Goals for Today

Basic Definitions:

- Data Science
- Machine Learning

The "Two Cultures" Problem

- Data modelers
- Algorithm modelers

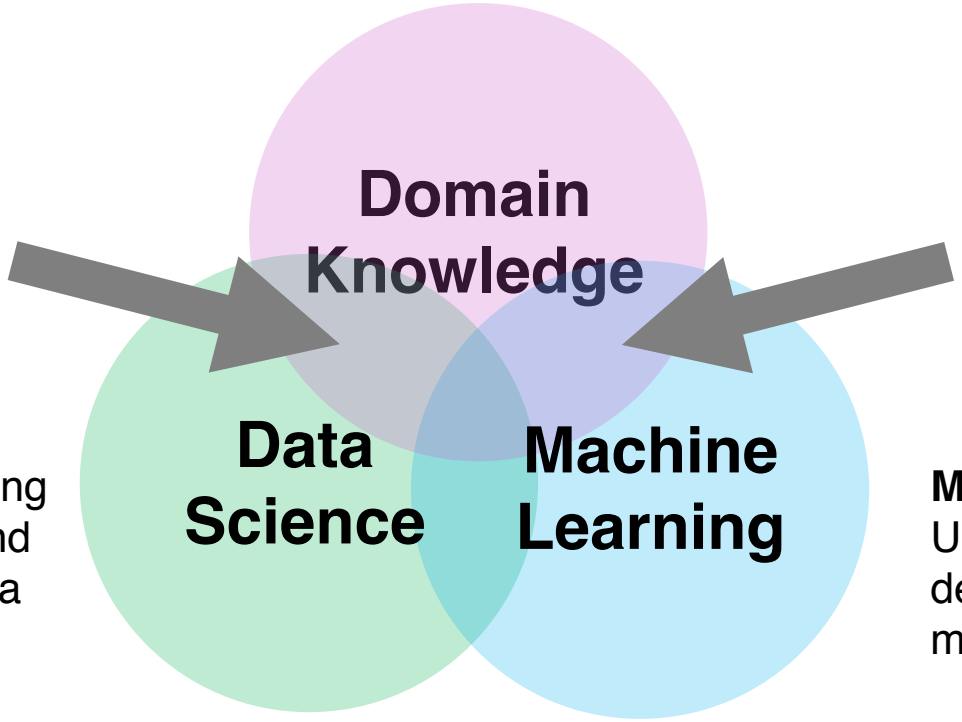
Types of Machine Learning:

- Supervised
- Transfer
- Unsupervised
- Reinforcement

Data Science and ML Overlap with Traditional Engineering

One of Two Venn Diagrams for today:

Domain Knowledge: The stuff that most of you have or will typically learn at school and on the job.



In the current environment, these overlaps are rapidly growing

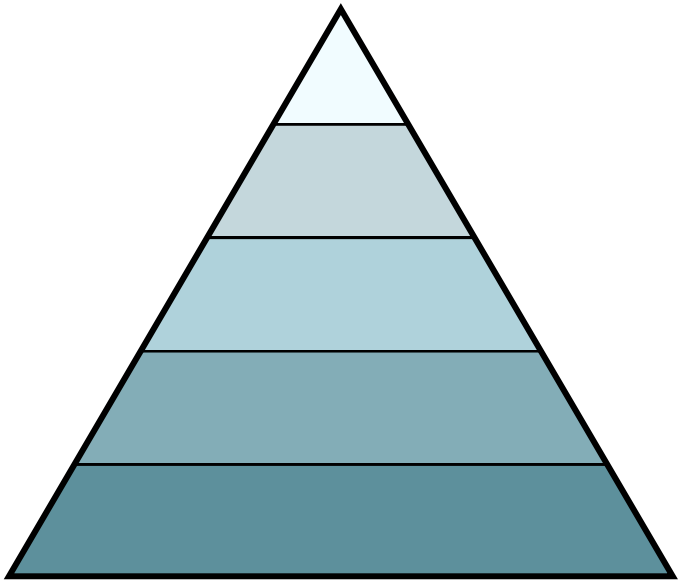
Data Science: curating analyzing, storing, and generally making data data useful.

Machine Learning: Using algorithms to develop predictive models from data

These overlaps matter: data literate domain experts are essential to effectively leveraging data

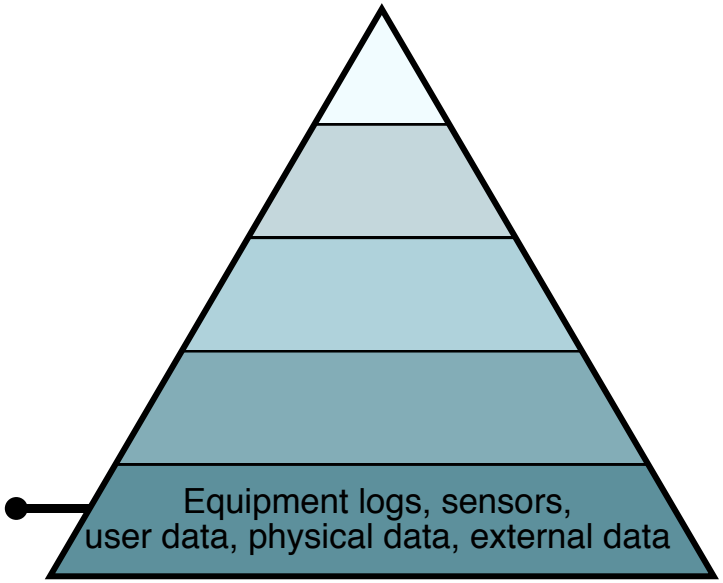
The Engineer and the Data Scientist

Hierarchy of Data Science Needs:



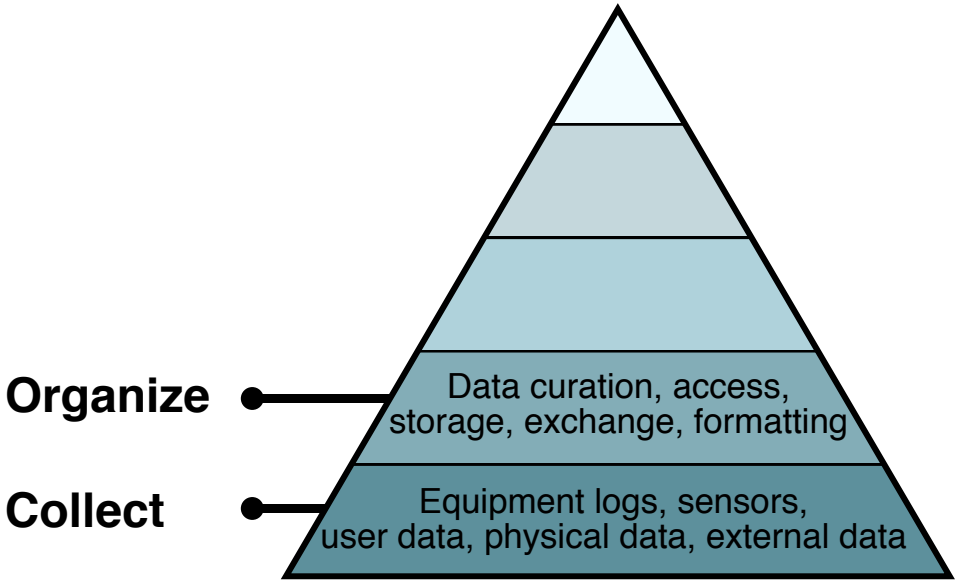
The Engineer and the Data Scientist

Hierarchy of Data Science Needs:



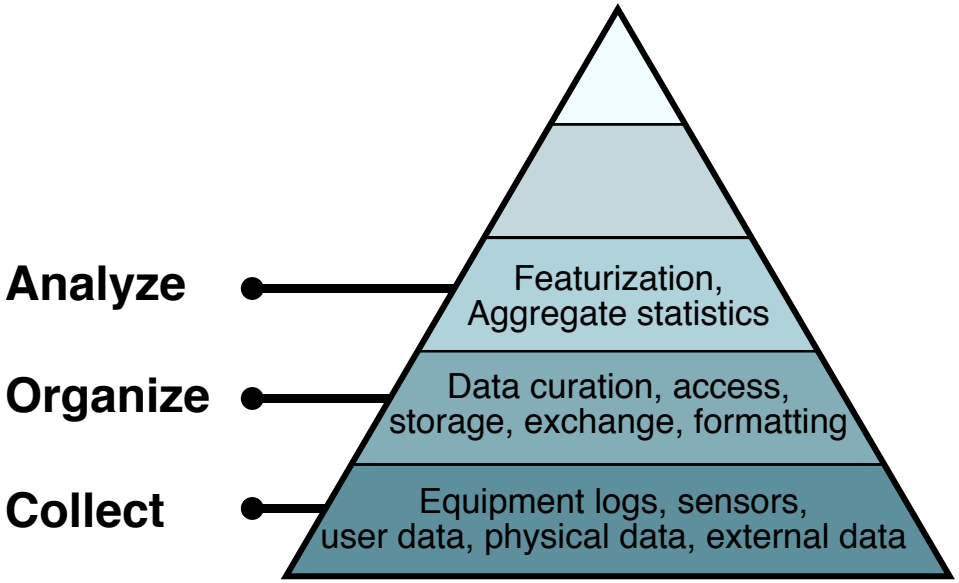
The Engineer and the Data Scientist

Hierarchy of Data Science Needs:



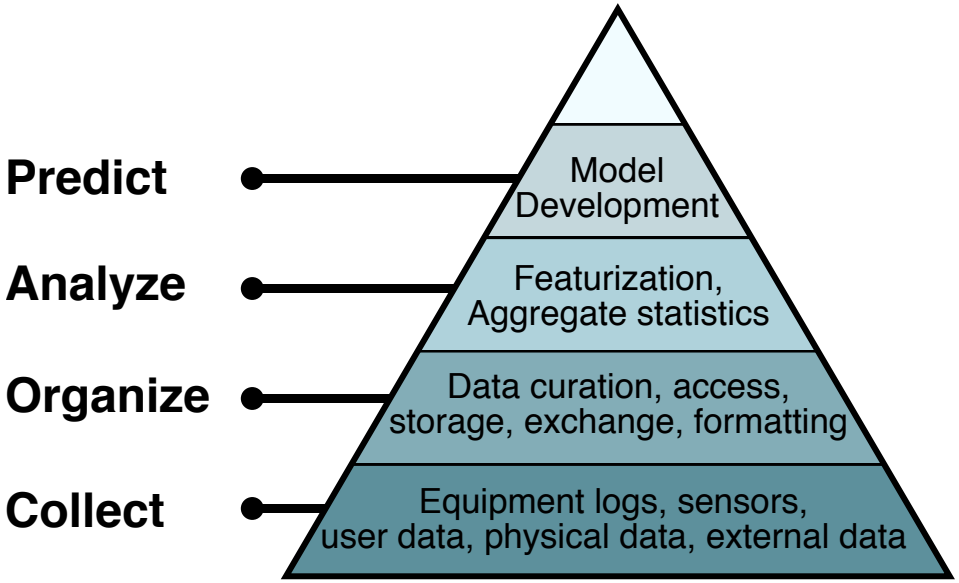
The Engineer and the Data Scientist

Hierarchy of Data Science Needs:



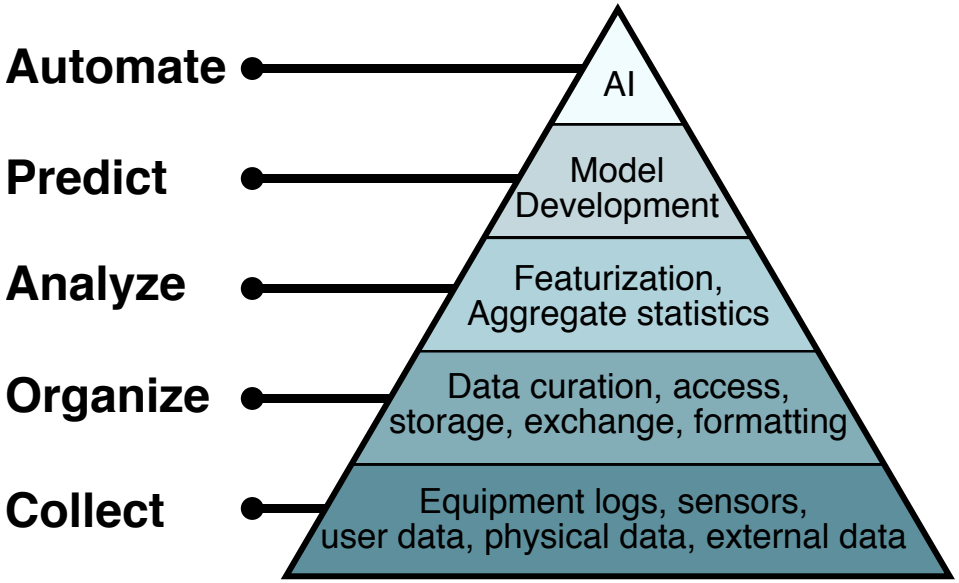
The Engineer and the Data Scientist

Hierarchy of Data Science Needs:



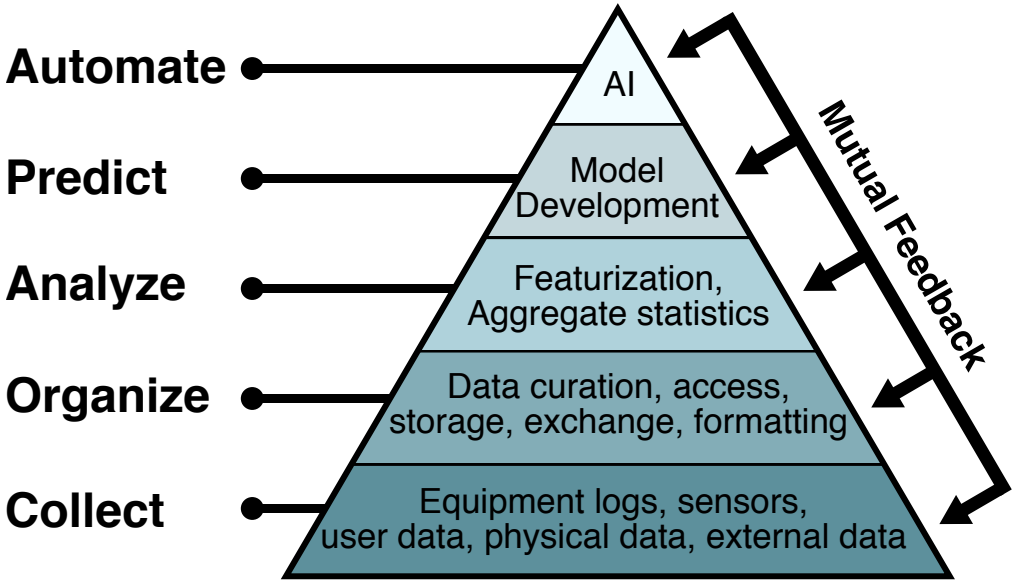
The Engineer and the Data Scientist

Hierarchy of Data Science Needs:



The Engineer and the Data Scientist

Hierarchy of Data Science Needs:



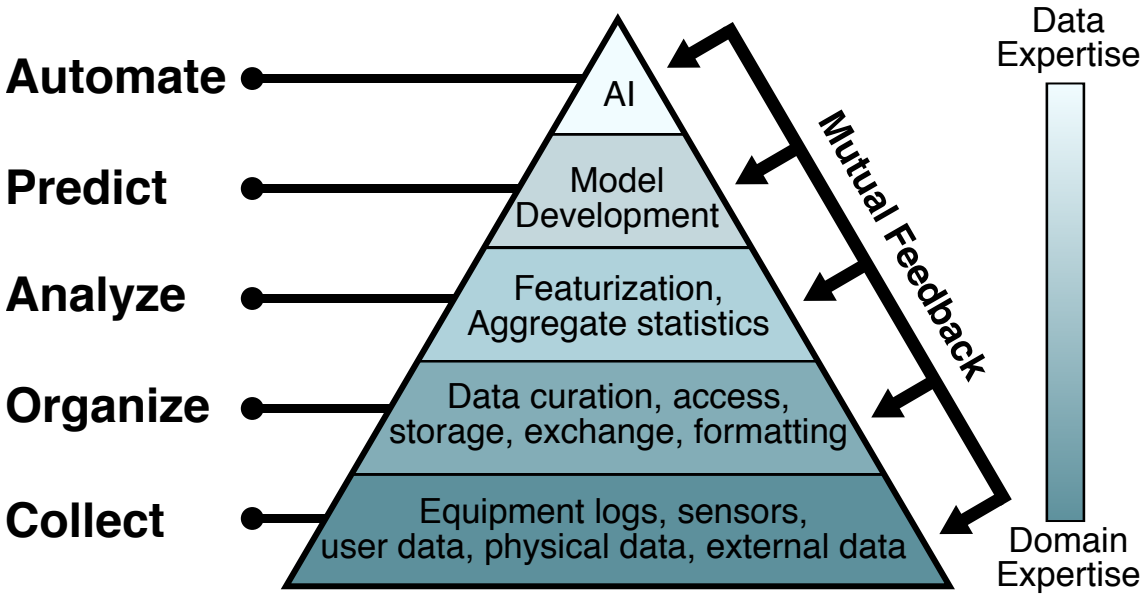
In this course we will cover the middle three tiers:

1. Organizing data (cleaning, standardizing, inputation, visualization)
2. Analyzing data (statistics, outlier detection, dimension reduction)
3. Prediction (model training and evaluation, supervised, unsupervised)

1 and 2 will occupy the first half and be utilized in 3 for the second half

The Engineer and the Data Scientist

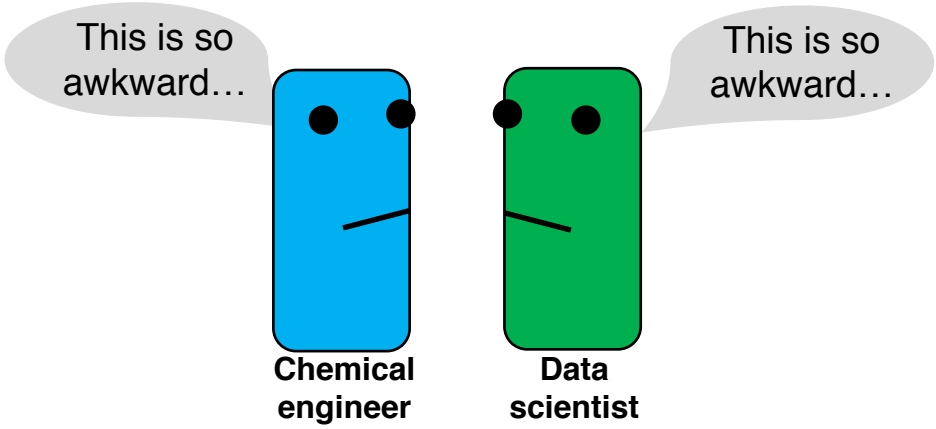
Hierarchy of Data Science Needs:



In the existing paradigm, the domain experts (like chemical engineers) and data scientists work from opposite ends of what needs to be a single unified framework

What does this mean for education?

Is it easier to teach an engineer data science, or a data scientist engineering?



The Engineer and the Data Scientist

Don't fall for the false dichotomy. In any effective application, they work together and need to speak a common language.

How would you integrate the output of reactor sensors, from different vendors, across multiple locations?

After building a data flow infrastructure, how would you model it in real time to optimize economic, safety, and quality constraints?

How would the model results be validated and implemented into reactor control?

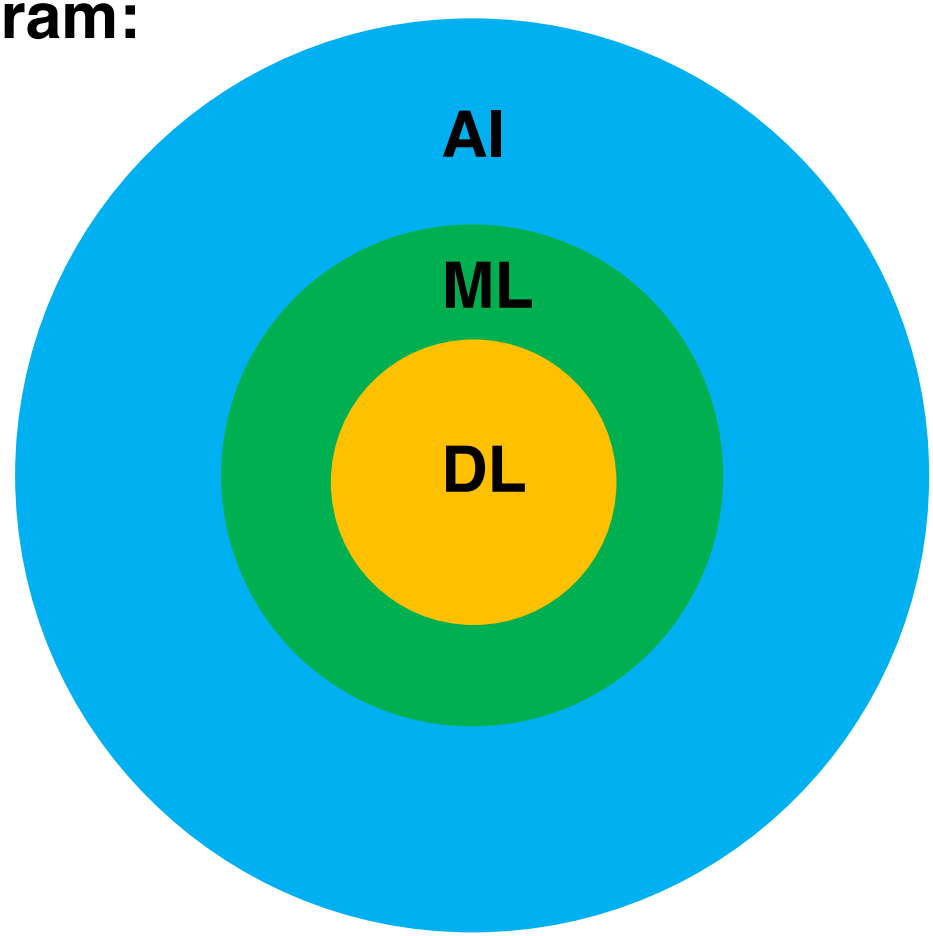
Who Knows the Difference Between AI/ML/DL?

These terms are confused in common use, but the most consistent definitions are based on the following diagram:

Artificial intelligence: The broadest class of activities where computers simulate things humans do.

Machine Learning: Programs that can modify themselves in response to more data.

Deep Learning: A special sub-branch of ML that uses neural networks and is not easily interpretable (more soon)

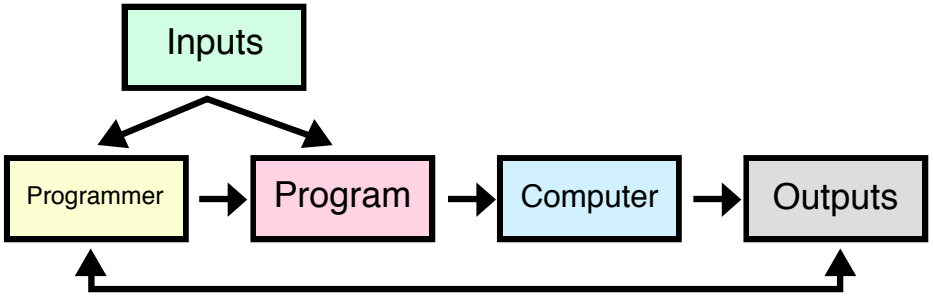


boundaries can be fuzzy!

The Big Difference Between AI and ML

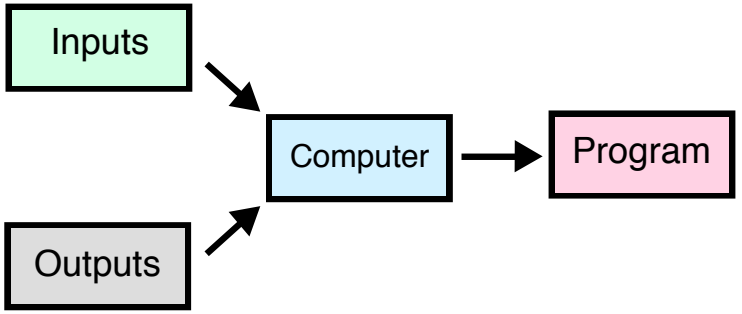
Traditional Programming:

Instructions (even if they are complicated) are hard-coded.



Machine Learning:

Instructions are generated based on the data



Where is the programmer?

"A breakthrough in machine learning would be worth ten Microsofts" – Bill Gates (Founder of Microsoft)

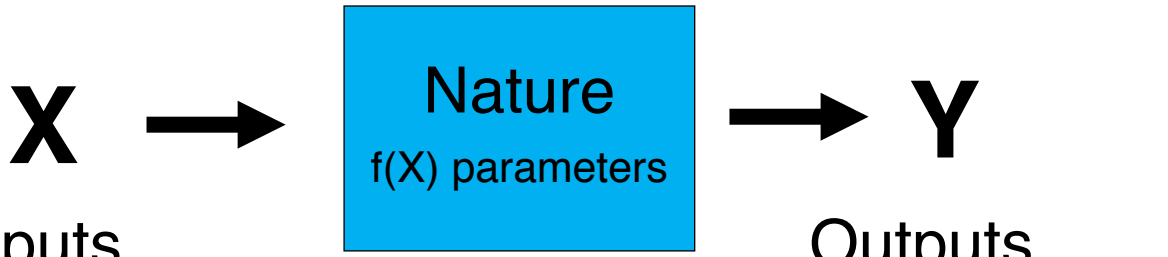
The "Two Cultures"

Leo Breiman, "Statistical Modeling: The Two Cultures", *Statistical Science* (2001)

"There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown"

Over 2000 citations, a highly influential article. Let's unpack this…

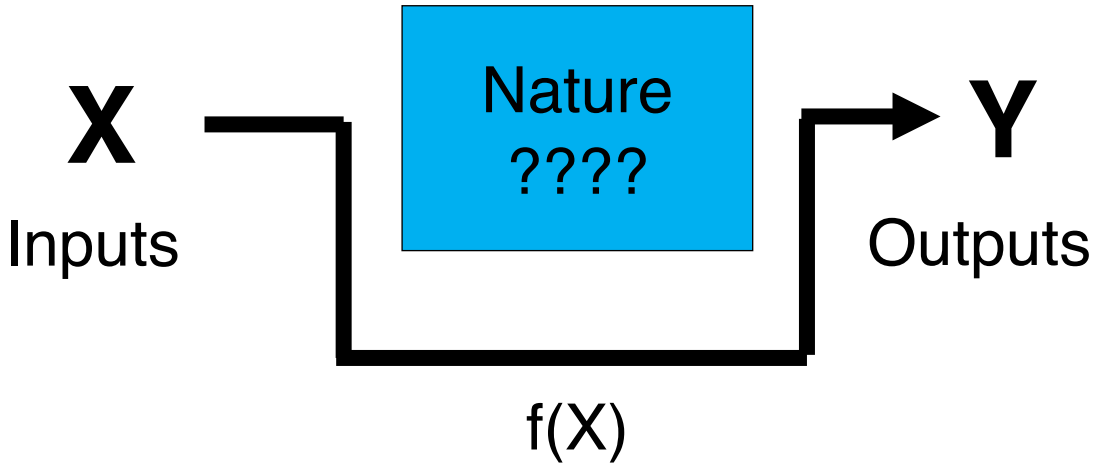
Culture 1: "The Data Modelers"



When the data modelers see data (X,Y), they starting thinking about models f(X) = Y that can map X→Y.

they evaluate f(X) based on physical mechanisms, how well it explains nature, and they give physical significance to the parameters

Culture 2: "The Algorithm Modelers"

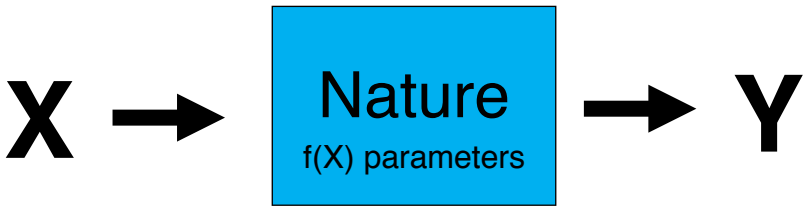


When the algorithm modelers see data (X,Y), they starting thinking about algorithms for choosing the best model of f(X) = Y that can map X→Y.

they evaluate f(X) based on how well X predicts Y, and they give don't usually try and interpret the significance of the parameters

The "Two Cultures"

The Data Modelers



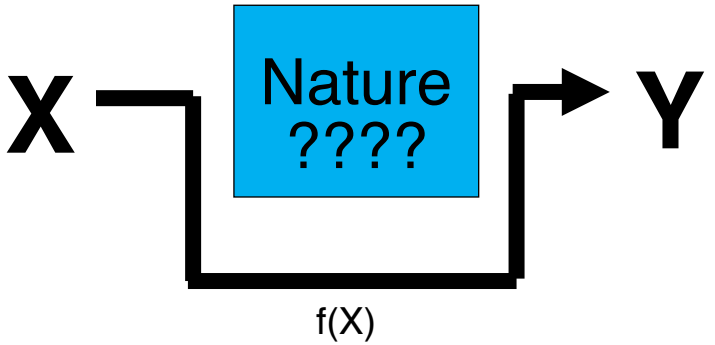
Try to find "the" model

Interpretability is very important

f(x) "explains"

Accept lower performance for simplicity

The Algorithm Modelers



Try to find a model

Interpretability is not important

f(x) "predicts"

Do not value simplicity only performance

The "Two Cultures"

Who is *right* ?

It depends on what we are talking about, doesn't it?

Four Types of Machine Learning

Supervised

- Labeled data
- Accuracy is easy to assess
- Typical goal: predict future

Transfer

- Multiple sets of labeled data
- Accuracy is easy to assess
- Typical goal: predict future using less data or less training.

Unsupervised

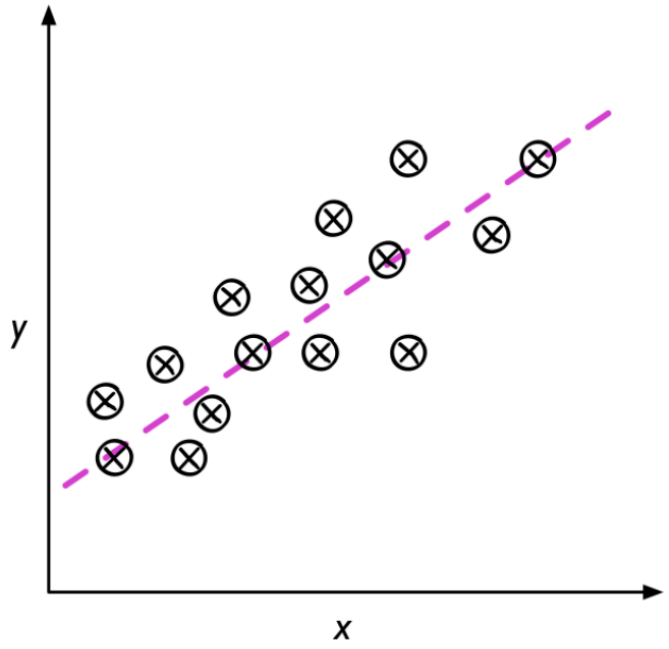
- No labels (what is what?)
- Accuracy is **not** easy to access
- Typical goal: find structure in data

Reinforcement

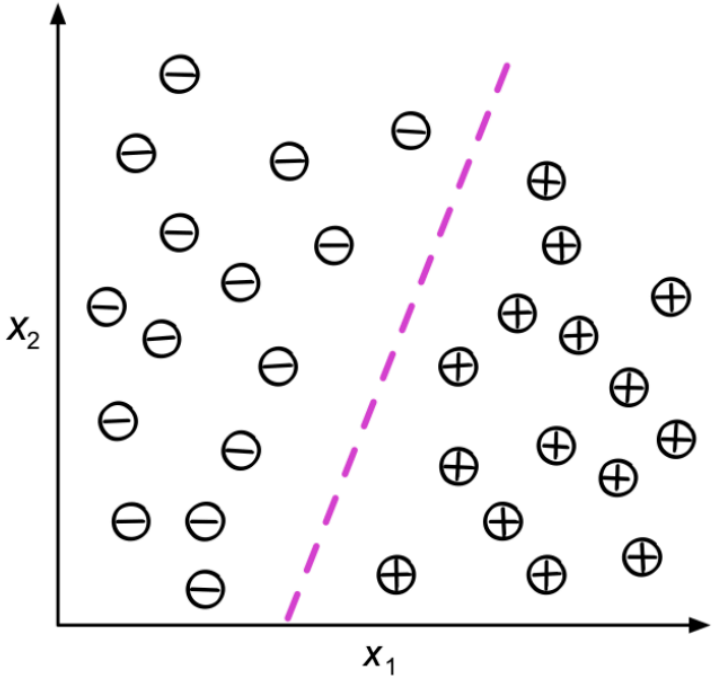
- Just a state (e.g., chess)
- Reward training (indirect)
- Typical goal: make a good move or series of moves.

Supervised Learning

Regression



Classification



- Labeled data
- Accuracy is easy to assess
- Typical goal: predict future

Adapted from Sebastian Raschka

Supervised Learning

Let's say you have reactor data and product pass/fail data from expensive quality controls studies. Building a model to predict pass/fail based on reactor data would be a **supervised learning** problem for **classification**.

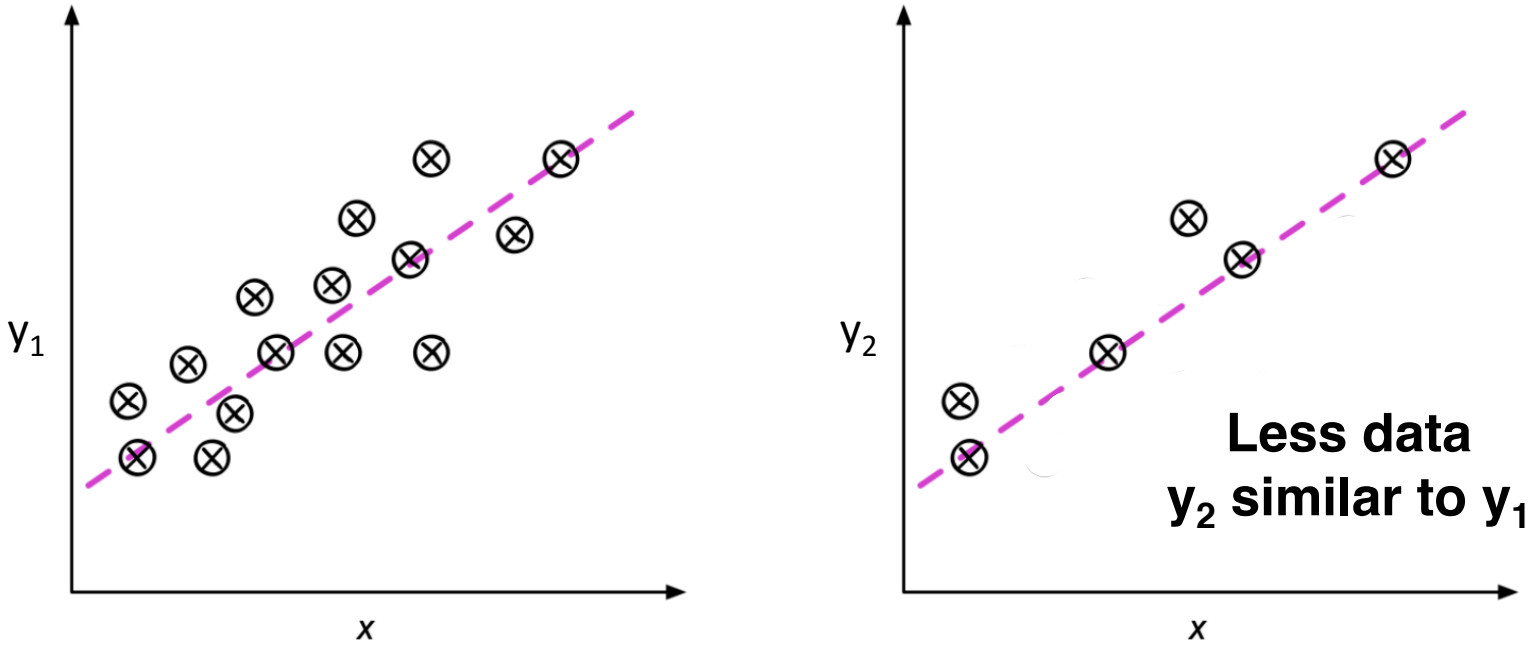
Let's say you have solubility data for 100,000 small molecules in diethyl ether. Building a model to predict the solubility of new molecules would be a **supervised learning** problem for **regression**.

- Typical goal: predict future

Adapted from Sebastian Raschka

Transfer Learning

Imagine y_1 and y_2 are different but x is the same



- Multiple Sets of labeled data
- Accuracy is easy to assess
- Typical goal: predict future using less data or less training

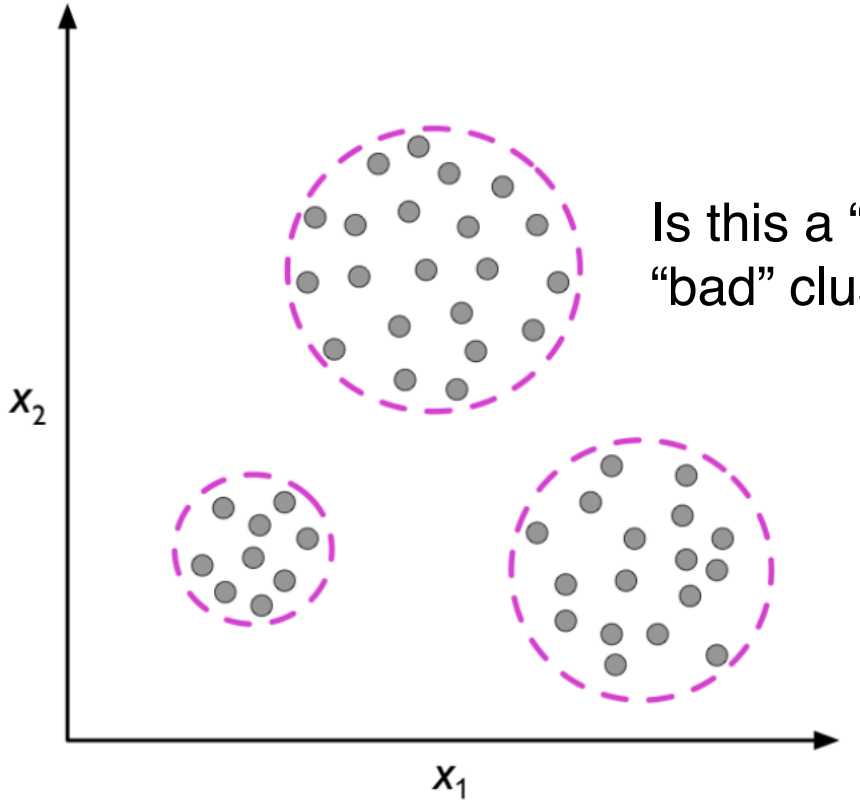
Transfer Learning

Let's say you have a small amount of reactor data and product pass/fail data from a new plant, but you have a lot of data from similar plants. Building a model to predict pass/fail based on reactor data from all plants would be a transfer learning problem.

Let's say you have an existing model for predicting solubility in diethyl ether, and now you want to train a new model for predicting solubility in mixed ethers. Retraining the diethyl ether model for this new task would be a transfer learning problem.

Unsupervised Learning

Clustering

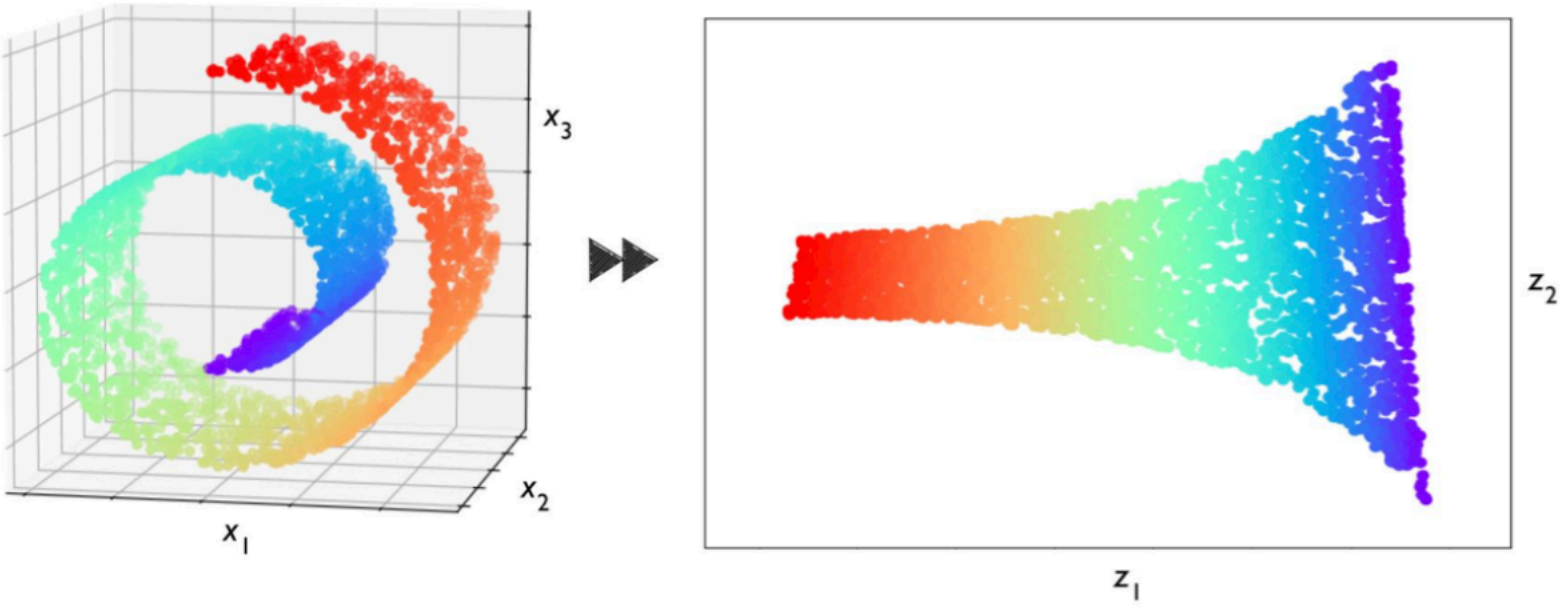


Is this a “good” or “bad” clustering?

- No labels (what is what?)
- Typical goal: find structure in data
- Accuracy is **not** easy to access

Adapted from Sebastian Raschka

Dimension Reduction (3rd dimension is redundant)



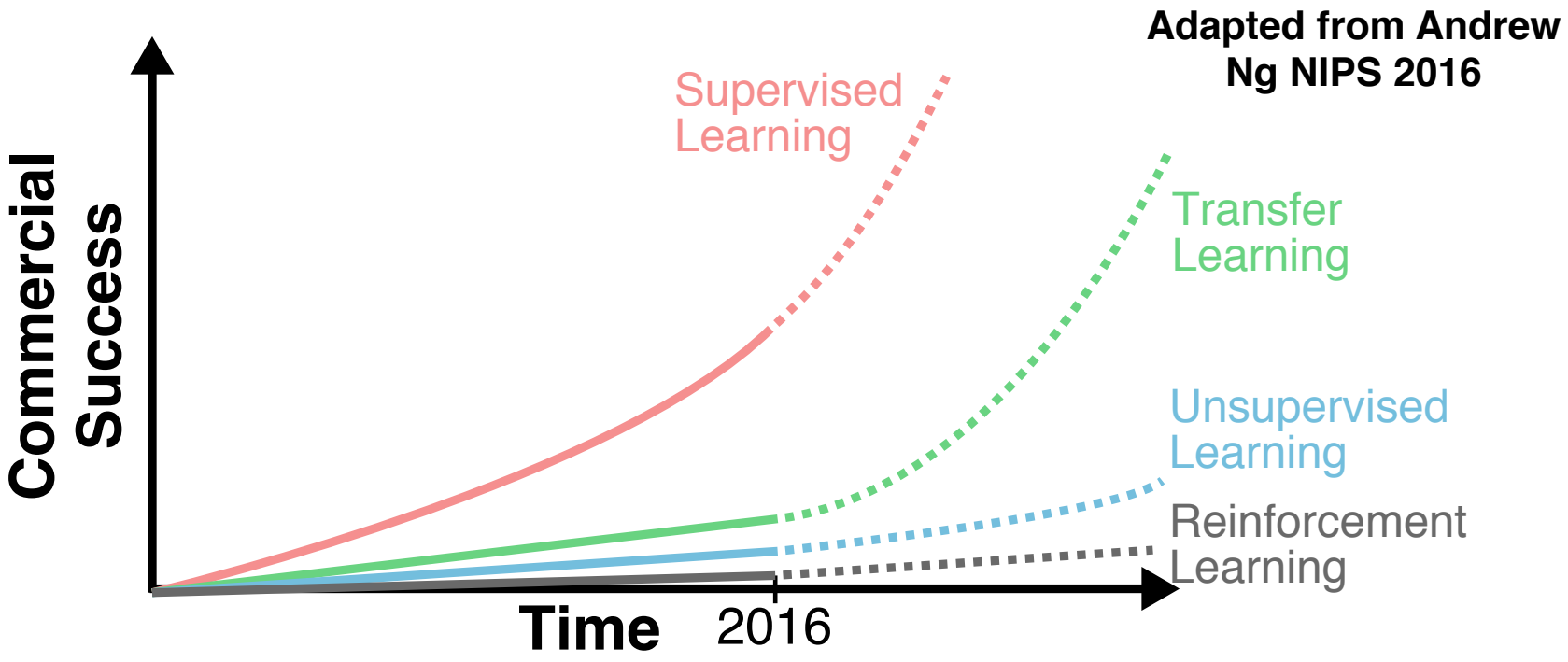
- No labels (what is what?)
- Typical goal: find structure in data
- Accuracy is **not** easy to access

Unsupervised Learning

Let's say you have logistics data with a massive number of variables and you are trying to schedule production for your plant. Only a small number of these variables will probably be useful for predicting demand. Finding out which ones are useful can be framed as an **unsupervised** learning problem.

Let's say you are analyzing pharmaceutical bioactivity assays and you want to find outliers in your data, but you don't have a physical model of the assay. Finding outliers and clustering similar compounds is an **unsupervised** learning problem.

Where are we heading?



- Supervised learning has dominated the most visible applications in industry (e.g., image and speech recognition)

- Transfer learning is already standard in many applications for effectively boosting mature supervised learning models. But many more substantial near-term opportunities exist.

Machine Learning in Engineering Applications

Popular Machine Learning*	Machine Learning in Engineering*
1) Data Rich	1) Data Scarce
2) Negative results reported	2) Systematic bias against publishing negative results
3) Low stakes for inaccuracies	3) High stakes for inaccuracies.
4) Non-hierarchical. Sloppy data curation and feature selection is forgiven.	4) Intrinsically hierarchical. Data curation and feature selection play a large role.
5) Weak desire for interpretability	5) Strong desire for interpretability
6) Testing and validation are cheap	6) Testing and validation are expensive

*These are illustrative not universal.