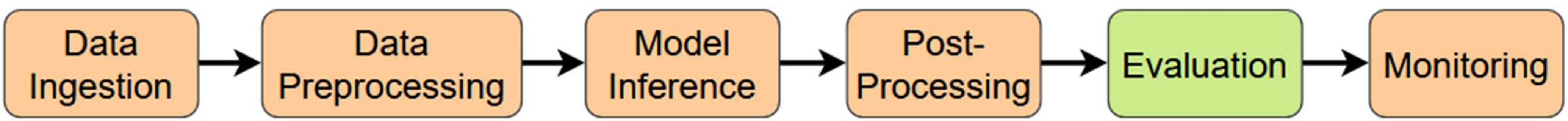
Common Oversights or Problems

- Reporting raw model output to users is often not best-practice
- Not logging and storing useful data on model outputs



Components of an AI Workflow – Evaluation

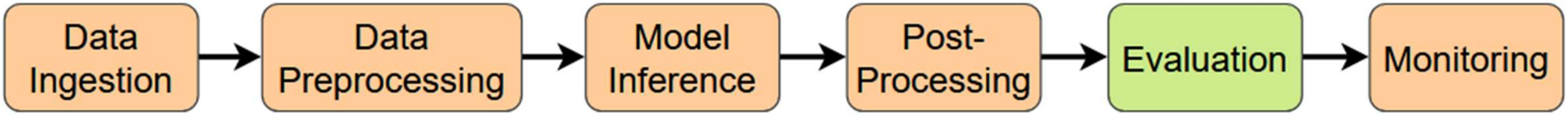
- Measuring correctness of model output
- Receiving human feedback
  - RL/HF
  - Make model better based on human review
- Handling incorrectness
- Identifying sources of incorrectness



Components of an AI Workflow – Evaluation

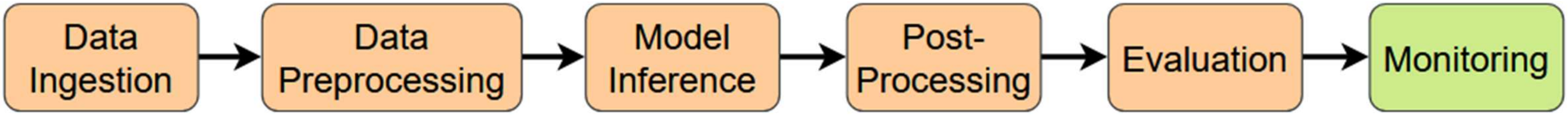
Common Oversights or Problems

- Continuous evaluation often overlooked
- Miss out on opportunities to improve model



Components of an AI Workflow – Monitoring

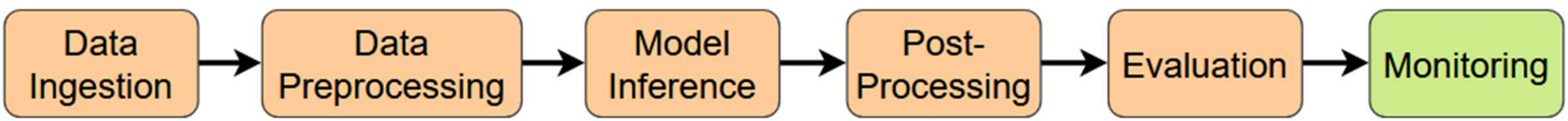
- Ensuring uptime
- Identifying any pipeline errors
- Identifying chokepoints



Components of an AI Workflow – Monitoring

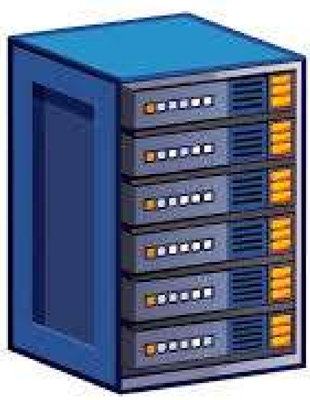
Common Oversights or Problems

- Ensure balance between each phase of the pipeline
  - Track what is useful
  - Don't over-monitor
    - Can stall pieces
    - Can drastically increase storage requirements



Deployment Strategies

Local Deployments vs. Cloud Deployments vs API Services



Deployment Strategies – Local Deployments

Host models on your hardware

Requirements:

- Server with GPU(s)
- Enough VRAM for your model
- Inference server software
  - vLLM
  - SGLang
- Management of servers

Deployment Strategies – Cloud Deployments

Host models on Cloud Hardware

- AWS
- Google Cloud Platform
- Microsoft Azure
- NIH Cloud Lab
- And many more

Available as NAIRR resources!

Deployment Strategies – Cloud Deployments

https://submit-nairr.xras.org/resources

Resource Catalog

Filters

AI Capabilities

- [ ] AI tools and support
- [x] Model inference services
- [ ] Model training services (GPU)
- [ ] Model training services (non-GPU)
- [ ] Research Collaboration

Resource Type

- [ ] CPU Compute
- [x] Cloud
- [x] Commercial Cloud
- [ ] GPU Compute
- [ ] Innovative / Novel Compute
- [ ] Sensors / Instruments
- [ ] Service / Other

Resources

- Amazon Web Services
- CloudBank Classroom
- CloudBank Research
- Google Cloud Platform
- Microsoft Azure
- NIH Cloud Lab
- SambaCloud (API inference services, Llama, DeepSeek, Qwen, etc.)

Deployment Strategies – API Services

Access models through existing APIs

- OpenAI API (ChatGPT)
- Anthropic API (Claude)
- Google API (Gemini)

Available as NAIRR resources!

Deployment Strategies – API Services

https://submit-nairr.xras.org/resources

Resource Catalog

Filters

AI Capabilities

- [ ] AI tools and support
- [x] Model inference services
- [ ] Model training services (GPU)
- [ ] Model training services (non-GPU)
- [ ] Research Collaboration

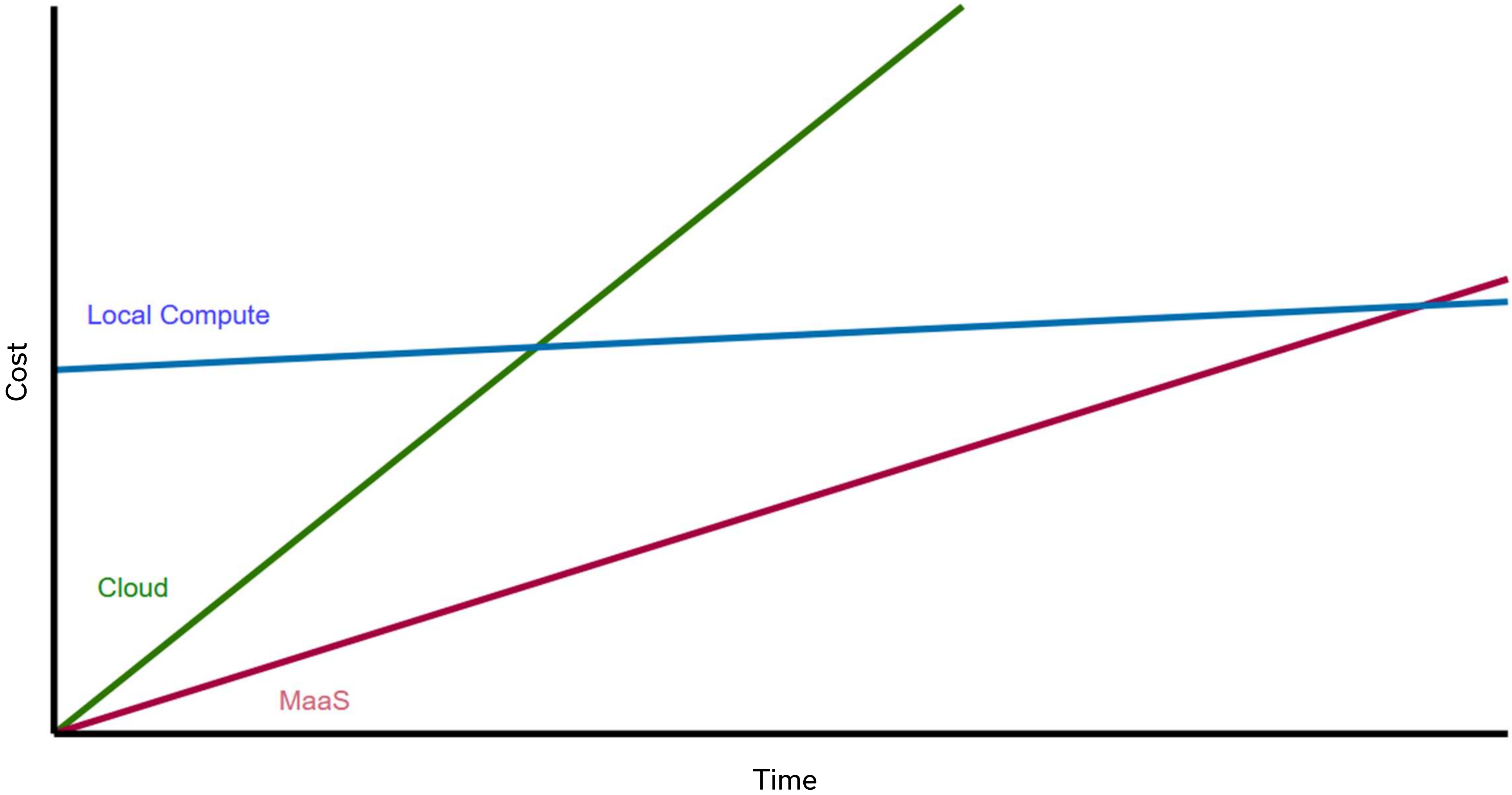
Resource Type

- [ ] CPU Compute
- [ ] Cloud
- [ ] Commercial Cloud
- [ ] GPU Compute
- [ ] Innovative / Novel Compute
- [ ] Sensors / Instruments
- [x] Service / Other

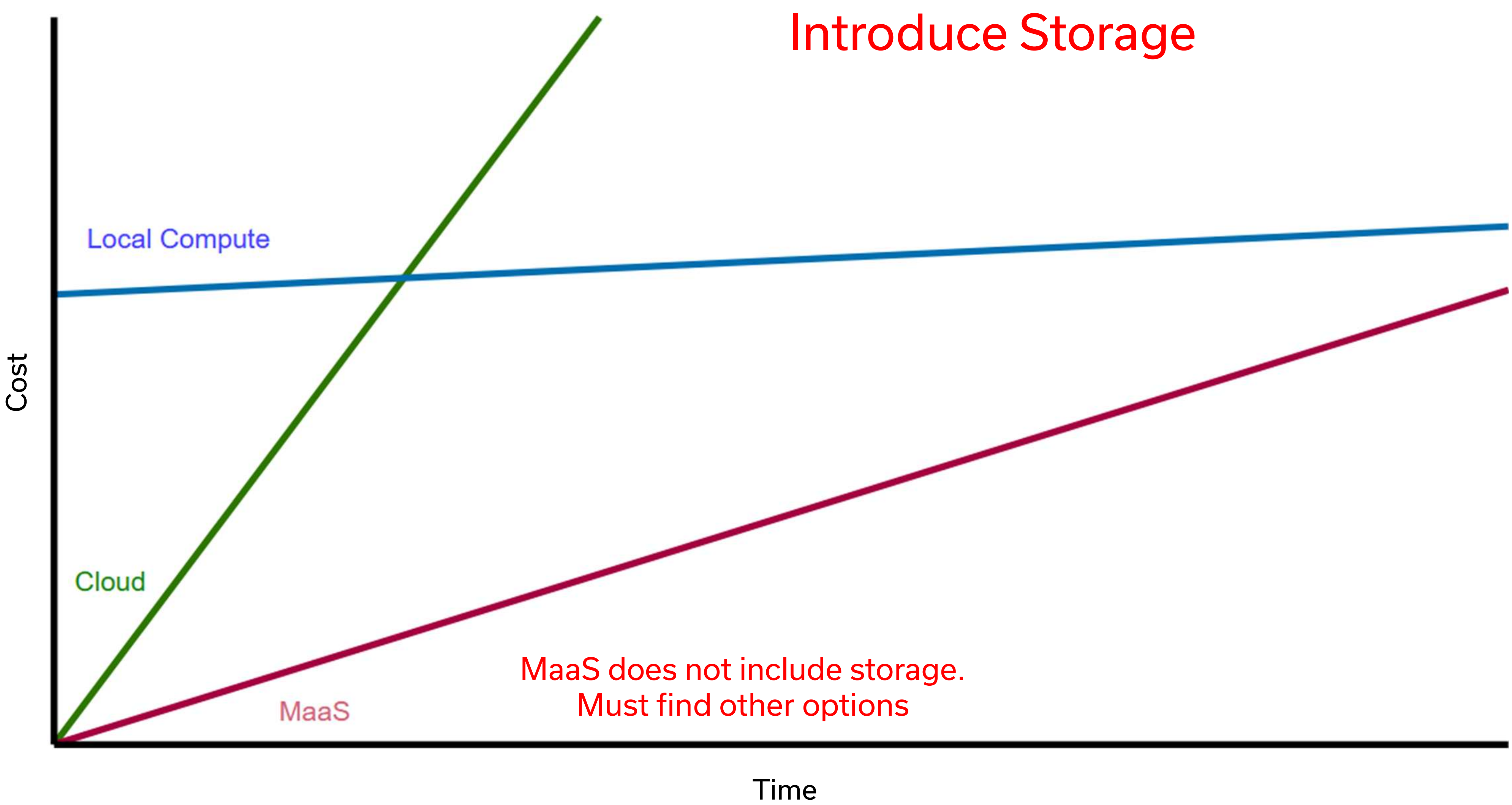
Resources

- Anthropic Model API
- OpenAI API
- SambaCloud (API inference services, Llama, DeepSeek, Qwen, etc.)

Deployment Strategies – Cost



Deployment Strategies – Cost



Applied AI Panel

CAAI Team

Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

NAIRR Pilot
KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM
CPE Higher Education Matters
University of Kentucky

Panel: Share Your Work

- How has AI impacted your work or research? (Good and Bad!)
- What parts of your work have you thought might be improved with AI?
- Have you incorporated AI solutions anywhere in your work?
  - Successes? Failures?

Conclusion

Caroline Leach
Program Manager, Democratizing AI Training & Access Statewide
Data Management Specialist, Center for Applied AI

Jefferson Community & Technical College
Louisville, Kentucky
January 23rd, 2026

NAIRR Pilot

KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM

CPE Higher Education Matters

University of Kentucky


THANK
YOU
JEFFERSON COMMUNITY AND
TECHNICAL COLLEGE
NAIRR Pilot
NSF
CPE
Higher Education Matters
JEFFERSON
COMMUNITY & TECHNICAL COLLEGE
UK University of Kentucky.

Workshop Goals

- Understand core AI concepts and how AI is used in real-world settings
- Explore how AI is shaping research and the workforce in Kentucky
- Gain confidence in foundational ideas like machine learning, large language models, and multimodal AI
- Experience applications of AI through the lens of research workflows
- Learn how to access local and national AI computing resources through the NAIRR Pilot
- Spark new ideas and collaborations across AI and related research fields
- Connect Kentucky researchers and innovators to grow a shared AI ecosystem

Please fill out our Post-Workshop Survey:

NAIRR Pilot | KENTUCKY COMMUNITY & TECHNICAL COLLEGE SYSTEM | CPE Higher Education Matters | University of Kentucky

Post Event Survey

- Let us know how we did today
- Help us in our reporting
- Your responses will be used to shape future events

Don't Forget!

- Resources are available to you after this event
  - Office Hours
  - Online Forum
  - YouTube
  - Demos on GitHub
  - NAIRR Virtual Assistant Chatbot – *Coming Soon!*
- NAIRR Pilot allocation requests are open!

- nairrpilot.org/
- caai.ai.uky.edu/about-us/news-and-events/nairr-pilot-expansion-project/

Thank you for attending!

Scan to access the event webpage:

Scan to access the survey:

ai_workshops@uky.edu

NAIRR Pilot | Kentucky Community & Technical College System | CPE Higher Education Matters | University of Kentucky