Carnegie Mellon University
Software Engineering Institute

Data Science for Cybersecurity Workshop

INFOSEC WORLD

INFOSECWORLDUSA.COM

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Thomas Scanlon

Technical Manager – CERT Data Science

Software Engineering Institute

Carnegie Mellon University

Carnegie Mellon University (CMU)

Pioneering discoveries that enrich the lives of people on a global scale

- Turning disruptive ideas into success through leading-edge research
- 2021 *U.S. News and World Report* rankings
  - #1 in computer engineering, artificial intelligence (AI), cybersecurity, and software engineering
  - #2 in overall computer science
  - #3 in data analytics/science

CMU Software Engineering Institute (SEI)

Carnegie Mellon University
Software Engineering Institute

Bringing innovation to the U.S. Government

- A Federally Funded Research and Development Center (FFRDC) chartered in 1984 and sponsored by the Department of Defense (DoD)
- Leader in researching complex software engineering, cyber security, and AI engineering solutions
- Critical to the U.S. Government's ability to acquire, develop, operate, and sustain software systems that are innovative, affordable, trustworthy, and enduring

The CERT Division: The Birthplace of Cybersecurity

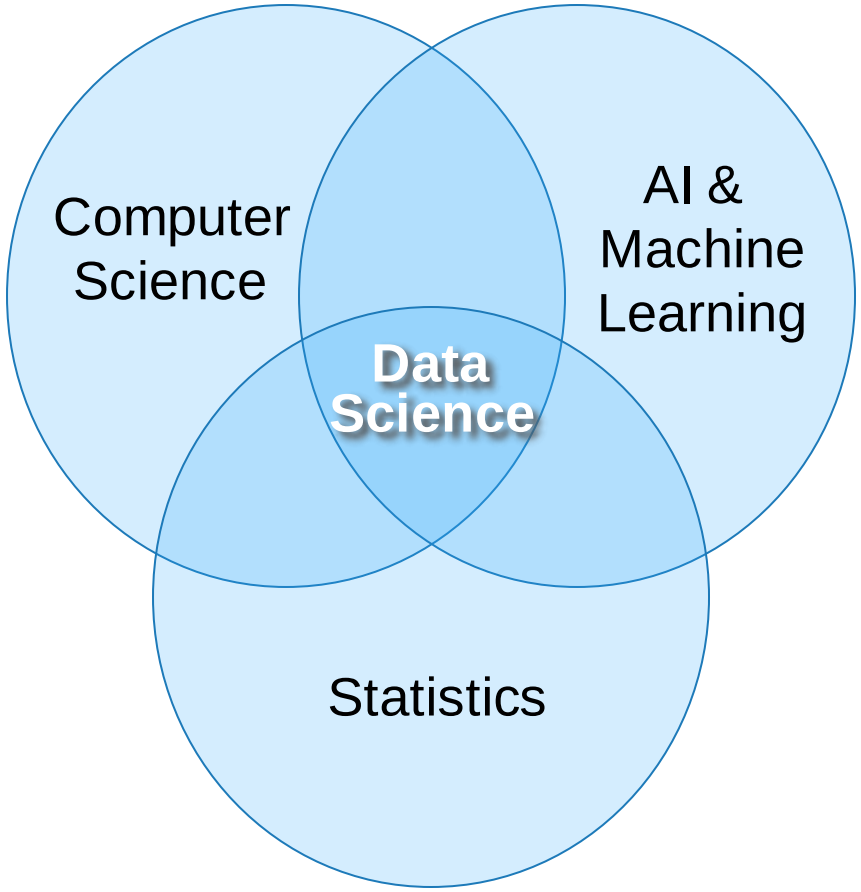
- **Trusted**
  Conducting research for the U.S. Government in a non-profit, public-private partnership
- **Valued**
  Collaborating with military, industry, and academia globally to innovate solutions
- **Relevant**
  Achieving technology and talent results for our mission partners

Data Science Overview

Data Science

XKCD Machine Learning: https://xkcd.com/1838/

Data Science



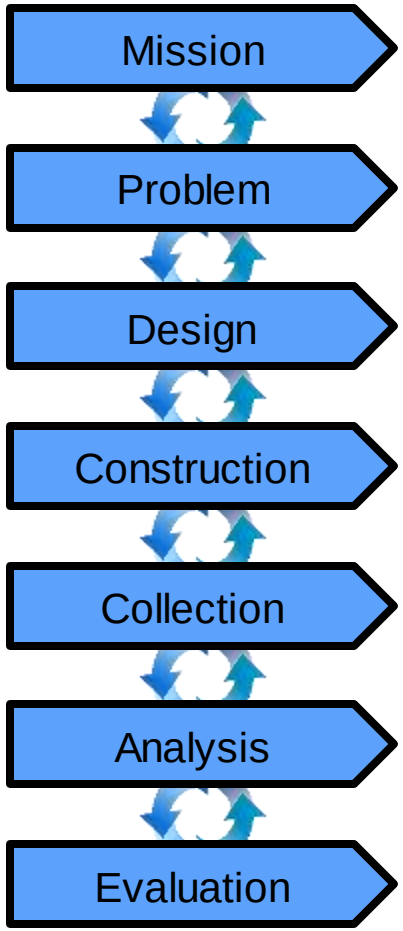
Analysis Techniques

- Prediction
- Classification
- Deep Learning
- “Big Data”
- Outlier detection
- Feature extraction

Methods:

- Regression
- Neural nets
- Bayesian networks
- Structural equation modeling
- Latent Dirichlet allocation
- Hidden Markov models
- Gradient boosting
- …

Data Analytics



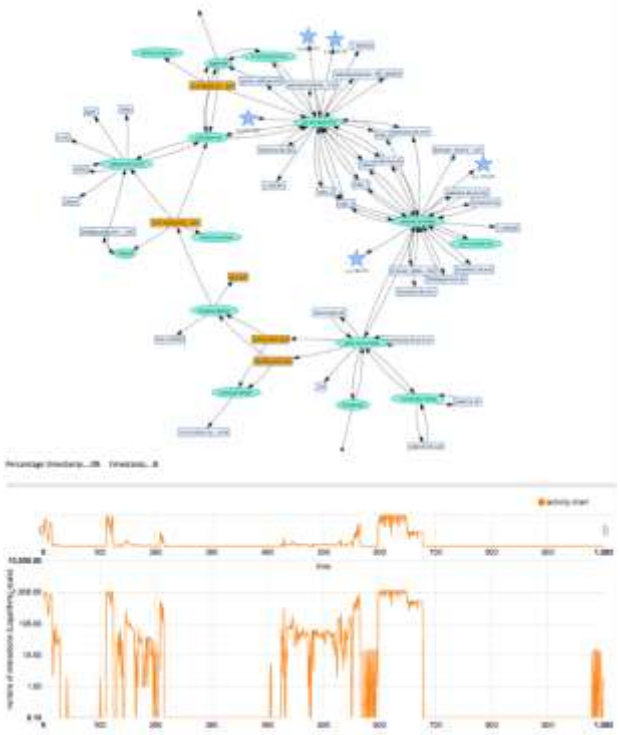
Data Types

Image	Static Video
Time series	Financial data Event counts
Network data	Netflow PCAP DNS BGP
Structured text	Web forms Structured data (JSON, XML) Source code
Free text	News Tweets Email
many more…	

Finding Features in High-Dimensional Data

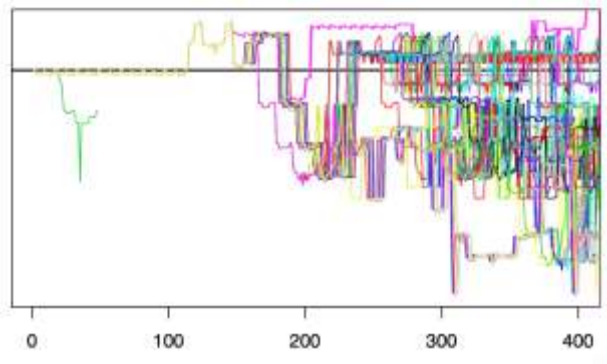
- Dimensionality Reduction
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)
  - Classifier-adjusted density estimation (CADE)
  - Principal component analysis (PCA)
    - Kernel
    - Graph-based
  - Linear discriminant analysis (LDA)
  - Generalized discriminant analysis (GDA)

Malware family classification

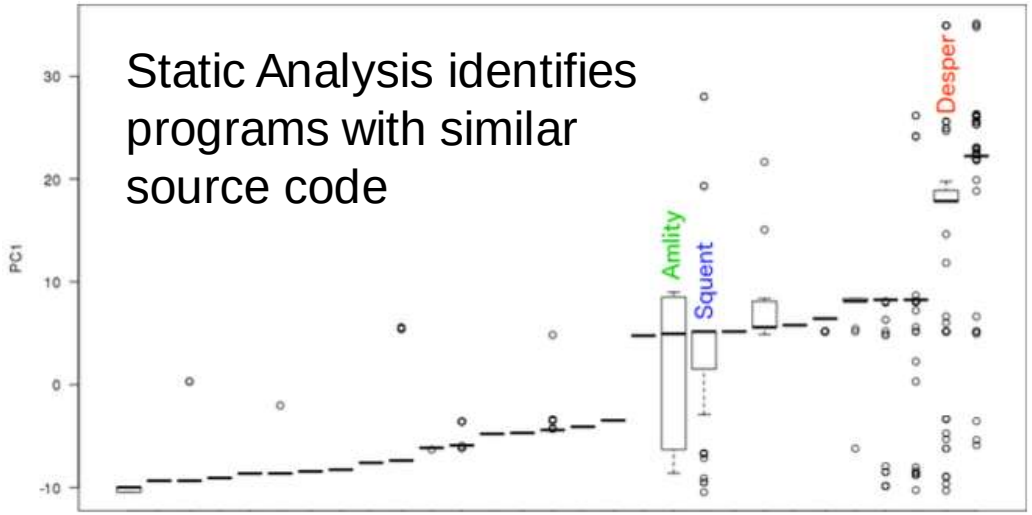


Signal Flow graph highlights behavior relating different malware families

Program instruction analysis shows similarity and diversion of behavior

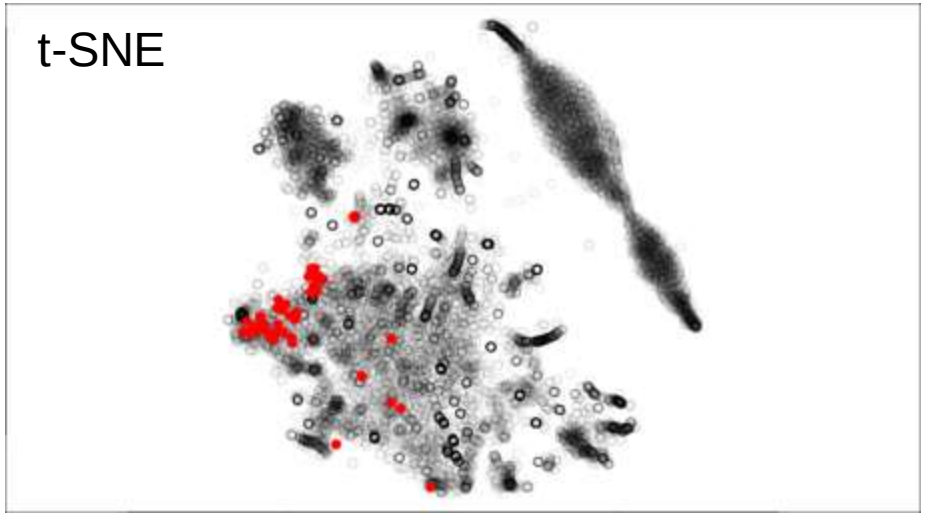


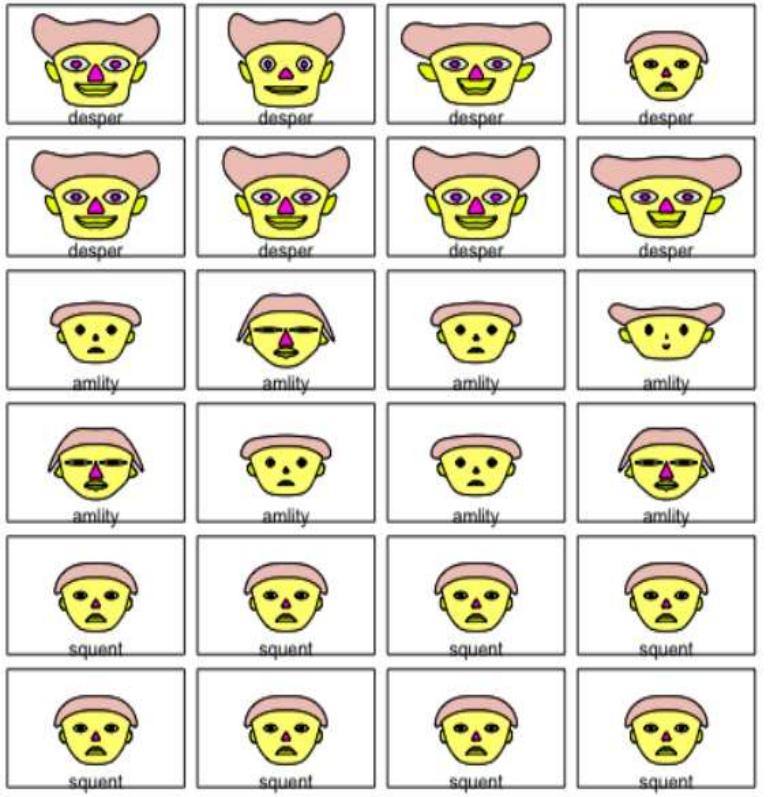
Static Analysis identifies programs with similar source code



Malware family classification – Visualization

Simplify visualization of extremely complex data through the use of dimensionality reduction and associated visualization techniques

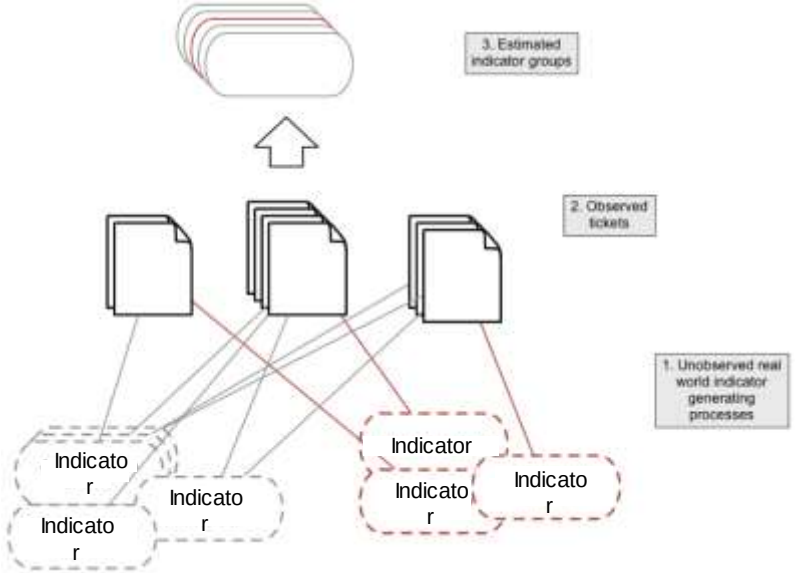




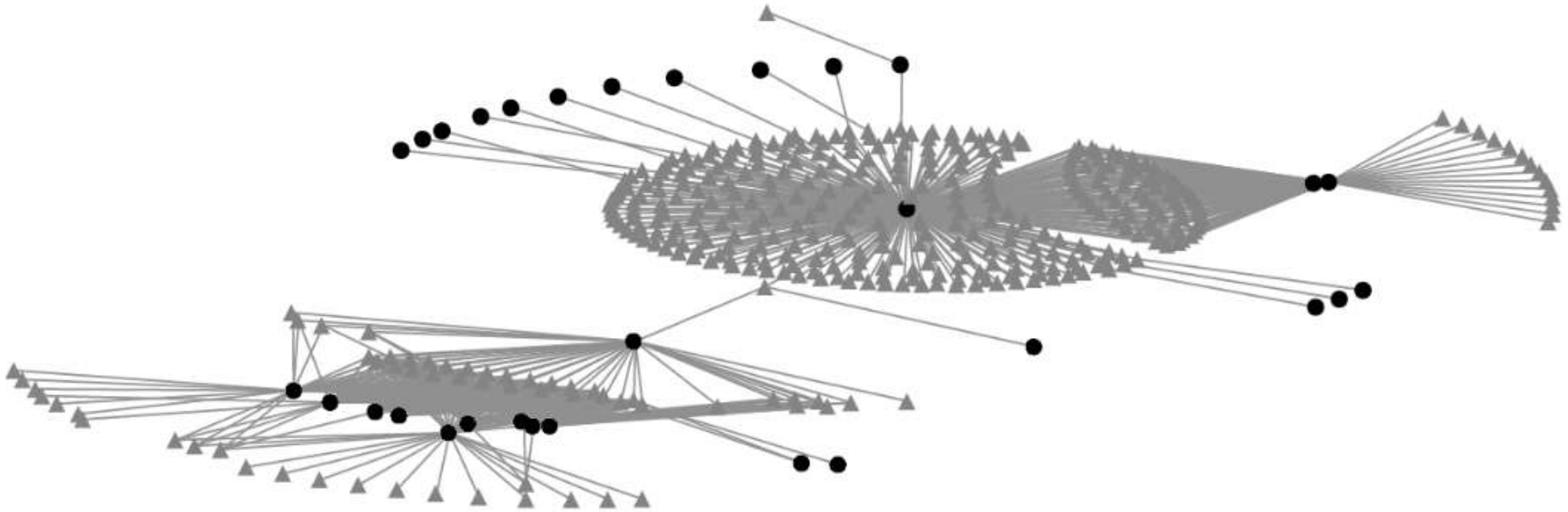
Chernoff face experiment

Incident Ticket analysis

Collections of incident tickets often contain hidden trends, revealing attacker methods and techniques that are invisible in individual tickets. Natural Language Processing helps find these trends.



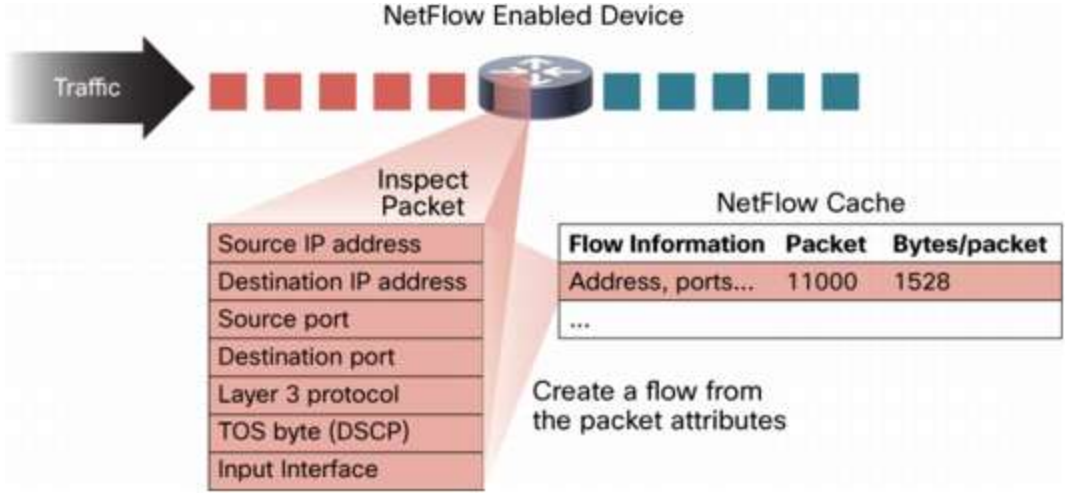
Incident Ticket analysis – Visualization



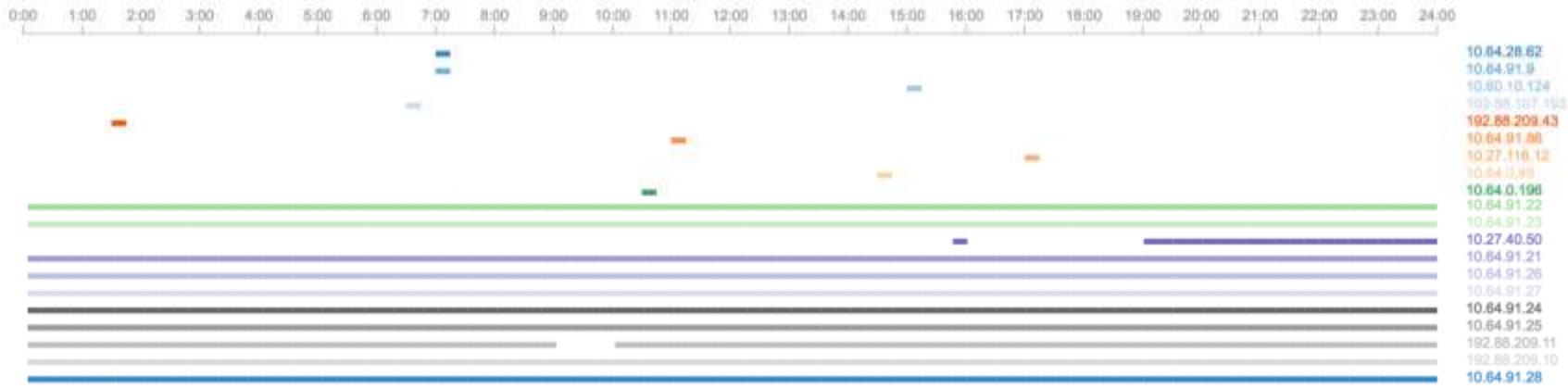
A subset of the ticket-indicator graph
(for a small set of selected indicators)

- Tickets are grey triangles
- Indicators are black circles
- Edges connect tickets to the indicators they contain

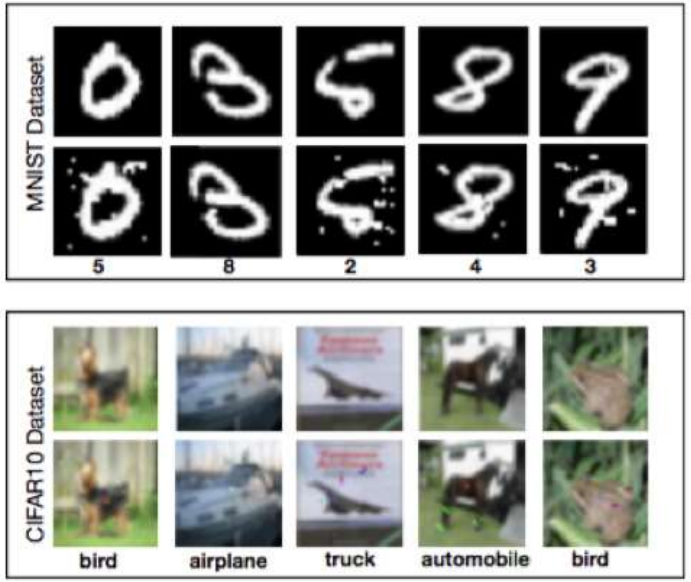
Network traffic classification



Netflow data summarizes network traffic, losing significant information. Numerous techniques exist to infer information from the flows themselves.



Protecting against AI Poisoning



Accepted to the 37th IEEE Symposium on Security & Privacy, IEEE 2016. San Jose, CA.

Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks

Nicolas Papernot*, Patrick McDaniel*, Xi Wu§, Somesh Jha§, and Ananthram Swami†
*Department of Computer Science and Engineering, Penn State University
§Computer Sciences Department, University of Wisconsin-Madison
†United States Army Research Laboratory, Adelphi, Maryland
{ngp5056,mcdaniel}@cse.psu.edu, {xiwu,jha}@cs.wisc.edu, ananthram.swami.civ@mail.mil

***Abstract*—Deep learning algorithms have been shown to perform extremely well on many classical machine learning problems. However, recent studies have shown that deep learning, like other machine learning techniques, is vulnerable to adversarial samples: inputs crafted to force a deep neural network (DNN) to provide adversary-selected outputs. Such attacks can seriously undermine the security of the system supported by the DNN, sometimes with devastating consequences. For example, autonomous vehicles can be crashed, illicit or illegal content can bypass content filters, or biometric authentication systems can be manipulated to allow improper access. In this work, we introduce a defensive mechanism called *defensive distillation* to reduce the effectiveness of adversarial samples on DNNs. We analytically investigate the generalizability and robustness properties granted by the use of defensive distillation when training DNNs. We also empirically study the effectiveness of our defense mechanisms on two DNNs placed in adversarial settings. The study shows that defensive distillation can reduce effectiveness of sample creation from 95% to less than 0.5% on a studied DNN. Such dramatic gains can be explained by the fact that distillation leads gradients used in adversarial sample creation to be reduced by a factor of 10^{30}. We also find that distillation increases the average minimum number of features that need to be modified to create adversarial samples by about 800% on one of the DNNs we tested.**

I. Introduction

Deep Learning (DL) has been demonstrated to perform exceptionally well on several categories of machine learning problems, notably input classification. These *Deep Neural Networks* (DNNs) efficiently learn highly accurate models from a large corpus of training samples, and thereafter classify unseen samples with great accuracy. As a result, DNNs are used in many settings [1], [2], [3], some of which are increasingly security-sensitive [4], [5], [6]. By using deep learning algorithms, designers of these systems make implicit security assumptions about deep neural networks. However, recent work in the machine learning and security communities have shown that adversaries can force many machine learning models, including DNNs, to produce adversary-selected outputs using carefully crafted inputs [7], [8], [9].

Specifically, adversaries can craft particular inputs, named *adversarial samples*, leading models to produce an output behavior of their choice, such as misclassification. Inputs are crafted by adding a carefully chosen adversarial perturbation to occur after training is complete and therefore do not require any tampering with the training procedure.

To illustrate how adversarial samples make a system based on DNNs vulnerable, consider the following input samples:

The left image is correctly classified by a trained DNN as a car. The right image was crafted by an adversarial sample algorithm (in [7]) from the correct left image. The altered image is incorrectly classified as a cat by the DNN. To see why such misclassification is dangerous, consider deep learning as it is commonly used in autonomous (driverless) cars [10]. Systems based on DNNs are used to recognize signs or other vehicles on the road [11]. If perturbing the input of such systems, by slightly altering the car's body for instance, prevents DNNs from classifying it as a moving vehicule correctly, the car might not stop and eventually be involved in an accident, with potentially disastrous consequences. The threat is real where an adversary can profit from evading detection or having their input misclassified. Such attacks commonly occur today in non-DL classification systems [12], [13], [14], [15], [16].

Thus, adversarial samples must be taken into account when designing security sensitive systems incorporating DNNs. Unfortunately, there are very few effective countermeasures available today. Previous work considered the problem of constructing such defenses but solutions proposed are deficient in that they require making modifications to the DNN architecture or only partially prevent adversarial samples from being effective [9], [17] (see Section VII).

Distillation is a training procedure initially designed to train a DNN using knowledge transferred from a different DNN. The intuition was suggested in [18] while distillation itself was formally introduced in [19]. The motivation behind the knowledge transfer operated by distillation is to reduce the computational complexity of DNN architectures by transferring knowledge from larger architectures to smaller ones. This facilitates the deployment of deep learning in resource constrained devices (e.g. smartphones) which cannot rely on

Other Applications

- APT Defense
- Relation of kinetic and cyber actions
- Automated forecasting and detection of cyber-attacks
- Static code analyzer behavior
- Technical debt estimation
- Cognitive support for assurance using Watson
- Optimizing planning budgets for travel
- Email sentiment analysis
- IoT based search-and-rescue

Data Science Do's and Don't's

Do

- Have a question in mind
- Utilize your subject experts
- Think of what data you **need** vs. **have**
- Interrogate your data
- Document all collection, cleaning, and transformation steps
- Justify models used
- Interrogate your model(s)
- Be ready for 'negative' results

Don't

- Force your data to fit your hypothesis
- Forsake model interpretability to do a 'machine learning / AI model'
- Overfit
- Overinterpret

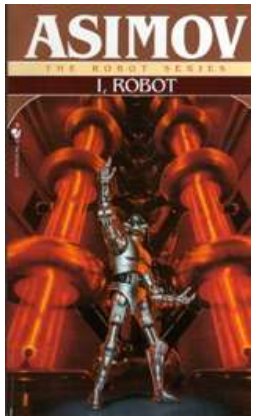
AI/ML Overview

When People Say "AI," They Might be Referring to Either a Narrow AI or a General AI

- **Narrow AI (Hard)**
- An algorithm to carry out one particular task.

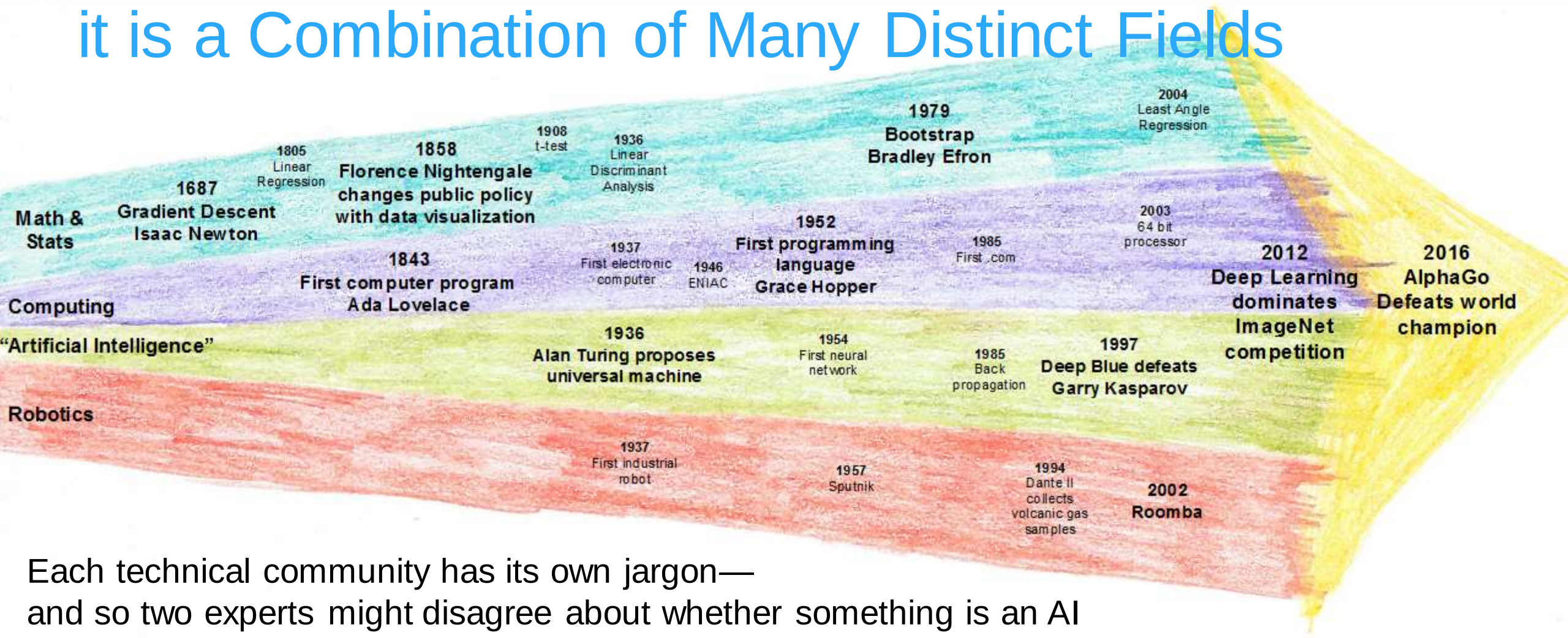
- *Examples:*
- Google Translate
- Autonomous Vehicles
- Spam Filters

- **General AI (Soft)**
- A machine that exhibits human intelligence.



- *Examples:*
- Doesn't exist yet.
- e.g., one goal should be to be able to monitor the understanding of its audience and make adjustments. This is something human children can generally do by the age of 5.

Modern Artificial Intelligence is not a Monolith, it is a Combination of Many Distinct Fields



Each technical community has its own jargon—
and so two experts might disagree about whether something is an AI

Example of Language Confusion

- **"I want to use _____ to build a speech recognition algorithm."**
- Different people would fill in the blank differently, and mean the same thing:

"Artificial Intelligence" – In this example, probably refers to using ML to set up a narrow AI

"Machine Learning" – A large set of methods for extracting information from data, including neural networks

"Neural Network" – one type of machine learning algorithm, originally patterned after how humans were thought to think

"Deep Learning" – A strategy for building neural networks that are easy to write down and can model complex behavior

Recommendation: When somebody says AI, ask them to be more specific and define what they mean.

Related Definitions: Statistics and Data Science

- **Statistics** is the art and science of learning from data.
- ***Data science*** refers to managing and analyzing large amounts of data.

Statistics + Data Management

ML and AI are not the Same Thing, but There is Overlap

- These could all be considered narrow AI, but do not use any ML:
  - Navigation (Google Maps) – Calculates optimum path based on an algorithm
  - Roomba – uses sensors to map the room, then follows a pre-programmed algorithm to most efficiently cover it
  - Non-Player Characters (NPCs) in video games – dialog and interactions between player characters and NPCs are pre-programmed and constrained for game play.

- ***ML is currently one of the best ways to create a successful narrow AI from data:***
  - Image recognition (used in autonomous vehicles)
  - Speech recognition (used in Alexa, Siri, and Echo)
  - Recommendation engines (used by Netflix and Amazon)

Machine Learning

If you ask a **computer scientist** to define **Machine Learning** you might get:

The science of getting computers to act without being explicitly programmed, by extracting information from data.

Statistics + Automation

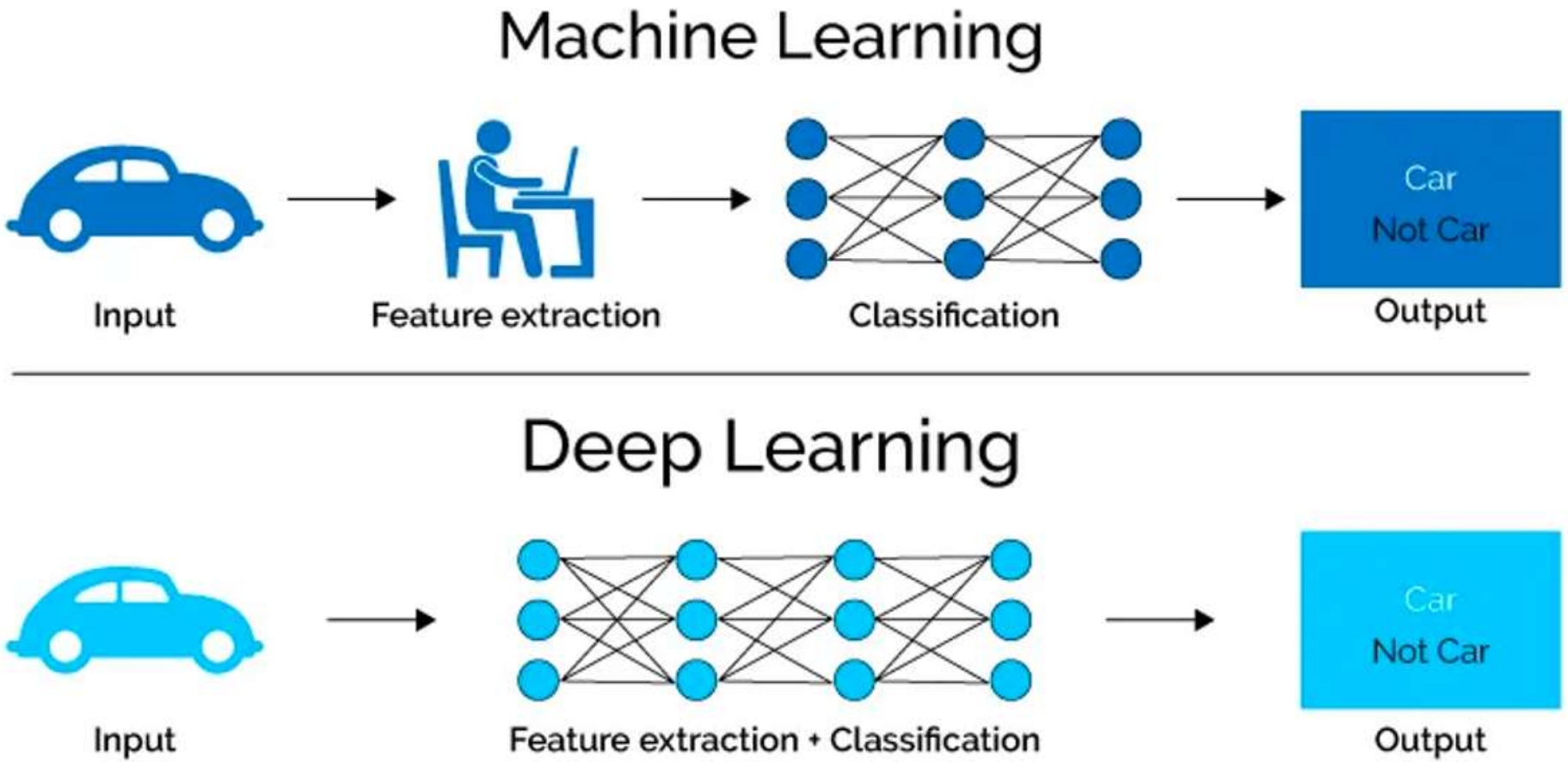
If you ask a **statistician** to define **Machine Learning**, you might get:

A set of models and algorithms for extracting information from data, developed primarily after 1980 that require intense computational power.

Statistics + Computing Power

Implication: Definitions matter. Different things fit in these two definitions.

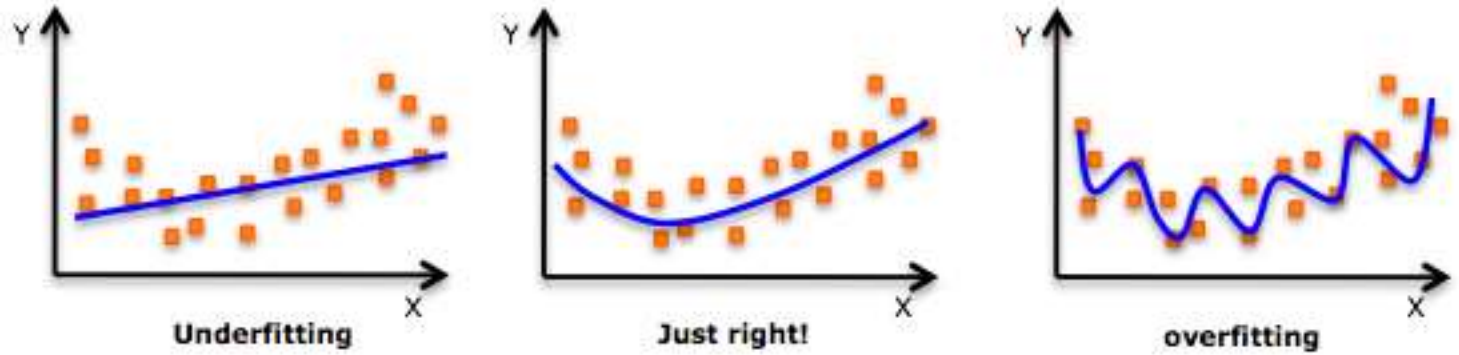
Machine Learning & Deep Learning



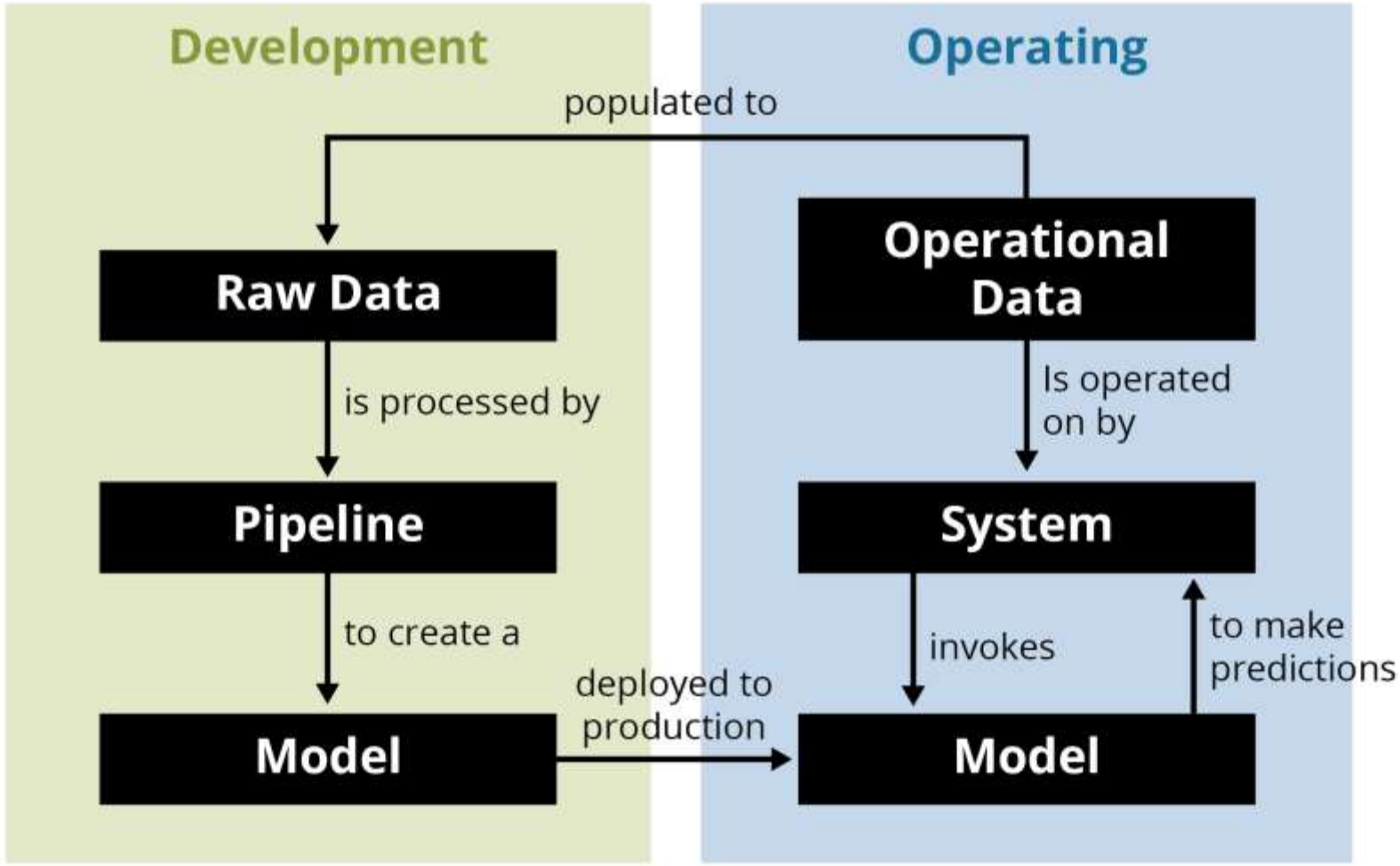
Deep learning is machine learning using a neural network.

https://semiengineering.com/deep-learning-spreads/

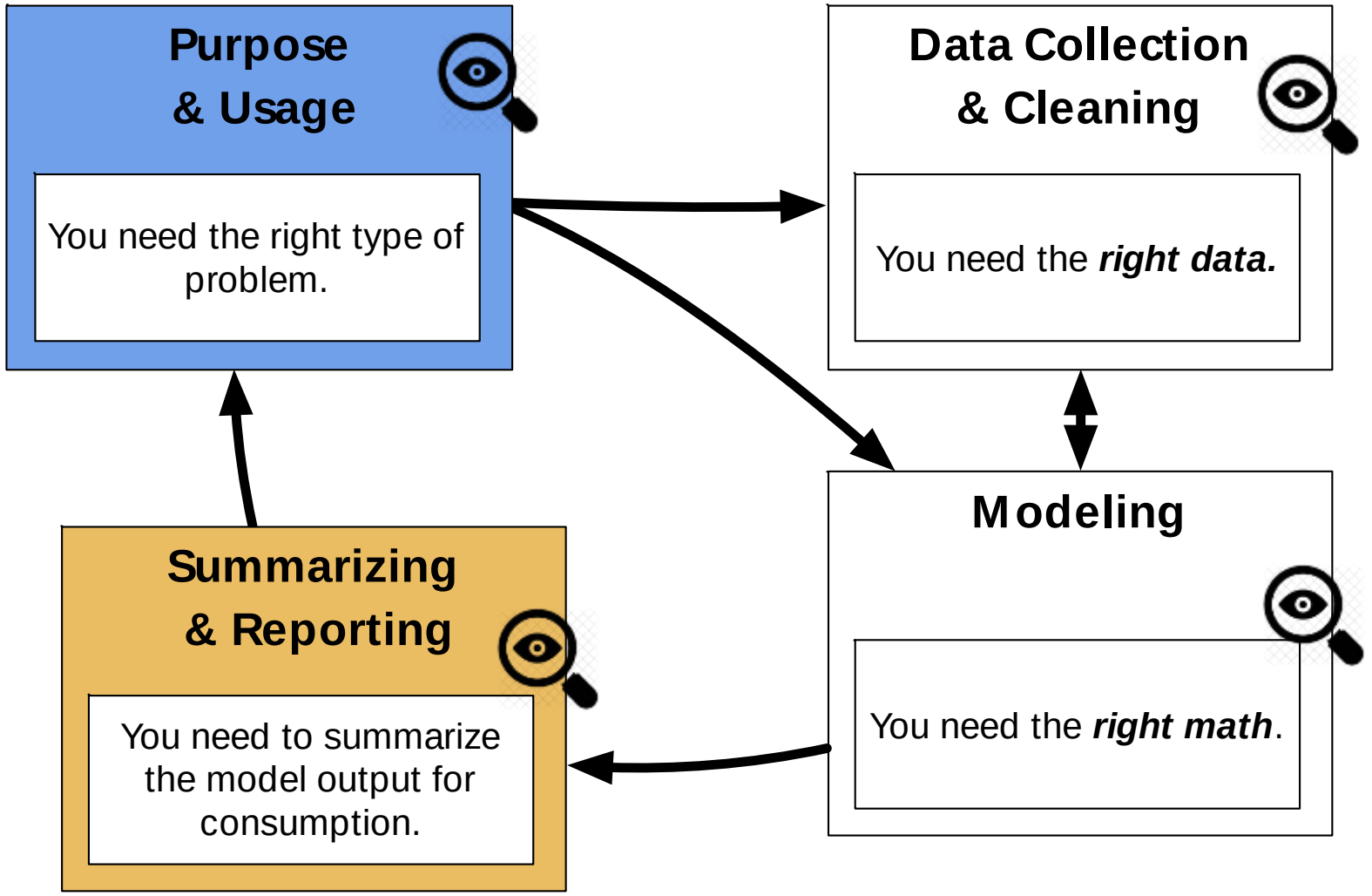
Underfitting and Overfitting



Notional ML Pipeline



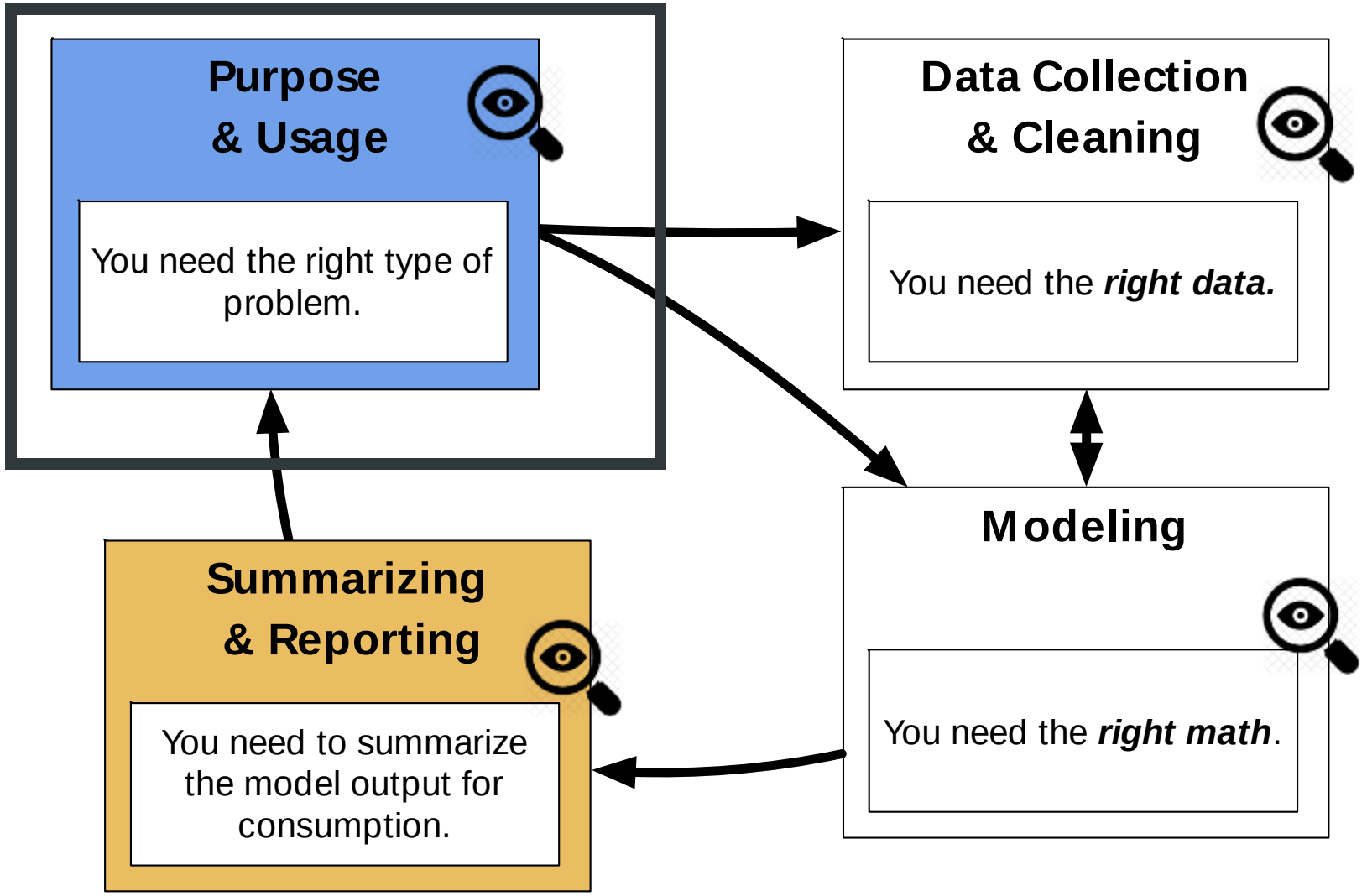
Process for an ML/Data Science Project



You need to monitor each part of the process.

For most projects, data collection and cleaning require a majority of the time and effort.

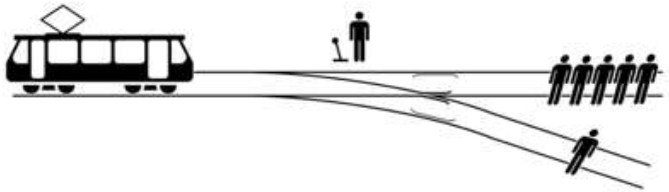
Purpose and Usage



You need to monitor each part of the process.

Not all Problems can be Solved with AI (or ML)

- AI cannot solve the trolley problem.
- AI can help optimize a chosen criteria.
- It cannot tell you what criteria to optimize.



You have two options:

1. Do nothing and allow the trolley to kill the five people on the main track.
2. Pull the lever, diverting the trolley onto the side track where it will kill one person.

Moreover: To implement an automatic system (like a self-driving car), we must make these choices ahead of time.

https://www.technologyreview.com/s/612341/a-global-ethics-study-aims-to-help-ai-solve-the-self-driving-trolley-problem/

Translating a Real World Problem into Something Tractable is not Always Straightforward

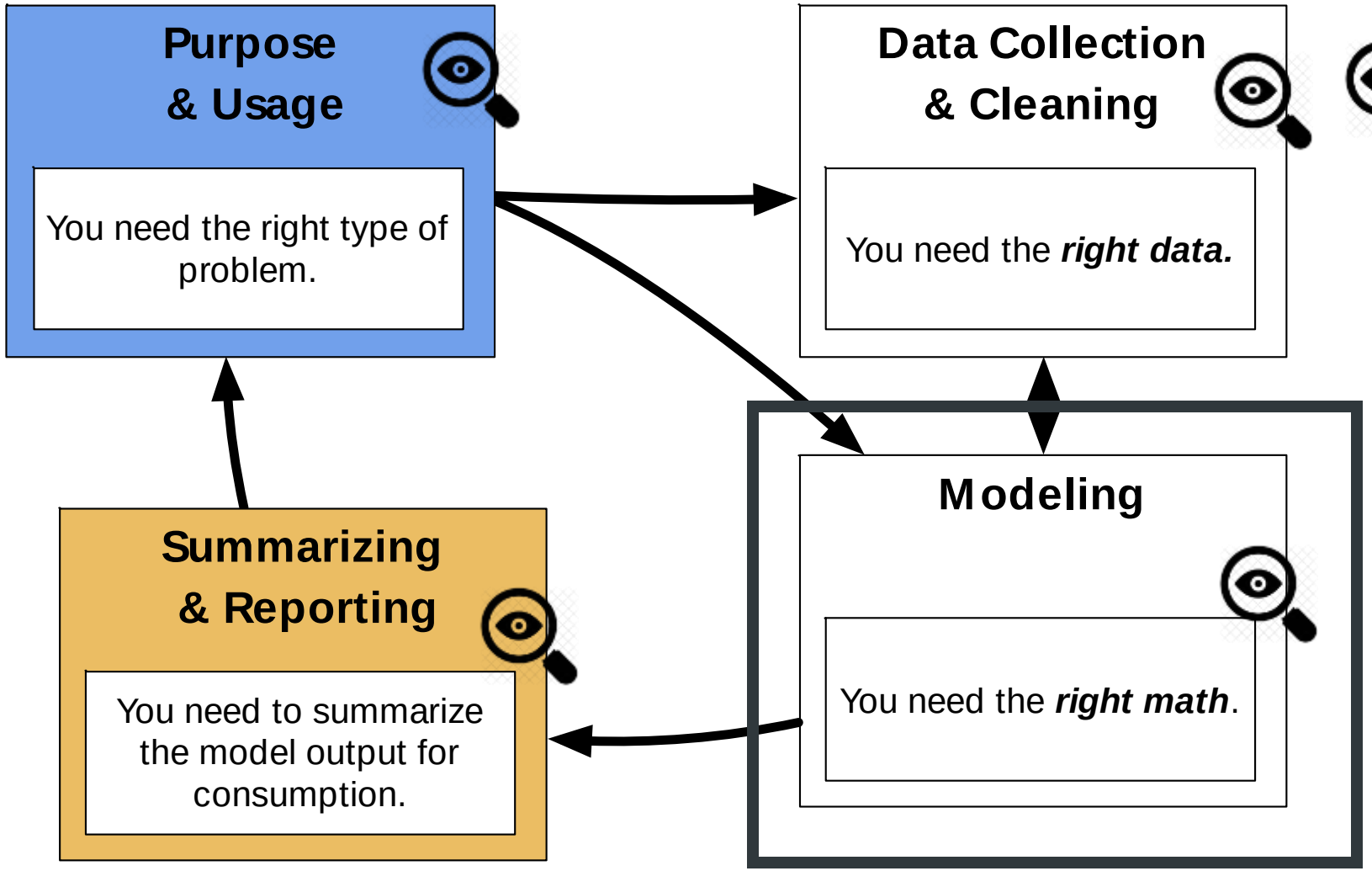
- Example: What does it mean to “jump”?
- Researchers were trying to have an AI design creatures that were optimized for “jumping”

When they defined jumping as “maximum elevation reached by the center of gravity of the creature during the test,” They got tall skinny creatures with enormous heads (top).

When they defined jumping as "furthest distance from the ground of the block that was originally closest to the ground” they got tall skinny creatures that flipped over (bottom).

arXiv:1803.03453

Modeling

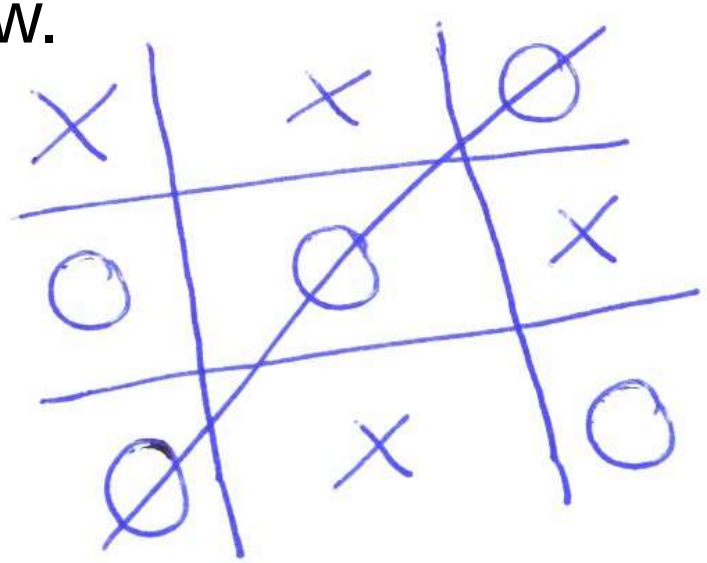


There are Different Types of ML

	Supervised Machine Learning	Unsupervised Machine Learning	Reinforcement Learning
Useful for	Making A->B predictions	Discovering previously unknown patterns in data	Optimization in complex, but constrained tasks
Example Uses	Determine whether an image contains a ship. Determine whether a set of financial documents indicate fraud. From a baseball player's prior performance, predict performance in the next game.	Discover customer profiles Identify clusters of malware Identify anomalous network activity	Optimizing logistics chain management Optimizing strategy in a game
Common Methods	**Regression** (Linear, Regression Trees, Kernel Regression, …) **Classification** (Support Vector Machines, Logistic Regression, Discriminant Analysis) Neural Networks, Ensemble Methods…	**Clustering** (K-means, DBSCAN, Mixture modeling) Association Rule Mining Anomaly Detection Neural Networks	Q-Learning Policy Optimization State-Action-Reward-State-Action Deep Deterministic Policy Gradient
Notes	By far the most common	Data widely available, implementation and verification are tricky	Only beginning to move into commercial space, still largely academic.

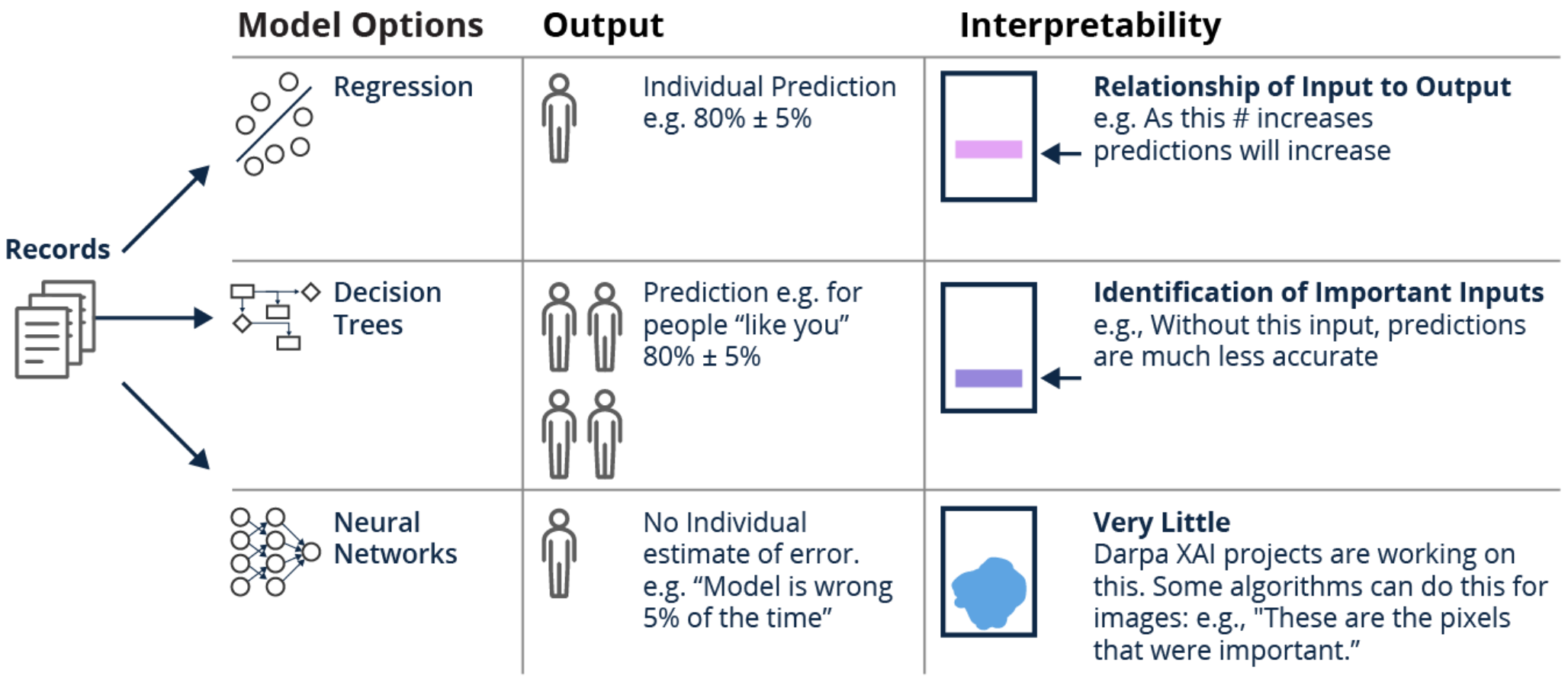
If You use the Wrong Math, You can get Pure Nonsense

- An algorithm that always takes the “best” move on the next turn will never lose at ***tic-tac-toe.*** It will either win or draw.



- An algorithm that always takes the next “best” move in ***chess***, will always fall into any trap set by its opponent. In order to see the trap and avoid it, the algorithm must be able to consider more than one move ahea

There are Usually Several Options of "Right Math," but Only Some of Them will give you the Summary You Need



Lots of Performance Metrics can seem Similar, Even if They're Quite Different

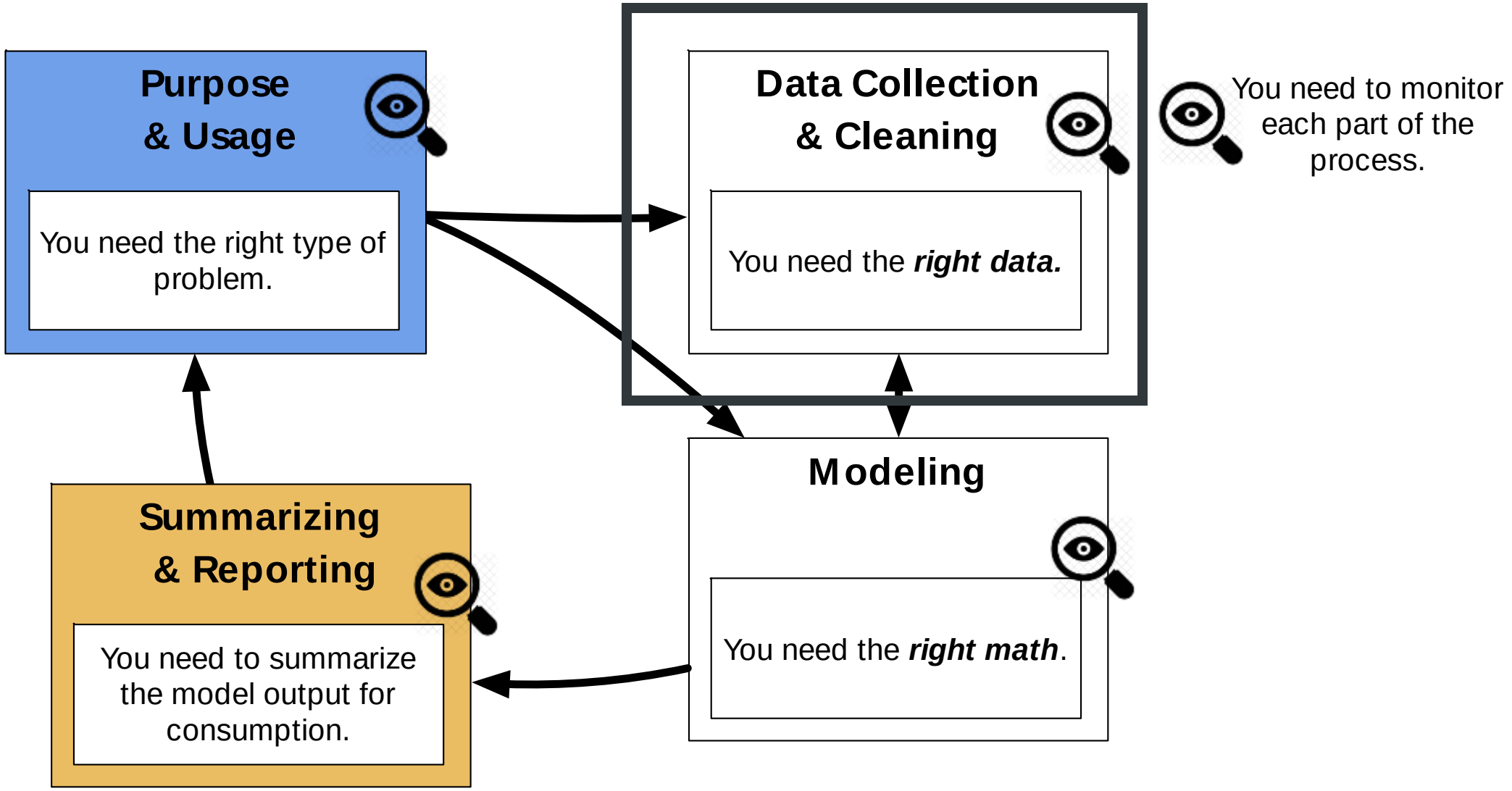
Different kinds of errors may have different kinds of implications.

Metric	What it measures
Error Rate	How often is the algorithm wrong?
False Positive	Algorithm predicted "yes," But the truth is "no"
False Negative	Algorithm predicted "no," But the truth is "yes"
Positive Predictive Value	Of the "yes's" that were predicted, How many are actually "yes"

The metrics you choose for an ML project are a policy statement about what kind of systematic errors are acceptable, and which should be minimized.

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Data Collection and Cleaning



Your Data has to Contain the Right Information, Otherwise, It Doesn't Matter how Much You Have

- If you want to be able to input a pile of financial documents and detect fraud, that's a supervised learning problem.
- You need to train your model on financial documents that have been labeled
-  contains fraud  no identified fraud
- Without this labeled data you cannot do ***this*** analysis.
- (There are other analyses that can still move you towards that goal, but they're more complex and less certain to produce useful results.)

Data Collection Must Reflect Usage Conditions

- ML algorithms learn what's in the data.
- ***So, "what's in the data" must match the conditions where the results will be used.***
- Train (your model) the way you fight.

• Google Flu Trends

- Designed to predict severity of flu outbreak, during real time.
- ***In 2008***, it worked very well, correctly predicting the CDC results weeks earlier.
- ***In 2013***, the predictions were off by 140%.
- The application conditions changed over time, reducing the accuracy of predictions.
- Project was discontinued.

https://www.wired.com/2015/10/can-learn-epic-failure-google-flu-trends/

• Dog/Wolf Images

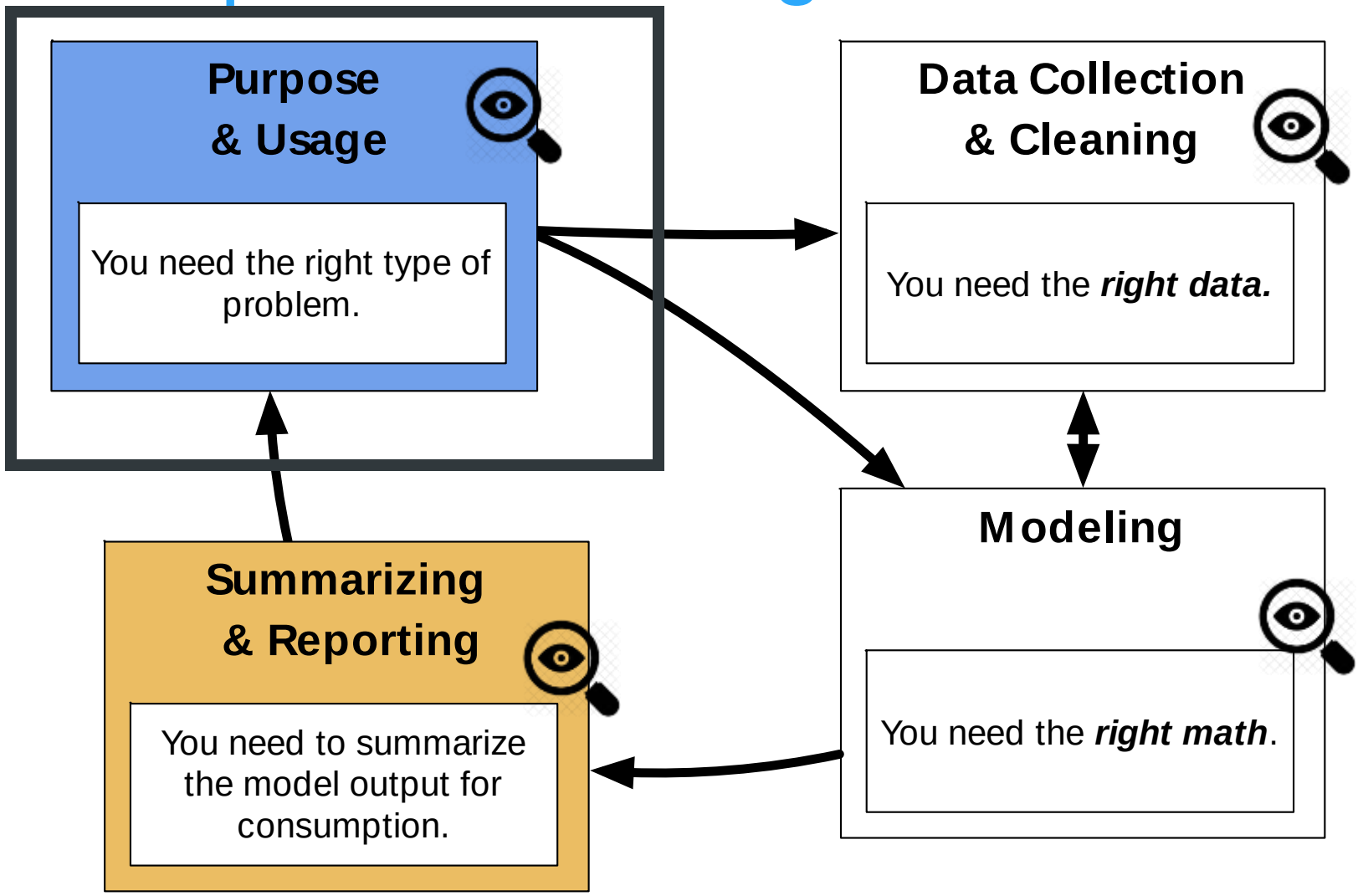
(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

- This model learned to distinguish between wolves and dogs by looking at the background, because a snow background indicated "wolf."
- Implications for training data in Project Maven.

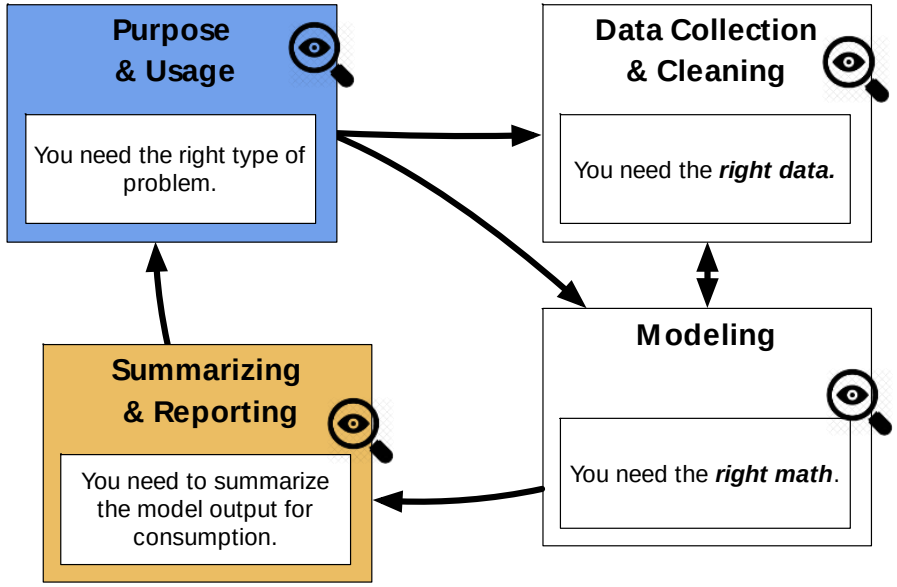
https://arxiv.org/pdf/1602.04938.pdf

Purpose and Usage



You need to monitor each part of the process.

If Your Use-Case Changes…



- You definitely need a new model.
- Because the math must match the use.

- You very likely need new data.
- Because the data must match the use.

Possible mitigations:

- Get end user involved from the beginning.
- Do feasibility studies.
- Collect extra data.

Usability, Interpretability and Explainability

Cautionary Tale:

Flint, Michigan developed an ML system to predict which homes had lead pipes that needed to be replaced.

Overall, performance was reasonably good, with about 80% accuracy.

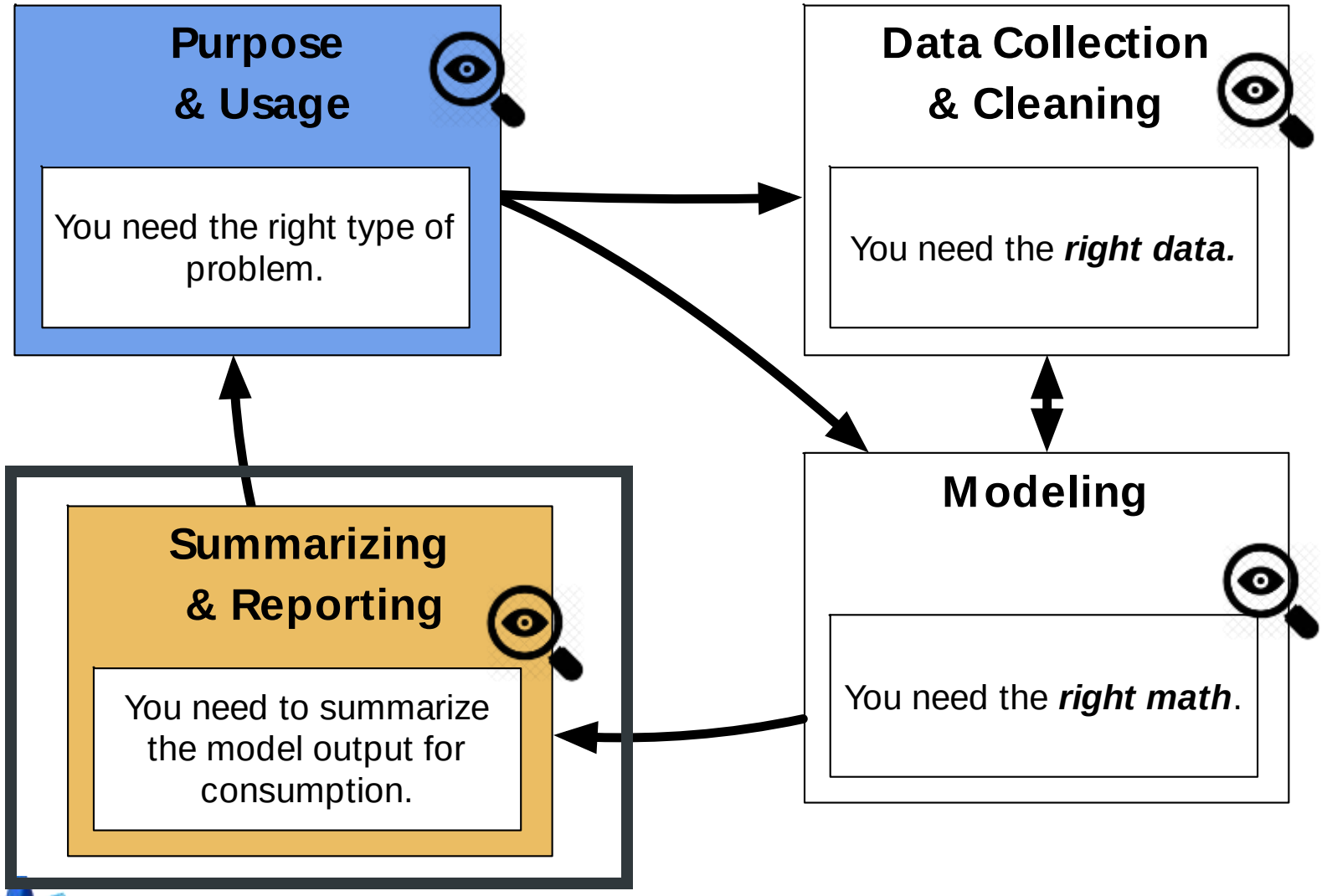
The city scrapped the system. One of the primary reasons was that they could not convey to the public how the algorithm worked, and why some houses were prioritized over others.

"When we started this, people would say, 'You did my neighbor's house and you didn't do mine,'" [Mayor] Weaver said.

- **Recommendation: Usability, interpretability and explainability must be designed into an ML/AI system. It cannot be added as an afterthought.**

https://www.theatlantic.com/technology/archive/2019/01/how-machine-learning-found-flints-lead-pipes/578692/

Summarizing & Reporting



You need to monitor each part of the process.

ML System User Interfaces Must Communicate Inherent Uncertainty

- ***ML systems are based on probability.***
- ***There is inherent uncertainty in any probabilistic system.***
- ***Output from an ML system must communicate this uncertainty to the users***.

Hypothetical:
Her mom is 5’10” and her dad is 6’1”. Can you predict how tall she will be as an adult?
You can make a good guess, but with some uncertainty.
e.g., *She’ll probably be about 5’10”*

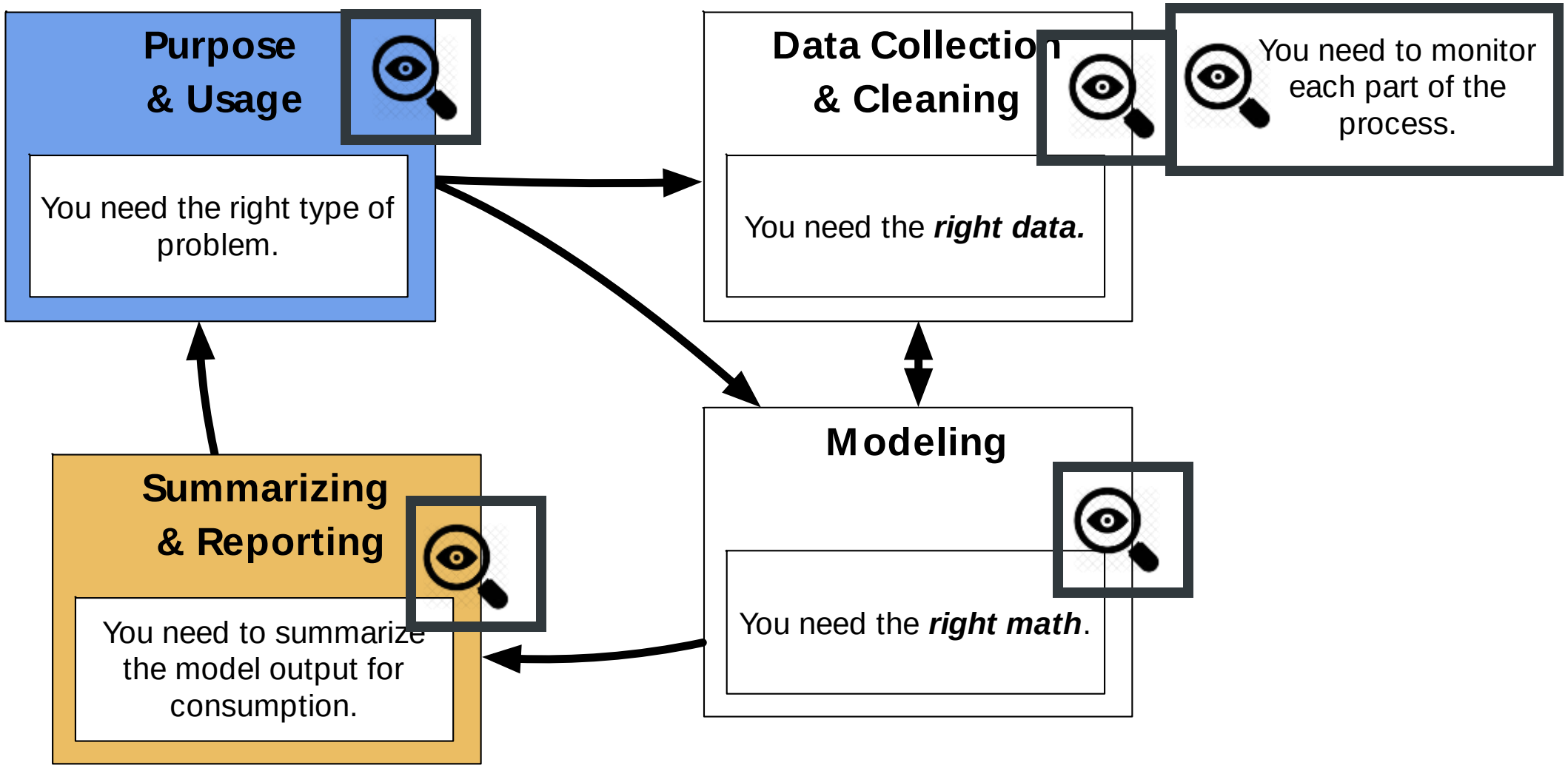
This is the kind of prediction (supervised) ML is doing. It’s doing it faster, with more data, and more precise math. So there’s (usually) less uncertainty than a human prediction.

e.g., *There’s an 80% probability she’ll be between 5’8” and 5’10”.*

ML Systems Must be Designed to Work Within a Larger System

- Risks are different in different contexts.
  - Low risk: An ML system makes an error in predicting which new movie you will like.
  - Medium risk: An ML system makes an error in predicting whether an individual is an insider threat.
- Risk evaluations should be considered in system design.
  - Performance metrics must be chosen to measure context-dependent risks.
  - If the risks are acceptable, an autonomous system might be appropriate.
  - If the risks of an autonomous system are too high, then there need to be appropriate procedures in place to verify a decision and escalate as appropriate (checks and balances) .
- Tactics, techniques and procedures must be carefully designed to mitigate risk.
  - A human-in-the-loop is often suggested as one way to mitigate risk, but this may not be sufficient if there are incentives to "just go with what the algorithm says"

Monitoring



AI/ML Implement Policy

- Examples of where algorithms are implementing policy:
  - Determining whether applicants are eligible for Medicare.
  - Identifying and locating men delinquent in paying child support.
  - Identifying and removing citizens registered to vote in multiple locations.
  - Predicting whether an individual is a threat and should be detained.

- In each of these examples (and many more), we must be able to verify that the algorithm is implementing policy as intended, and that policy is not being set by software developers.

Danielle Keats Citron, Technological Due Process, 85 Wash. U. L. Rev. 1249 (2008).

Any Deployed AI/ML System Needs Checks and Balances

- In the same way that we require inspections for food safety, automobile safety, and aviation safety, we must be able to determine whether a deployed AI/ML system is functioning as designed.
- We must be able to validate and adjust:
  - Training data
  - Model choice
  - Model implementation
  - The data pipeline for the deployed model
  - Performance in the field

Anybody can download Tensorflow and throw a pile of data at it. That does not mean the output is useful or correct.

Validation metrics don't really exist right now. This is one place to invest in research.

- Good monitoring and validation practices help ensure good performance, making the system more secure against both adversaries and poor implementation.

Adversarial Issues Are

1. An active area of research – we don't even fully understand what kinds of attacks might be possible right now.
2. Every neural network ML system is susceptible to attack in infinite variations on a theme.
3. Adversarial attacks are becoming more widespread, we are starting to see them "in the wild."
4. Current best practices for mitigating adversarial risks are ongoing monitoring and validation of the ML system, and having well-designed procedures around it.

Many Adversarial AI applications leverage "usage doesn't match training" in different ways.

This 3D printed turtle was built to be classified as a rifle from any angle.

It was created to fit into a gap in what the algorithm had learned a rifle looked like. The turtle is not from the same population of images that the algorithm was trained on.

https://www.theverge.com/2017/11/2/16597276/google-ai-image-attacks-adversarial-turtle-rifle-3d-printed

Useful Questions to Begin an Oversight Discussion

- What policy is this algorithm implementing?
  - What are the intended consequences of a policy?
  - What unintended consequences can be anticipated?
- What checks and balances are in place?
  - How will field performance be evaluated?
  - What is the procedure for monitoring and validation? Who will be doing the monitoring?
  - Are there historic problems (e.g., racial bias) in this area that could be perpetuated?
- What procedures are in place for handling inherent uncertainty?
  - How is uncertainty communicated to the end user?
  - How can the end user check or verify a prediction? (e.g., If you're uncertain about a rain forecast, you might look at a radar map.)
  - How does the end user make a decision when told a prediction has low confidence? (e.g., the ML system only has 60% confidence in its prediction.)

Checklist to Identify Good Candidates for ML Projects

- Can you state your problem as either:
  - I would like to use ___ data to predict ___.
  - I would like to understand the structure of the features recorded in ___ data.
  - I would like to optimize ____ well defined process.

- Is it a large scale problem?

- Have you already done exploratory analysis on available data?

- Have you considered the broader context?

Please complete the session evaluation in the event mobile app.

Thank you!