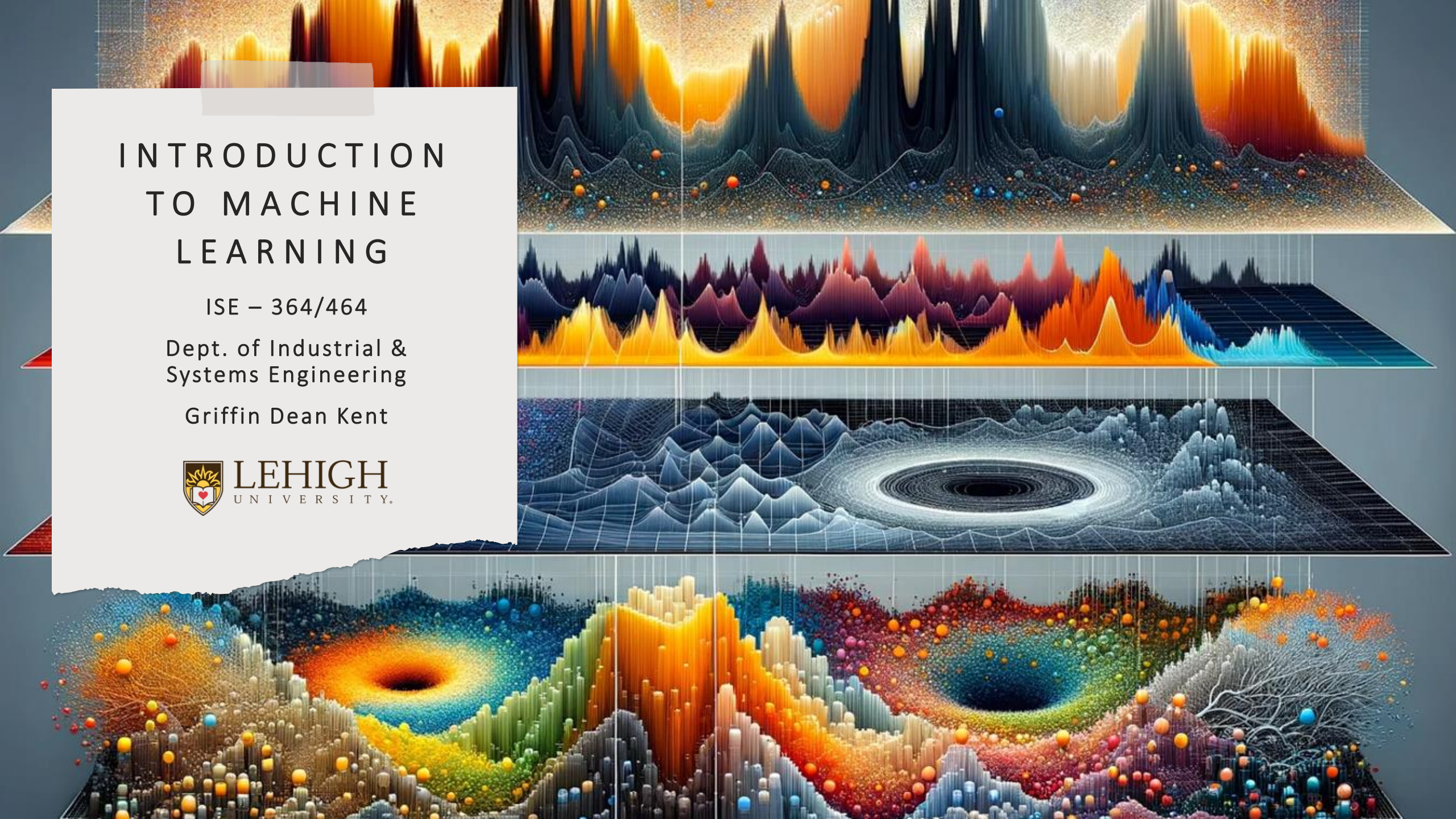
INTRODUCTION TO MACHINE LEARNING

ISE – 364/464

Dept. of Industrial & Systems Engineering

Griffin Dean Kent

LEHIGH UNIVERSITY



WELCOME TO ISE – 364 / 464!

Course Description (for 364 section):
Techniques of applied machine learning rather than deep theory behind the algorithms and methods. Programming solutions for machine learning problems using a high-level programming language and associated machine learning libraries. Regression, clustering, principal component analysis, Bayesian methods, decision trees, random forests, support vector machines, and neural networks.

→

Translation (for 364):
Assignments will mostly consist of mathematically less-rigorous problems as well as coding problems.

Course Description (for 464 section):
Techniques of applied machine learning with some deep theory behind the algorithms and methods. Programming solutions for machine learning problems using a high-level programming language and associated machine learning libraries. Regression, clustering, principal component analysis, Bayesian methods, decision trees, random forests, support vector machines, and neural networks.

→

Translation (for 464):
Same requirements for the 364 section. However, there will be added more-difficult problems in assignments (proofs, and more involved coding problems).

Class Time: Monday & Wednesday 3:00 - 4:15 PM.

Office Hours: Monday & Wednesday 4:15 – 5:00 PM.

A Note for Both Sections:
The mathematical formulations /derivations as well as some theory of the algorithms will be covered in lectures and will be some topics of the lectures.

GRADING

Homework Assignments (40%):

- Bi-weekly (every 2 weeks) typed sets of 4 problems (5 for 464). Approximately 7 assignments: ~1.1%-1.5% per question.
- These problems will either consist of either math problems or coding questions.
- The other additional question for section 464 will be a more theoretical-based proof question.

Final Project (30%):

- You will be given a real dataset which you will data-mine for a supervised learning problem.
- You will write a report detailing all your methodology as well as fully justify your reasoning.

Final Exam (20%):

- 1 on 1 meetings with me. I will ask you interview-style questions regarding your homework assignments.

Participation (10%):

- Self-explanatory… Be involved, show me you're trying, talk with your class-mates, answer questions in lecture.

Extra-Credit (for section 364 only):

- Section 364 students can answer the added proof-question meant for the 464 students (adds ~1.1% per question).

A Note on Participation:

This class may be difficult… Machine Learning is a field that requires a decent amount of overhead knowledge. As such, we will be utilizing concepts from several different arenas of applied mathematics, such as calculus, linear algebra, analysis, probability, statistics, and computer science.

Please feel free to talk amongst each other and form study groups if need be. Also feel free to contact me if you have questions about anything, but please understand that I only have so much time to help (as I do not have a TA for this class).

My job here is to help you to succeed in this class, to the best of my ability! Your success as a student is my success as an instructor, so this is a team effort!

ON PRE-REQUISITE KNOWLEDGE (FOR THIS CLASS)

Desirable Background Knowledge:

- **Multivariate Calculus** (Derivatives, Gradients, Hessians)
- **Linear Algebra** (Matrix-Vector Operations, Dot-Products, Norms, Matrix Inversion)
- **Probability** (Expected Value, Variance, Distributions)
- **Statistics** (Hypothesis Testing, Exploratory Data Analysis, Philosophy of Statistics)
- **Coding** (Python)

Placement "Exam":

Please complete the placement exam by the end of this first week! There are two sections to it: "Minimum Background" and "Intermediate Background".

"Minimum": If you are struggling with the "Minimum" section, you should maybe wait a semester or two before taking this class to obtain the required math background.

"Intermediate": Students in the 464 section should be able to answer all these questions. If 464 students struggle with this section, then their assignments may be on the difficult end for them (they may take you more time). It would be great if 364 students could answer some of the "Medium" questions, but not mandatory for success.

Review Lectures:

The first few weeks of the class will indeed be reviewing essential topics that everyone should already be familiar with, but the students should not view these lectures as a substitute to taking those classes and should plan accordingly.

CODING

Fundamentals

- NumPy → Math / Matrices / Linear Algebra
- pandas → Dataframes / Data Manipulation
- matplotlib, seaborn → Plotting / Graphis

(1st Month of Classes)

Specialized

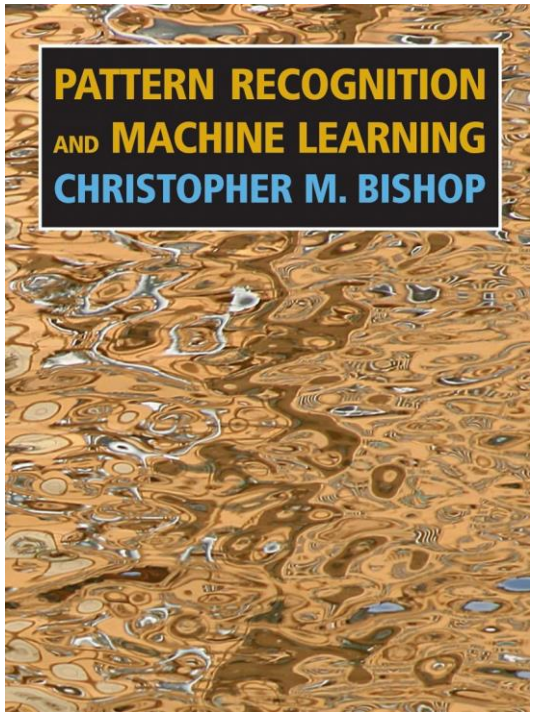
- SciPy → Optimization / Statistics
- scikit learn → Machine Learning
- PyTorch → Deep Learning

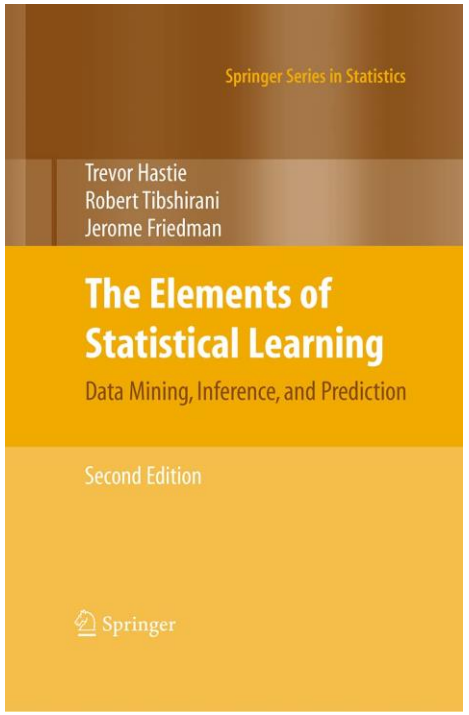
(As Needed)

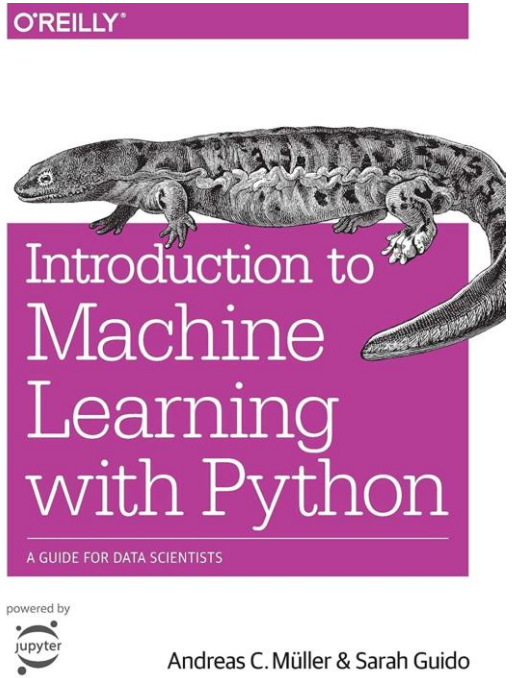
GENERAL OUTLINE OF THIS CLASS

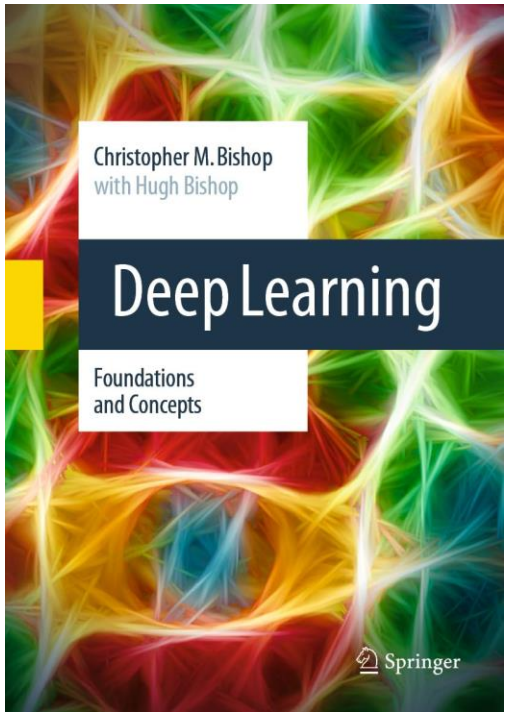
- Introduction and General Overview (Today!)
- Review Material
    - Multi-Variable Calculus
    - Linear Algebra
    - Probability & Statistics
    - EDA with Python
- The Paradigm of Mathematical Optimization and its Connection to Learning
- Supervised Learning
    - Discriminative Models
        - Linear Regression
        - Logistic Regression
        - K-Nearest Neighbors
        - Decision Trees
        - Random Forests and Boosted Trees
        - SVM
        - Neural Networks
    - Generative Models
        - Gaussian Discriminative Models
        - Naïve Bayes
- Unsupervised Learning
    - K-Means Clustering
    - Gaussian Mixture Models
- Dimensionality Reduction
    - PCA
    - T-SNE and UMAP *(Optional)*
- Data Mining
    - Data Cleaning
    - Feature Engineering
    - Dealing with Different Types of Data
    - Feature Encoding
    - Model Selection
    - Cross-Validation
    - Hyperparameter Tuning

GOOD REFERENCE TEXTBOOKS FOR THIS CLASS (OPTIONAL)







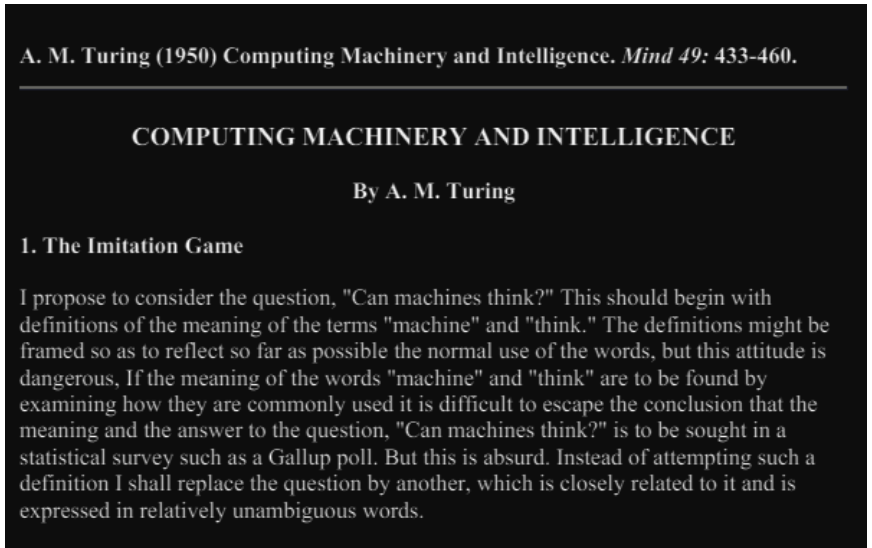


SOME HISTORY OF MACHINE LEARNING

A BRIEF HISTORY OF ML

Alan Turing (1912 – 1954)

- Regarded as the "Father of Computer Science".
- Turing was the **first** to conduct substantial research in the field that he referred to as **"Machine Intelligence"**.
- In his seminal paper titled **"Computing Machinery and Intelligence"** (1950), Turing proposed what he called the *Imitation Game* (now known as the *Turing Test*) as a proxy to identify the point at which a machine essentially **attains consciousness**.



A. M. Turing (1950) Computing Machinery and Intelligence. *Mind 49:* 433-460.

COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. The Imitation Game

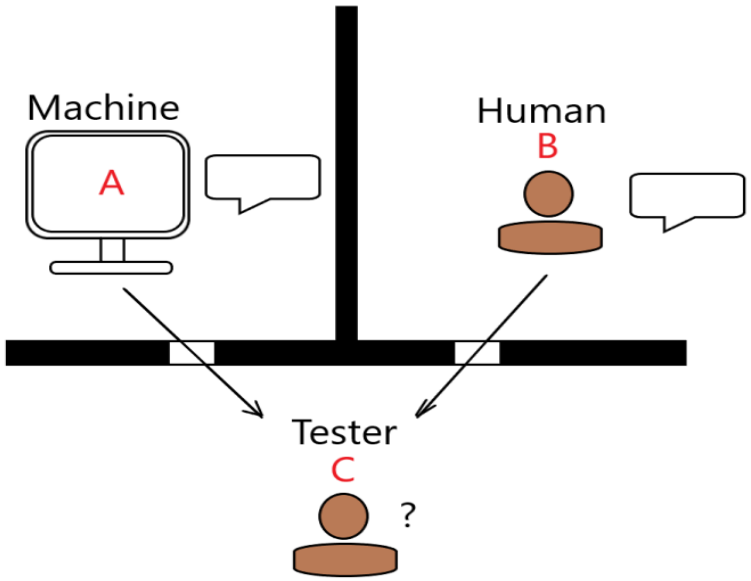
I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous, If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.

A BRIEF HISTORY OF ML

The Turing Test

Original Version

- **Setup:** Three rooms with a person in two of the rooms (a tester A and a testee B) and a machine A in the third room.
- **Goal:** Can the tester C, without knowing which is the machine A or the testee B, correctly identify the two via remote interaction?



- There have been a series of "goal-post shifting" when it comes to TT.
- The original version of TT was surpassed long ago…

Modern Version (AGI?)

- What if the tester interacts directly with the machine but **does not know** that it is a machine. Can the tester **identify** if they are interacting with a real person or a machine?

Hyper-Modern Version (???)

- What if the tester interacts directly with the machine and **knows** that it is indeed a machine. Is the machine advanced enough to surpass that barrier and **still be convincing** enough to elicit human-like treatment?

A BRIEF HISTORY OF ML

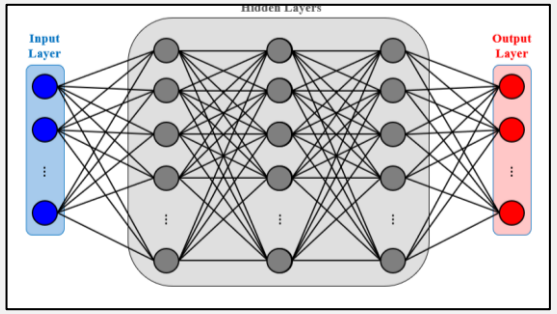
1950: Alan Turing writes the paper "**Computing Machinery and Intelligence**", proposing the Turing Test.

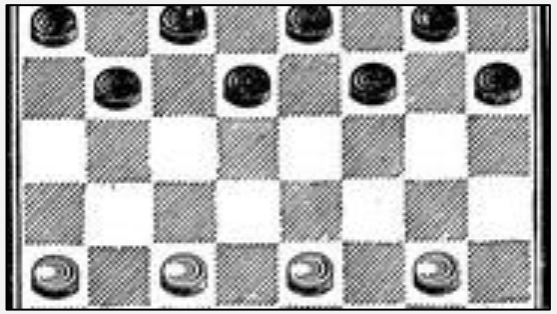
1952: Arthur Samuel creates one of the **first self-learning programs**; it plays checkers and gets better with experience.

1956: John McCarthy coins the term "**Artificial Intelligence**" during the Dartmouth Conference.

1957: Frank Rosenblatt develops the **Perceptron** model; the fundamental building block of modern-day artificial neural networks (NNs).

1959: Arthur Samuel coins the term "**Machine Learning**" to describe the field focused on giving computers the ability to learn from data and improve their performance over time in an automated fashion.





A BRIEF HISTORY OF ML

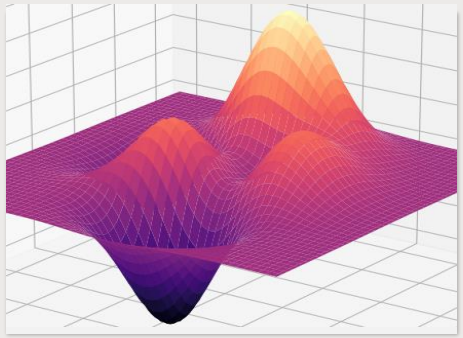
1967: Shun'ichi Amari published a **deep-neural network** trained via **stochastic gradient descent** for the **first time** and can classify non-linearly separable classes of data.

1970: Seppo Linnainmaa develops the modern **backpropagation algorithm** for computing the gradients of NNs.

1970s – 1980s: The **"A.I. Winter"**. During this period, interest and funding (and therefore research) in A.I. and ML began to fall flat due to unmet expectations and the limitations of the technology of the time.

1995: Vladimir Vapnik and co. develop the **Support Vector Machine** (SVM) as a simpler alternative to NNs.

Early 2000s: **Ensemble** models begin to emerge such as the **Random Forest** (Leo Brieman, 2001) as well as **Boosting** methods.



THE DEEP LEARNING REVOLUTION & THE RISE OF REINFORCEMENT LEARNING

2003: Interest in **Deep Learning** is **revitalized** due to the success of DNNs being applied to language modeling. One of the researchers that pushed these developments is one of the "Godfathers of A.I.", Yoshua Bengio.

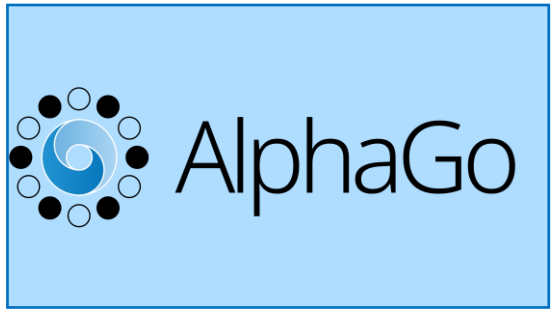
2012: The deep convolutional NN (CNN) **"AlexNet"** wins the ImageNet competition, demonstrating the power of CNNs.

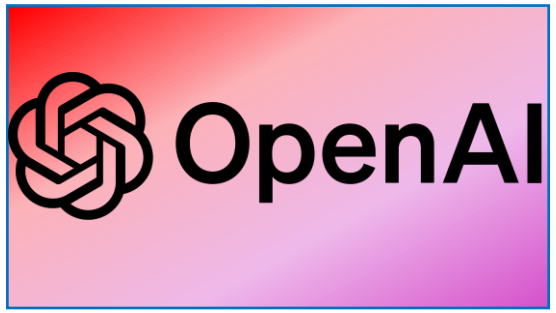
2014: Ian Goodfellow and co. develop **Generative Adversarial Networks** (GANs).

2016: Google DeepMind's **"AlphaGo"** defeats the world champion Go player, demonstrating the power of deep RL.

2017: The **Transformer architecture** is developed, the beginning of large-language models (LLMs).

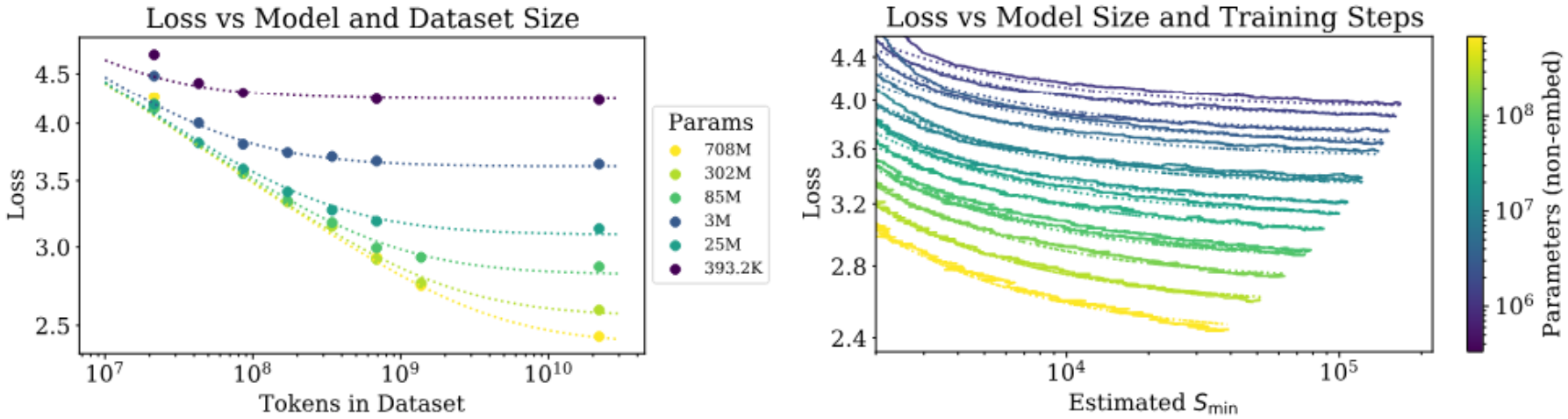
2018 & 2020: Google develops **BERT** for language understanding and OpenAI releases **GPT-3** ushering in the "Era of LLMs" to the public.





THE CURRENT LLM "GOLDRUSH"

"The Scaling Laws for Neural Language Models" (2020)



- Essentially: Performance of LLMs has not "bottomed out" yet.

  Model Size + Training Iterations = Lower Loss.

- Thus: The more compute power you have, the better your LLM will be.

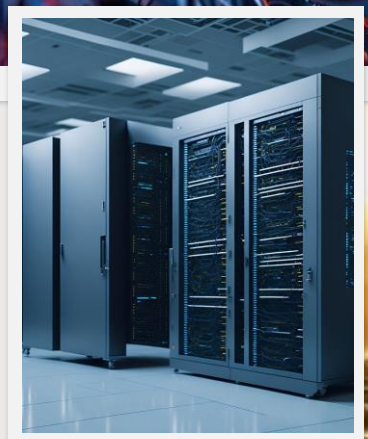
THE "IMMINENT" FUTURE OF ML

Summary of Training Meta's Llama 2 70B (relatively small)

- ~10 Terabytes of text (from the internet)
- ~6,000 GPUs
- ~12 days
- ~$2,000,000

General Bottlenecks in Training LLMs

- GPUs
- Data storage
- Power



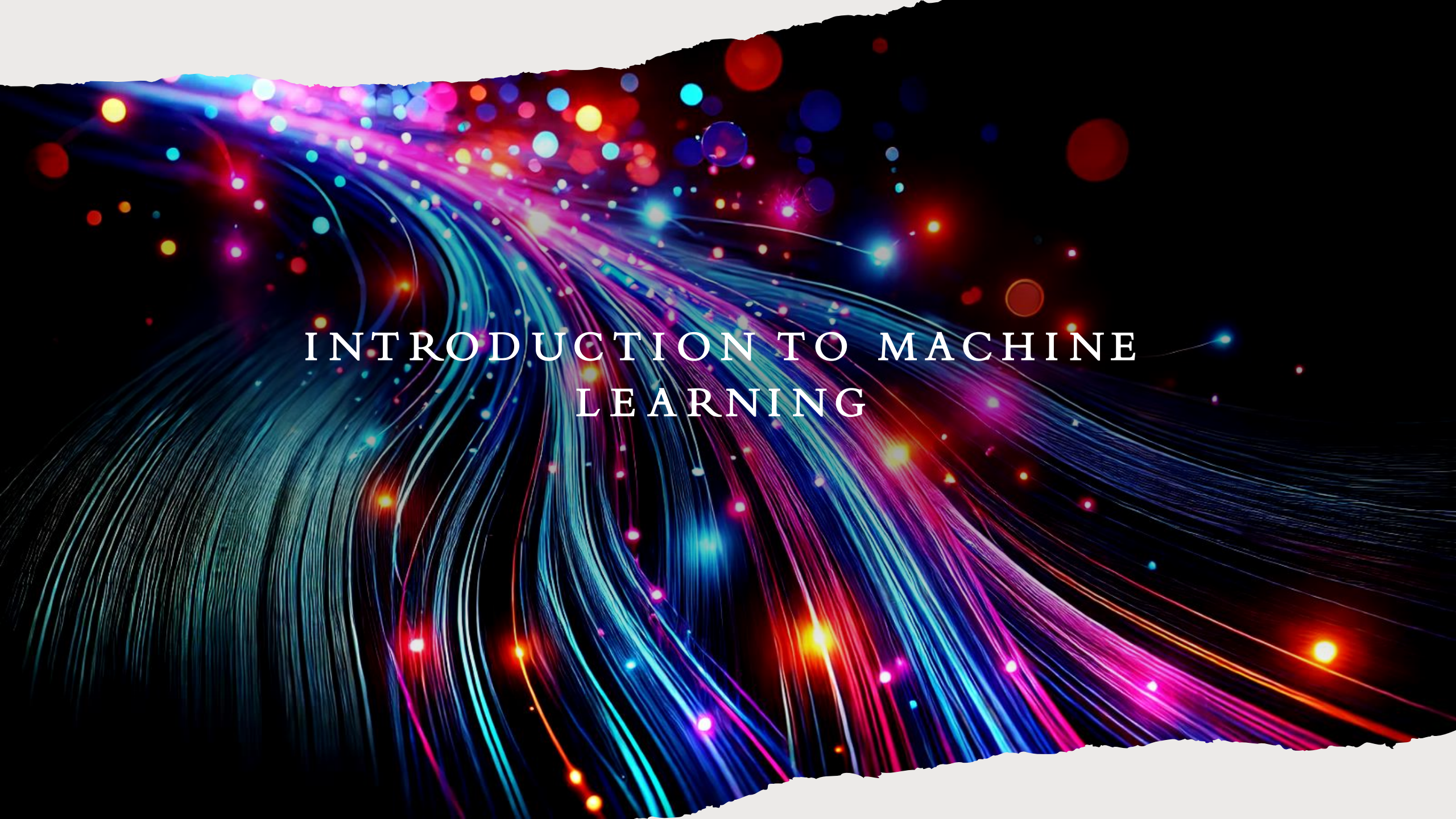
The Register

Amazon goes nuclear, acquires Cumulus Data's atomic datacenters for $650M

The Verge / Tech / Reviews / Science / Entertainme

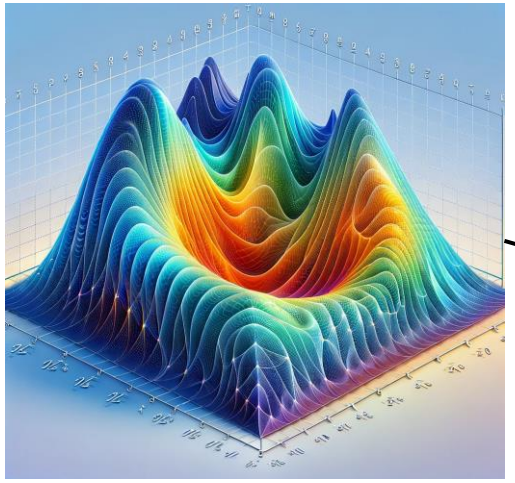
CLIMATE / ENERGY / SCIENCE

Microsoft is going nuclear to power its AI ambitions


INTRODUCTION TO MACHINE LEARNING

WHAT IS MACHINE LEARNING?

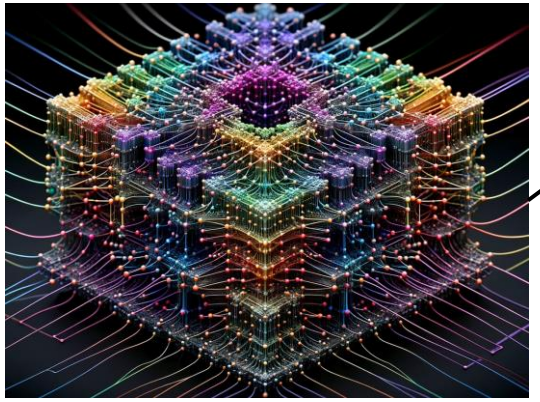
Optimization Theory



Mathematics / Analysis

Machine Learning

Hardware / Computation



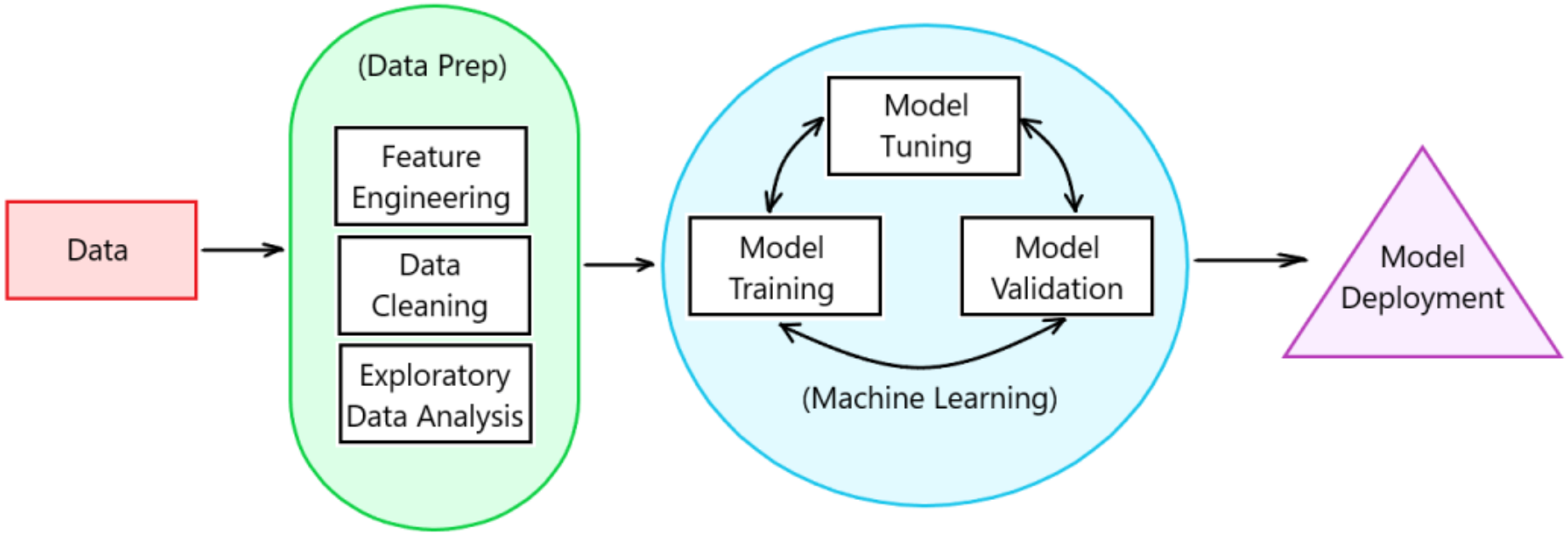
Probability & Statistics

WHAT IS MACHINE LEARNING?

"A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*." – **Tom M. Mitchel**, computer scientist at Carnegie Mellon University

Learning (in the context of ML): The process of identifying patterns in data. This is typically achieved by iteratively updating (via an algorithm) the parameters in a mathematical model in such a way that optimizes some performance metric.

The Typical Data Mining / ML Pipeline



THE CATEGORIES OF LEARNING

Supervised Learning: A category of problems where one has a set of datapoints that have features along with corresponding target labels. The goal of these problems is to "learn" the best model that can accurately determine the target class label (for a categorical target) or the target numerical value (for a numerical target) given the feature vectors of the dataset.

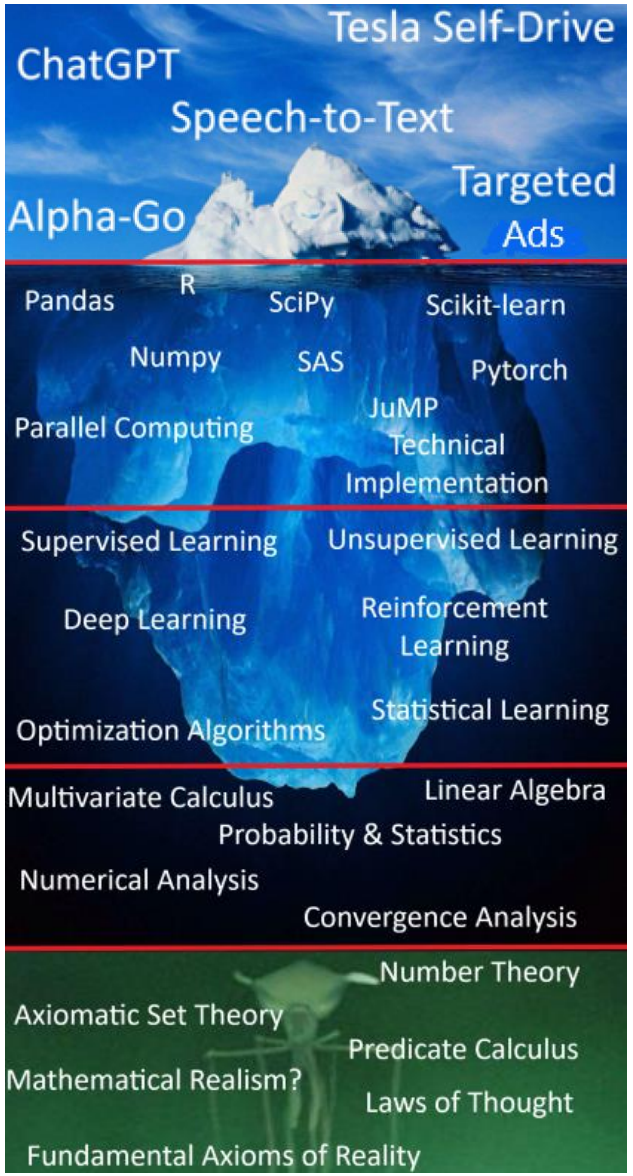
- **Discriminative Models:** Models that aim at directly learning (estimating) the posterior probability distribution $\mathbb{P}(Y|X)$. Some of these models would be linear & logistic regression, K-NN, tree-based models, support vector machines (SVM), deep neural networks (DNNs), etc.
- **Generative Models:** Models that aim at learning the joint probability distribution $\mathbb{P}(X, Y)$, from which the model can then generate posterior probabilities $\mathbb{P}(Y|X)$ by utilizing Bayes' Theorem. Some of these models would be gaussian discriminative analysis (GDA), naïve Bayes, generative adversarial networks (GANs), etc.

Unsupervised Learning: A category of problems where one has a set of datapoints that have only features and does not have any corresponding target labels. The goal of these problems is to "learn" some overall patterns that are present in the features of the dataset such as natural groupings or similarities amongst the datapoints. Some of these models would be K-means clustering, gaussian mixture models (GMMs), etc.

Reinforcement Learning (not in this class): A category of problems where one is concerned with imbuing a deciding agent with an optimal strategy to achieve some predefined goal through a series of rewards and punishments. This essentially amounts to the agent "learning" the best choices (otherwise known as the policy) and, as a result, is the most similar to the way that humans learn.

Dimensionality Reduction: A category of problems that aim at reducing the feature space of a dataset and/or condensing and preserving the most amount of information if a dataset into a reduced set of new features. Some of these models would be principal component analysis (PCA), linear discriminant analysis (LDA), t-distributed stochastic neighbor embedding (t-SNE), uniform manifold approximation and projection (UMAP), etc.

WHAT IS THE KNOWLEDGE HIERARCHY OF ML?



High-Level ML
& A.I. Applications

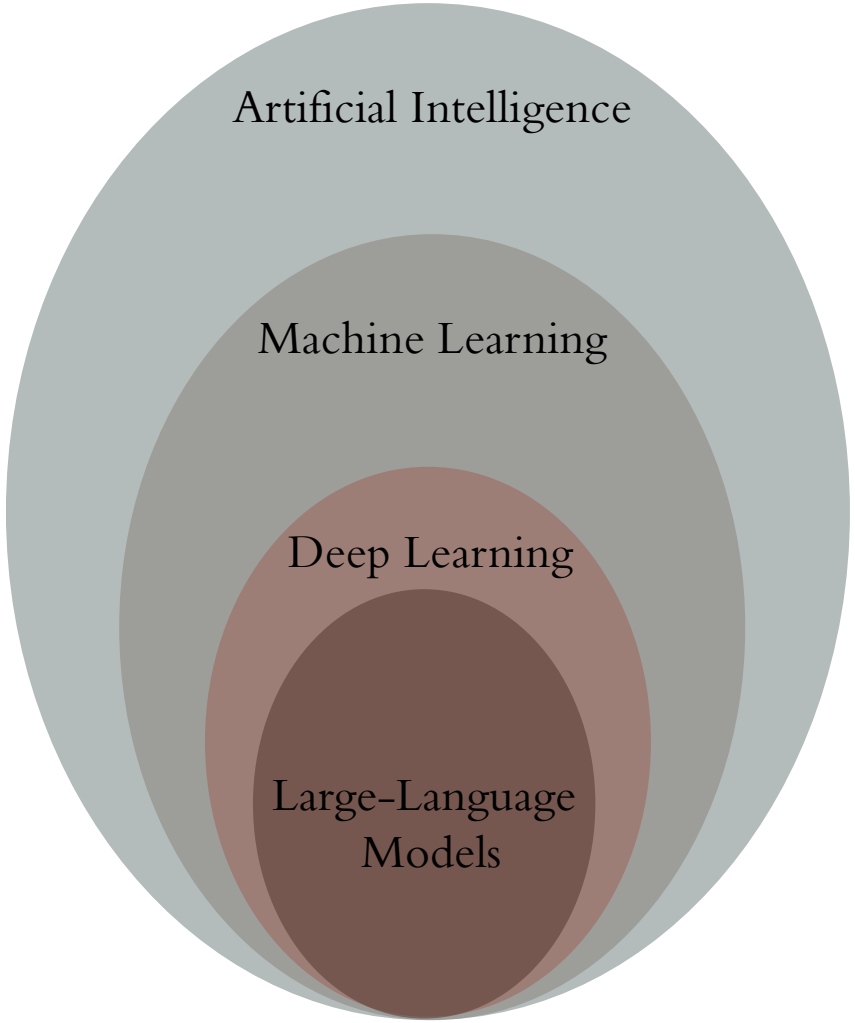
Technical
Implementations
of ML Algorithms

Optimization & Learning
Algorithms

All the Underlying
Mathematics

Foundational Logic &
Meta-Mathematics

The Modern-Day Hype Within A.I.



FOUNDATIONAL PHILOSOPHY

A WORD FROM FEYNMAN

(AN APPROACH TO LEARNING)

How do you find out what's True?

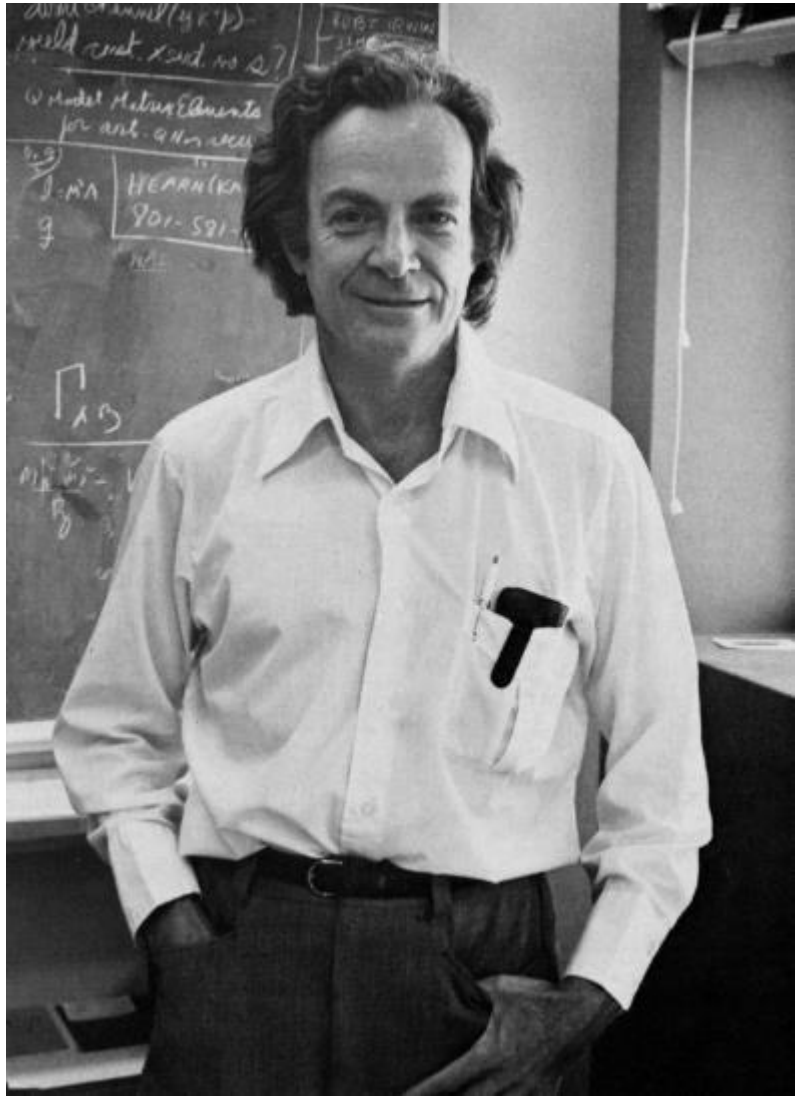
Be skeptical…

"**We never are right; we can only be sure that we're wrong.**" –
Richard Feynman lecturing on the Scientific Method

- In this class, we are interested in understanding the "**why**" of machine learning: "**why**" does it work, "**how**" can a machine learn, and "**what**" is going on at the fundamental level?
- To answer a "**why**" question is a difficult task.
- To start, you need to be in a framework where you allow something to be **true**.
- The natural question is: **how do we justify** that truth at the fundamental level?

The Feynman Method (i.e., Cartesian Skepticism)

(1) Cast doubt on all the beliefs that you hold (clear the slate). **(2)** Re-examine all these propositions carefully and one at a time to ensure that they can be rigorously justified at a fundamental level. **(3)** Proceed to rebuild a consistent belief system.



Richard Feynman (1918 – 1988)

RENE DESCARTES

"COGITO ERGO SUM"

In the work *Meditations on First Philosophy*, **Rene Descartes** (the father of the cartesian coordinate system) wanted to refute skepticism, but only after he had formed the most powerful case for it that he could.

Skepticism

The epistemological position that questions the possibility of human knowledge.

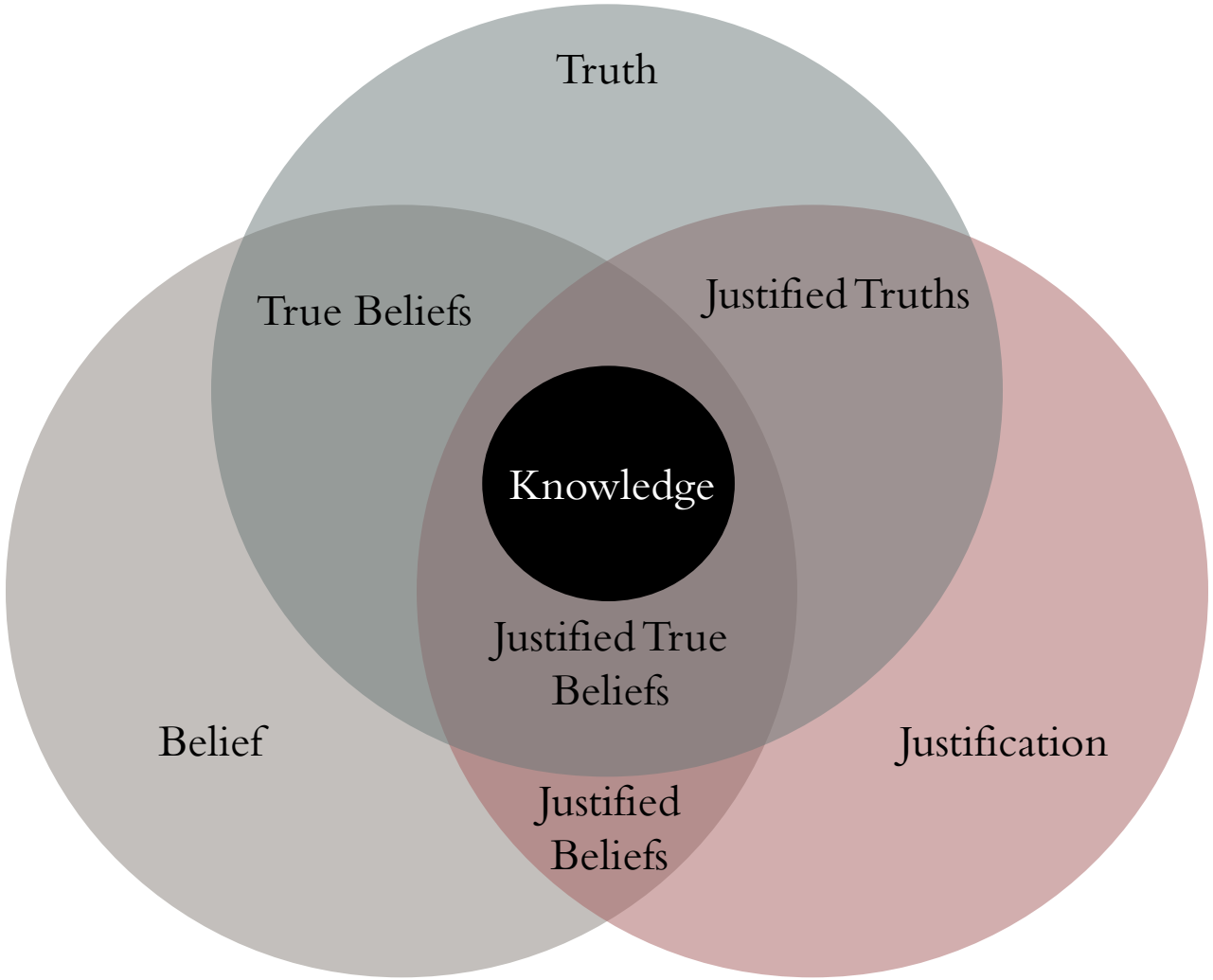
Since all our beliefs hinge on presuppositions which in turn rely on other beliefs all the way down, the only way to ensure that all our beliefs are true is to cast doubt on all of them, and then rebuild them one at a time under careful consideration.

"Cogito Ergo Sum" → "I think; therefore, I am."

Is this classical statement the most fundamental "truth"? Is there anything else more fundamental?

Rene Descartes (1596 – 1650)

DEFINING KNOWLEDGE



THE TWO MOST FUNDAMENTAL LAWS

After some consideration, it is typically arrived upon that one needs to accept two fundamental laws. These laws essentially provide a way through which we can obtain what we call truths.

The Law of Deduction

Anything that is implied by a true proposition is true.

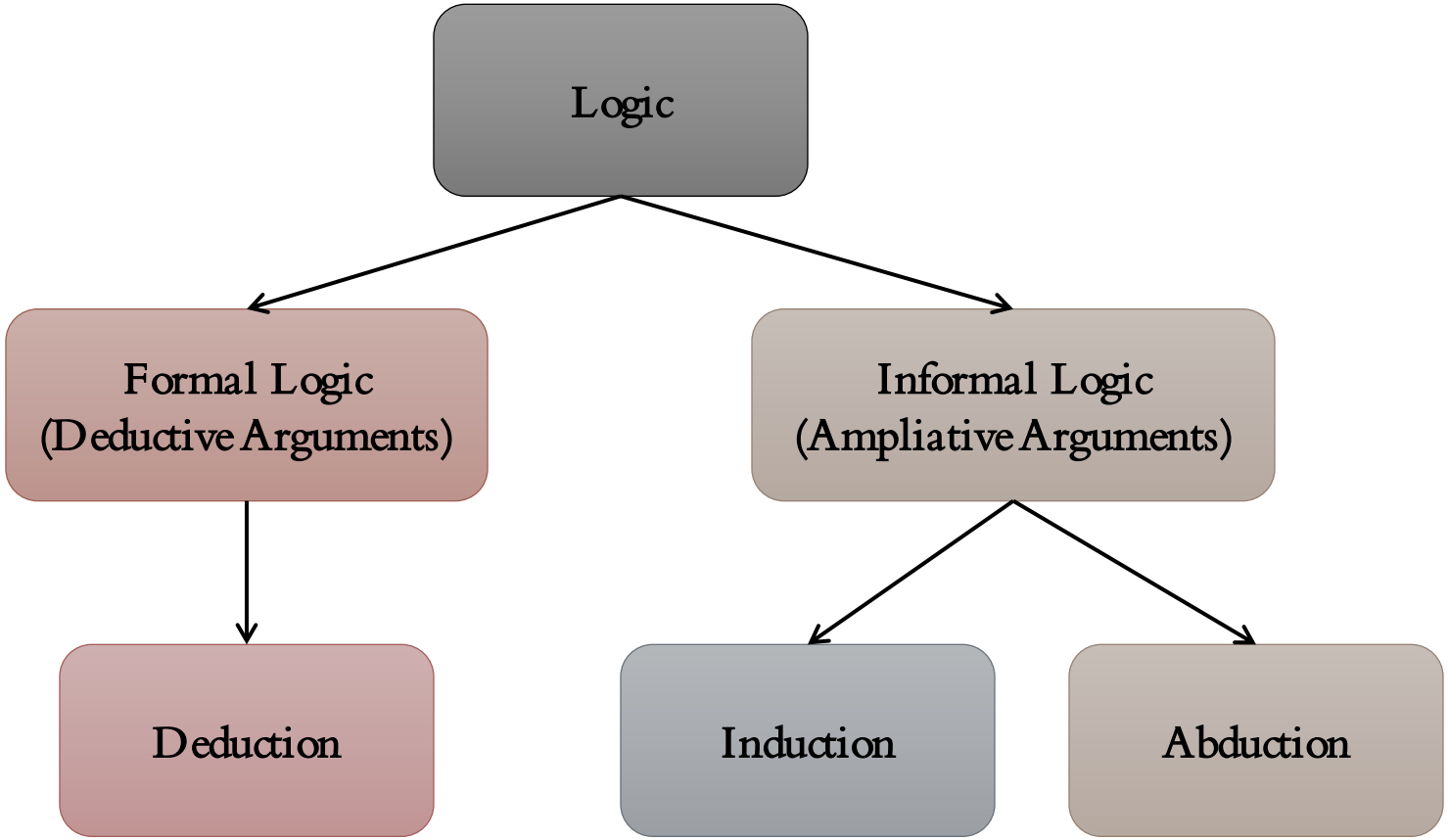
The Law of Induction

The greater the number of cases in which a thing of the sort *A* has been found associated with a thing of the sort *B*, the more probable (if no cases of failure are known) that *A* is always associated with *B*. Thus, as the number of cases of association increases to infinity without contradiction, the general law of association between *A* and *B* will approach certainty.

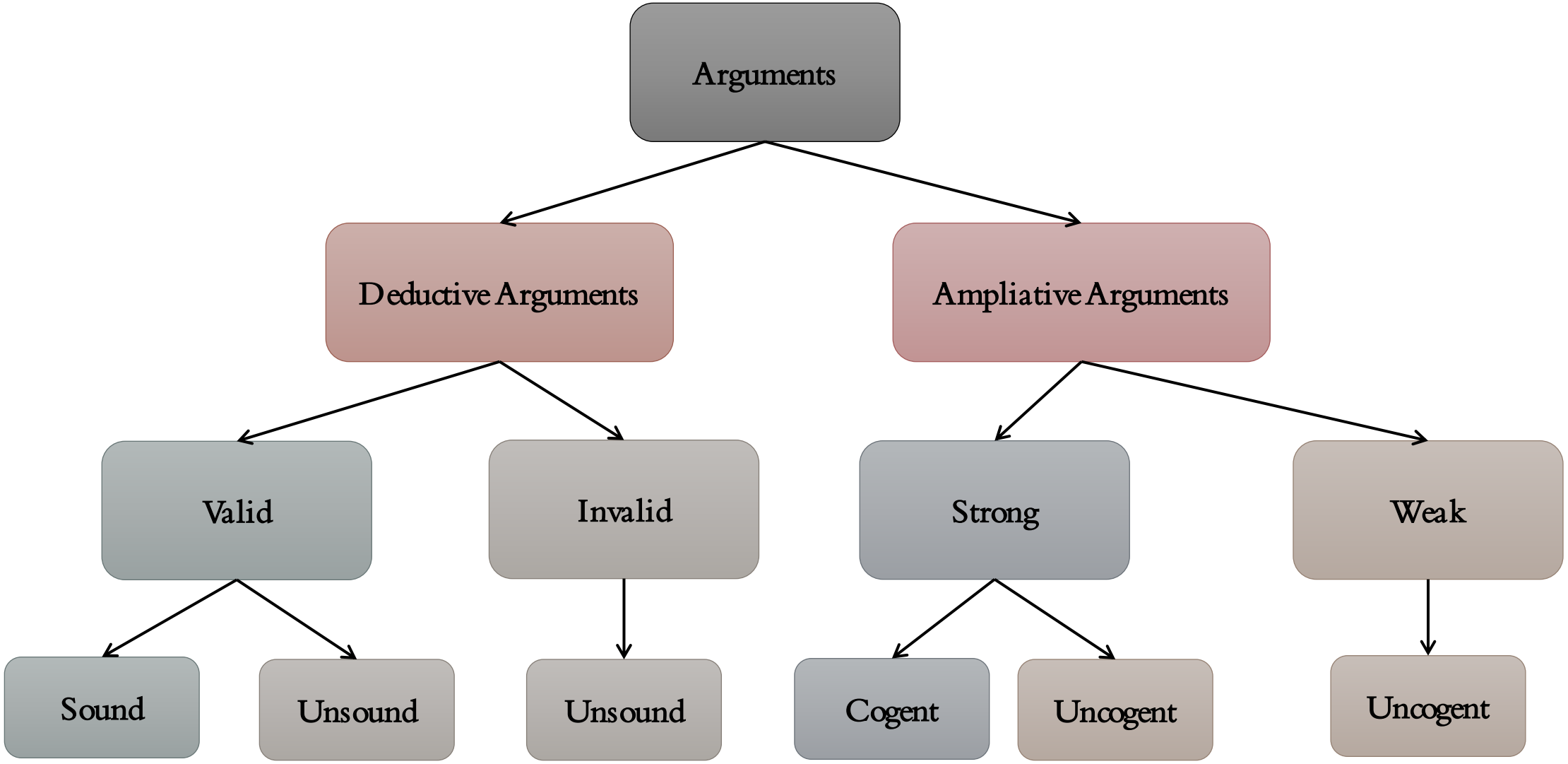
These two laws comprise the **two divisions** of the study of logic: **Formal Logic** and **Informal Logic**.

- Both of these sub-divisions of logic revolve around these two fundamental laws.
- Specifically, Formal Logic provides a way of formalizing the Law of Deduction.
- Similarly, to the way that formal logic formalizes deduction, so to does the **Scientific Method** formalize the practice and implementation of the Law of Induction. As such, the scientific method falls under the umbrella of informal logic.

THE DIVISIONS OF LOGIC



THE TYPES OF LOGICAL ARGUMENTS



FORMAL SYSTEMS

Formal System

A Formal System is an abstract structure for inferring theorems from a set of axioms according to a set of rules. These rules (known as rules of inference when considering systems of logic) are used for carrying out the inference of theorems and are referred to as the logical calculus of the system. A formal system $\mathcal{L}$ is typically defined as $\mathcal{L} \coloneqq (\mathrm{A}, \Omega, \mathrm{Z}, \mathrm{I})$, where

- **(The Alpha Set *A*)** A countably infinite set of propositional variables (symbols). These are the most basic elements of the formal language $\mathcal{L}$ and are typically referred to as atomic formulas. For example, the elements of A could be p, q, r, s, and so on.
- **(The Omega Set *Ω*)** A finite set of elements called operator symbols (or logical connectives). The set Ω is partitioned into disjoint subsets as follows: $\Omega = \Omega_0 \cup \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_j \cup \cdots \cup \Omega_n$, where Ω_j is the set of operator symbols with arity j (the number of arguments or operands taken by a function).
  - For example, in most logical systems, Ω is typically partitioned with the following familiar symbols:

$$\Omega_0 = \{\top, \bot\}, \qquad \Omega_1 = \{\neg\}, \qquad \Omega_2 = \{\wedge, \vee, \rightarrow, \leftrightarrow\}.$$

- **(The Zeta Set *Z*)** A finite set of transformation rules (inference rules when used in a logical system).
- **(The Iota Set *I*)** A countable set of initial points (called axioms when used in a logical system).

The Language (Formulas) of Formal Systems

The language of the system $\mathcal{L}$, also known as the formulas (otherwise known as well-formed formulas (wff)), is inductively defined:

- (Base) Any element of the alpha set A is a wff.
- If $p_1, p_2, \ldots, p_j$ are wffs, and f is in Ω_j, then $f(p_1, p_2, \ldots, p_j)$ is a wff of $\mathcal{L}$.
- (Closed) Nothing else is a wff of $\mathcal{L}$.

PROPOSITIONAL & PREDICATE LOGIC
(CLASSICAL LOGIC)

"To discover truths is the task of all sciences. It falls to logic to discover the laws of truth." – Gottleb Frege

Formal Logic (also known as Symbolic Logic) is perhaps the most fundamentally important field of study as it provides the formal approach to studying reasoning. As such, formal logic deals with the study of those things which can be either true or false, i.e., propositions.

Proposition

A statement, that is either true or false, which makes a claim about how things are. Further, a proposition must, by definition, be well-formed in the sense that it must be either true or false.

Propositional Logic (Propositional Calculus)

The collection of formal systems that deals with propositions and the relations between them. As such, propositional logic serves as the foundation of first-order logic and all higher-order logics as well as mathematics. As we will see, the laws of propositional logic are fundamental.

Predicate Logic (First-Order Logic or Classical Logic)

The collection of formal systems that extend propositional logic to now include quantifiers (such as the quantifier $\exists$ "there exists" as well as the quantifier $\forall$ "for all") and predicates (these are functions of the form $P(x)$, where P is a "predicate" or some property that applies to some x).

ILLUSTRATION OF SYSTEMS OF LOGIC

- The axioms provide the seed of truth.
- The inference rules specify how the truth grows, branching out into infinity.
- The branching growth is never finished.
- Every branch of truths also highlights an opposing tree of falsehoods.
- How?

THE NATURAL DEDUCTION SYSTEM

The most "intuitive" and universal formal system of logic is the **Natural Deduction System (NDS)** and is the one that is typically used when deriving new theorems by hand due to its intuitive nature.

It is a formal system that utilizes the five typical logical operands $\Omega_1 = \{\neg\}$, $\Omega_2 = \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ as well as the "truth" and "falsity" operands $\Omega_0 = \{\top, \bot\}$. However, it utilizes **no axioms**, but instead has an "**introduction**" and "**elimination**" rule of inference for every single one of the five logical operators.

There are other systems of logic (such as Hilbert-type systems) that utilize a small number of axioms and typically just a single rule of inference (Modus Ponens, i.e., Conditional Elimination). However, these systems are typically what are used in proof checking software systems, as they are more efficient in that setting.

(Conditional Introduction)
$$(p \vdash q) \vdash (p \rightarrow q)$$
(Conditional Elimination)
$$\{(p \rightarrow q), p\} \vdash q$$
(Negation Introduction)
$$\{(p \rightarrow q), (p \rightarrow \neg q), \} \vdash \neg p$$
(Negation Elimination)
$$\neg p \vdash (p \rightarrow r)$$
(Conjunction Introduction)
$$\{p, q\} \vdash (p \wedge q)$$
(Conjunction Elimination)
$$(p \wedge q) \vdash p \quad \text{and} \quad (p \wedge q) \vdash q$$
(Disjunction Introduction)
$$p \vdash (p \vee q) \quad \text{and} \quad q \vdash (p \vee q)$$
(Disjunction Elimination)
$$\{(p \vee q), (p \rightarrow q), (q \rightarrow r)\} \vdash r$$
(Biconditional Introduction)
$$\{(p \rightarrow q), (q \rightarrow p)\} \vdash (p \leftrightarrow q)$$
(Biconditional Elimination)
$$(p \leftrightarrow q) \vdash (p \rightarrow q) \quad \text{and} \quad (p \leftrightarrow q) \vdash (q \rightarrow p)$$

META-LOGIC

Now that we have introduced proof systems, we would like to know if a specific system has certain desirable properties. To start, notice that we have two different qualities of interest: "**truth**" ($\vDash$) and "**provability**" ($\vdash$).

- Ideally, we would like these two things to coincide; that is, we should only be able to prove things that are true and, if they are true, we should be able to prove them. These two properties are referred to as **Soundness** and **Completeness**.

Soundness

A formal (proof) system is said to be **sound** if everything that is provable is in fact true. Symbolically, letting $\Gamma \coloneqq (\phi_1, \phi_2, \dots, \phi_n)$ denote a sequence of n well-formed formulas, then if $\Gamma \vdash \psi$, it follows that $\Gamma \vDash \psi$, where ψ is a well-formed formula of the system.

Consistency

A formal (proof) system is said to be **consistent** if cannot syntactically prove a contradiction within itself. Symbolically, letting $\Gamma \coloneqq (\phi_1, \phi_2, \dots, \phi_n)$ denote a sequence of n well-formed formulas, then Γ is said to be consistent if it cannot prove both ψ and $\neg\psi$, where ψ is a well-formed formula. Thus, Γ is consistent if $!\exists\psi$ such that $\Gamma \vdash \psi$ and $\Gamma \vdash \neg\psi$.

Completeness

A formal (proof) system is said to be **complete** if there exists a proof of all true statements in the language of that system. Symbolically, letting $\Gamma \coloneqq (\phi_1, \phi_2, \dots, \phi_n)$ denote a sequence of n well-formed formulas, then if $\Gamma \vDash \psi$, it follows that $\Gamma \vdash \psi$, where ψ is a well-formed formula of the system.

PHILOSOPHY OF LOGIC

It can be shown that the classical system of logic is **both sound and complete**.

- This is one of the **most important facts** in the field and is what makes the system of **classical logic** the gold standard.

Is it reasonable to be skeptical of logic?

- Essentially by definition, no…
- What would doubting the validity of the laws of logic look like?
- "If someone does not value reason or evidence, what reason and what piece of evidence are you going to use to convince them that they should value these things?"
- There comes a point where you hit rock bottom and faith must take over.

A Note on Ultimate Skepticism

Skepticism is essentially a **destructive force** and by itself is not really a way to live.

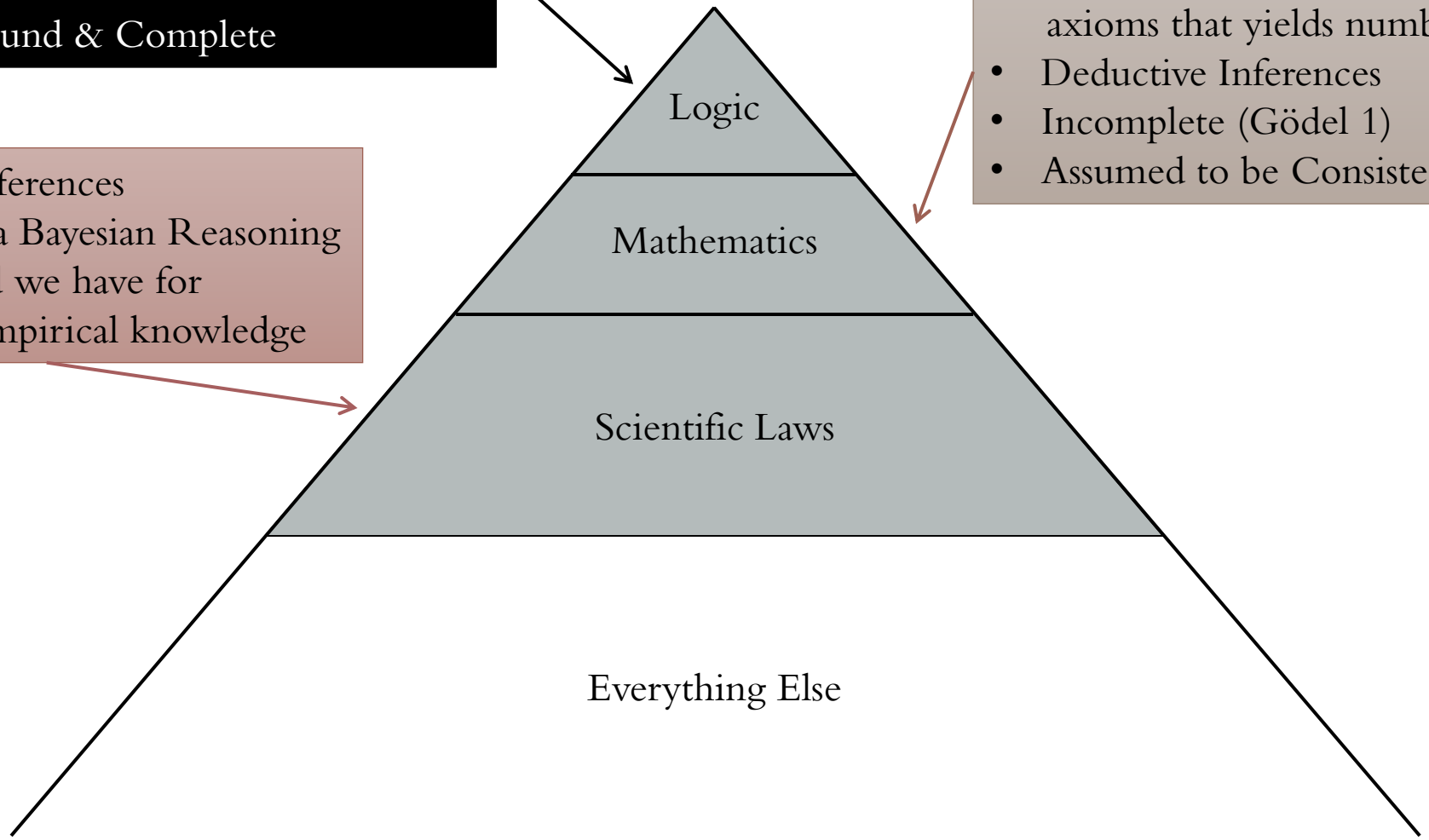
- It should be viewed as a **tool** to be used in the greater search for truth.
- "Skepticism is not an end in itself; it is a tool for the discovery of truths."
- The goal that we wish to get out of skepticism is to fully understand exactly where our assumptions and beliefs come in the picture and to understand at what point things can ultimately no longer be further justified.
- This ruthless adherence to the truth is what will yield better insights, in analytics and in life.

THE HIERARCHY OF TRUTH

- Propositional & Predicate Logic
- Tautologies
- Deductive Inferences
- Sound & Complete

- Builds on Classical Logic by adding axioms that yields numbers
- Deductive Inferences
- Incomplete (Gödel 1)
- Assumed to be Consistent (Gödel 2)

- Inductive Inferences
- Obtained via Bayesian Reasoning
- Best method we have for obtaining empirical knowledge



REFERENCES

- Amini, Amini, et al. "Spatial Uncertainty Sampling for End-to-End Control." *arXiv*, 2019, https://doi.org/10.48550/arXiv.1805.04829.
- Kaplan, Jared, et al. "Scaling Laws for Neural Language Models." *arXiv*, 2020, https://doi.org/10.48550/arXiv.2001.08361.
- Turing. "Computing Machinery and Intelligence." Mind 49: 433 – 460, 1950.
- https://image-net.org/index.php
- https://www.theregister.com/2024/03/04/amazon_acquires_cumulus_nuclear_datacenter/
- https://www.theverge.com/2023/9/26/23889956/microsoft-next-generation-nuclear-energy-smr-job-hiring

ATTRIBUTION & LICENSE

These lecture slides are part of the "*Introduction to Machine Learning*" course materials created by **Griffin Dean Kent**.

For the latest version, updates, and additional resources, visit the GitHub repository: https://github.com/GdKent/Introduction-to-Machine-Learning.