Expert System and Artifical Intelligance

BM-766

Lecture Outline

- Personal introduction
- What is artificial intelligence (and course focus)?
- Review course syllabus
- Introducing and Installing:
  - Python
  - Anaconda
- Coding Environments
- Jupyter Notebooks
- Learning Python Coding
- Coding Help

My Background

Education:

B.Sc. Agriculture Engineering - University of Agriculture Faisalabad
M.Sc. Computer Sciecne - University of Agriculture Faisalabad
MS Computer Sceicne - University of Agriculture Faisalabad
Ph.D Computer Sceicne – University Teknologi Malaysia
Post-Doc – Gangzhou University China

Currently:

Assistant Professor - University of Agriculture Faisalaabd
Project PI – Blockchain & IoT (1.3 Million)
Project PI – CNC Kitchen Gardeing Robot (13.2 Million)
Project Co-PI – Precision Agriculture Lab NCBC (111 Million)
Project Co-PI – Computation & Analytics PKNC (1000 Million)

Research Intreast:

AI & Data Analytics
Privacy
Blockchain & IoT

Goals as Instructor

- Clarify ‘why’ each topic is relevant
- Present the topics from a primarily conceptual perspective, while providing opportunities for students to apply these topics *in silico*.
- Welcome feedback throughout the year
- Any unanswered question, I’ll do my best to track down
- Improving the course this year and beyond
  - Integrate useful topics or resources students may identify

Goals for Students

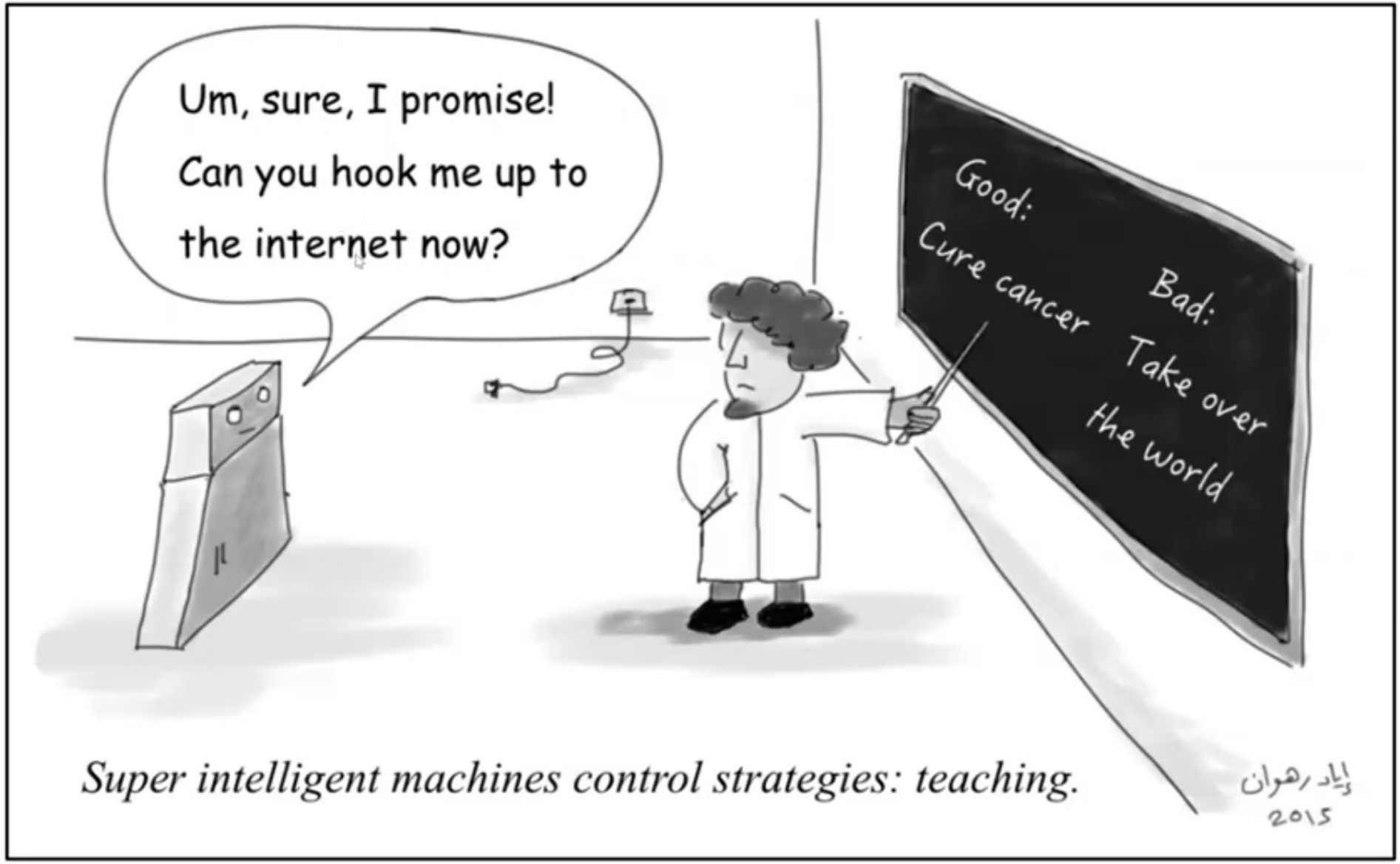
- Understand and develop an appreciation for the unique challenges of representing and applying human knowledge within a machine
- Understand the conceptual differences between deduction and induction and the role of both in AI.
- Develop an understanding and appreciation of what it takes to develop an expert system for real world application
- Learn and/or apply the basics of Python coding and markdown

What is Artificial Intelligence (AI)?


AI


CHAPPIE

Concerns about AI

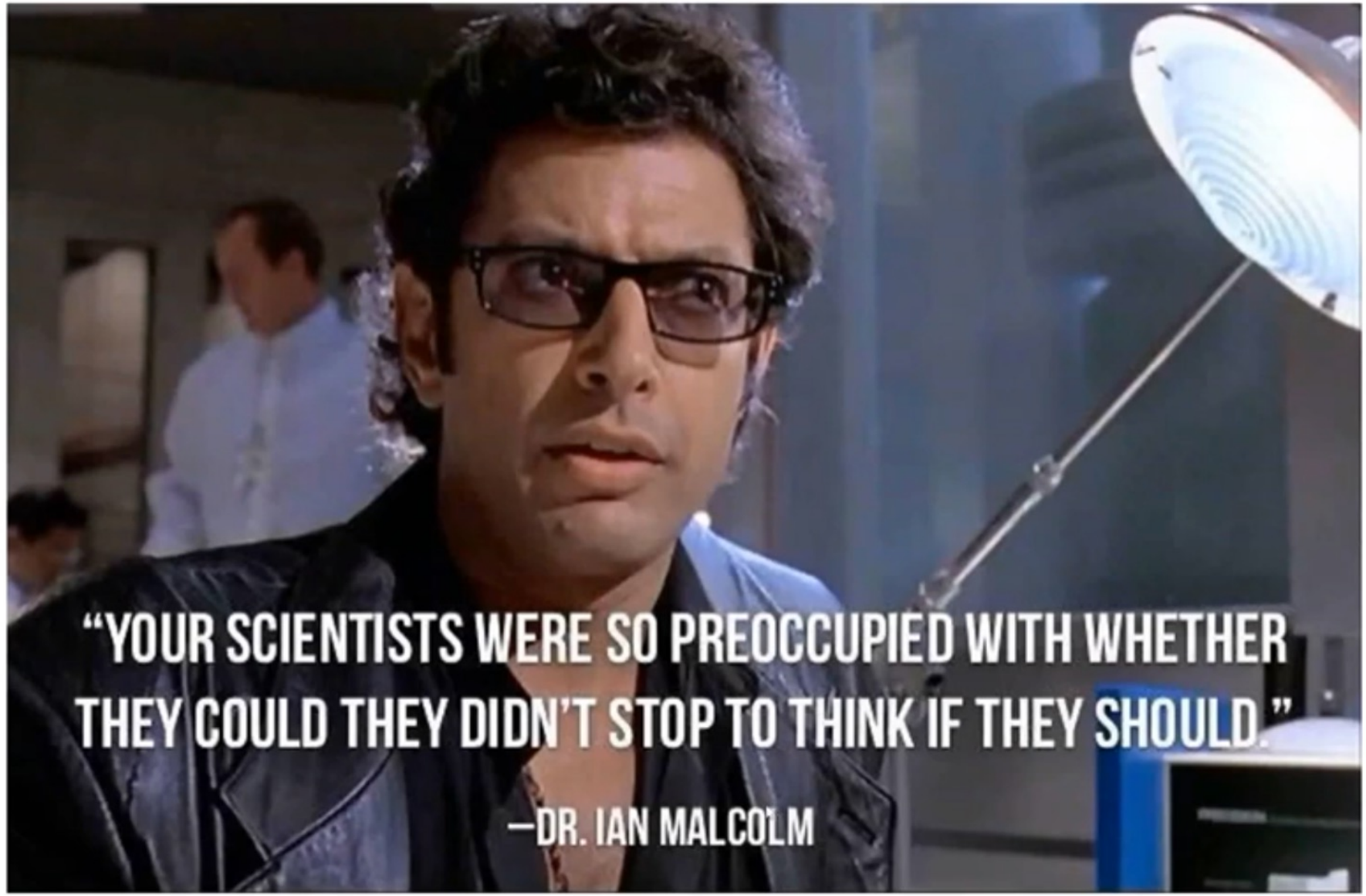


Technological Singularity

- The **singularity:** is a hypothetical point in time at which technological growth becomes uncontrollable and irreversible, resulting in unforeseeable changes to human civilization.

- AKA: **intelligence explosion**, an upgradable intelligent agent will eventually enter a "runaway reaction" of self-improvement cycles, each new and more intelligent generation appearing more and more rapidly, causing an "explosion" in intelligence and resulting in a powerful superintelligence that qualitatively far surpasses all human intelligence.

Ethical Responsibility in Developing AI



WIRED Opinion

Elon Musk is wrong. The AI singularity won't kill us all

Elon Musk has stirred up fear, yet again, over the threat of killer AI. But he's missing the point completely, argues professor Toby Walsh

By **TOBY WALSH**

Wednesday 20 September 2017

Credit: Brendan Smialowski / Getty

- "Most people working in AI like myself have a healthy skepticism for the idea of the singularity."
- "We know how hard it is to get even a little intelligence into a machine, let alone enough to achieve recursive self-improvement."

Definitions of Artificial Intelligence (AI)

- **AKA: Machine Intelligence**

- *"Intelligence demonstrate by machines…"* - [Russell & Norvig 2009]
  - Machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving"

Definitions of Artificial Intelligence (AI)

- **AKA: Machine Intelligence**

- *"Intelligence demonstrate by machines…"* - [Russell & Norvig 2009]
  - Machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving"

- *"Machines that behave as though they were intelligent"* - [McCarthy 1955]

Definitions of Artificial Intelligence (AI)

- **AKA: Machine Intelligence**

- *"Intelligence demonstrate by machines…"* - [Russell & Norvig 2009]
  - Machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving"

- *"Machines that behave as though they were intelligent"* - [McCarthy 1955]

- *"The ability of digital computers or computer controlled robots to solve problems that are normally associated with higher intellectual processing capabilities of humans"* - [Encyclopedia Britannica 1991]

Definitions of Artificial Intelligence (AI)

- **AKA: Machine Intelligence**

- *"Intelligence demonstrate by machines…"* - [Russell & Norvig 2009]
  - Machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving"

- *"Machines that behave as though they were intelligent"* - [McCarthy 1955]

- *"The ability of digital computers or computer controlled robots to solve problems that are normally associated with higher intellectual processing capabilities of humans"* - [Encyclopedia Britannica 1991]

- *"The study of how to make computers do things at which, at the moment, people are better"* - [Rich 1983]

Definitions of Artificial Intelligence (AI)

- **AKA: Machine Intelligence**

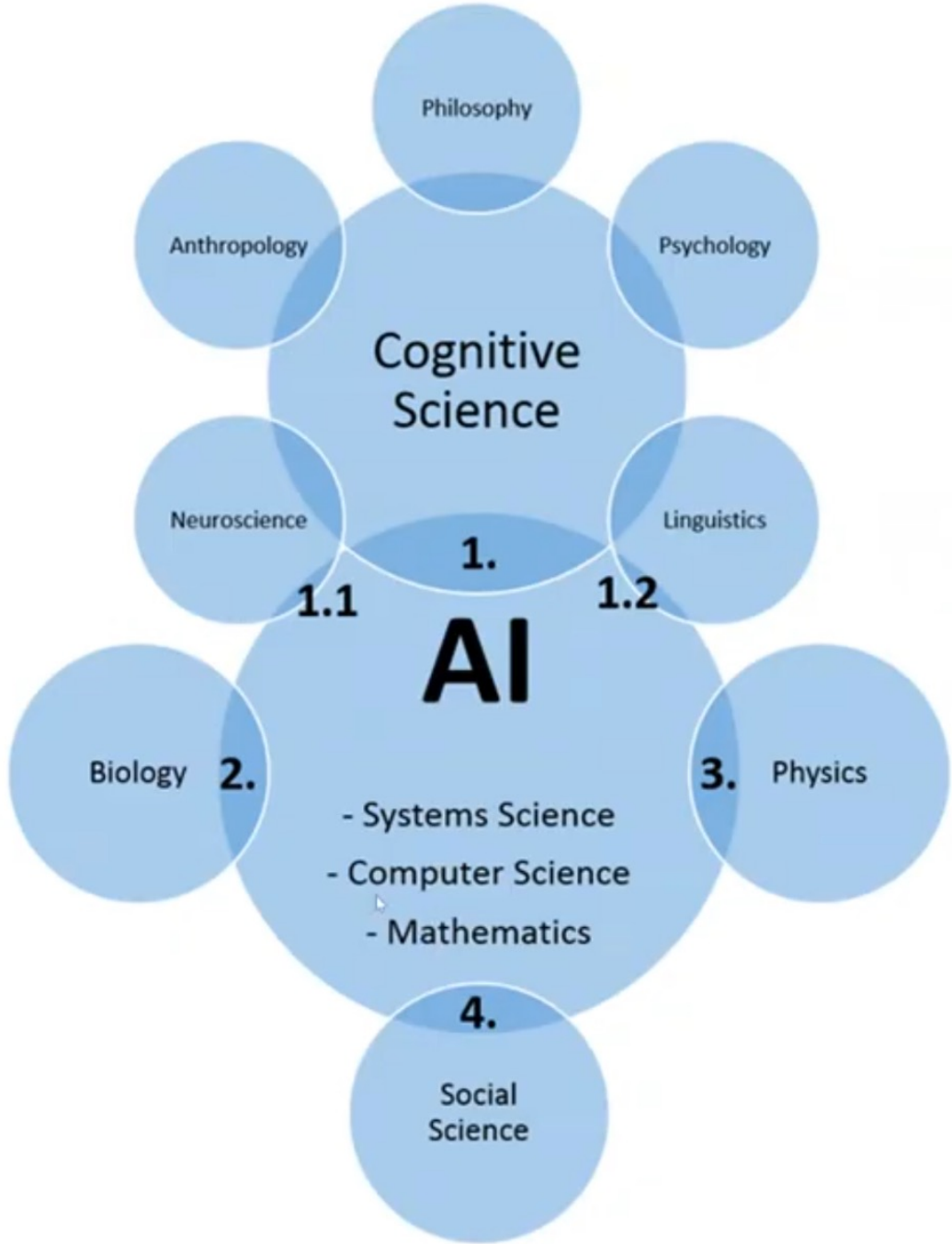
- *"Intelligence demonstrate by machines…"* - [Russell & Norvig 2009]
  - Machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving"

- *"Machines that behave as though they were intelligent"* - [McCarthy 1955]

- *"The ability of digital computers or computer controlled robots to solve problems that are normally associated with higher intellectual processing capabilities of humans"* - [Encyclopedia Britannica 1991]

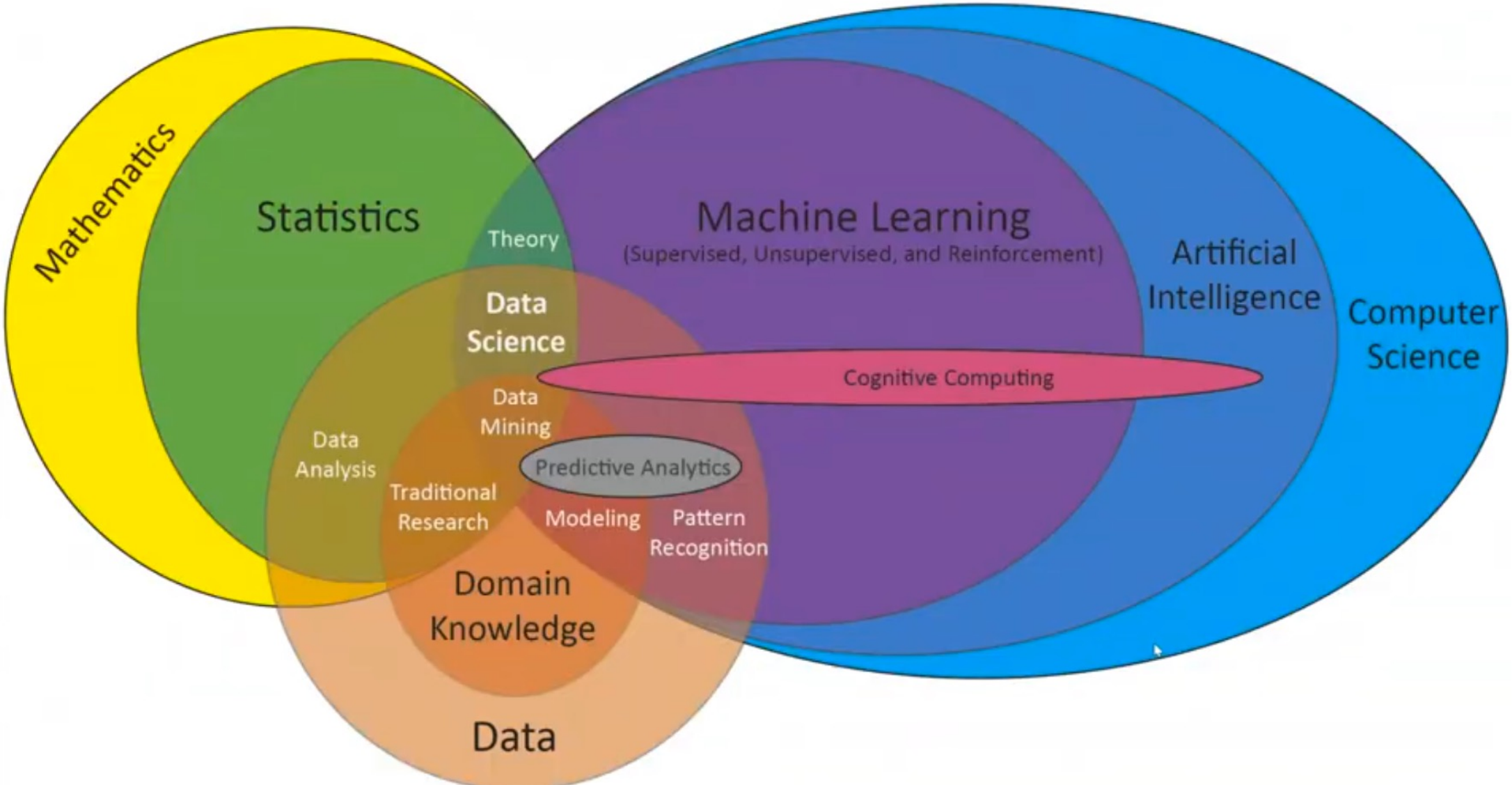
- *"The study of how to make computers do things at which, at the moment, people are better"* - [Rich 1983]

- *"Getting computers to do tasks that require human intelligence"* – [Cawsey 1997]



https://opentechai.blog/2017/11/10/navigating-in-the-ai-sciences-jungle/

Fields & Terms Related to AI & ML



What is Machine Learning (ML)?

- Computational strategies that, given data, are designed to progressively improve performance on a specific task without being explicitly programmed.

What is Machine Learning (ML)?

- Computational strategies that, given data, are designed to progressively improve performance on a specific task without being explicitly programmed.

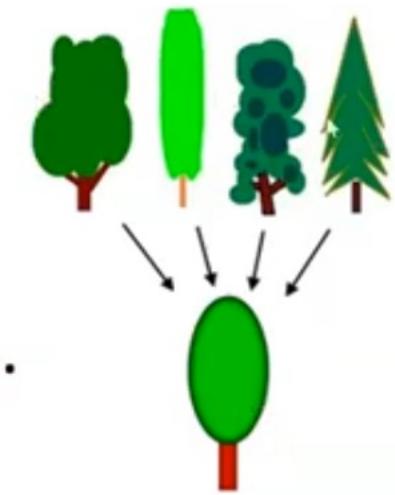
- ML includes many methods/algorithms.

What is Machine Learning (ML)?

- Computational strategies that, given data, are designed to progressively improve performance on a specific task without being explicitly programmed.

- ML includes many methods/algorithms.

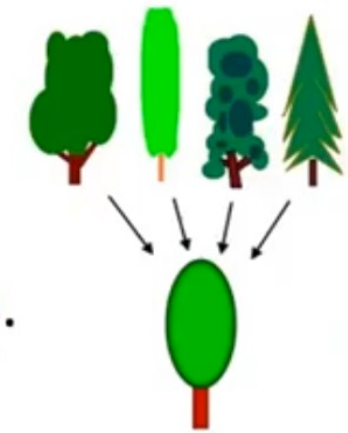
- Big Picture Goal: Learning useful generalizations.



What is Machine Learning (ML)?

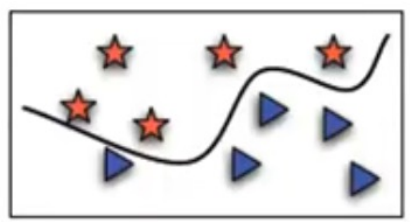
- Computational strategies that, given data, are designed to progressively improve performance on a specific task without being explicitly programmed.

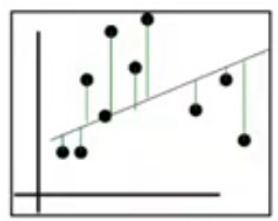
- ML includes many methods/algorithms.



- Big Picture Goal: Learning useful generalizations.

- Supervised Learning Labeled Data

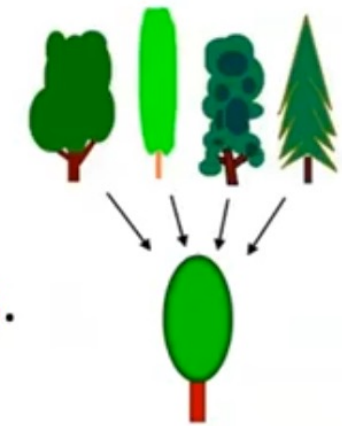




What is Machine Learning (ML)?

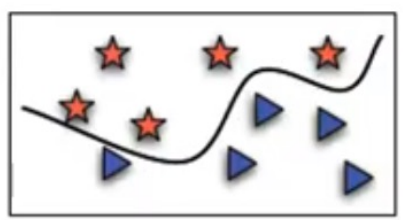
- Computational strategies that, given data, are designed to progressively improve performance on a specific task without being explicitly programmed.

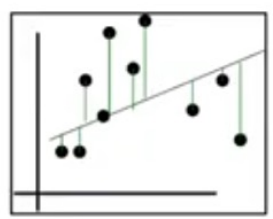
- ML includes many methods/algorithms.



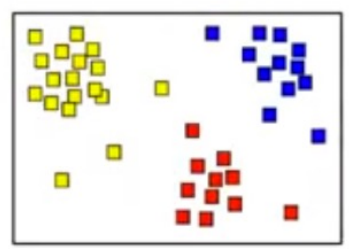
- Big Picture Goal: Learning useful generalizations.

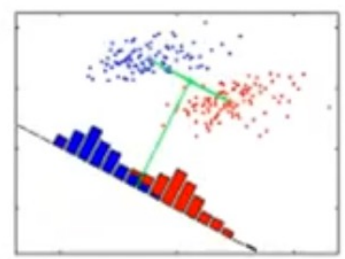
- Supervised Learning Labeled Data





- Unsupervised Learning Unlabeled Data





Why Artificial Intelligence?

- Many tasks which we might reasonably think require intelligence
  - E.g. complex arithmetic – easy for computers

- Some tasks, easy for people, but difficult to automate
  - E.g. recognizing a face, interpreting text

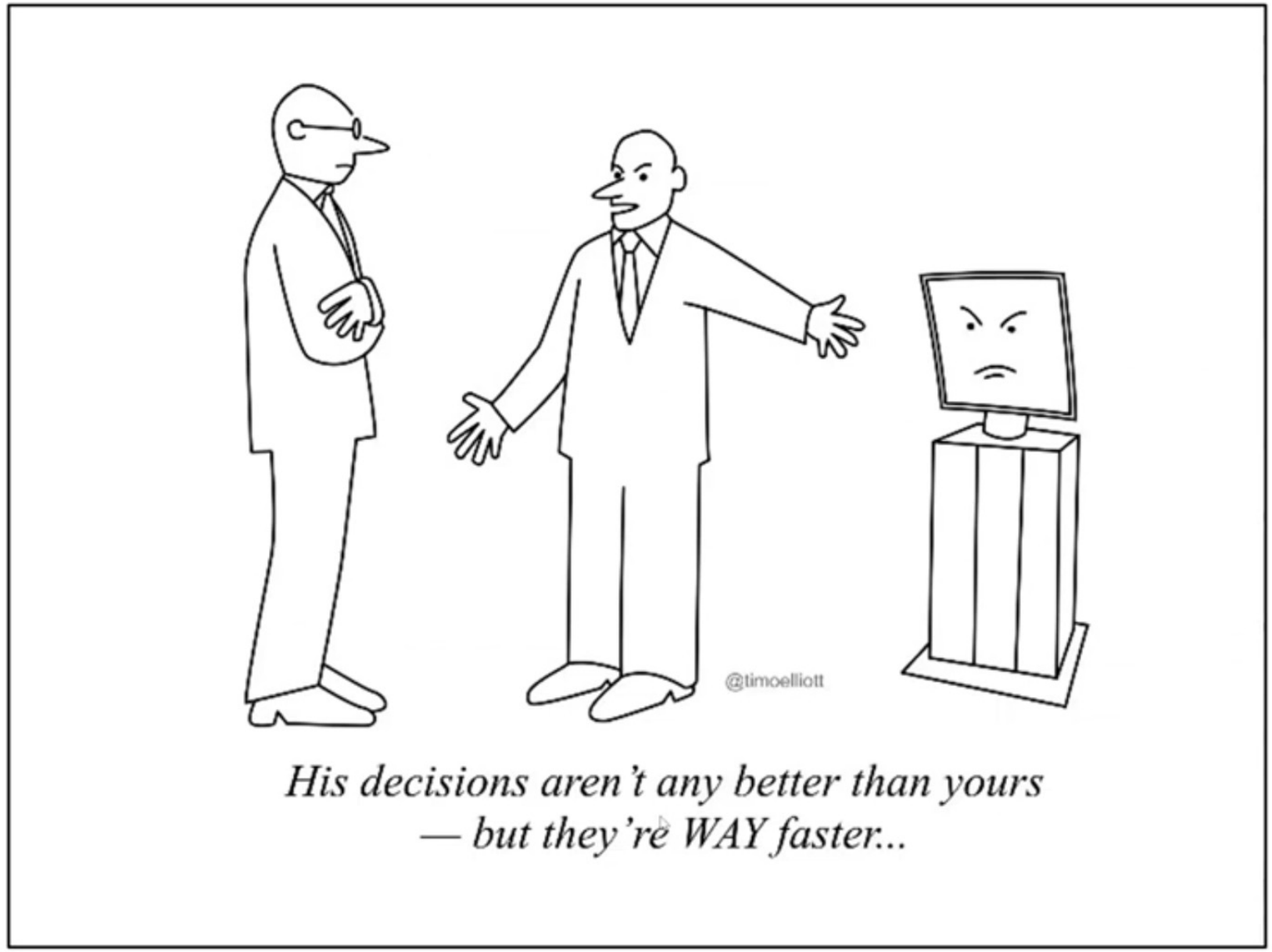
Why Artificial Intelligence?

- Many tasks which we might reasonably think require intelligence
  - E.g. complex arithmetic – easy for computers
- Some tasks, easy for people, but difficult to automate
  - E.g. recognizing a face, interpreting text
- AI is concerned with these difficult tasks which require complex and sophisticated reasoning processes and knowledge.

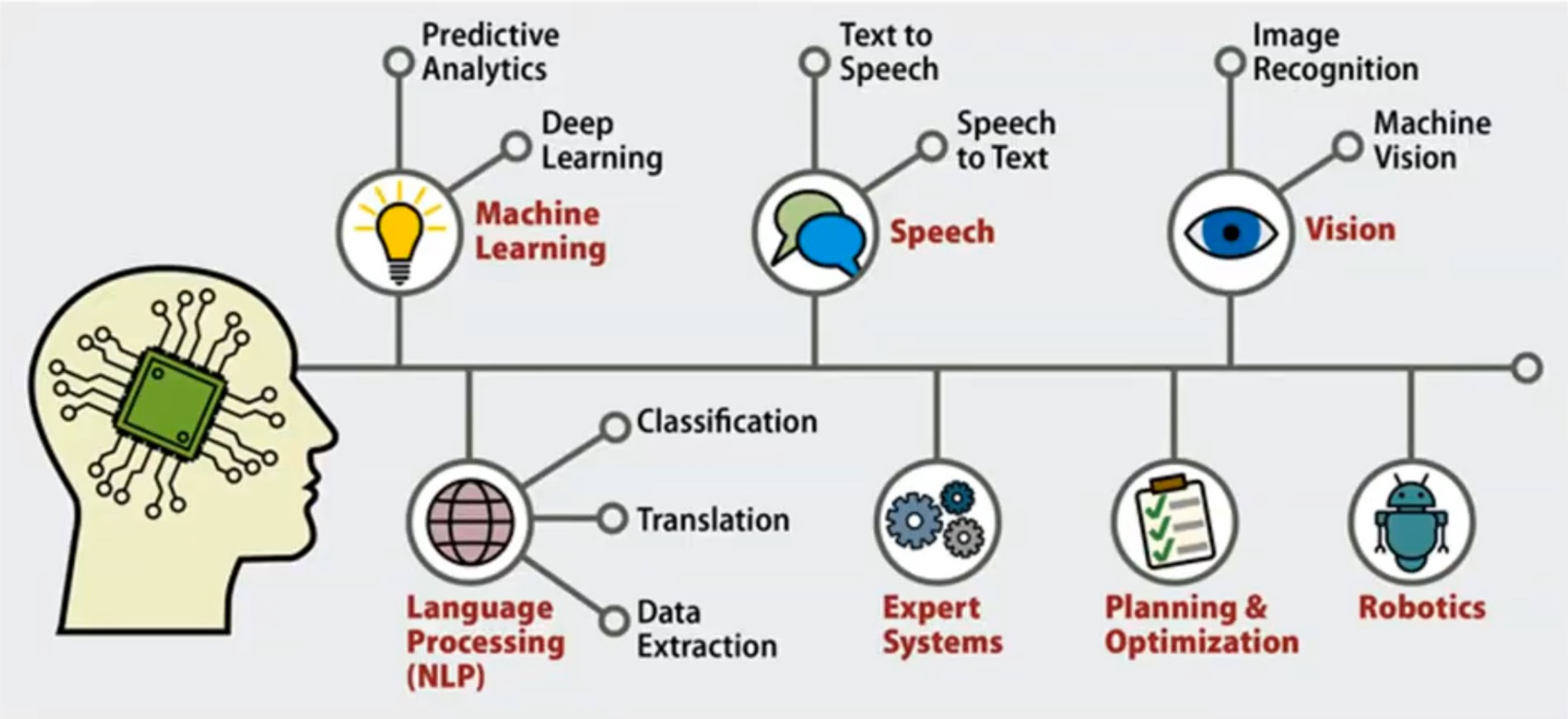
Why Artificial Intelligence?

- Many tasks which we might reasonably think require intelligence
  - E.g. complex arithmetic – easy for computers
- Some tasks, easy for people, but difficult to automate
  - E.g. recognizing a face, interpreting text
- AI is concerned with these difficult tasks which require complex and sophisticated reasoning processes and knowledge.
- The promise of AI:
  - **Automation:** speed, efficiency, and reduction of manual human effort
  - **Scale:** Performing tasks at a scale that most/all humans cannot achieve
  - **Discovery & Innovation:** Integrating information to find novel patterns, design, and strategies

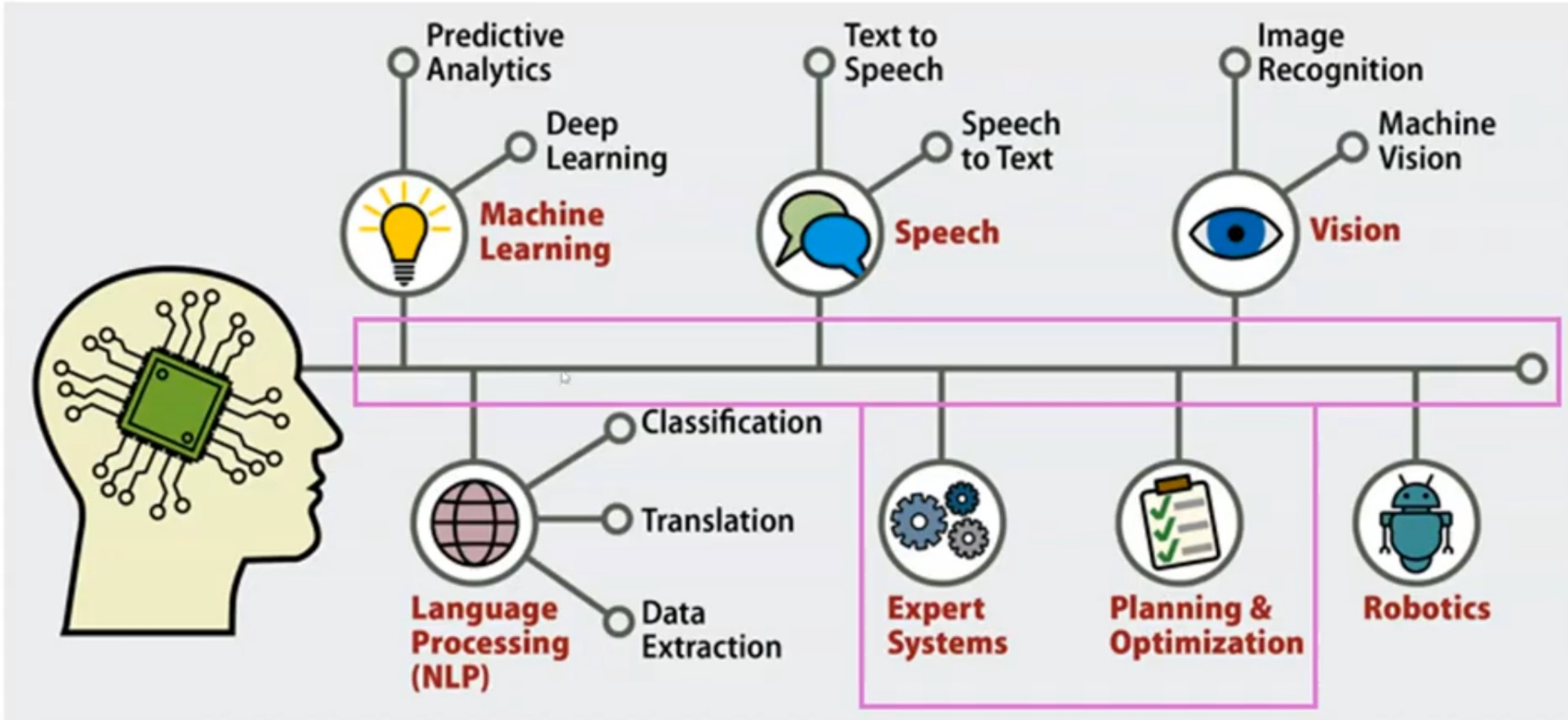
Automation Can Be Enough



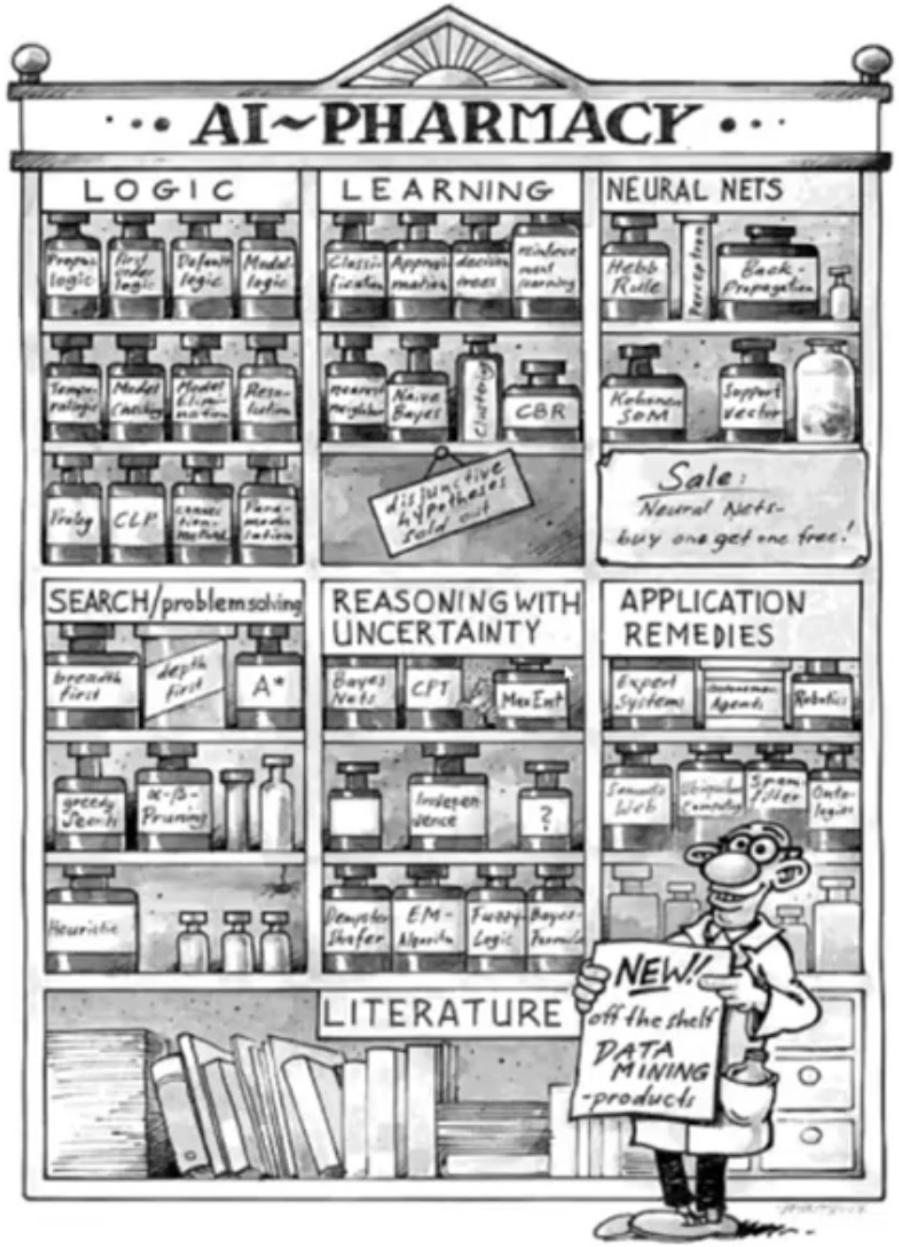
Active Areas of AI Study and Application



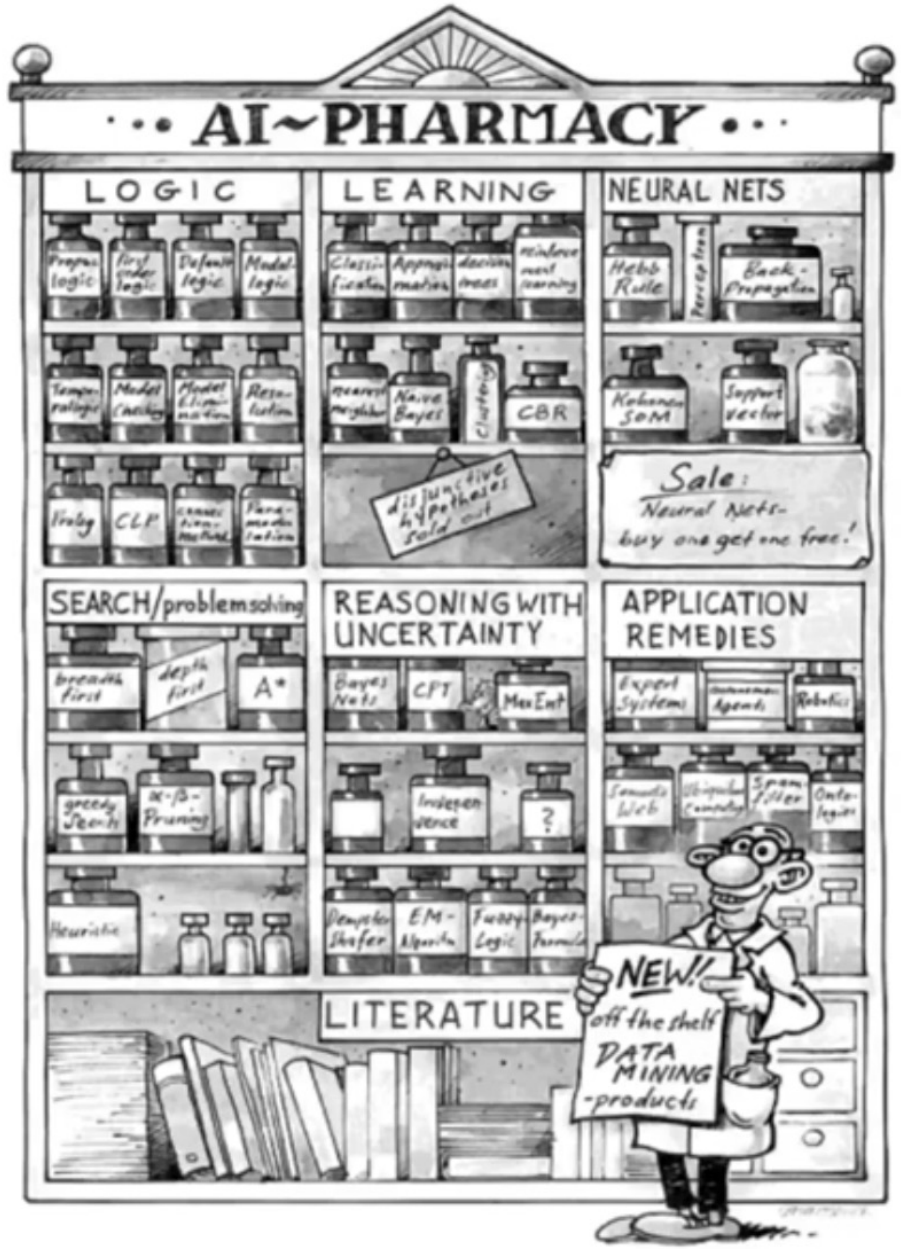
Active Areas of AI Study and Application

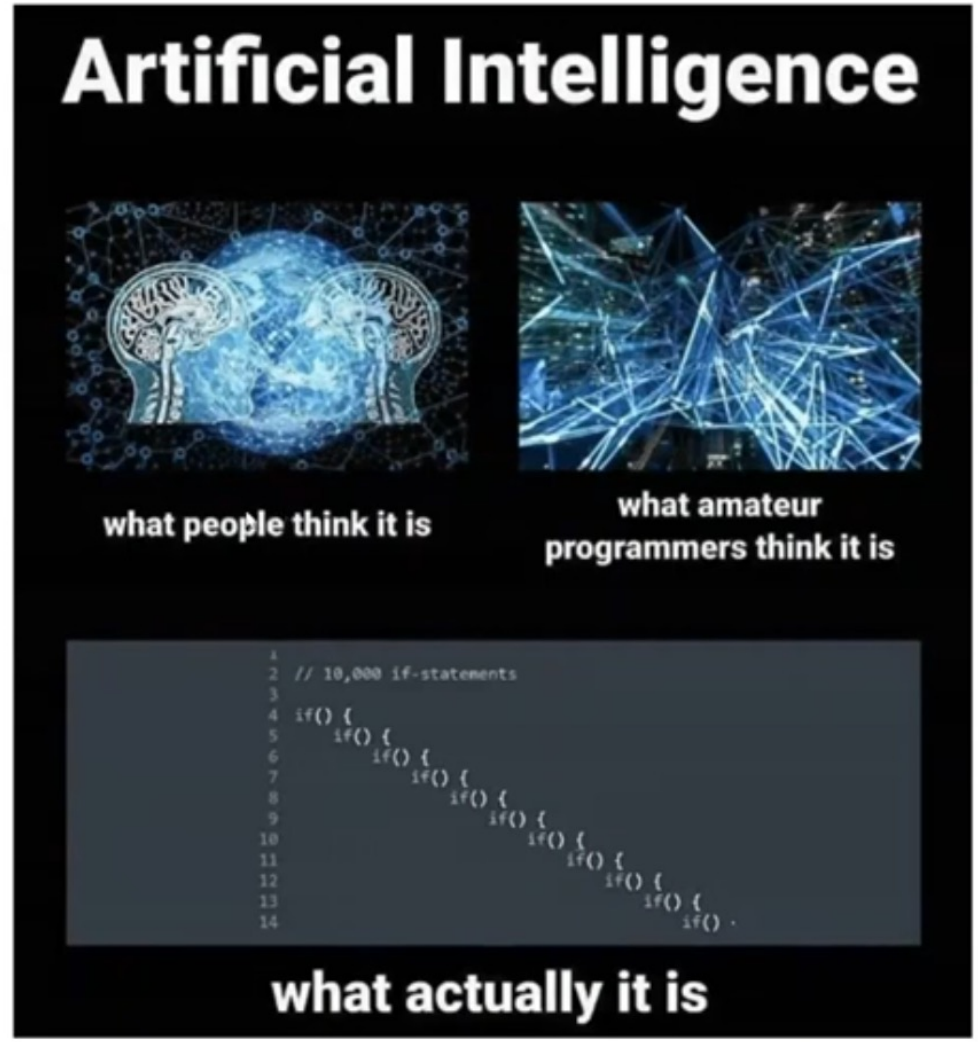


Fundamentals of AI



Fundamentals of AI





Cognitive Computers

- **Definition:** simulate human thought processes in a computerized model.

- **Are:**
  - Made with algorithms
  - Knowledgeable ONLY about what is taught
  - Control ONLY what we give them to control
  - Can continue to learn more – given data/environmental sensors

- **Can:**
  - Do very boring work for you
  - Often make better, more consistent, objective decisions than humans
  - Perform tasks faster and without tiring

AI Largely Dependent on Experts

- Subject matter experts (SMEs)
  - Clinicians
  - Researchers
  - Lawyers
  - Machinists
  - Insurance Adjustors
  - Etc..

AI Largely Dependent on Experts

- Subject matter experts (SMEs)
  - Clinicians
  - Researchers
  - Lawyers
  - Machinists
  - Insurance Adjustors
  - Etc..

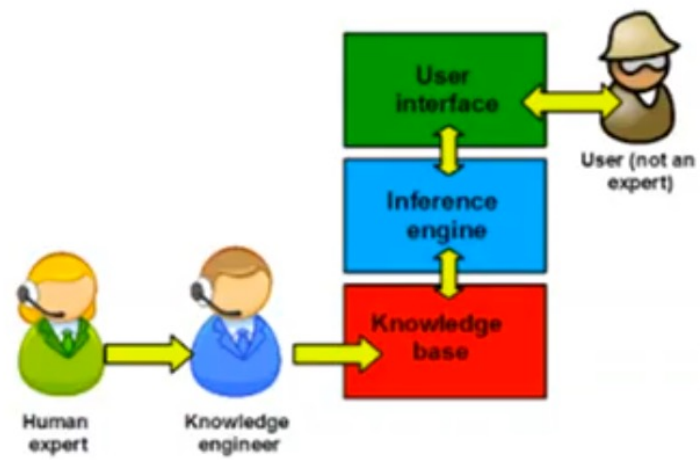
AI Largely Dependent on Experts

- Subject matter experts (SMEs)
  - Clinicians
  - Researchers
  - Lawyers
  - Machinists
  - Insurance Adjustors
  - Etc..

- Usually not experienced in computer science/machine learning
  - A collaboration between domain experts and AI developers is often essential.

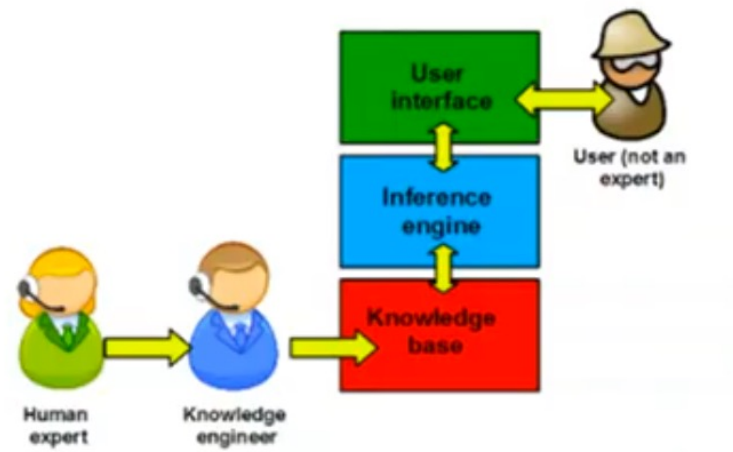
Cognitive Computers = Expert Systems

- What does it take to make one?
  - Content annotated and knowledge specified by experts



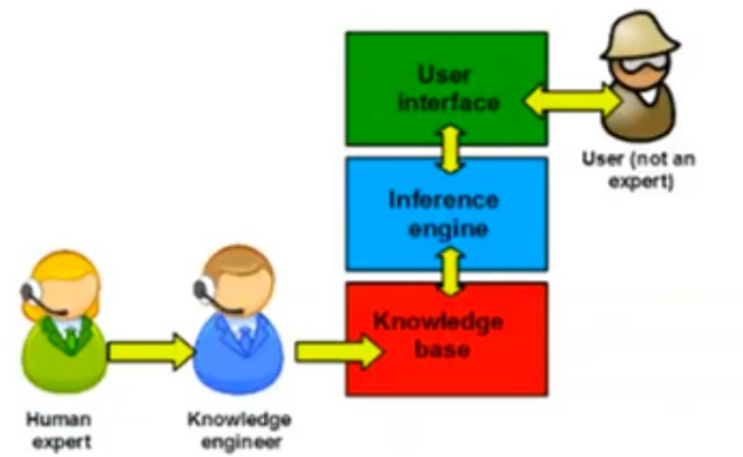
Cognitive Computers = Expert Systems

- What does it take to make one?
  - Content annotated and knowledge specified by experts
  - Organize the knowledge base using an appropriate representation



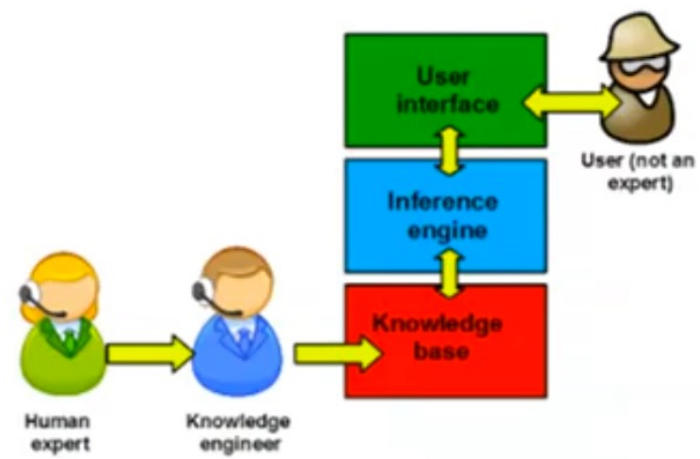
Cognitive Computers = Expert Systems

- What does it take to make one?
  - Content annotated and knowledge specified by experts
  - Organize the knowledge base using an appropriate representation
  - Select or develop a deductive inference engine for reasoning



Cognitive Computers = Expert Systems

- What does it take to make one?
  - Content annotated and knowledge specified by experts
  - Organize the knowledge base using an appropriate representation
  - Select or develop a deductive inference engine for reasoning
  - User interface



Example of Cognitive Computer

- Supporting medical doctors in their treatment of disease

Example of Cognitive Computer

- Supporting medical doctors in their treatment of disease
- IBM Watson for Oncology, for example, has been used at Memorial Sloan Kettering Cancer Center to provide oncologists with evidence-based treatment options for cancer patients
- When medical staff input questions, Watson generates a list of hypotheses and offers treatment options for doctors to consider

Course Syllabus

Syllabus Review: Course Topics

- Module 1: Introduction and Background
  - Course and Software Introduction
  - AI Concepts
    + Python Coding & GitHub (for those new to either)
  - History of Artificial Intelligence

Syllabus Review: Course Topics

- Module 2: Logic
  - Propositional Logic (Representation)
  - Propositional Logic (Reasoning)
  - First Order Predicate Logic (Representation)
  - First Order Predicate Logic (Reasoning)

Syllabus Review: Course Topics

- Module 3: Other Knowledge Representation
  - Semantic Networks
  - Frames
  - Knowledge Organization Systems and Ontologies
  - Semantic Webs and Trees

Syllabus Review: Course Topics

- Module 4: Essentials of Expert Systems
  - Rules & Introduction to Expert Systems
  - Rule-Based Systems
  - *Journal Club-Style Paper Presentations*
  - Building and Expert System and PyKE
  - Probability and Introduction to Uncertainty
  - Biomedical Expert Systems

Syllabus Review: Course Topics

- Module 5: Search
  - Intelligent Agents and Introduction to Search
  - Uninformed Search and Heuristic Search
  - Local and Population-based Search

Syllabus Review: Course Topics

- Module 6: Uncertainty
  - Entropy and Information Theory
  - Bayesian Networks
  - State Machines and Dynamic Models

Syllabus Review: Course Topics

- Module 7: Advanced and Auxiliary Topics
  - Learning and Planning
  - Introduction to Machine Learning
  - Rule-Based Machine Learning
  - *Final Project Presentations*

Syllabus Review: Materials

- Reference Textbooks:
  - Reading assignments strongly suggested from:
    - **Cawsey A, "The Essence of Artificial Intelligence (First Edition – 1998) - $18 on Amazon**
  - Additionally, I'd recommend the following two textbooks as excellent resources for additional reading/reference:
    - **Russell S and Norvig P, "Artificial Intelligence: A Modern Approach" (Third Edition - 2018)**
    - **Kalet I, "Principles of Biomedical Informatics" (Second Edition - 2014)**
  - For students looking for additional textbook resources, I'd recommend the following:
    - Ertel W, "Introduction to Artificial Intelligence" (Second Edition – 2017)
    - Shortliffe E, "Biomedical Informatics: Computer Applications in Health Care and Biomedicine" (Fourth Edition 2014)
    - Coppin B, "Artificial Intelligence Illuminated" (First Edition – 2004) – Available as PDF
  - Personal Computer
    - Installing free software
    - Working in Jupyter notebook

Student Expectations

- Watch respective lectures prior to class dates.
- Attend and participate in synchronous class sessions
  - Come to prepared to discuss that day's lecture
  - Have 2-3 questions prepared for Q&A
- Read assigned materials
- Complete assignments by specified date (or notify prior with cause)
- Mid-term: Journal club style presentation
- Final project: (report and presentation)

Mid-Term Journal-Club Presentation

- (1) Undertake a literature search (Apply this search to also laying the groundwork for your final project)
- (2) Identify a handful (e.g. 3-5) original research papers that illustrate examples of the AI topics covered in this course applied to the biomedical/clinical domain. These papers should ideally have all (or some) of the following properties (a) clearly written, (b) impactful, e.g. highly cited, (c) intuitive figures, (d) clear review of relevant/related research, and (e) clearly denote the advantages and disadvantages of the proposed work
- (3) Prepare a short (1-2 paragraph) synopsis/evaluation of each publication along with citation information for each paper save each paper as a PDF.
- (4) Turn in above materials by given deadline, ranking publications from 'most interested in presenting' to 'least'. From this I will identify a unique paper for each team/individual to present.
- (5) Prepare a journal-style format oral presentation with (max 10 slides). In-class presentations will be 10 minutes each with 5 minutes for discussion.
  - Exact time depends on class size

- Example Topics:
  - Expert systems, clinical decision support tools, ontologies, standards in biomedical informatics, data harmonization, collaborative data analysis, probabilistic reasoning, ethics, health information infrastructure, health care education, patient-centered care systems, patient monitoring systems, telehealth, information retrieval and digital libraries

Final Project

- Propose and attempt to implement a basic biomedical research or support tool to address a target biomedical task
  - For example, this could be an expert system to make deductions towards their chosen biomedical task, e.g. a clinical decision support system, alert system
  - Conduct a primary source literature search targeting a specific biomedical task
  - Identify relevant data, databases, and/or necessary recourses for developing their research/support tool
  - Tool should be implemented in Python using a Jupyter notebook to clearly annotate the implementation and application of their system.

Final Project

- A **final report** and a 10-15 minute **oral presentation** will be due on the last day of class.
- This final report should be presented as a 6-12 page summary of their work including:
  - (1) a **Background** that synthesizes relevant literature in the field and the need for and previous application of similar methodologies,
  - (2) **Materials and Methods** which discusses any data, databases, or other resources utilized by their system,
  - (3) the **Design** of their proposed tool and how it integrates multiple topics covered in this course, and
  - (4) a **Demonstration** which highlights the application of this tool to a specific task and the results of this application or discussion of where the tool is with respect to implementation progress/troubleshooting attempts, etc, and
  - (5) **Discussion and Conclusions** to review any challenges overcome, advantages and disadvantages of the tool, and a proposal of how the tool could be improved further or applied more broadly to alternative biomedical tasks. This report should include relevant citations. The oral presentation should offer a brief summary of this report and optionally a demonstration of the developed tool.

Final Project

- Final Project Proposal (see syllabus for due date):
  - One half to full page summary

- Final Report (due last day of class):

  - 6-12 page summary
    - Background
    - Materials and Methods
    - Design
    - Demonstration
    - Discussion and Conclusions

  - Oral Presentation (given last day of class):
    - 10-15 minutes

Introducing and Installing Python and Anaconda

python™ Coding Language

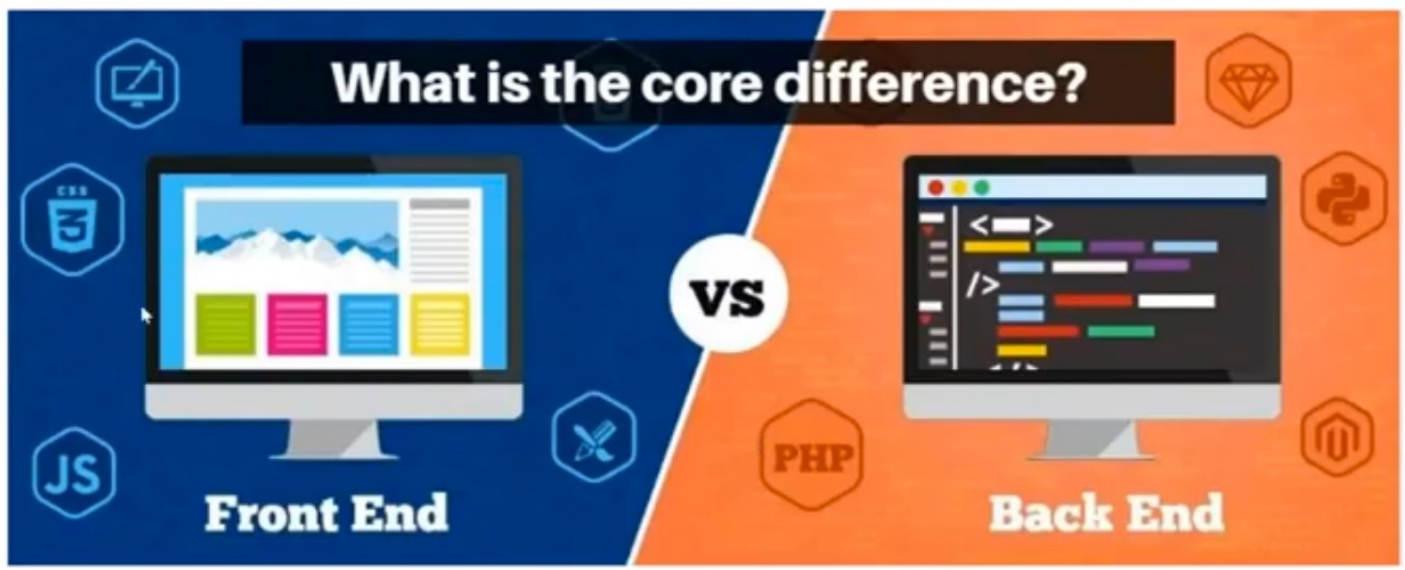
Python Benefits:

- Back-end and front-end development
- cross-platform language
- open-source
- Strong community base
- Plethora of tools
- Fewer and simple lines of codes

python™ Coding Language

Python Benefits:

- Back-end and front-end development
- cross-platform language
- open-source
- Strong community base
- Plethora of tools
- Fewer and simple lines of codes

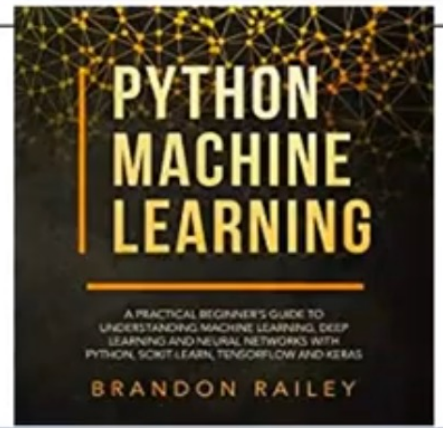


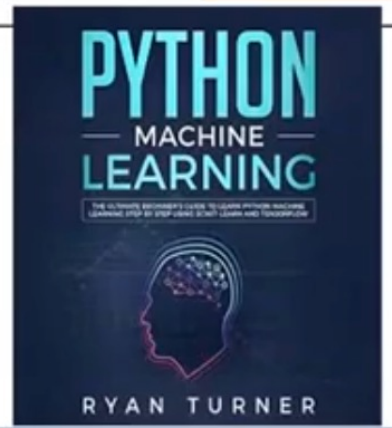
Difference Between R and Python

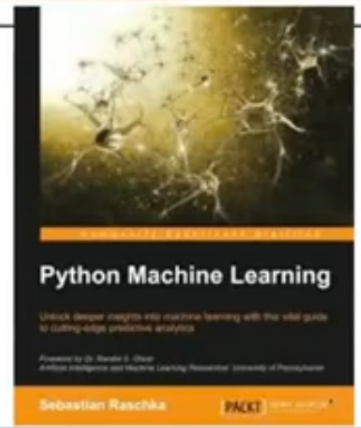
Features	R	Python
Scope	Used mainly for statistical modeling	Used for a variety of purposes like web-application development and data analysis
Used By	Statisticians, Analyst & Data Scientist	Developer, Data Engineers & Data Scientist
Suitable For	People with no prior experience in programming	Newbies to experienced IT professionals
Package Distribution	CRAN	PyPi
Visualization Tools	ggplot2, plotly, ggiraph	Matplotlib, bokkeh, seaborn

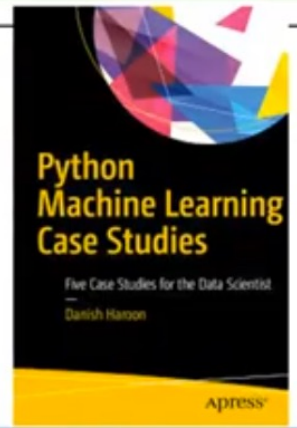
Difference Between R and Python

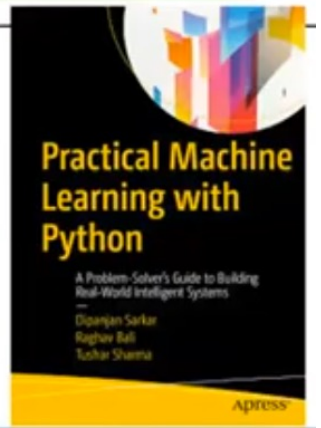
Features	R	Python
Scope	Used mainly for statistical modeling	Used for a variety of purposes like web-application development and data analysis
Used By	Statisticians, Analyst & Data Scientist	Developer, Data Engineers & Data Scientist
Suitable For	People with no prior experience in programming	Newbies to experienced IT professionals
Package Distribution	CRAN	PyPi
Visualization Tools	ggplot2, plotly, ggiraph	Matplotlib, bokkeh, seaborn







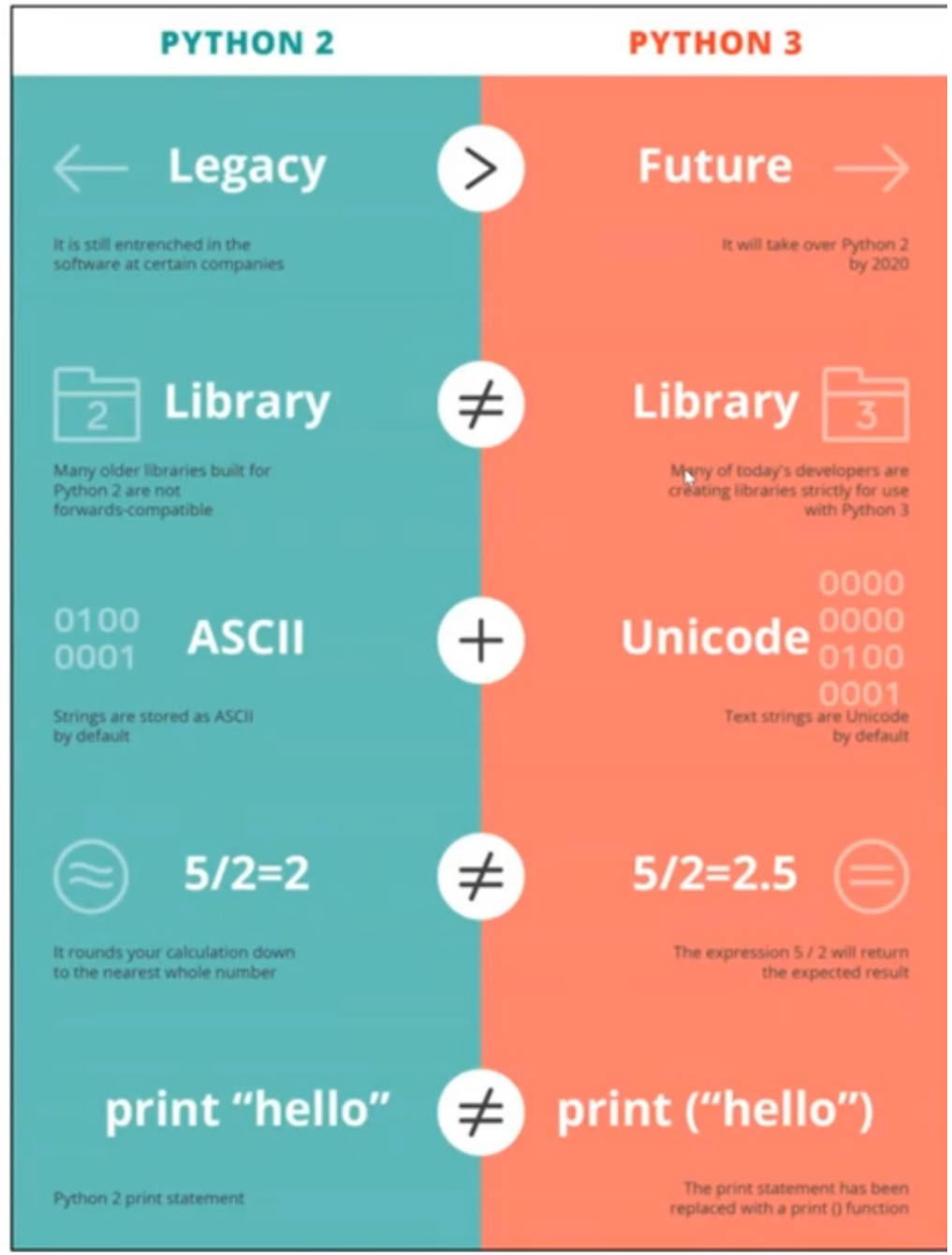




python™ 2 vs 3

- Recommend: Use latest version of Python 3.

- Exception:
  - Want to run Python code developed in (or only updated to) Python 2, or some other specific Python version
    - E.g. Python 2.7



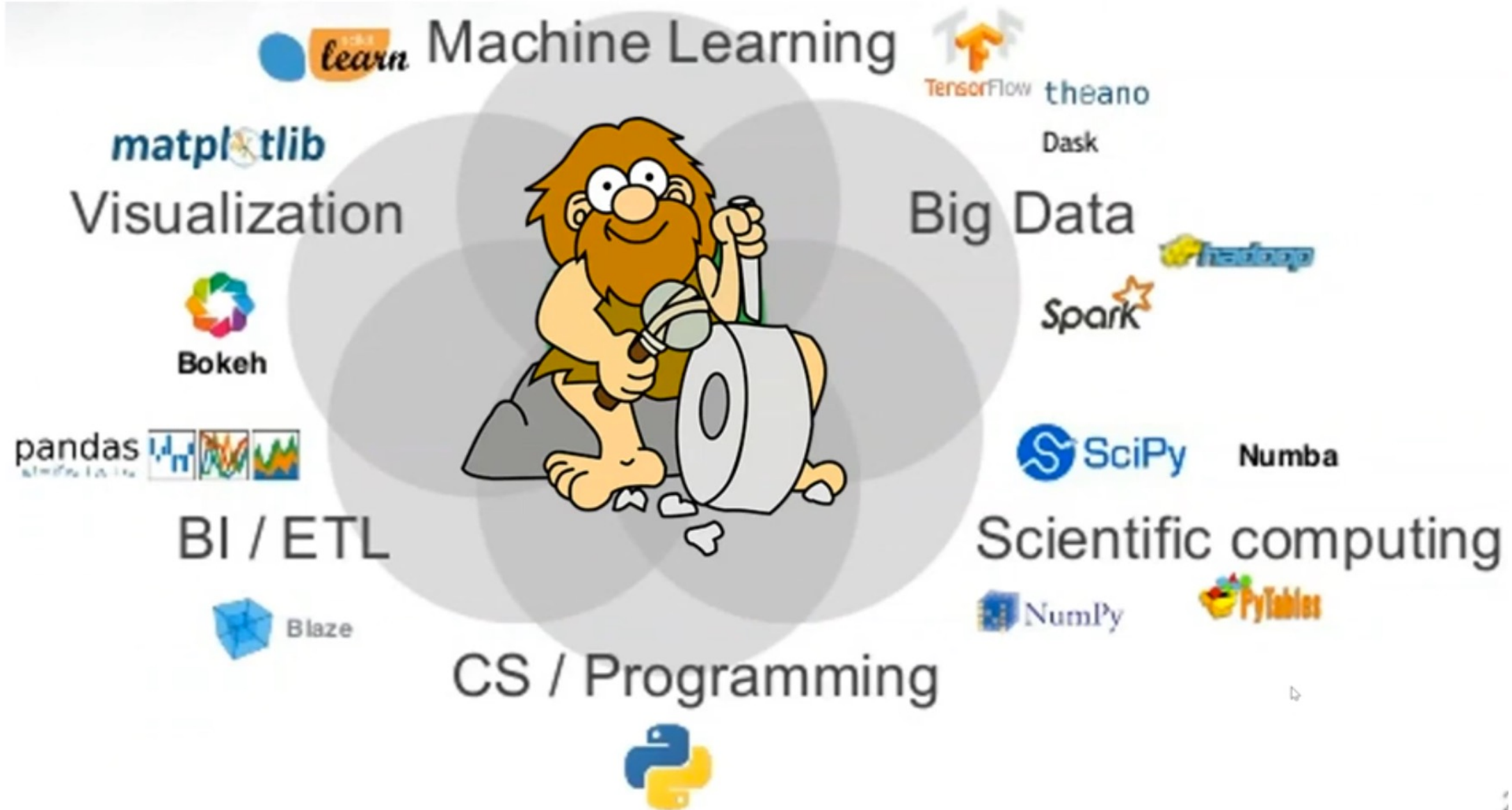
Python

- Choice is primarily based on your hardware architecture (CPU) and operating system (OS) installation
  - 64 vs 32 bit CPU
  - Windows, MacOS, Linux

- 64-bit processors:
  - Can run either 64 or 32 bit OS and Python installations.
  - Handles large amounts of random access memory (RAM) more effectively

- Choose 64-bit Python unless…
  - Want to run a ‘legacy’ code (i.e. code developed to run on 32-bit systems)

The Python ‘Ecosystem’ of Tools



- https://www.anaconda.com/

- Free, open source software distribution of Python and R (and associated packages) for scientific computing (e.g. data science and ML)

- One easy way to install Python (and keep it up-to-date) along with commonly used packages: (Click names for links to respective pages)
  - Jupyter Notebooks — for writing Python code, running experiments and communicating your work to others.
  - pandas — for exploring and manipulating data.
  - NumPy — for performing numerical operations on data.
  - Matplotlib — for creating visualizations of your findings.
  - scikit-learn — also called sklearn, for building and analyzing machine learning models.
  - TensorFlow – parallelization of analyses for efficiently handling ‘big data’

Useful Definitions for Python Coding

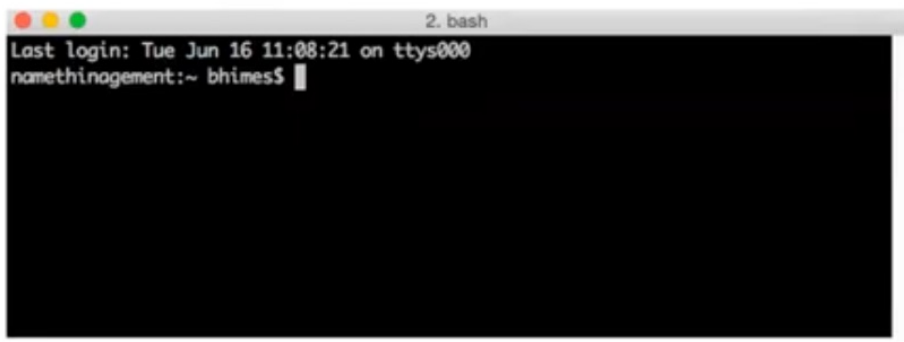
- Package:
  - A collection of (self contained)code someone else has written
- Conda:
  - A package manager that helps install, update and remove packages
  - pip – An alternative package installer for Python
- Environment:
  - The specific collection of packages or data science tools being accessed by your code
    - Each package has an associated 'version'.
- Build (of the package):
  - The Python version the package is made for
- Scripts:
  - Reusable code
  - Can be imported and called as a saved Python file (.py)

Useful Definitions for Python Coding

- Working Directory:
  - The directory from which a specific piece of code or software is run.
  - Usually = the Project directory/folder
  - Code run from the working directory looks in this local folder for other necessary files by default.

- Command Line Interface (CLI):
  - Interact with a computer with text commands
  - Faster, with more control, better for larger datasets
  - Link: Command Line Tutorial
    - How to open a CLI on Mac, Windows, Linux?
    - Basics on how to use CLI

- Graphical User Interface (GUI):
  - Interact with electronic devices through graphical icons and visual indicators instead of command lines



Anaconda ‘Products’

- MINICONDA:
  - Bare necessities of Python + CONDA
- Anaconda Navigator:
  - GUI to manage packages without command line.

Installing Anaconda

- Link: Introduction to Anaconda (Reading & Installation Instructions)
- Link: Guided tour of installing and testing Anaconda
- Link: Supplemental Guide to Installing Python and Packages

Anaconda 2018.12 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (614.3 MB)

32-Bit Graphical Installer (509.7 MB)

Python 2.7 version

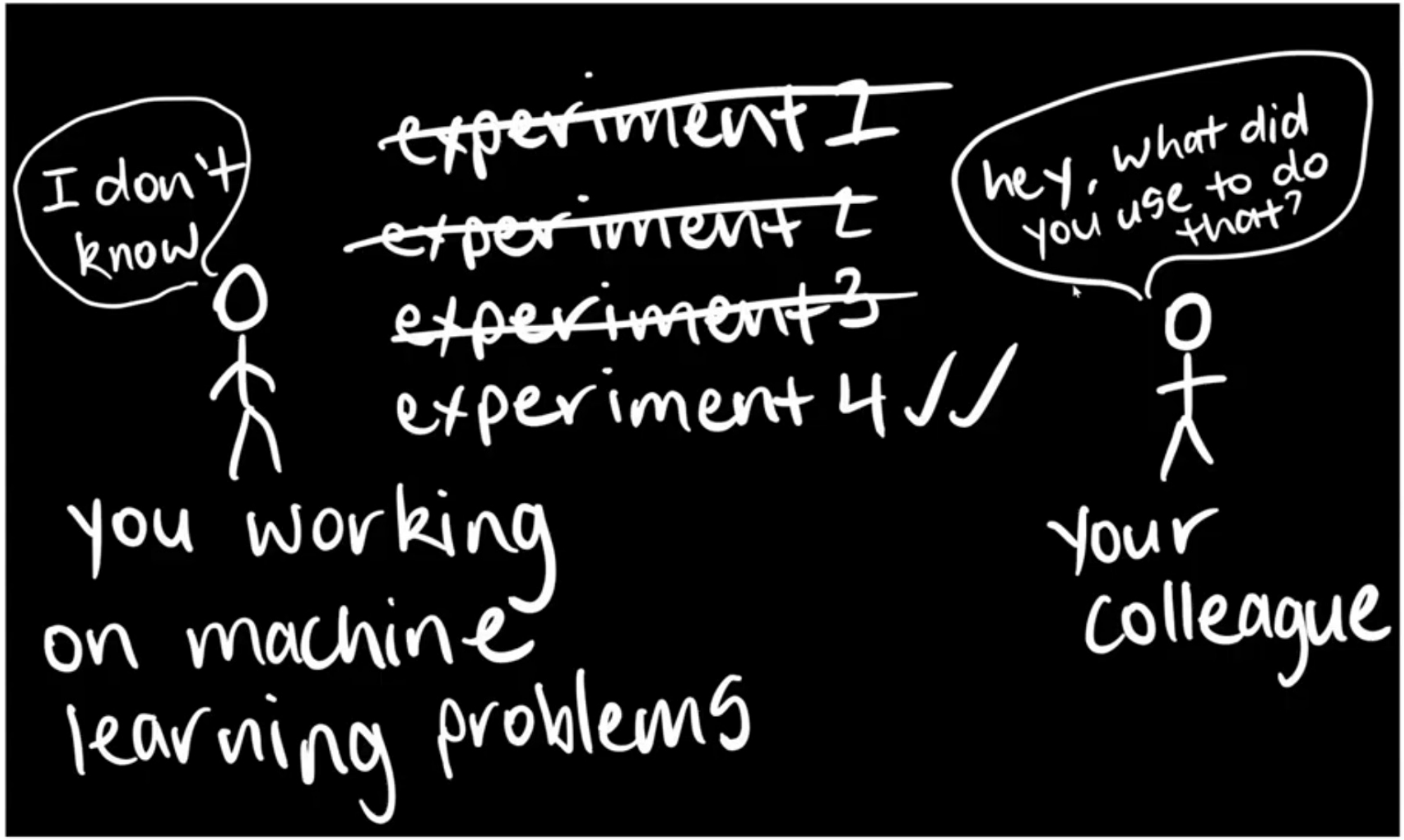
Download

64-Bit Graphical Installer (560.6 MB)

32-Bit Graphical Installer (458.6 MB)

Coding Environments

Value of a Reproducible Environment



Managing Custom Environments

- Virtual Environment:
  - An ‘local’, project environment that can be activated as an alternative to the ‘base’ environment installed in the ‘PATH’ of your computer
  - Switching between environments involves ‘activating’ an environment folder
  - The environment can be saved as a subdirectory in a given project directory
  - Environments can also be shared as files.

- PATH:
  - An environment variable on Unix-like operating systems, DOS, OS/2, and Microsoft Windows, specifying a set of directories where executable programs are located.

- Why?: Code dependencies may require:
  - A specific (usually) older version of:
    - Python
    - One or more packages

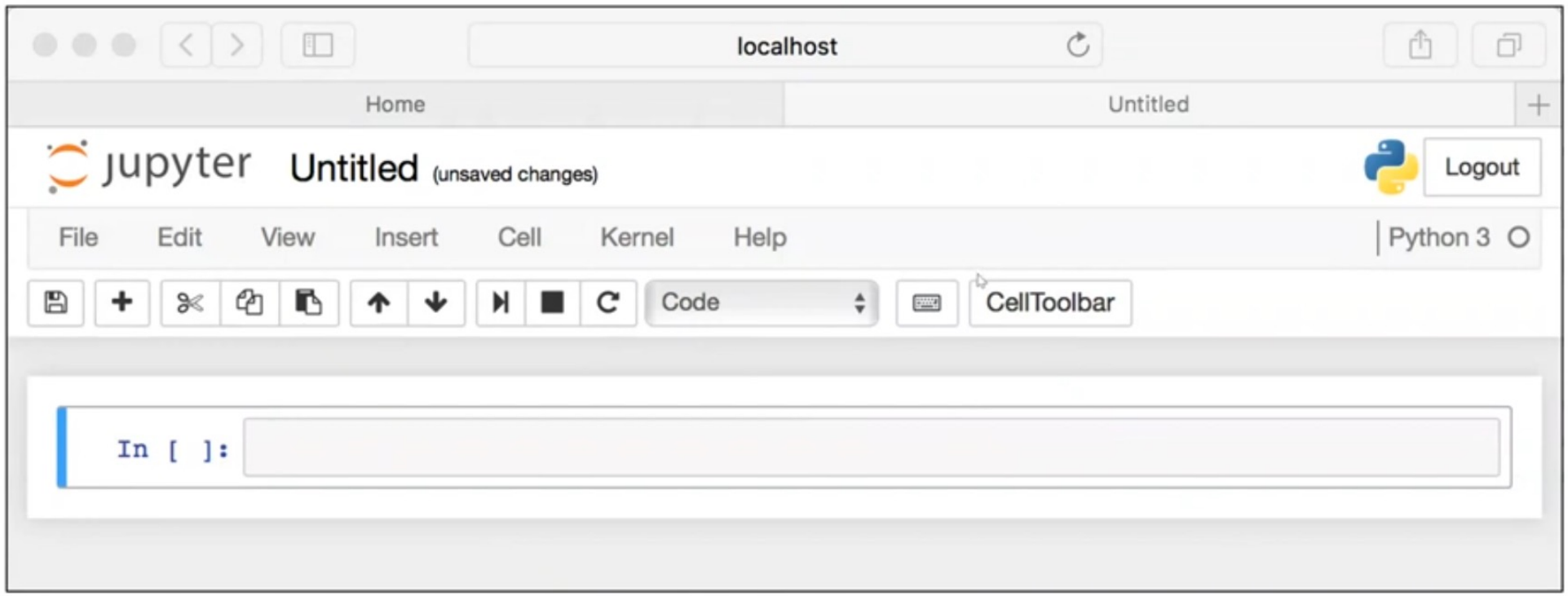
- MiniConda can be a starting point for creating a project environment with only the necessary packages

- Link: Conda Documentation on Managing Environments in Python

Jupyter Notebooks

Jupyter Notebooks

- A ‘scientific notebook’ for code
- Open source web application (with GUI):
  - Allows you to create and share documents that contain live code, equations, visualizations and narrative text
- Link: Jupyter Notebook Tutorial

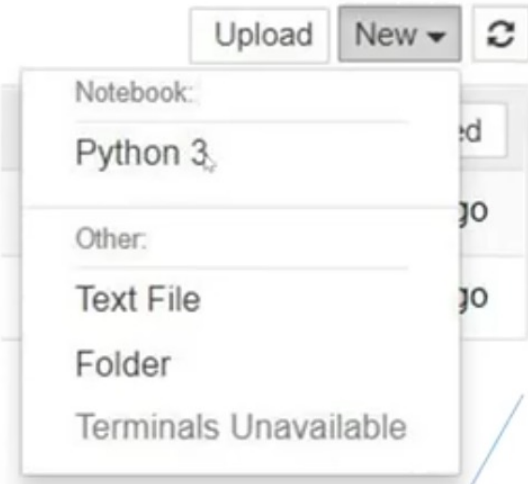


Jupyter Notebook

- .ipynb → Denotes jupyter notebook file
- To open:
  - New file:
    1. Navigate to desired working directory via CLI
       - Can use the anaconda prompt as the CLI
    2. Type 'jupyter notebook' + [Enter]
    3. Click 'New' and select Python 3
    4. Give a name to 'Untitled' notebook
  - Existing file:
    - Double click existing file (open automatically)
    - Or repeat steps 1-2 above and then select notebook file to open

Jupyter Notebook

- .ipynb → Denotes jupyter notebook file
- To open:
  - New file:
    1. Navigate to desired working directory via CLI
       - Can use the anaconda prompt as the CLI
    2. Type 'jupyter notebook' + [Enter]
    3. Click 'New' and select Python 3
    4. Give a name to 'Untitled' notebook
  - Existing file:
    - Double click existing file (open automatically)
    - Or repeat steps 1-2 above and then select notebook file to open



Jupyter Notebooks

- Cells:
  - A container for text to be displayed in the notebook or code to be executed by the notebook's kernel
  - Code cell: contains code to be executed in the kernel and displays its output below
  - Markdown cell: contains text formatted using Markdown and displays its output in-place when it is run
    - Link: Markdown Tutorial (in Jupyter Notebook)

- Kernels:
  - A "computational engine" that executes the code contained in a notebook document

- Run a cell (code or markdown): 

- Kernel Options:
  - Restart: restarts the kernel, thus clearing all the variables etc that were defined.
  - Restart & Clear Output: same as above but will also wipe the output displayed below your code cells.
  - Restart & Run All: same as above but will also run all your cells in order from first to last.

Jupyter Notebook Shortcuts

- Toggle between edit and command mode with Esc and Enter, respectively

- Once in command mode:
  - Scroll up and down your cells with your Up and Down keys
  - Press A or B to insert a new cell above or below the active cell
  - M will transform the active cell to a Markdown cell
  - Y will set the active cell to a code cell
  - D + D (D twice) will delete the active cell
  - Z will undo cell deletion
  - Hold Shift and press Up or Down to select multiple cells at once
  - With multiple cells selected, Shift + M will merge your selection

- Ctrl + Shift + -, in edit mode, will split the active cell at the cursor

- You can also click and Shift + Click in the margin to the left of your cells to select them

daffodil™
Better Answers

"By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."

Eliezer Yudkowsky
(American AI Researcher)

Assignments

- Assignment 1:

- Goals:
  - Get your computer set up (and equipped) with the tools needed for using Python and available packages
  - Understand the difference between Anaconda, MiniConda, and Conda
  - Understand how to access Python through your terminal (MacOS) or command line (Windows, Linux)
  - Learn how to use Git and GitHub
  - Learn basics of Python programming