

# Security Engineering for Machine Learning



@cigitalgem

NOVEMBER 25, 2020

**GARY MCGRAW, PH.D.**

<https://garymcgraw.com>

where I'm coming from



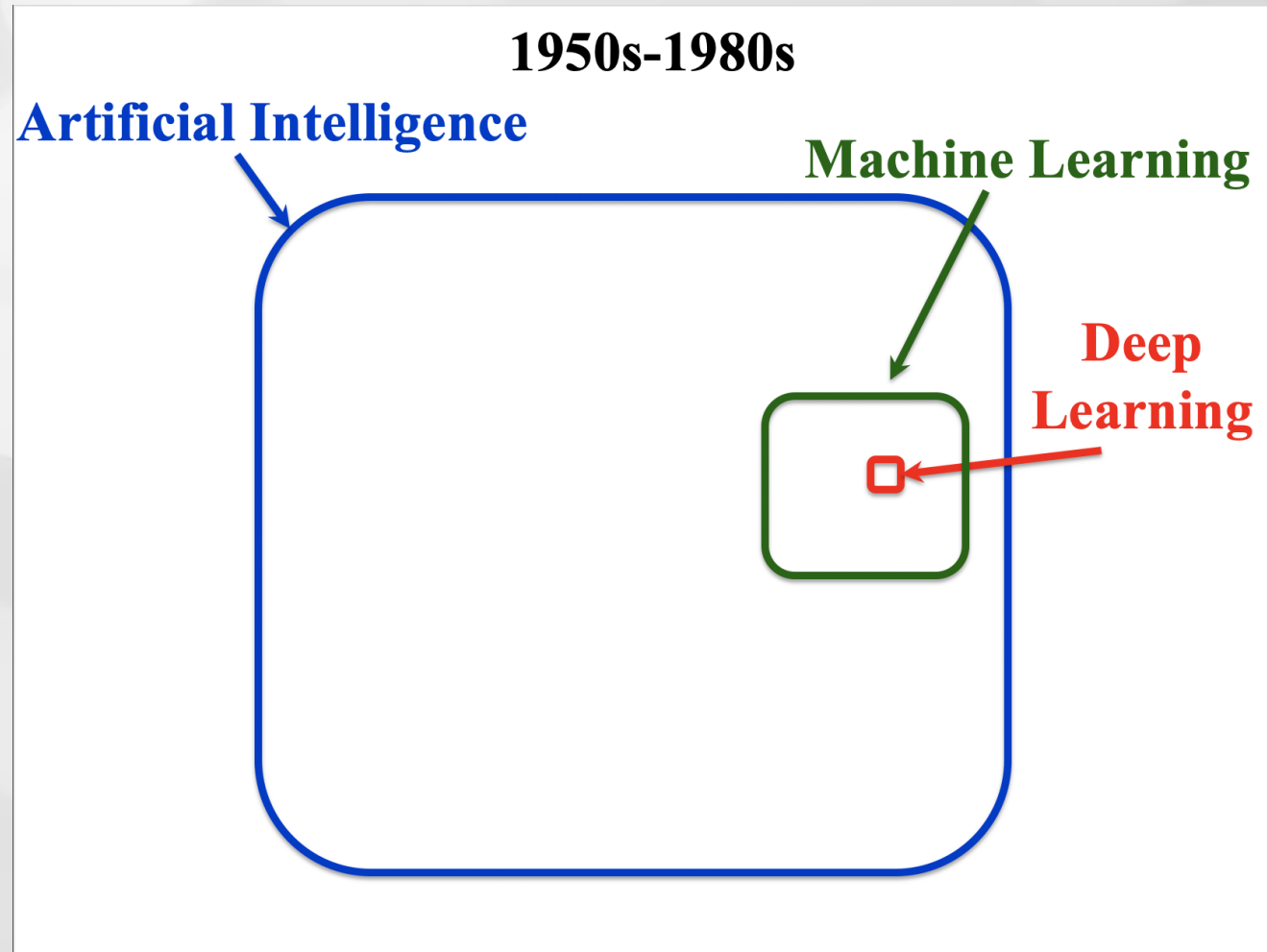
# berryville institute of machine learning



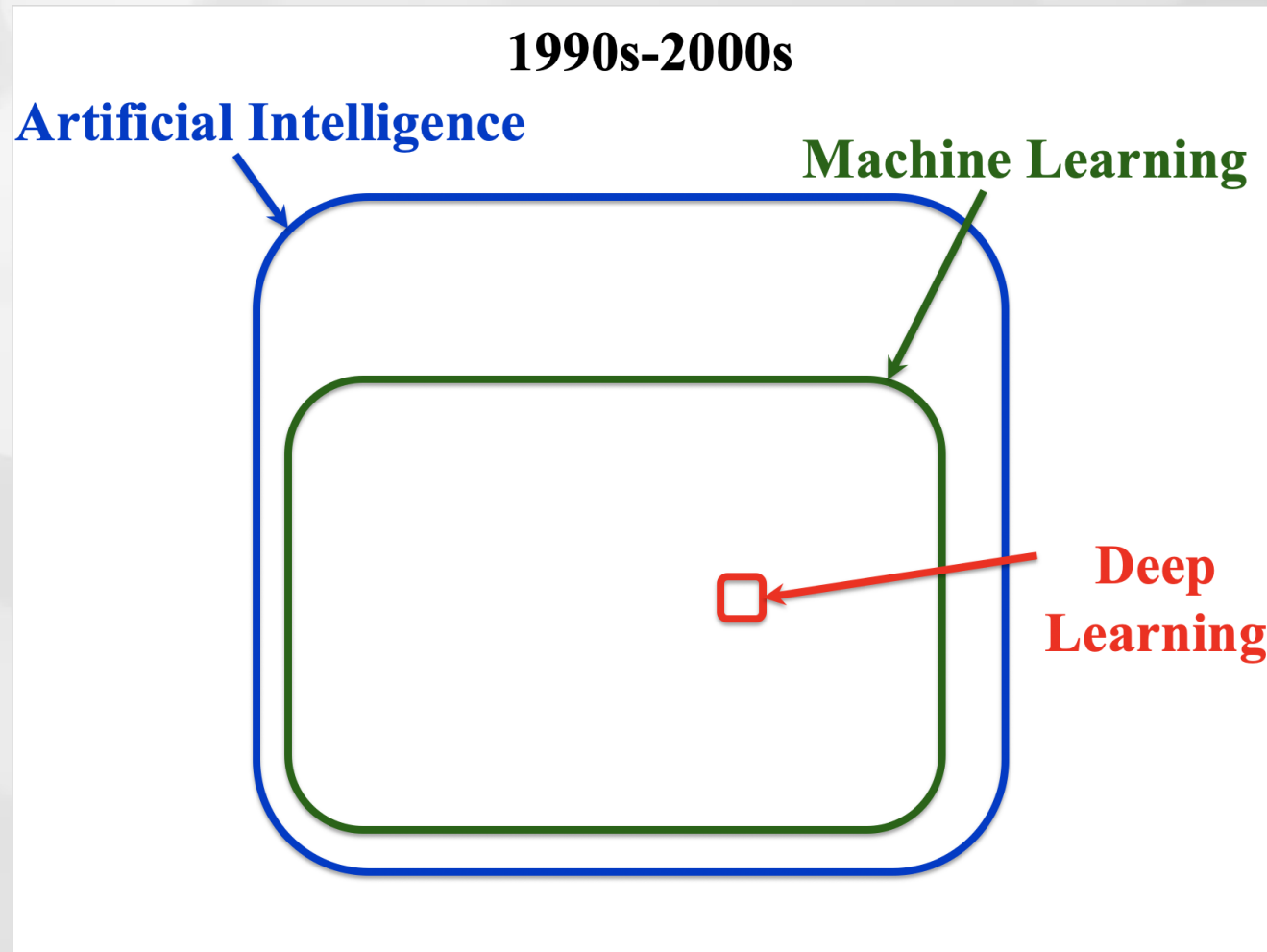
# intro to ML



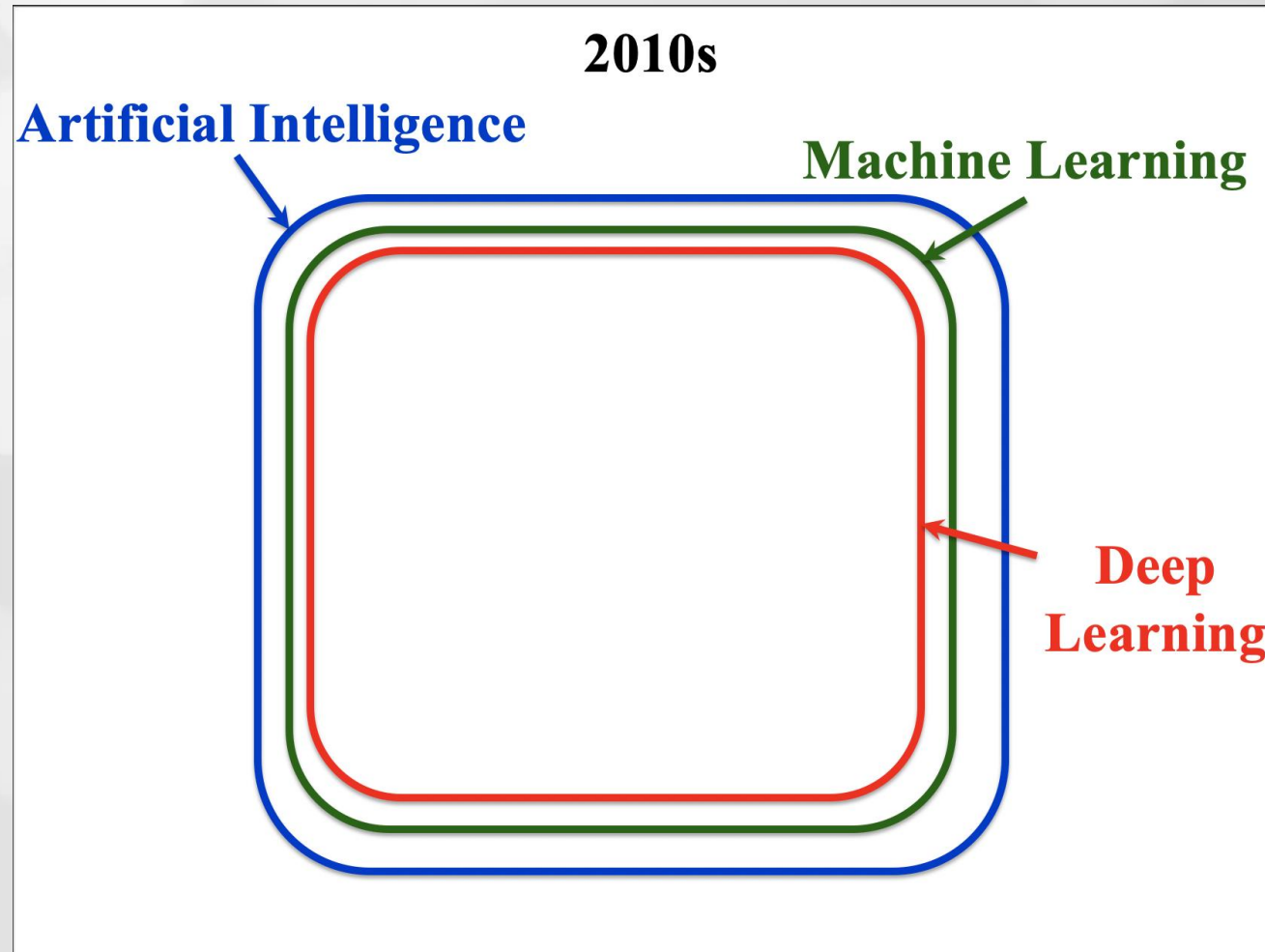
# on AI, ML, and other gobbledygook



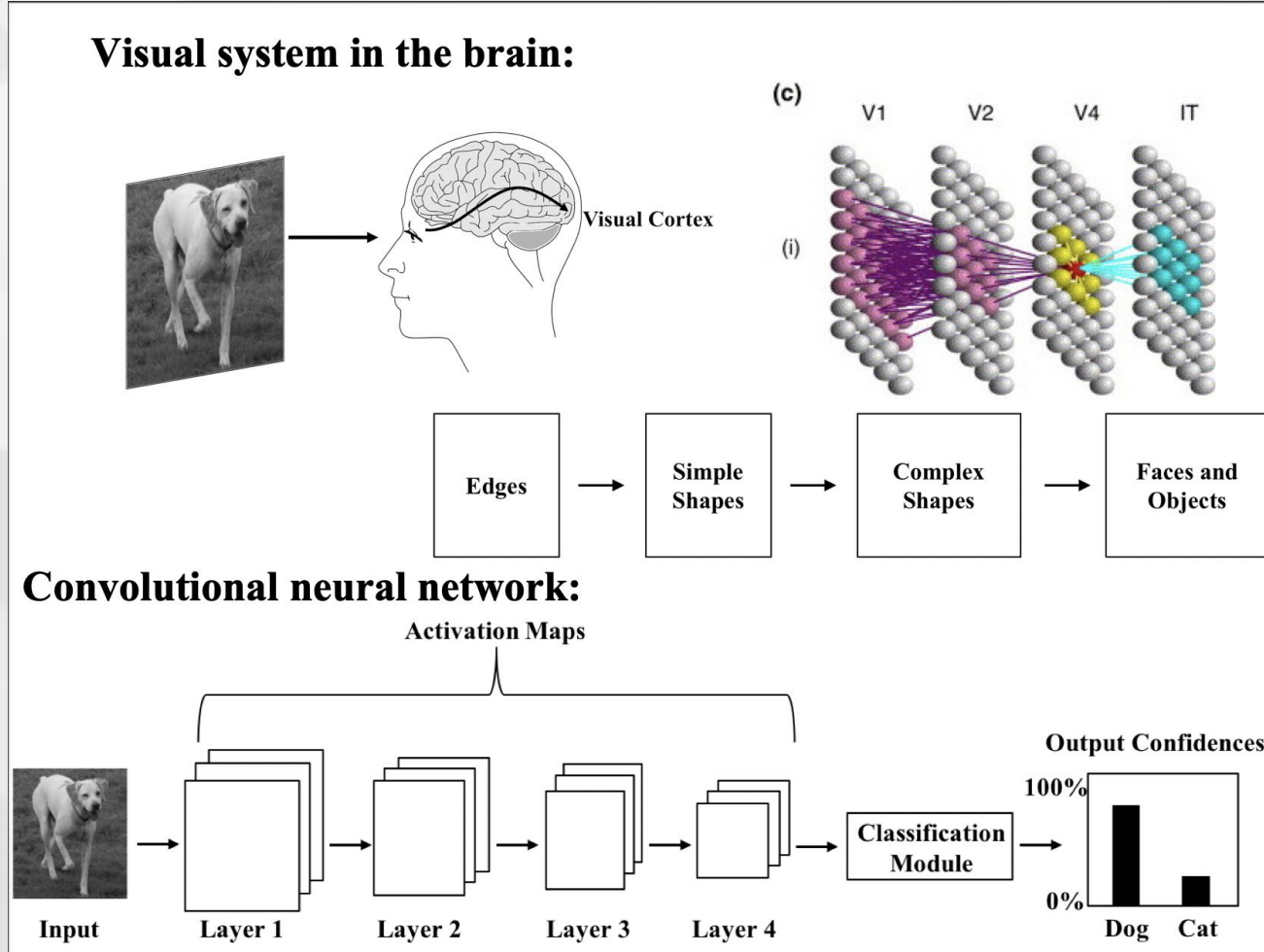
# on AI, ML, and other gobbledygook



# on AI, ML, and other gobbledygook

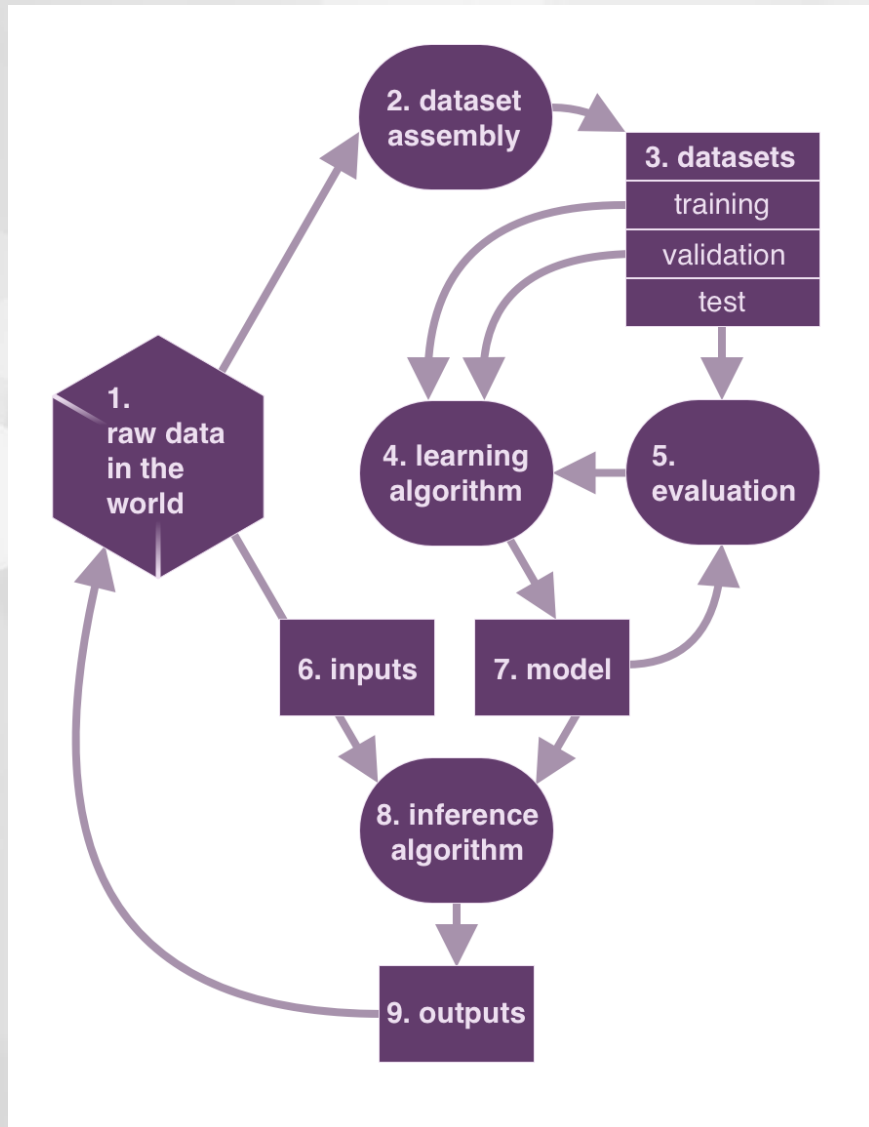


# on AI, ML, and other gobbledygook





# a generic ML model



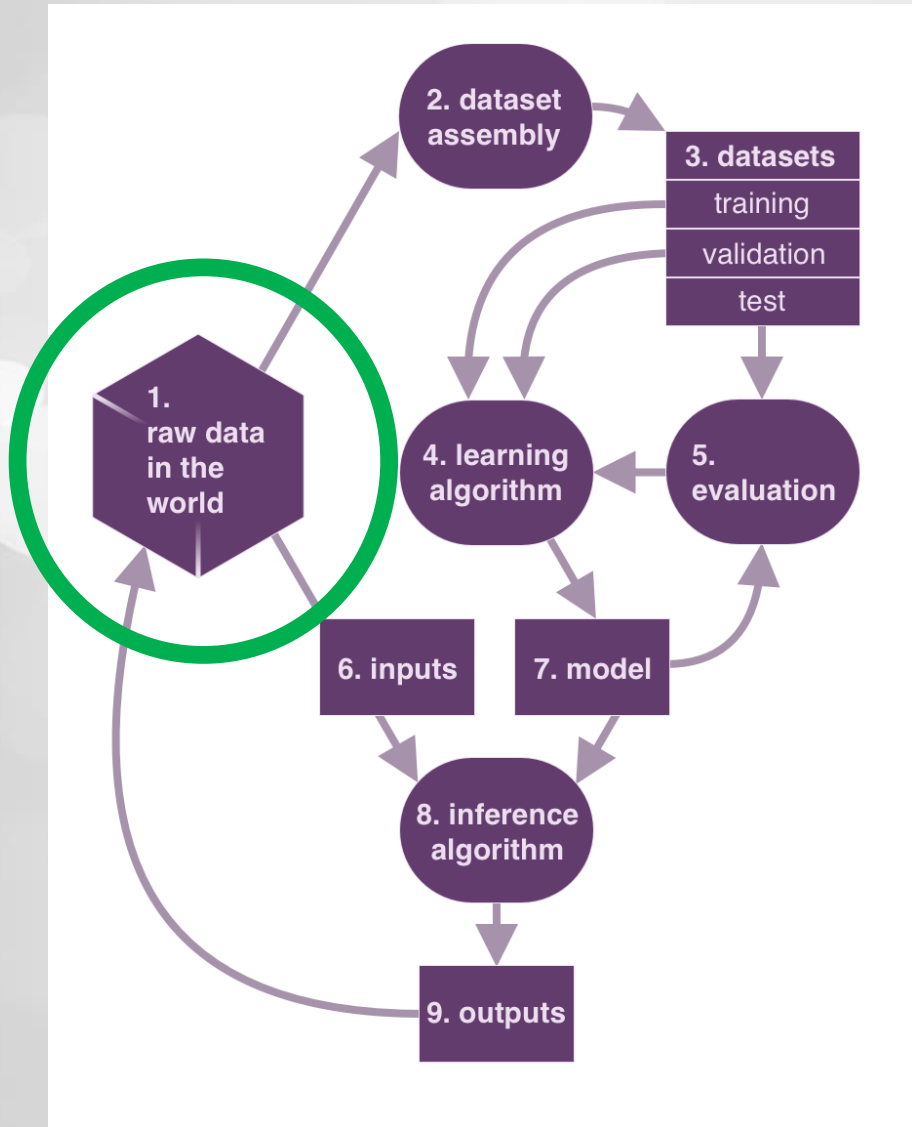
- Nine basic components
  - Processes are ovals
  - Collections are rectangles
- Arrows represent information flow
- We used this model to think about risks in each component



# ML risk analysis



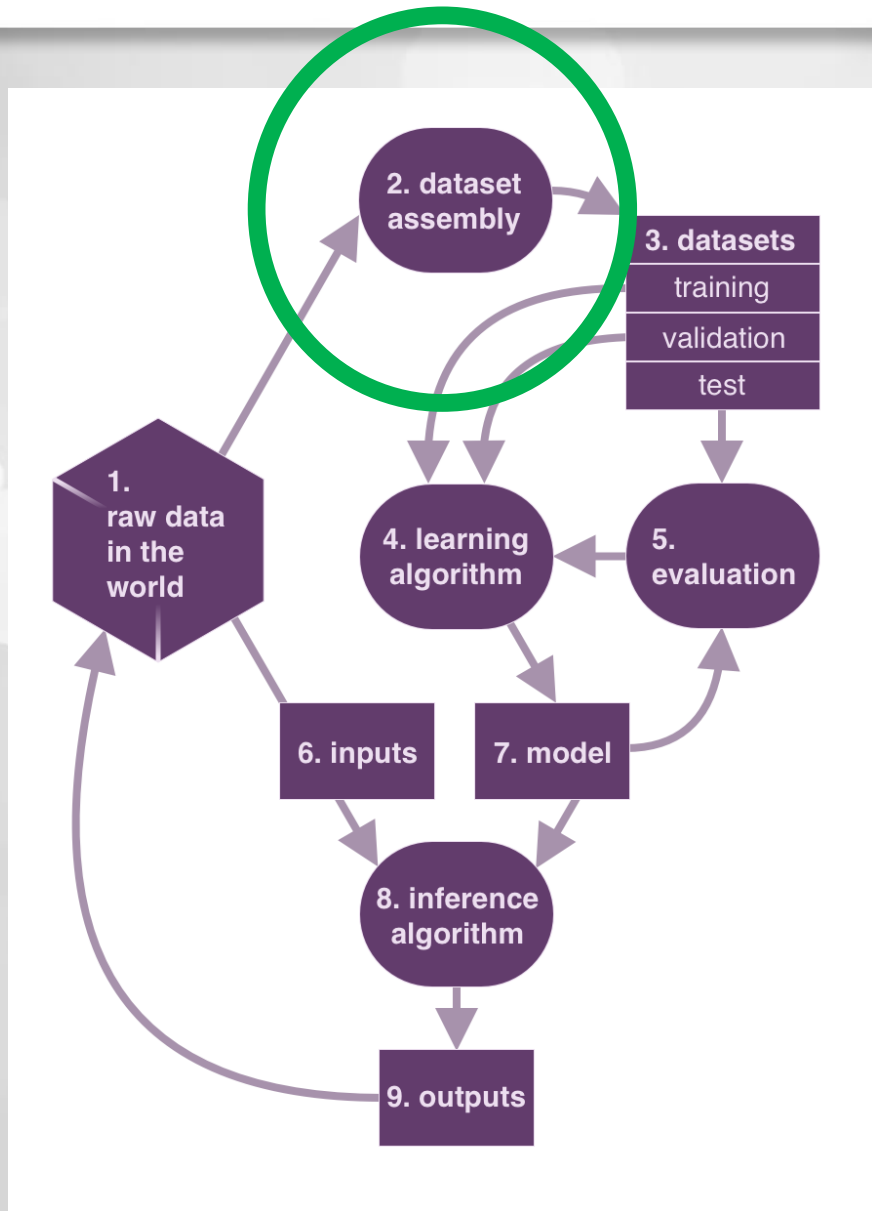
# 1: raw data in the world (13 risks)



- Data play a critical role in ML systems, in fact data are the **MOST IMPORTANT** aspect of ML security
- Lots of raw data out there to be manipulated
- Examples:
  - [raw:1:data confidentiality]
  - [raw:2:trustworthiness]



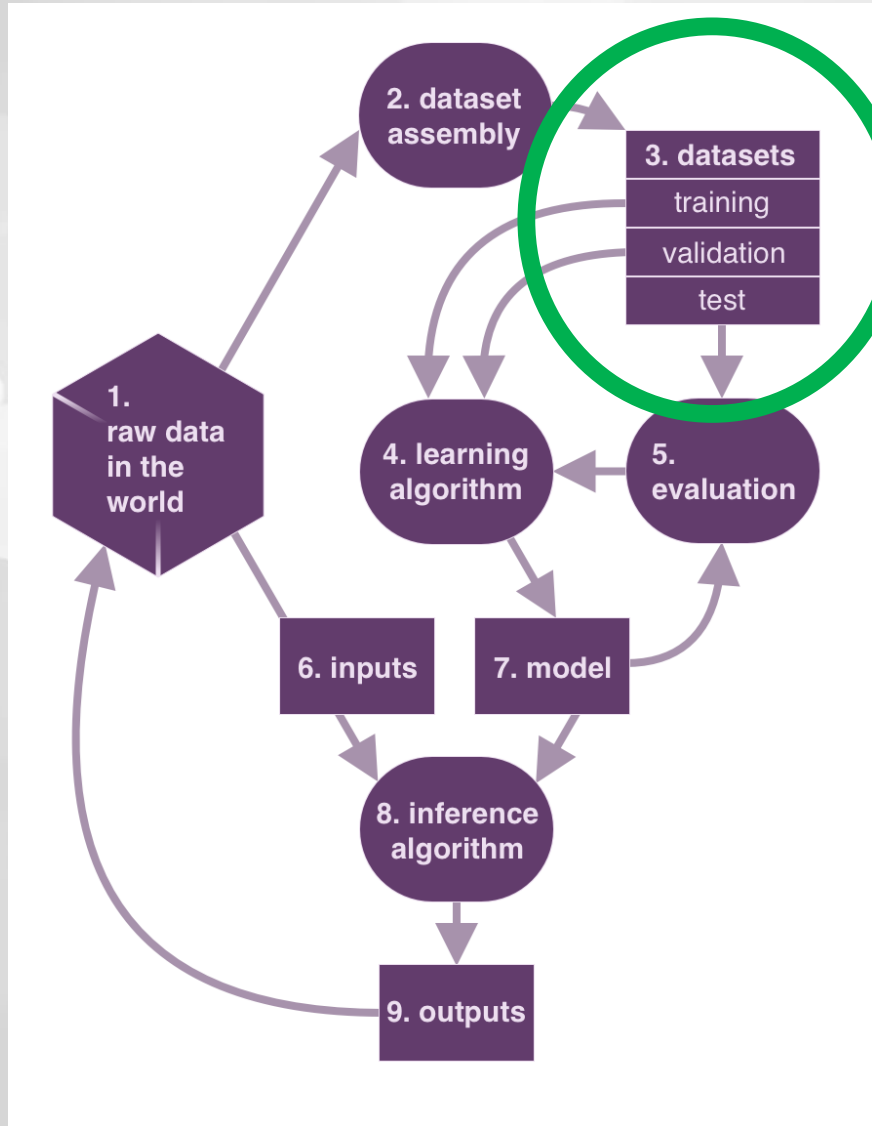
## 2: dataset assembly (8 risks)



- Raw data must be transformed into ML format
- Pre-processing is critical to security
- Online versus offline models (offline is easier to secure)
- Examples:
  - [assembly:1:encoding integrity]
  - [assembly:2:annotation]



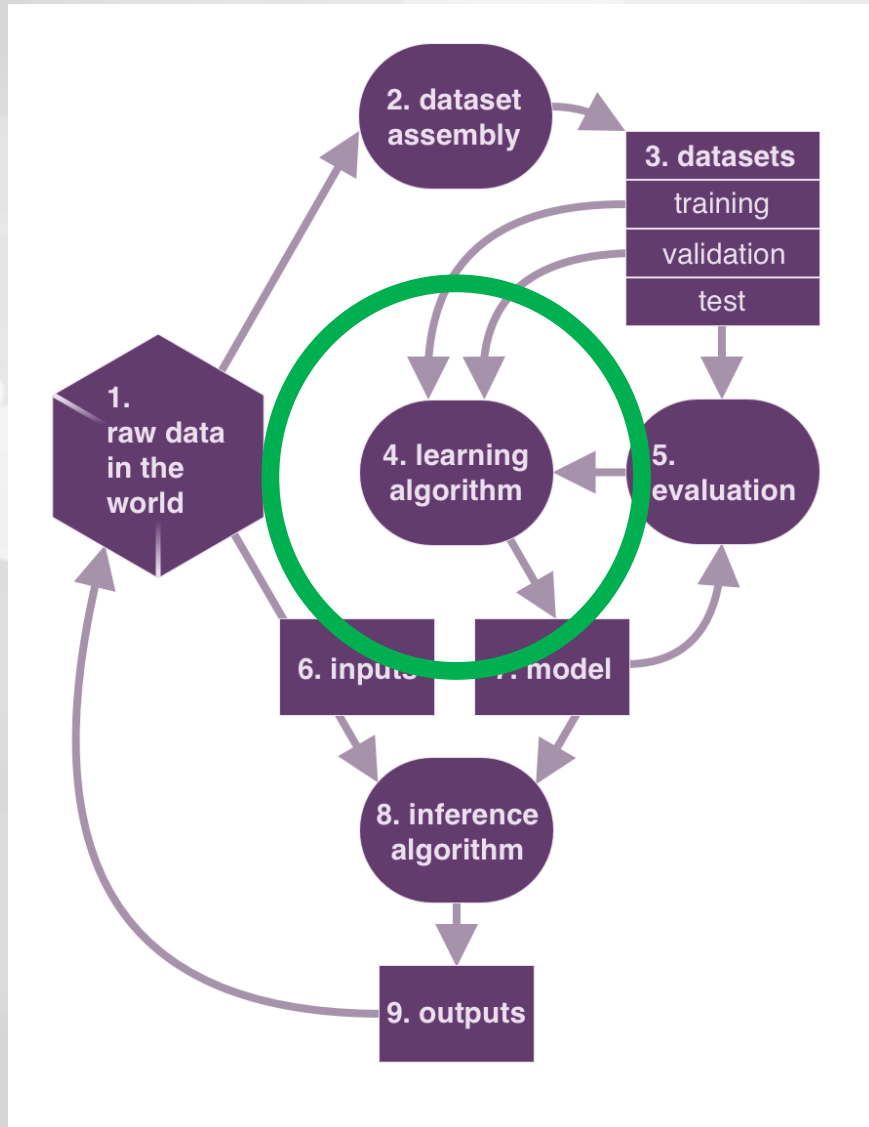
# 3: datasets (7 risks)



- Data are grouped into training, validation, and test sets
- Such partitioning is a tricky process that deeply impacts future ML behavior
- Examples:
  - [data:1:poisoning]
  - [data:2:transfer]



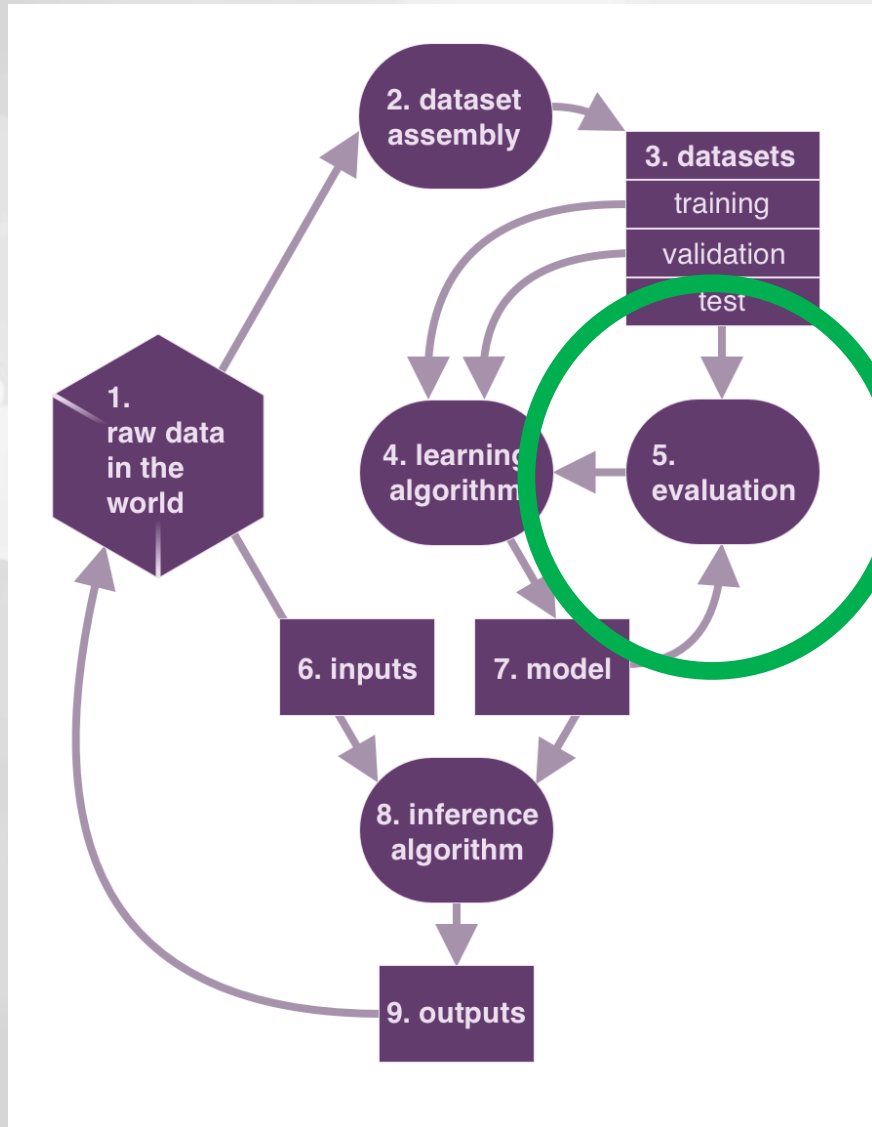
## 4: learning algorithm (11 risks)



- The technical heart of ML (but less security risk than the data)
- Online versus offline (offline is easier to secure)
- Examples:
  - [alg:1:online]
  - [alg:2:reproducibility]



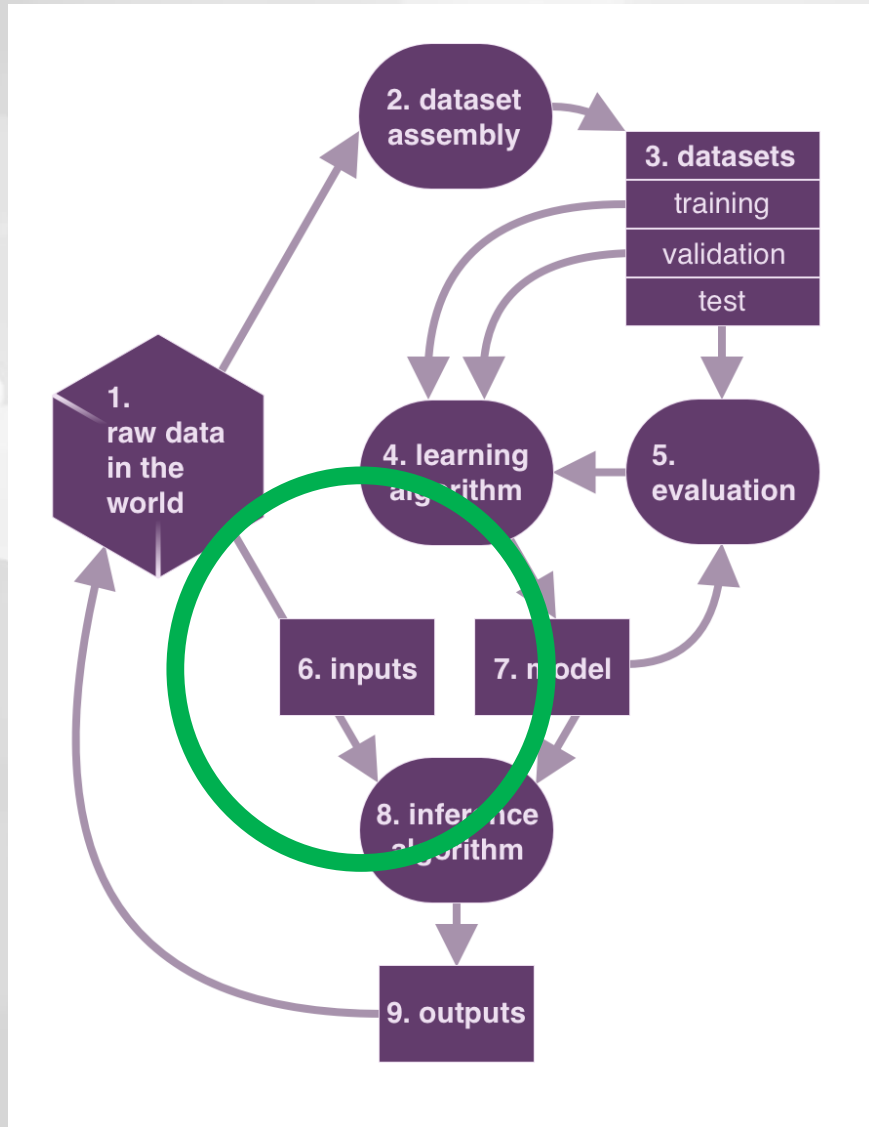
# 5: evaluation (7 risks)



- When is training “done”?
- How good is the trained model?
- Examples:
  - [eval:1:overfitting]
  - [eval:2:bad eval data]



# 6: inputs (5 risks)

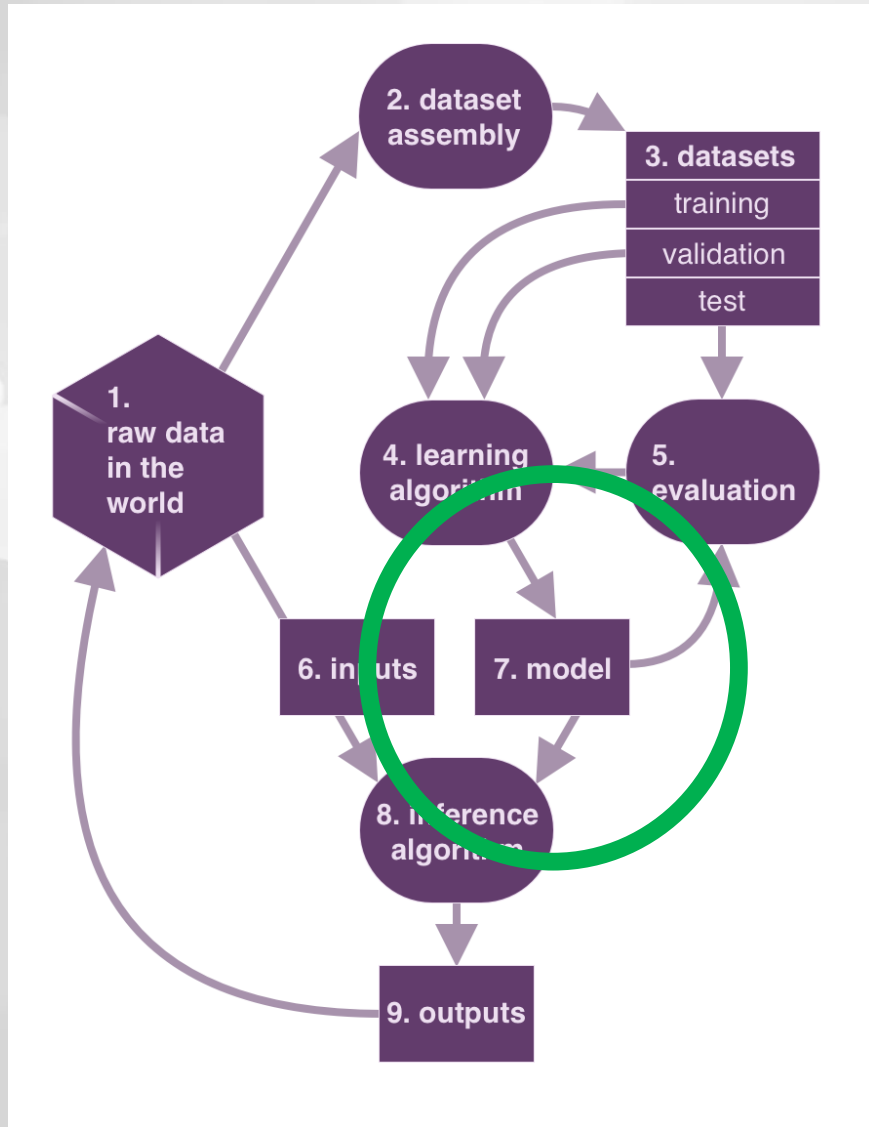


- What input is fed to the trained model during production?
- Very similar to dataset assembly risks and raw data risks
- Examples:
  - [input:1:adversarial examples]
  - [input:2:controlled input stream]





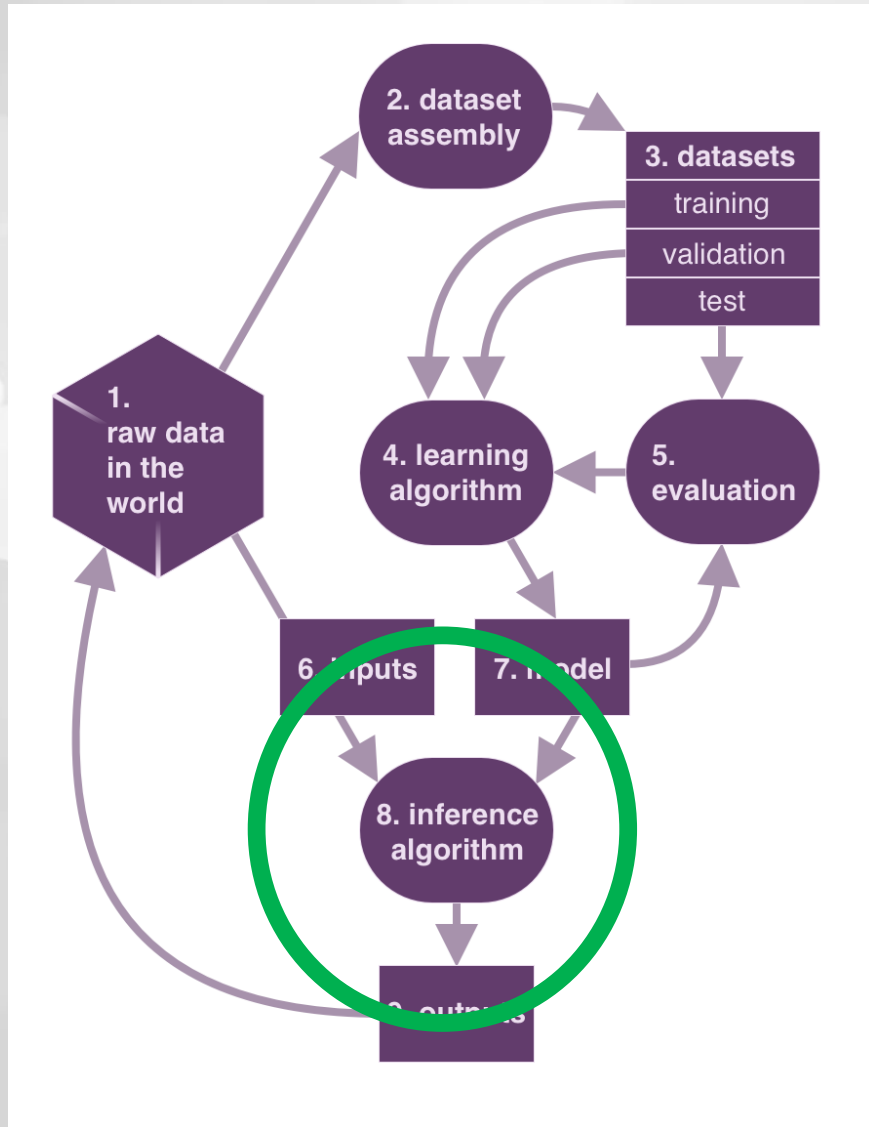
# 7: model (5 risks)



- Risks associated with a fielded model
- Similar to evaluation risks in many respects
- Examples:
  - [model:1:improper re-use]
  - [model:2:Trojan]



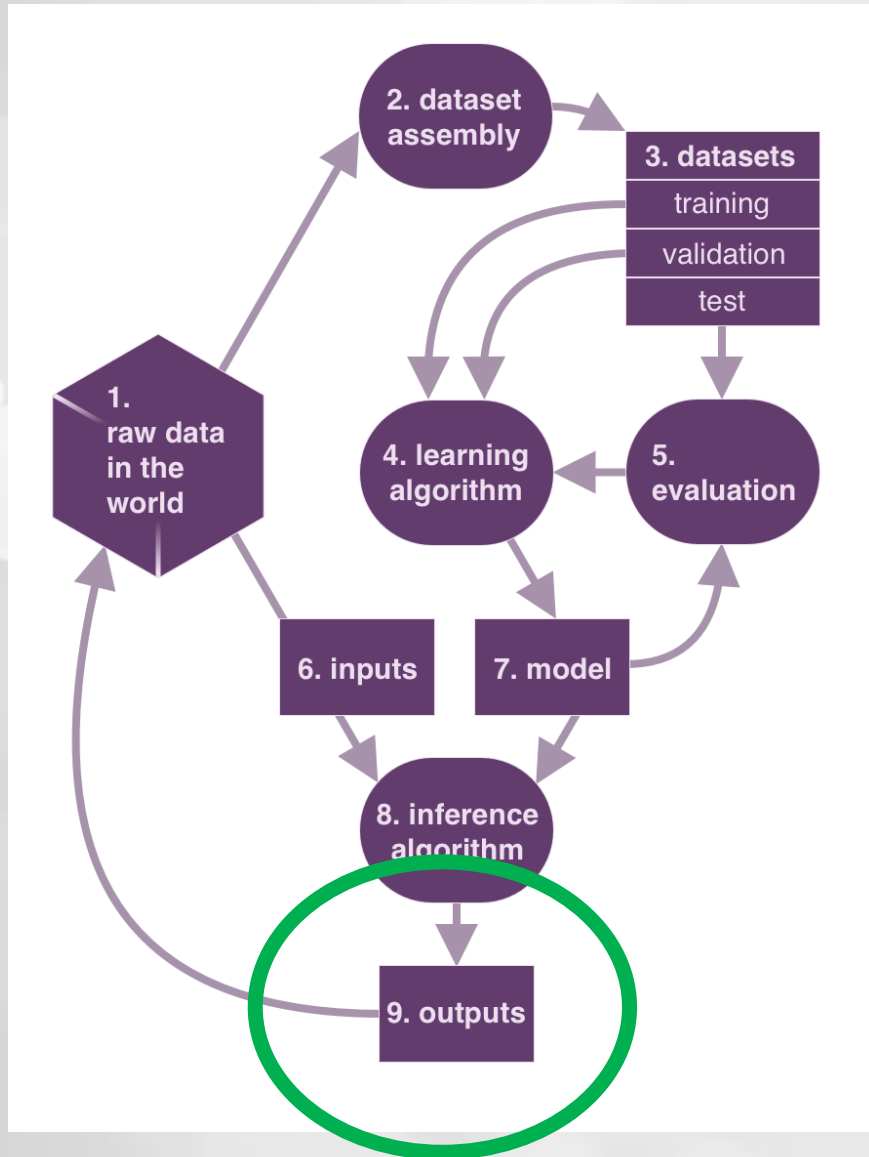
# 8: inference algorithm (5 risks)



- More risks associated with a fielded model
- Output risks arise
- Examples:
  - [inference:1:online]
  - [inference:2:inscrutability]



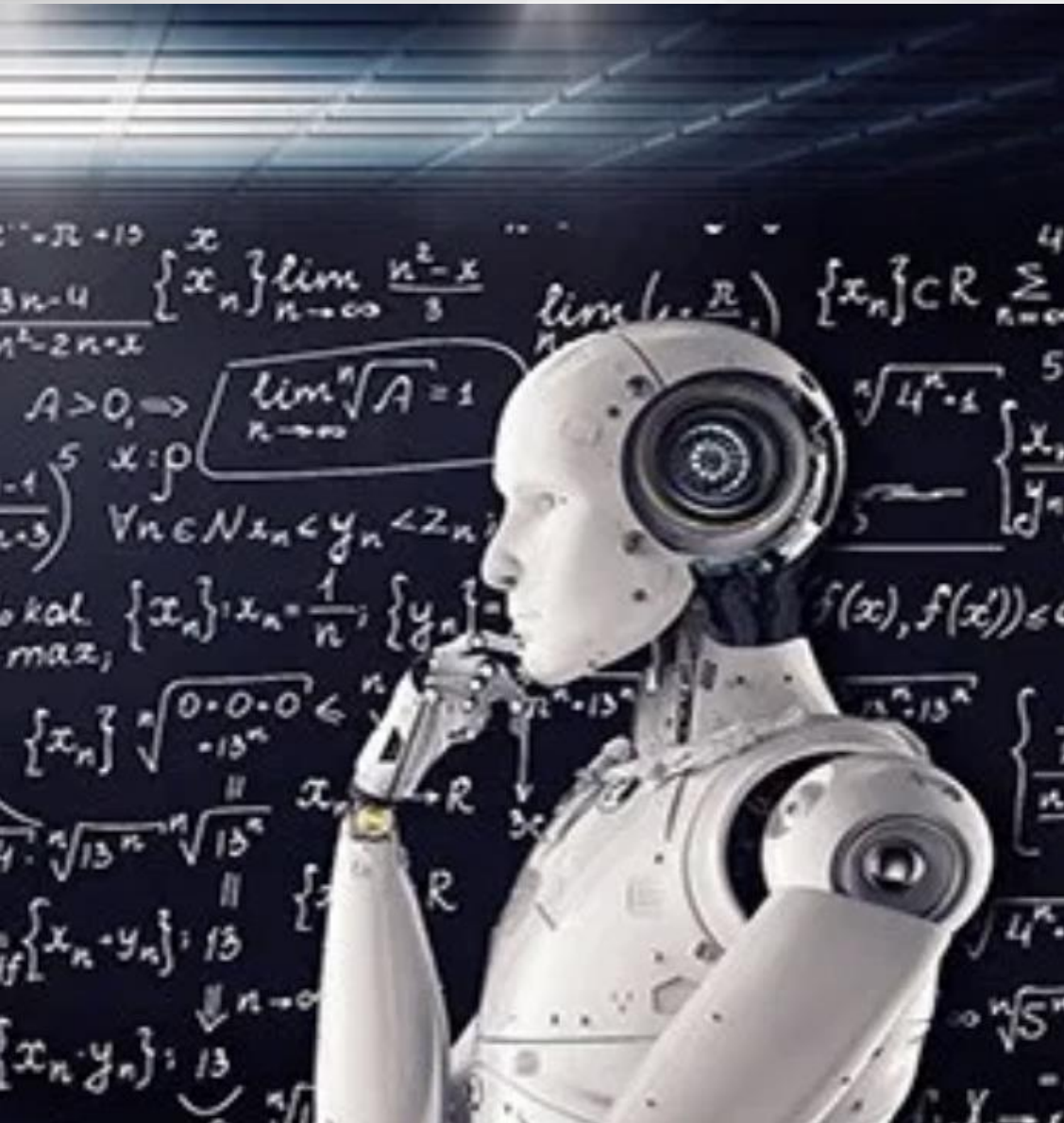
# 9: outputs (7 risks)



- System output is often the whole point
- Direct attack on the output is pretty obvious
- Examples:
  - [output:1:direct]
  - [output:2:provenance]



# system-wide risks (10 risks)



- Getting beyond (and over) a component view
- These risks happen between or across components
- Examples:
  - [system:1:black box discrimination]
  - [system:2:overconfidence]

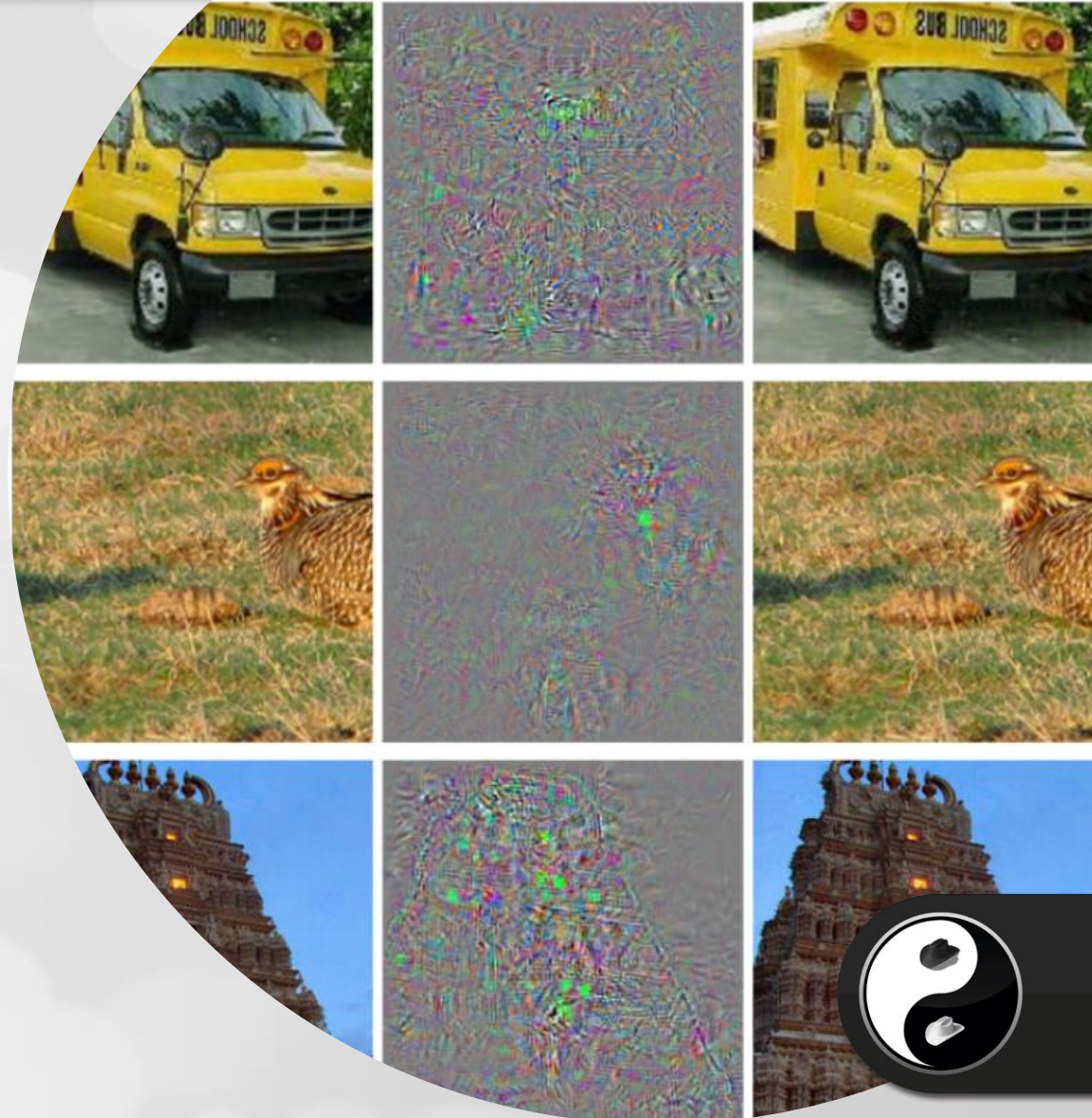


top five ML risks



# 1. adversarial examples

- Probably the most commonly discussed attacks
- Fool an ML system by providing malicious input often involving very small perturbations that cause the system to make a false prediction or categorization
- Though coverage and resulting attention might be disproportionately large, swamping out other important ML risks, adversarial examples are very much real



## 2. data poisoning

- Data play an outsized role in the security of an ML system
- If an attacker can intentionally manipulate the data being used by an ML system in a coordinated fashion, the entire system can be compromised
- Data poisoning attacks require special attention.
  - What fraction of the training data can an attacker control and to what extent?



### 3. online system manipulation



- An ML system is said to be “online” when it continues to learn during operational use, modifying its behavior over time
- A clever attacker can nudge the still-learning system in the wrong direction on purpose
- This slowly “retrains” the ML system to do the wrong thing
- This risk is complex, demanding that ML engineers consider data provenance, algorithm choice, *and* system operations in order to properly address it





## 4. transfer learning attack

- In many cases in the real world, ML systems are constructed by taking advantage of an already-trained base model which is then fine-tuned to carry out a more specific task
- A data transfer attack takes place when the base system is compromised (or otherwise unsuitable), making unanticipated behavior defined by the attacker possible



## 5. data confidentiality



- Data protection is difficult enough without throwing ML into the mix
- One unique challenge in ML is protecting sensitive or confidential data that, through training, are built right into a model
- Subtle but effective extraction attacks against an ML system's data are an important category of risk



where to learn  
more



# build security in

- Writings, Blogs, Music  
<https://garymcgraw.com>
- BIML  
<https://berryvilleiml.com/>
- Send e-mail:  
[gem@garymcgraw.com](mailto:gem@garymcgraw.com)



@cigitalgem

