Machine Learning in a Nutshell

15-488 Spring '20

Lecture 1:
Introduction

Teacher:
Gianni A. Di Caro

جامعة كارنيجي ميلون في قطر
Carnegie Mellon University Qatar

Outline

- Logistics and Admin issues
- ML?
- Some *motivations*
- General ML scheme
- ML pipeline: ML in the production process
- 15-488 vs. 10-315
- ML main paradigms, typical workflow (next time)
- Course road map
- What you'll take home
- Rules

Logistics and Admin

- This is a **newly designed course from scratch** (in a very short time) … be patient! ☺
- It's about ML + Data Science + Python programming + Practice of ML + Data publishing
- **Website**: https://web2.qatar.cmu.edu/~gdicaro/15488/

Key Information

Classes:	Lectures: UT 4:30 - 5:50pm - Room 2052
	Labs/Recitations: W 4:30pm - 5:50pm, Room 2062
Teacher	Gianni A. Di Caro
Units	9.0
Grading	35% In-class assessments (Quizzes, Labs), 35% Homework, 30% Project (Two Tasks)
Pre-requisites	15-112 or 15-110 passed with a C or a higher letter grade
Piazza	https://piazza.com/class/k53sxhx9tvt77b
Teaching Assistant	Aliaa Essameldin

Logistics and Admin

- **Software** you need to install on your laptops:

Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

Download

https://www.anaconda.com/distribution/

ANACONDA®

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews

jupyter | spyder | NumPy | SciPy | Numba | pandas | DASK | Bokeh | HoloViews | Datashader | matplotlib | scikit-learn | $H_2O.ai$ | TensorFlow | CONDA

Python 3.7 version

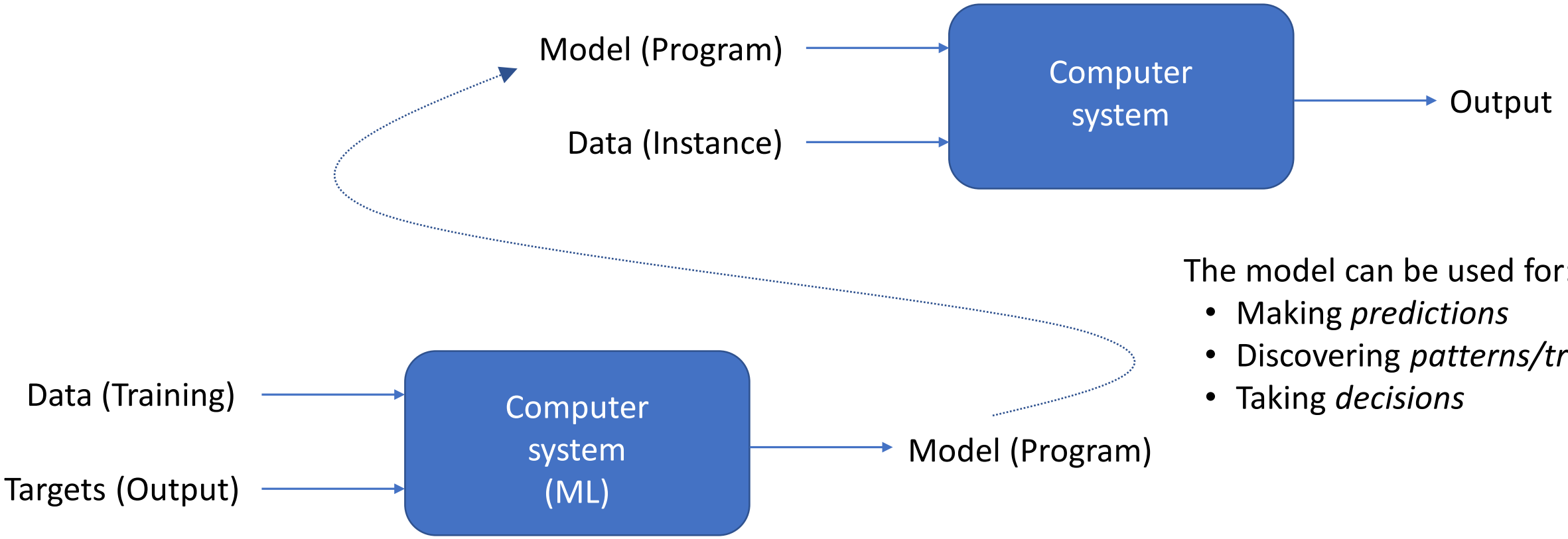
Windows | macOS | Linux

Machine Learning (ML)

- 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 Mitchell, 1997)*
  - **Machine Learning**: designing and understanding the properties of algorithms that can *incrementally* learn from experience (data, learning samples)
- **Ideal situation:** the machine is only fed with (raw) data and minimal (or zero) amounts of pre-built models and hypothesis
  - But … **Inductive biases** will always be there in some respect
- ML algorithms are heavily **data-driven**: avoid to input predefined rules, hard-code models, ...

Machine Learning

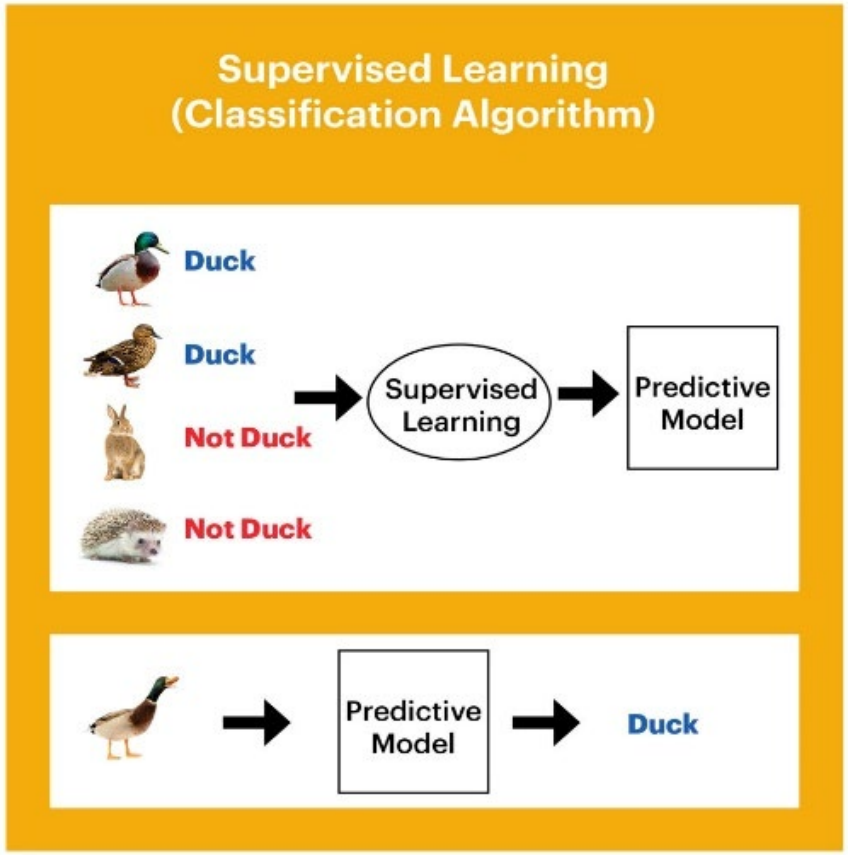
(Human) Model-based science vs. Data-based science



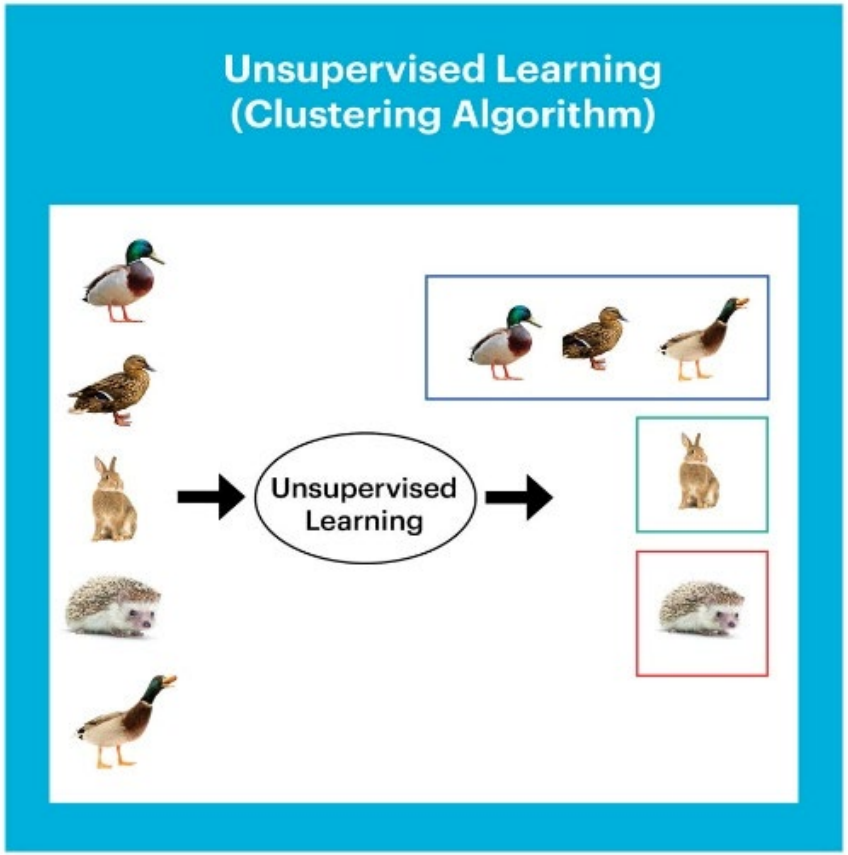
The model can be used for:

- Making *predictions*
- Discovering *patterns/trends*
- Taking *decisions*

Examples of ML model applications (will come back on this)



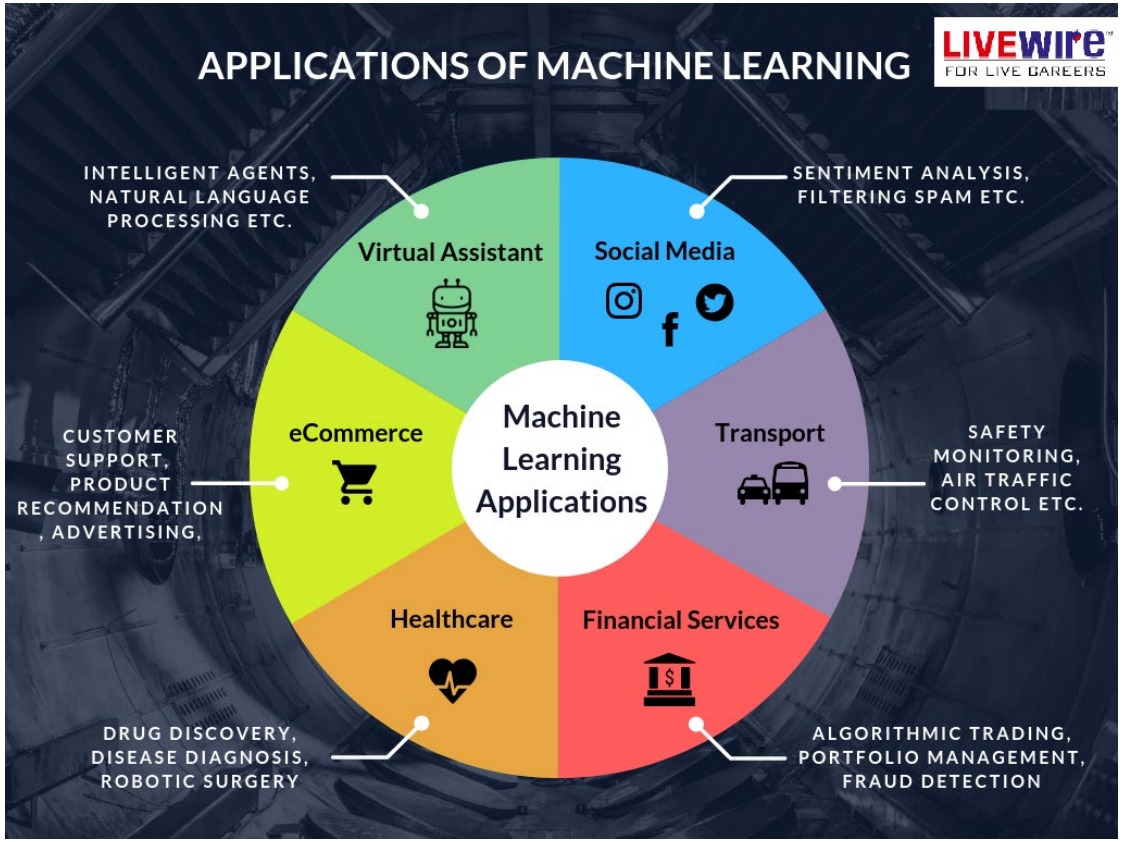
Predict (a class)



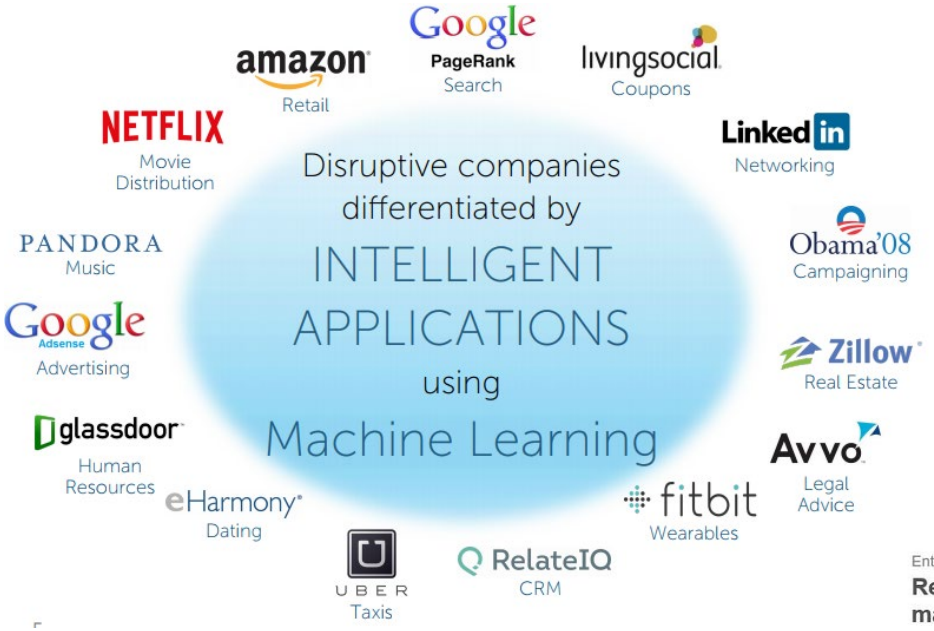
Find structure / Organize data

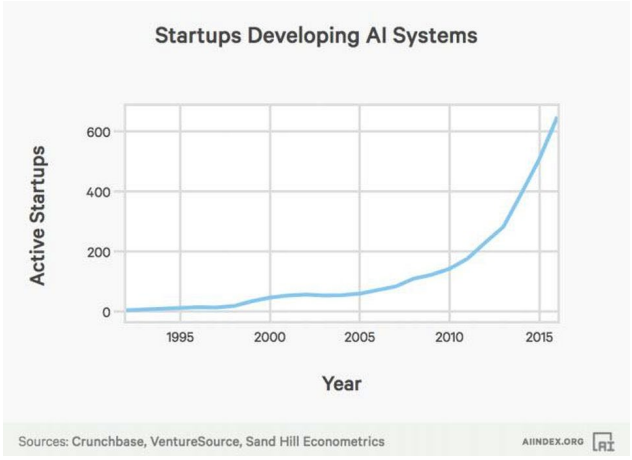
Real-world Applications

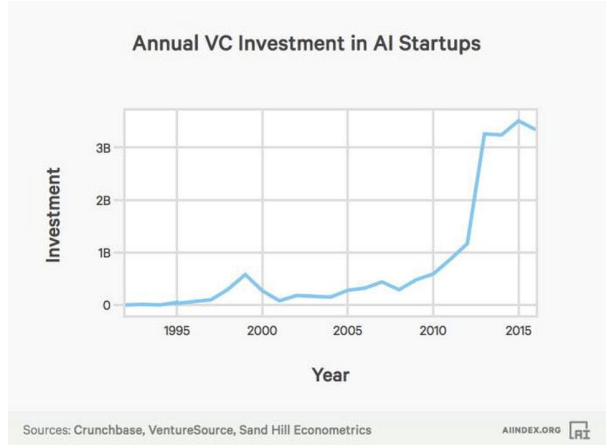
- Speech/handwriting recognition
- Virtual Personal Assistants (Siri, Alexa, Google Now)
- Machine Translation (e.g., Google Translate)
- Chatbots for online customer service
- Recommendation systems (e.g., Netflix, Amazon)
- Search engines (e.g, Google)
- Ad placement on websites
- Object detection/recognition
- Face recognition
- Weather prediction
- Traffic prediction
- Email Spam and Malware filtering
- Stock market analysis
- Credit-card fraud detection
- Automatic news generation
- Game playing (Atari, Go, Chess, StarCraft)
- Classifying DNA sequences
- Medical diagnosis
- Drug discovery
- Automatic vehicle navigation
- ... and many more

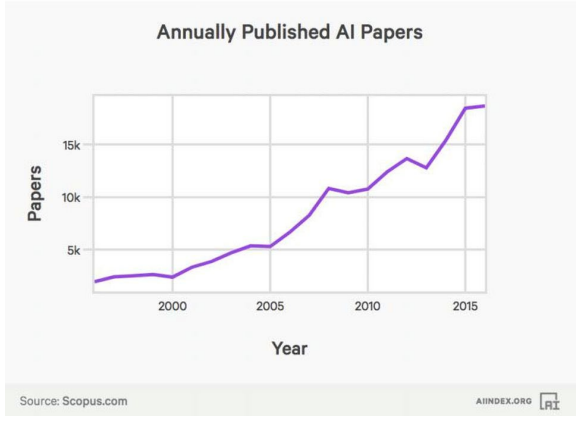


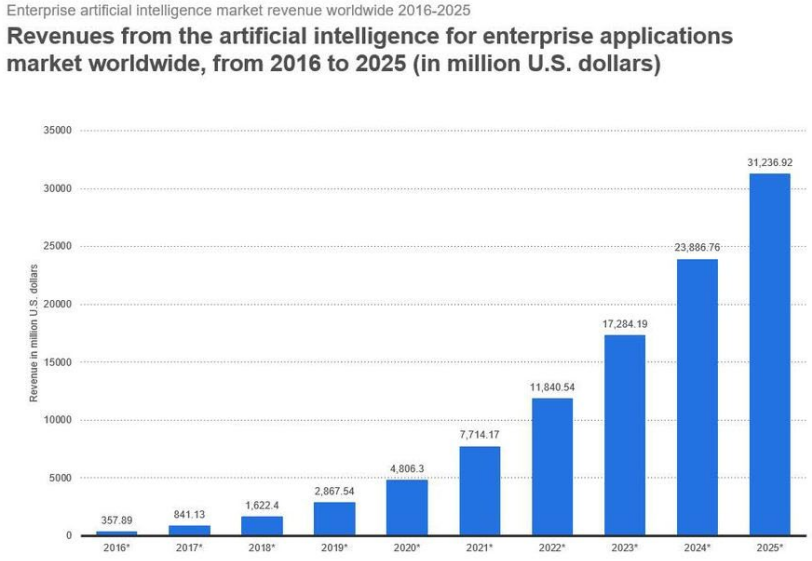
Economy of ML

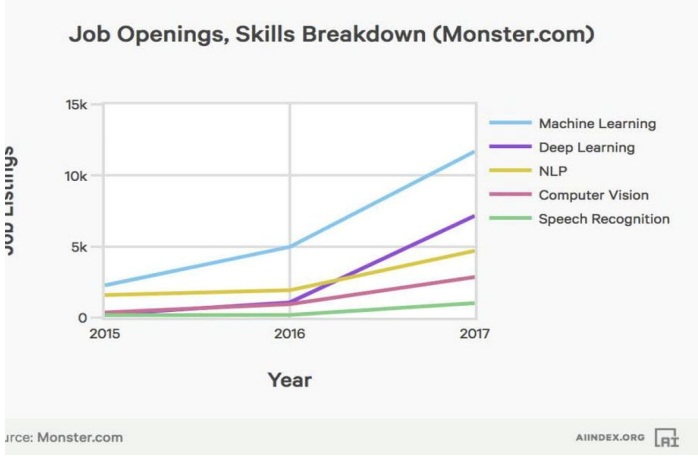




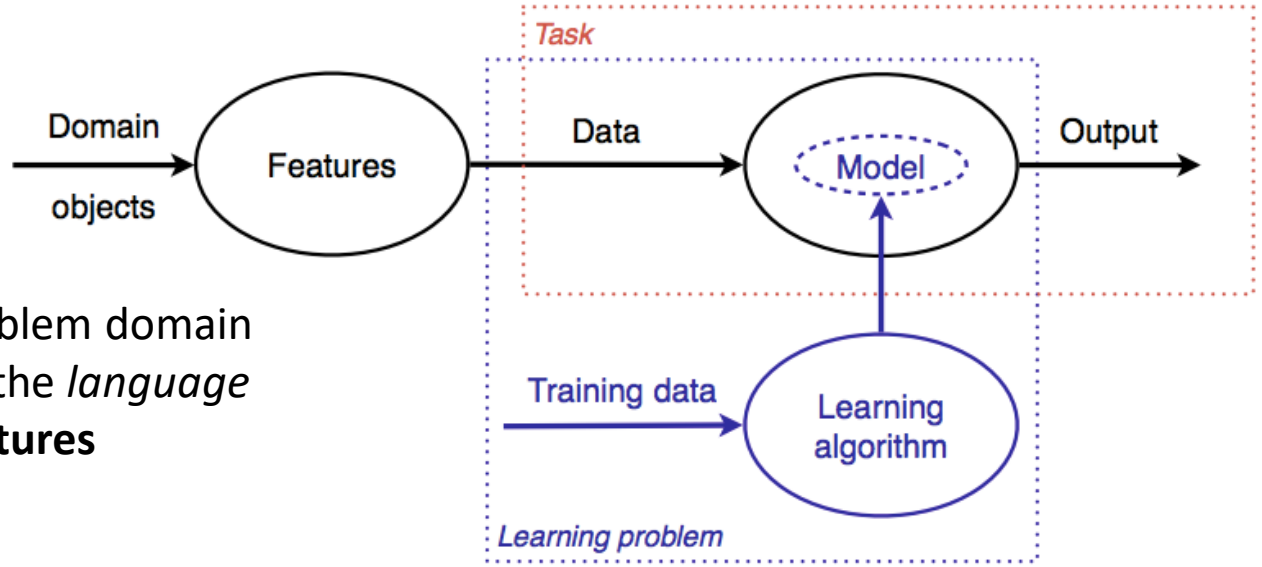








General ML Scheme



Data in the problem domain is described in the *language* of selected **Features**

Task: define an appropriate mapping from data to the **Outputs**

$$f: data \rightarrow Outputs$$

Learning Problem: Obtaining such a mapping from *training data*

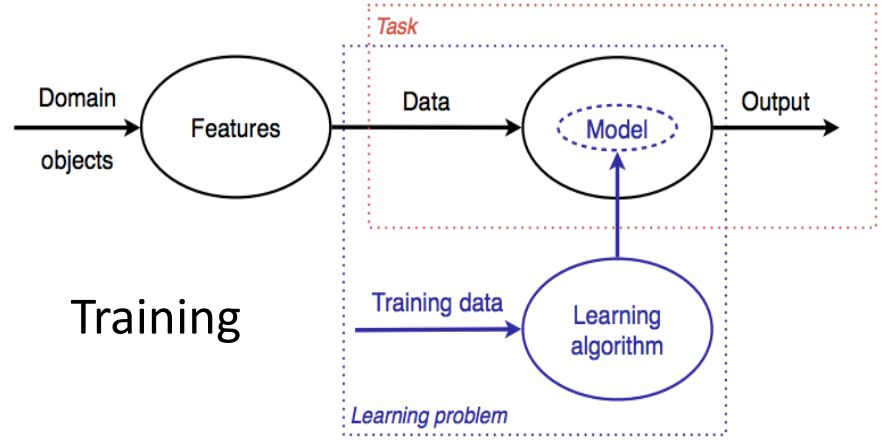
- ML Design: Use the right *features* (description language), to build the right *model*, that achieve the task according to the desired *performance*
- **Learning by examples**: Look at some data, *guess* at a general scientific hypothesis, make *statements* or *predictions* on test data, based on this hypothesis
- *Inductive learning* (from evidence) ≠ *Deductive learning* (logical, from facts)

Key aspect in ML: Generalization!

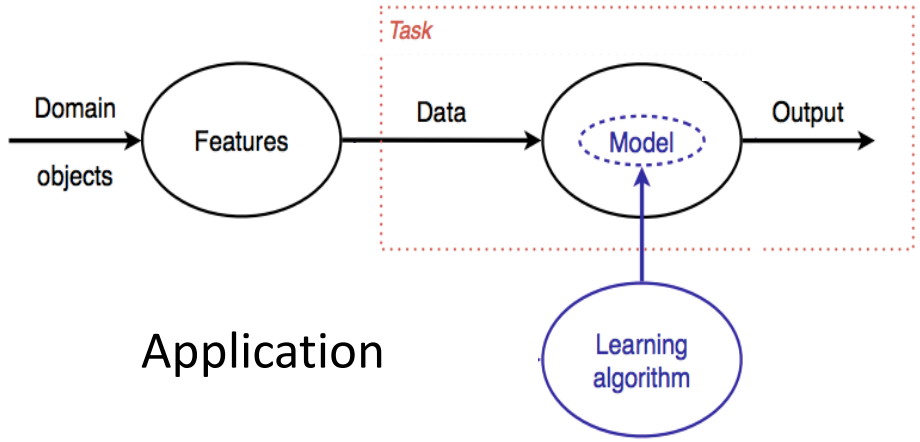
- ML share similarities, goals, and techniques with other fields, such as:
  - **Statistics**
  - **Function approximation:**
    - Approximation theory
    - Interpolation, extrapolation, curve fitting, regression

➢ Key characteristics of ML:

- The learned ML model isn't limited to the training set, but rather aims at *generalizing* the performance beyond the training set
- **Generalization**: Ability of an ML algorithm to do well on future *test* data
- Training data are just for learning the model. A good performance on training data doesn't ensure a good performance on future test data!
- **Core ML challenge**: *how to ensure generalization …*

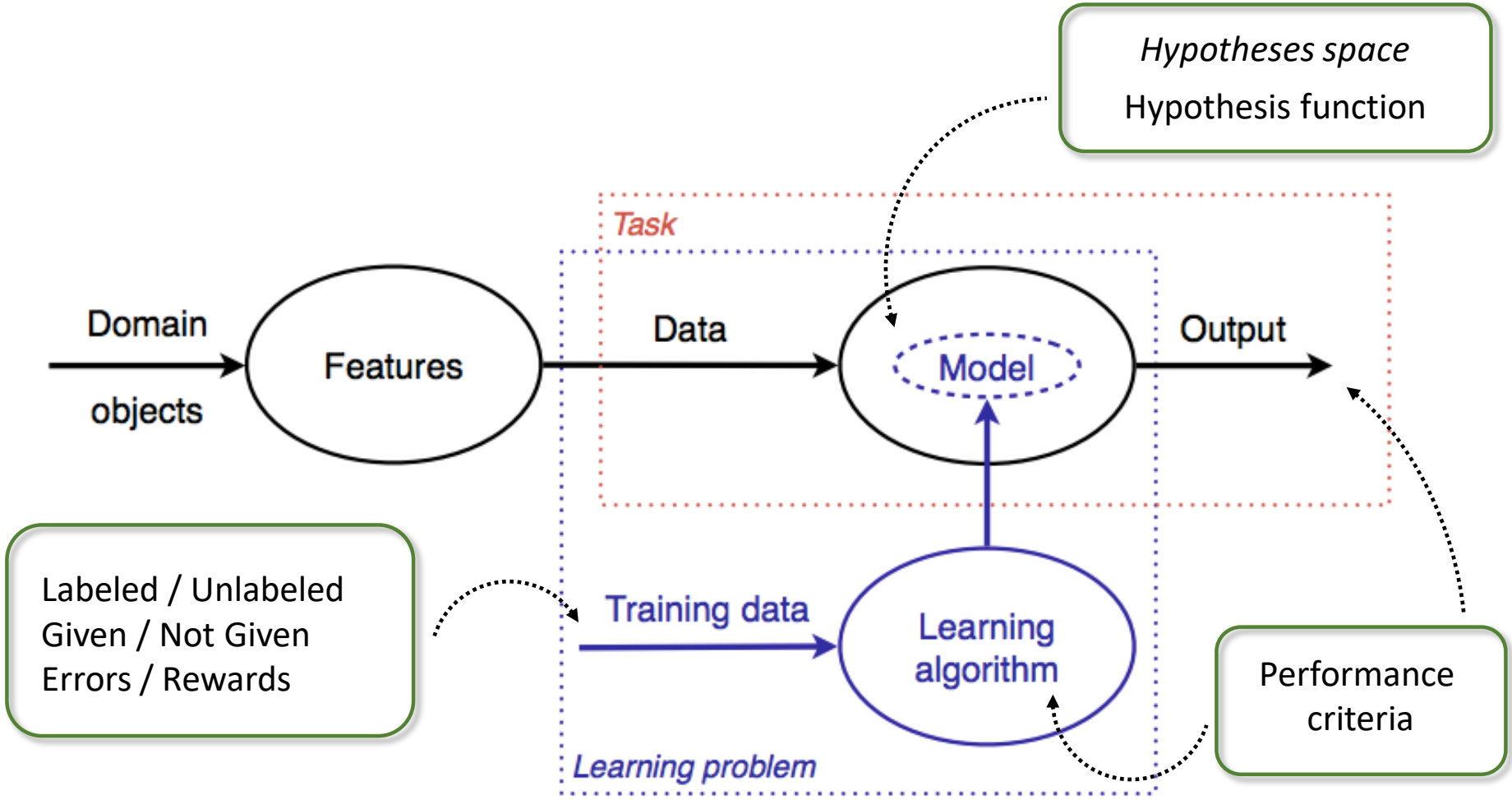


Training

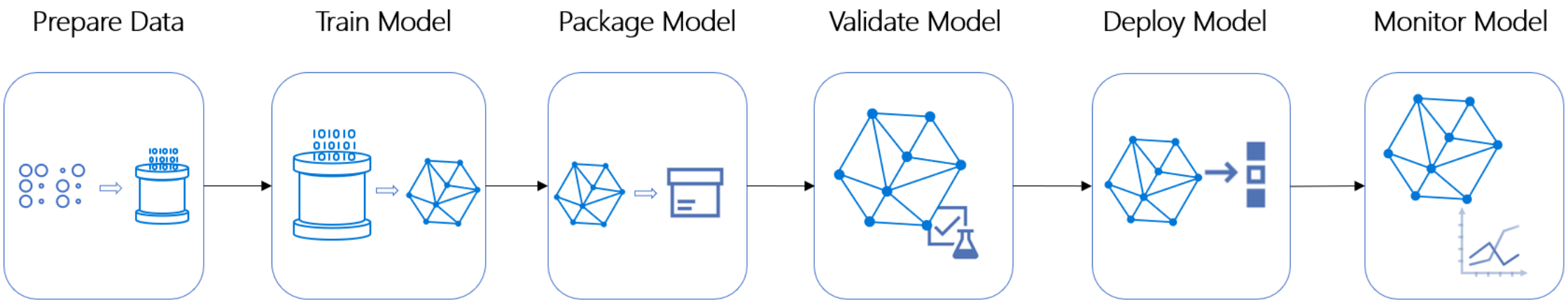


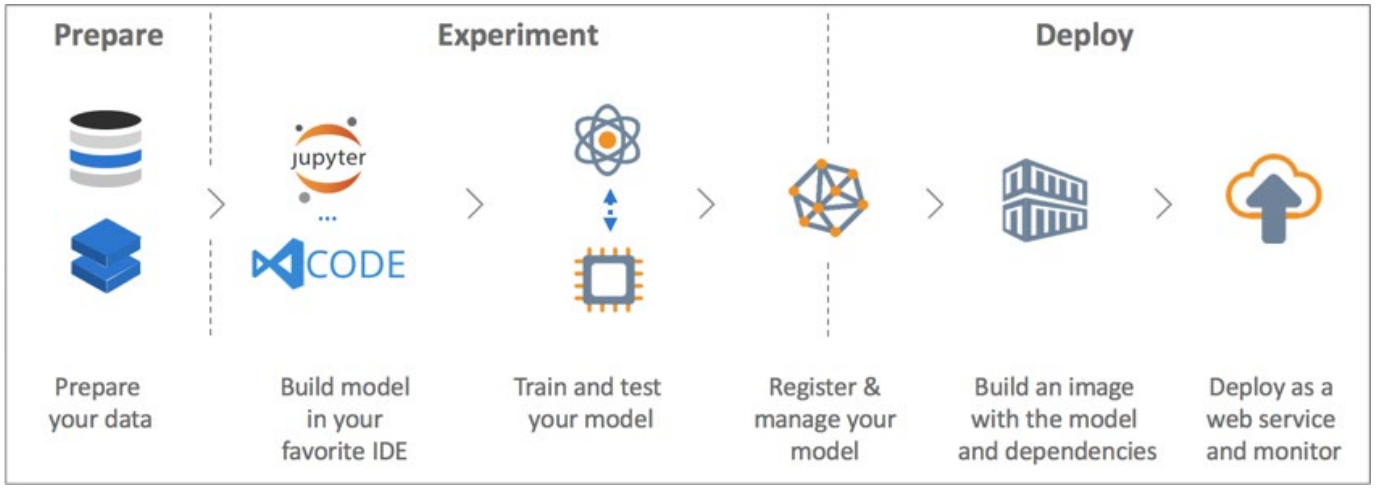
Application

General ML Scheme

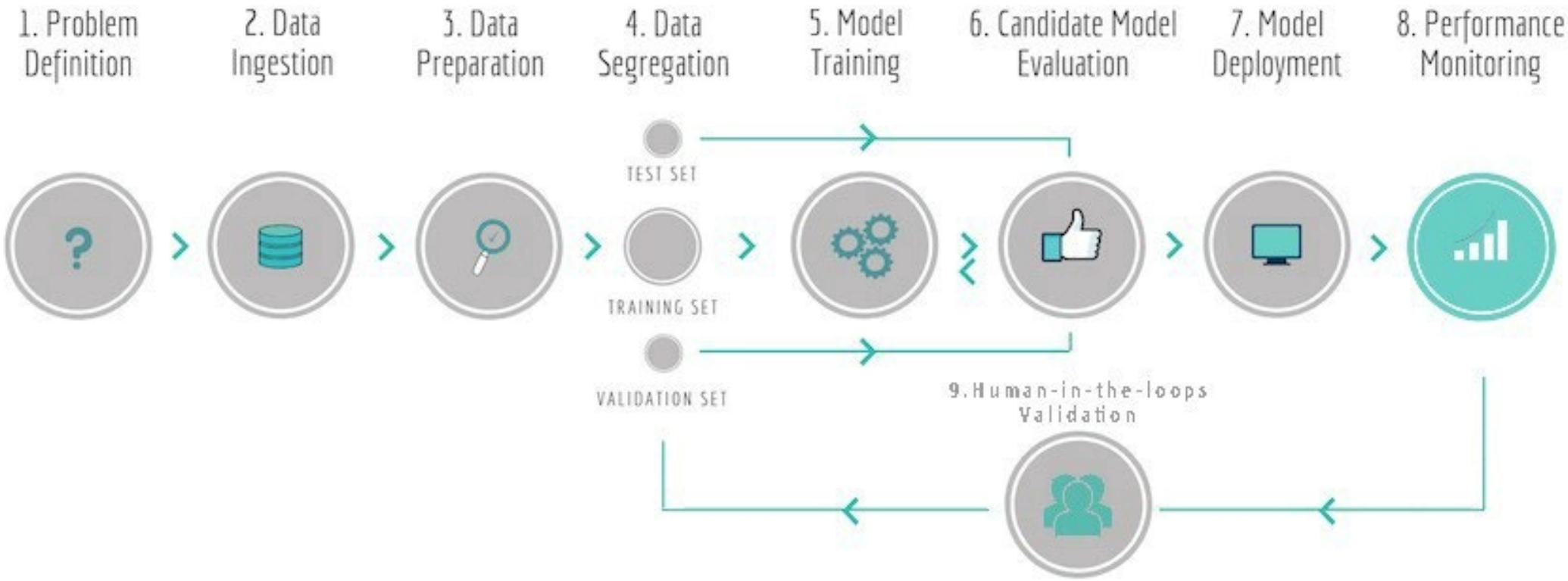


ML (pipeline) in the production process

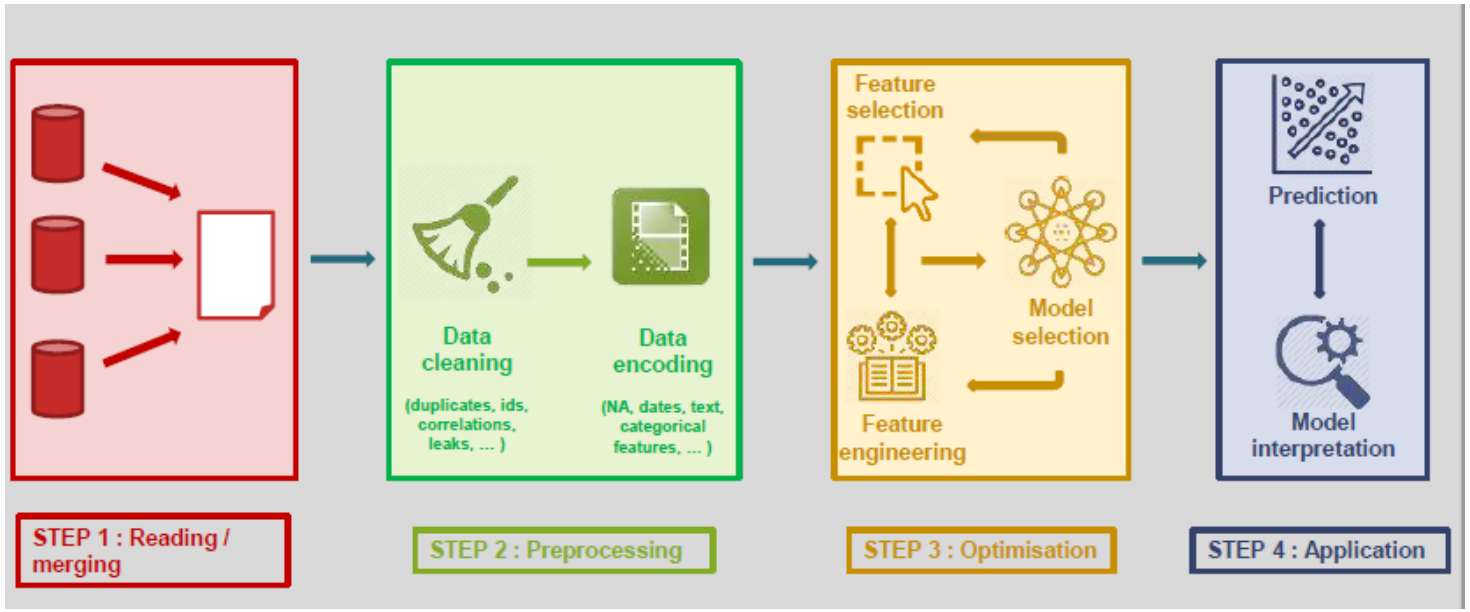




Not really a pipeline



Expanded view of the ML pipeline / workflow

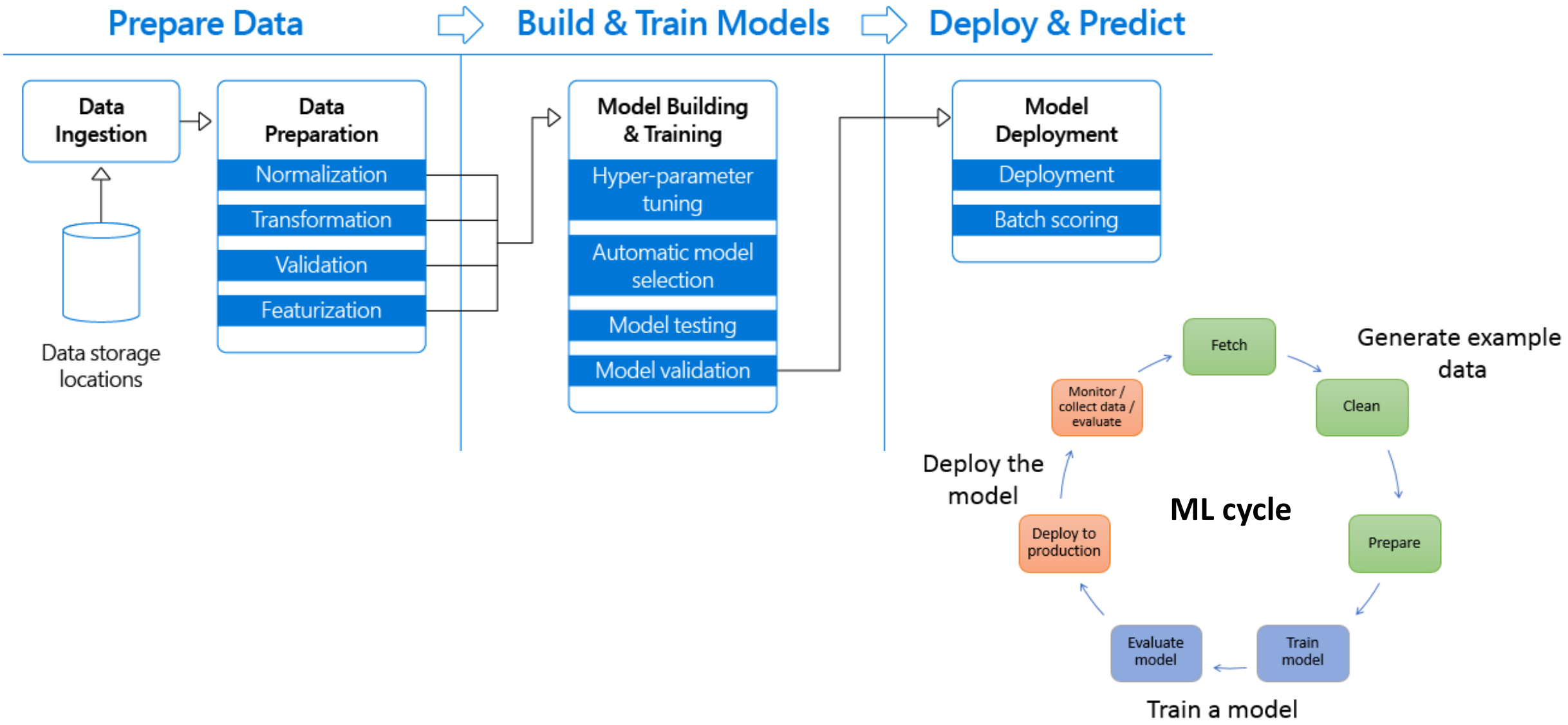


Machine Learning Pipeline

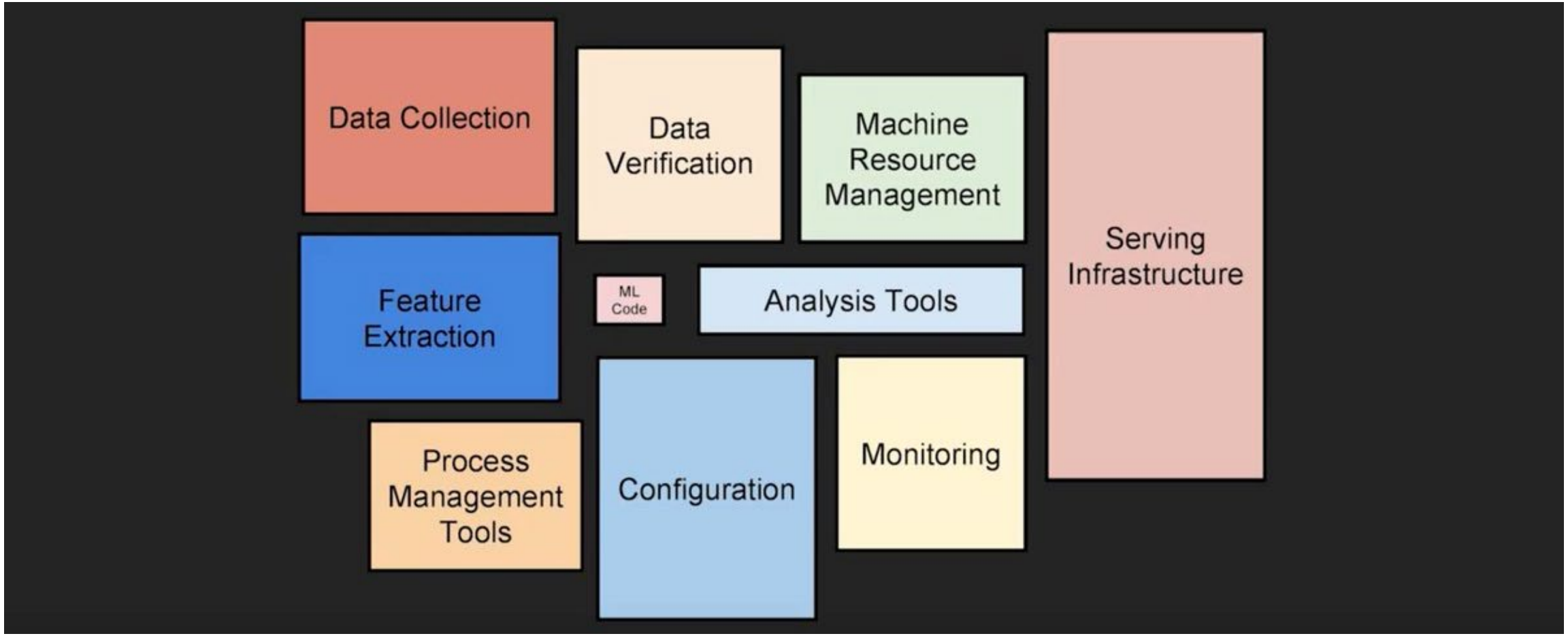
(Potentially) Automated Pipeline

Data Acquisition → Data Preparation → Feature Engineering → Preprocessing → Model Selection → Hyper-parameter Tuning → Model Training → Model Deployment

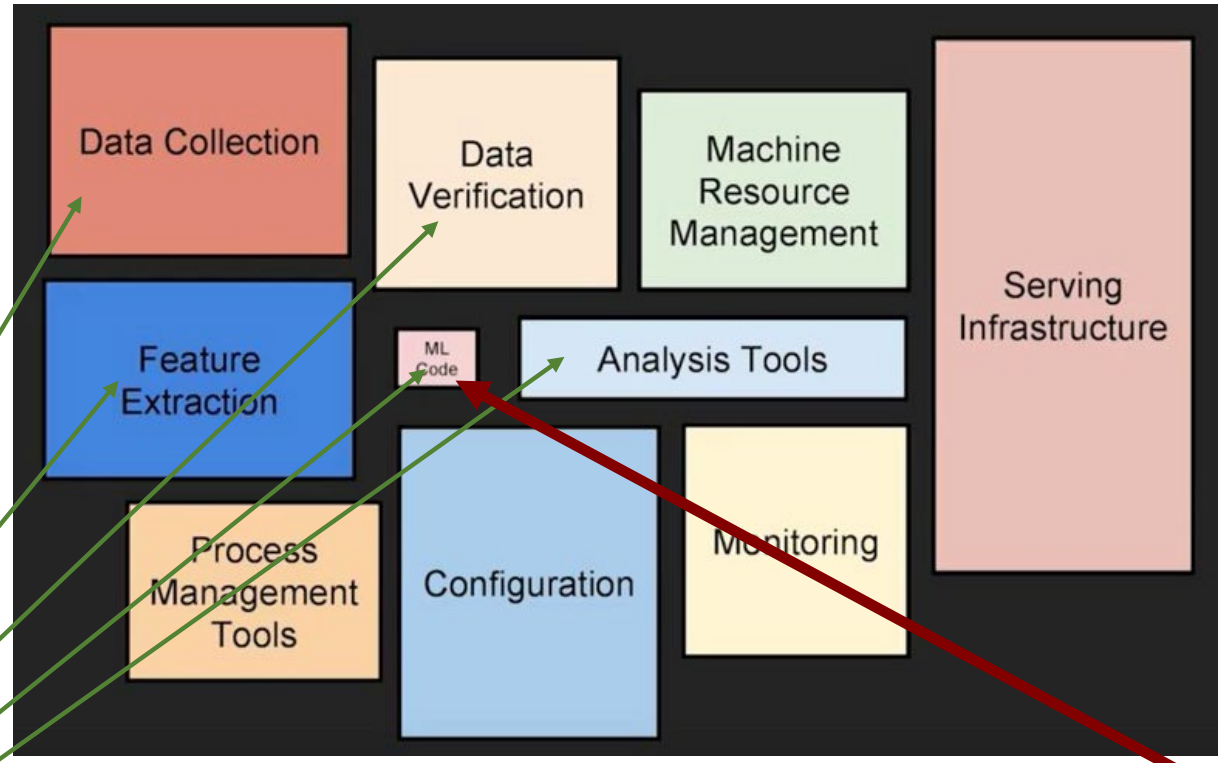
Expanded view of the ML pipeline / workflow / cycle



ML code / algorithms and the *rest* of the ML process



15-488 vs. 10-315



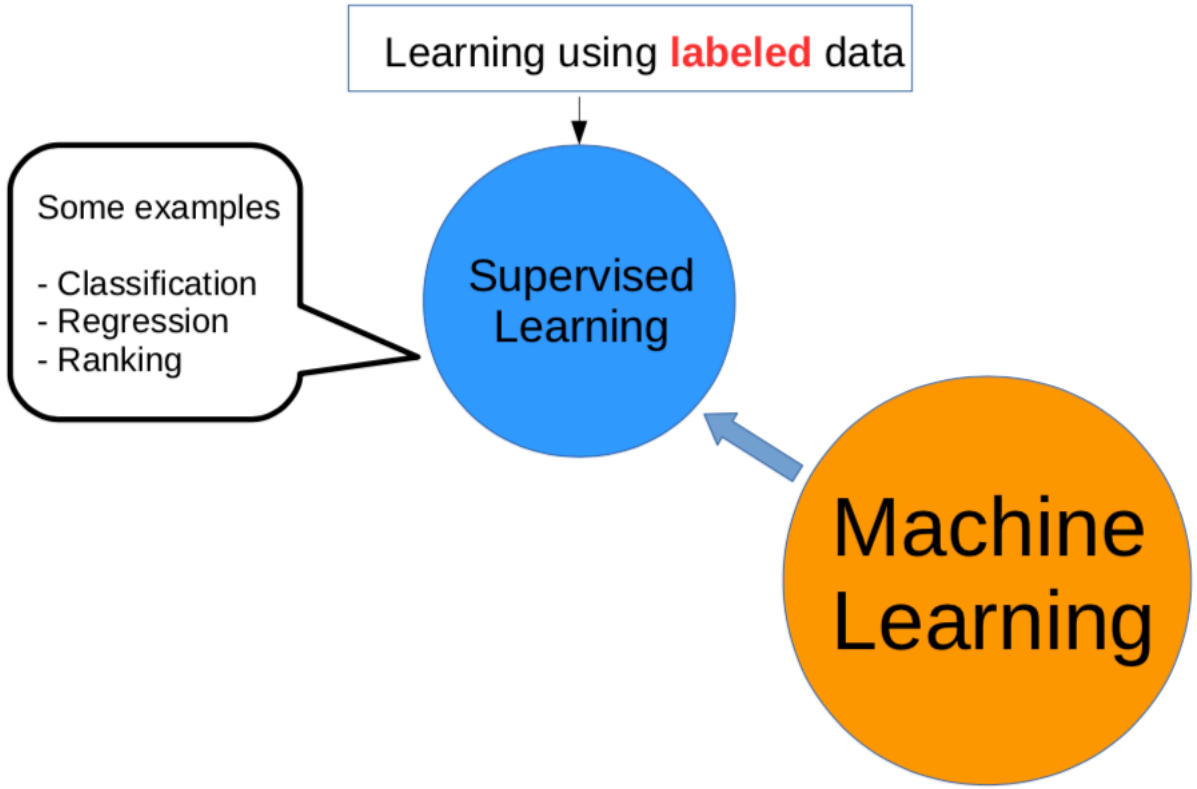
15-488

- ML pipeline for business production
- Data science issues and methods
- Software tools (Python ML/Data science ecosystem)
- Different data types (images, text, temporal) & scenarios
- Hands-on, Experimenting ≻ Theory

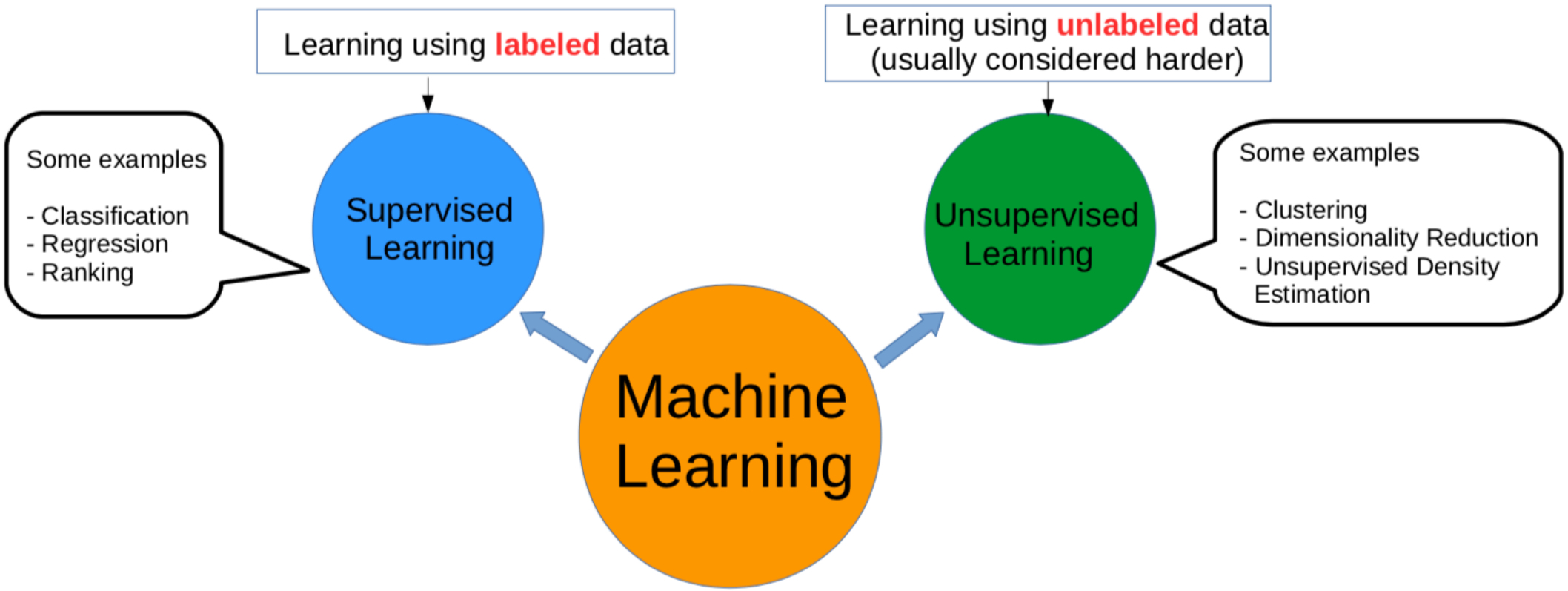
10-315

- ML problems
- ML algorithms
- Properties
- Formal methods
- Mathematical and probabilistic analysis
- Code implementation
- Theory ≻Practice

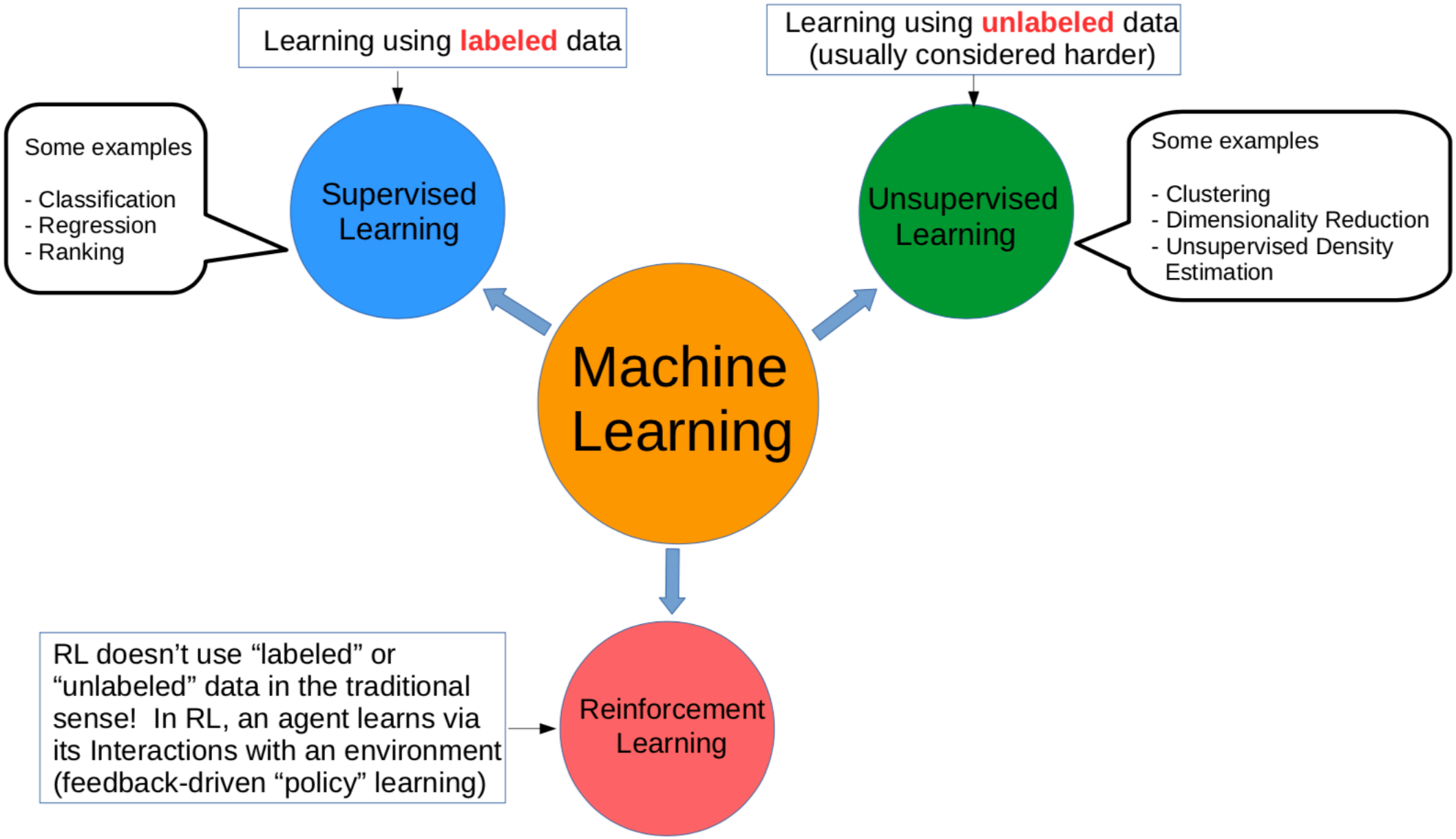
Basic taxonomy: SL, UL, RL



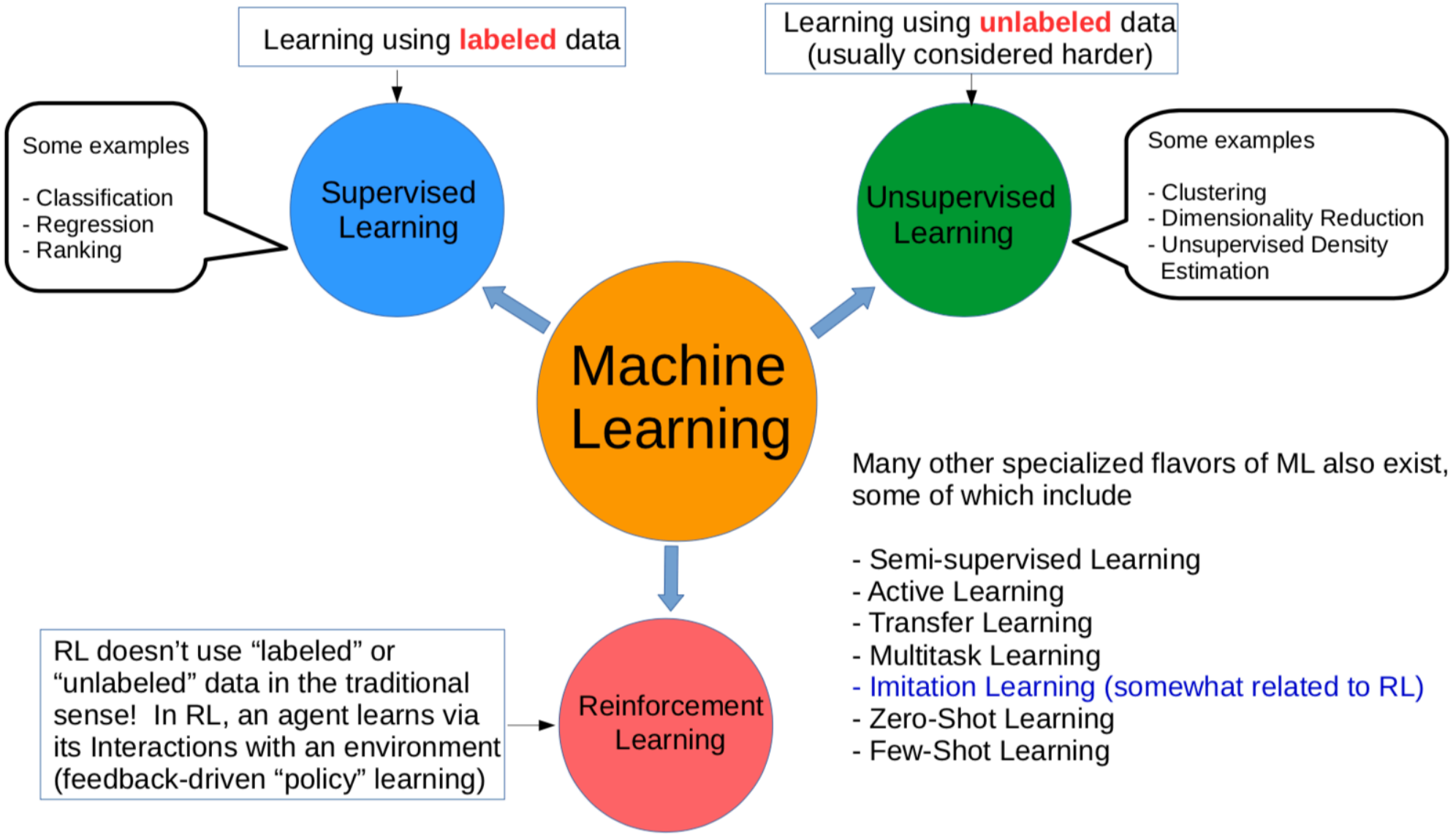
Basic taxonomy: SL, UL, RL



Basic taxonomy: SL, UL, RL



Basic taxonomy: SL, UL, RL



ML *approaches* that will be considered during the course

- **Supervised Learning** for classification and regression, **Deep learning**
- **Unsupervised Learning** for finding structure and for automatic feature extraction and dimensionality reduction

What will be *not* covered

- **Formal / probabilistic aspects of SL and UL, as well as more advanced ML techniques:** ML course, 10-315
- **Reinforcement Learning:** AI course, 15-381
- **Active Learning**: iterative SL where the learning machine can ask the user to add labels to selected unlabeled training samples
- **Semi-Supervised Learning**: learning out of small set of labeled data and large sets of unlabeled data, in-between UL and SL
- **Graphical models**: inference, prediction, control using Bayesian networks, Hidden Markov Models, Partially Observable Markov Models → AI course, 15-381, NLP course 11-411

Course road map

- **1. Definitions** of problems, objectives, performance metrics

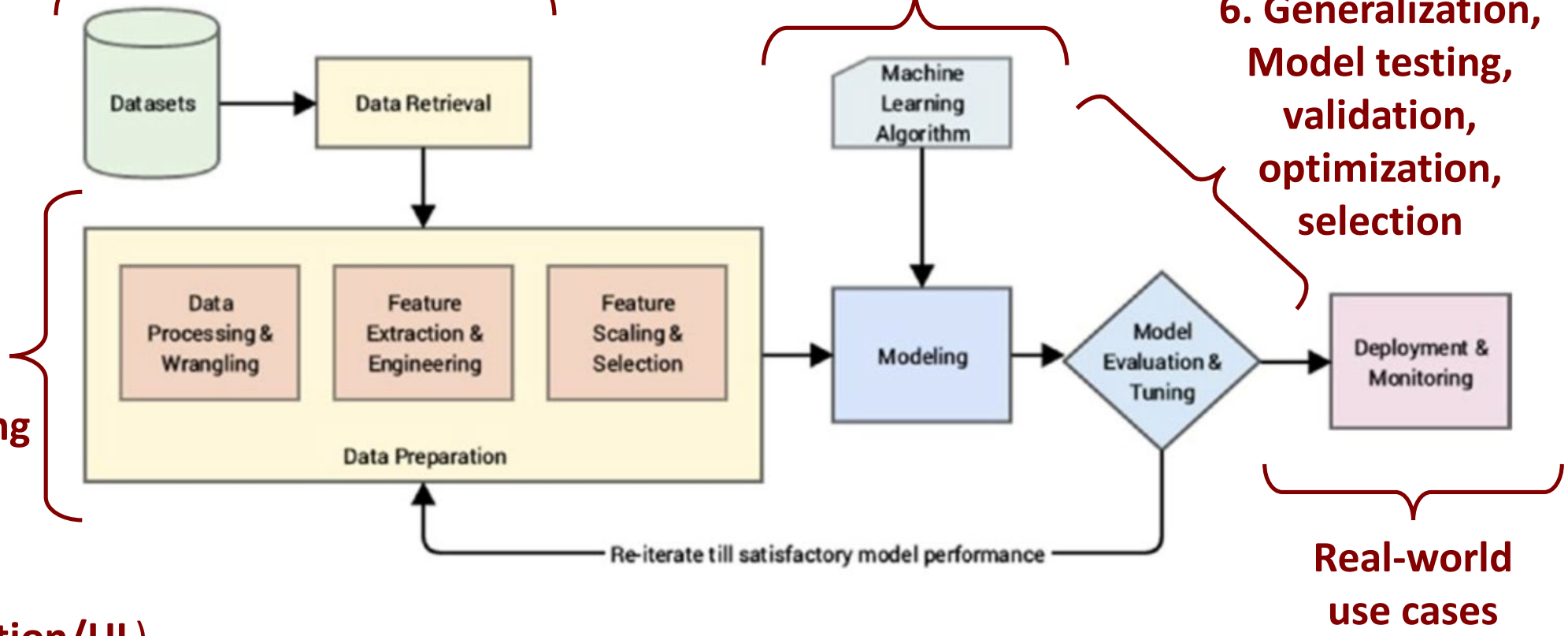
2 . Collection and **management** of relevant operational data

5. Machine Learning algorithms for classification and regression

6. Generalization, Model testing, validation, optimization, selection

3 . Data wrangling (transforming, cleaning, filtering, scaling, EDA, …)

4 . Feature engineering (feature selection, feature extraction, feature processing, **dimensionality reduction/UL**)



Real-world use cases

ML techniques

- ✓ **Unsupervised learning:**
  - Clustering models
  - Principal Component Analysis (PCA)
  - Autoencoders

- ✓ **Supervised Learning:**
  - Decision Trees
  - k-Nearest Neighbors
  - Naive Bayes
  - Logistic Regression
  - Support Vector Machines (SVMs)
  - Least Squares Linear
  - Regression
  - Regularization
  - Feature maps
  - Kernelization
  - Deep / Convolutional Neural Networks

Data science / ML software tools

Python 3.7

- CSV
- JSON
- HTTP

What you'll take home

- A toolkit of different skills useful to effectively go through the **entire ML / Data science pipeline**

- Conceptual and (mostly) practical knowledge about:
  - ✓ collecting, handling, exploring, and wrangling data in different formats (image, text, temporal) and originating from different sources
  - ✓ selecting, extracting and engineering data features using both manual and learning techniques;
  - ✓ identifying the most appropriate ML techniques for the problem and the data at hand;
  - ✓ implementing and using a set of core ML models;
  - ✓ testing and evaluating ML models;
  - ✓ using the Python ecosystem for ML and data science;
  - ✓ applying ML to problems from a range of different application domains.

Grading, rules

Key Information

Classes:	Lectures: UT 4:30 - 5:50pm - Room 2052
	Labs/Recitations: W 4:30pm - 5:50pm, Room 2062
Teacher	Gianni A. Di Caro
Units	9.0
Grading	35% In-class assessments (Quizzes, Labs), 35% Homework, 30% Project (Two Tasks)
Pre-requisites	15-112 or 15-110 passed with a C or a higher letter grade
Piazza	https://piazza.com/class/k53sxhx9tvt77b
Teaching Assistant	Aliaa Essameldin

- No smartphones / playing around!
- No late more than 5 minutes!
- Bored? Take a walk!

A Timeline of ML