# Lecture 4: Fundamentals of Machine Learning Pt. 1

INFO 1998: Introduction to Machine Learning

CDS Education

# Lecture 4: Fundamentals of Machine Learning Pt. 1

**INFO 1998: Introduction to Machine Learning**

*Attendance!*

**CDS Education**

# Announcements

- Office Hours Update
- A2 grades released Friday, A3 due tonight, A4 released + due next Friday
- Partner finding megathread
- Weeks 6-8: mid semester check-in

# Mid Semester Project Check-In 👀

- Starting in 2 weeks (can complete anytime from 10/20 - 11/7)!

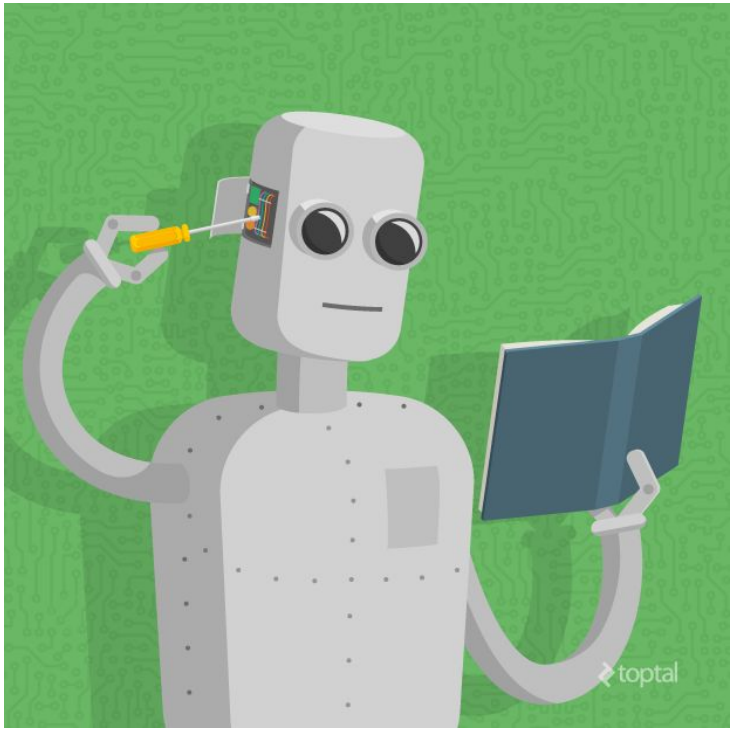
- Attend OH, and fill out a Google Form (will be posted on EdStem)
  - Expecting hypothesis/question/problem to solve
  - Chosen dataset
  - Some progress on data cleaning/data visualization

# Agenda

**Today's Goal:** be able to write code to do some kind of ML (to some extent)

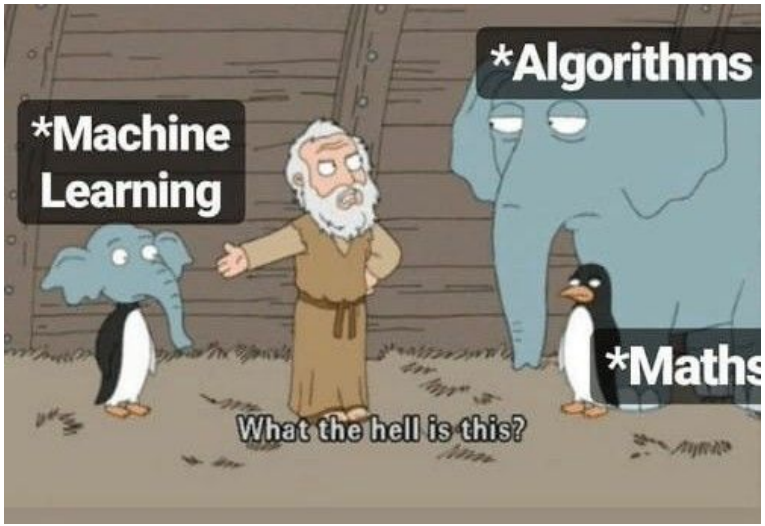
- **Define Machine Learning:** or like, 5 definitions
- **Start learning the language of ML:** There's a lot of terminology!
- **Try Linear Regression (via ScikitLearn)**: Our first ML algorithm!
- **Introduce our Workflow**: An outline for developing an ML model
- **Discuss Some Important Considerations**: What should we be thinking about as we're MLing?

# What's Machine Learning? Part 1: what does an ML engineer do




# Some Definitions of ML

- Give computers the ability to learn without being explicitly programmed
- Build a useful mathematical model, based on sample data, to make inferences
- Take in data and make predictions or decisions
- Help your computer learn patterns



# Machine Learning can involve:

- Preprocessing data
- Splitting and selecting pieces of data
- Doing mathematical analysis on the data
- Deciding what data structures are needed to efficiently implement algorithms
- Implementing accuracy metrics
- ...and a lot more

# What we're going to do:

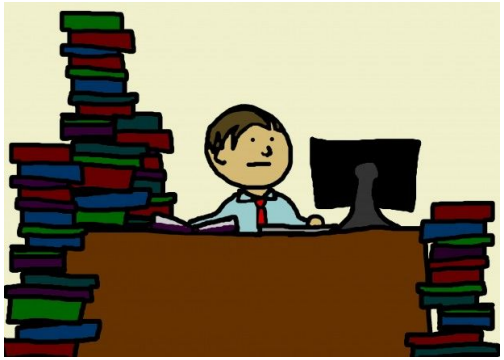
**Our main tasks:**

- **Formulate** a problem
- Find and **understand** data for that problem
- Choose a specific **algorithm class** to **solve** the problem
- Choose different parts of the data to **best** solve the problem
- Find which pandas, numpy, and scikit-learn functions do what we want
- Interpret the results and **fine-tune** our model

**Write as little code as possible!**

- Use pandas to deal with data
- Use numpy to do math
- Use scikit-learn ("sklearn") to make & analyze ML models

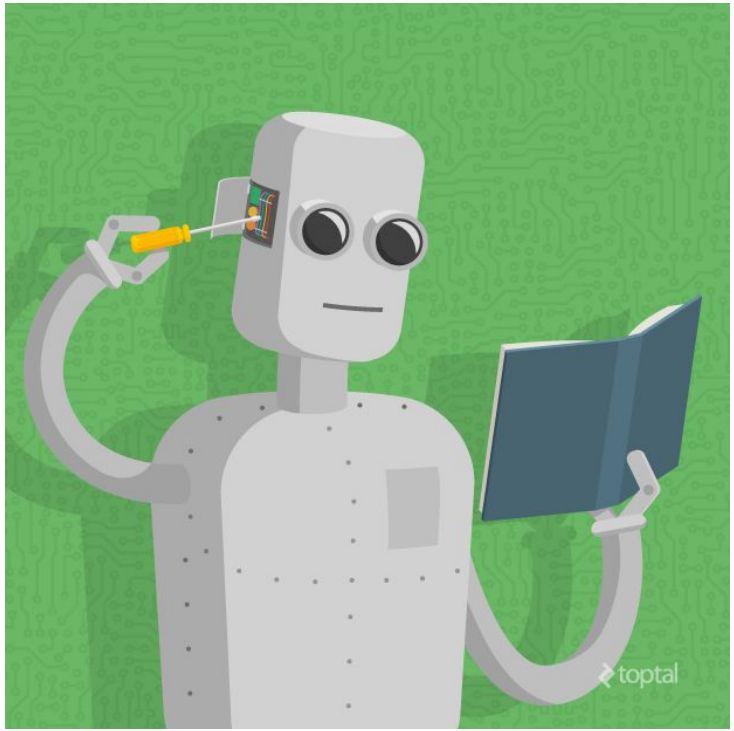
# Quick analogy: studying



- Setup
  - Goal: Be able to solve the test problems *well*
  - Resources: Practice problems + answers
- Method
  - You study those practice problems and answers. Given a problem, how do you get the answer?
- Result:
  - On the real test, the problems aren't the exact same as the practice problems. But they're similar!
  - Since you learned generally how to solve the practice problems, you can solve the similar test problems too :)

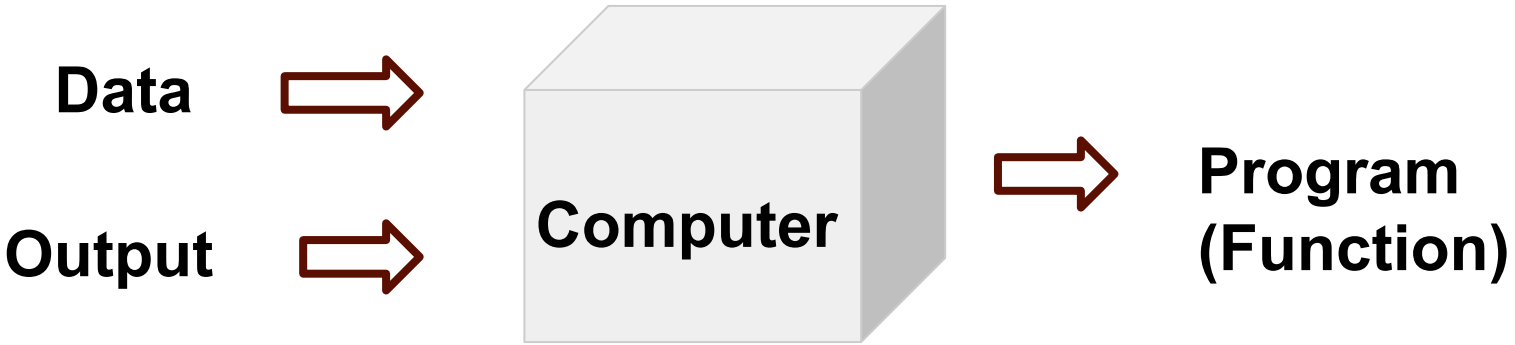
*Note: this analogy describes supervised learning

# What's Machine Learning? Part 2: like seriously what is it

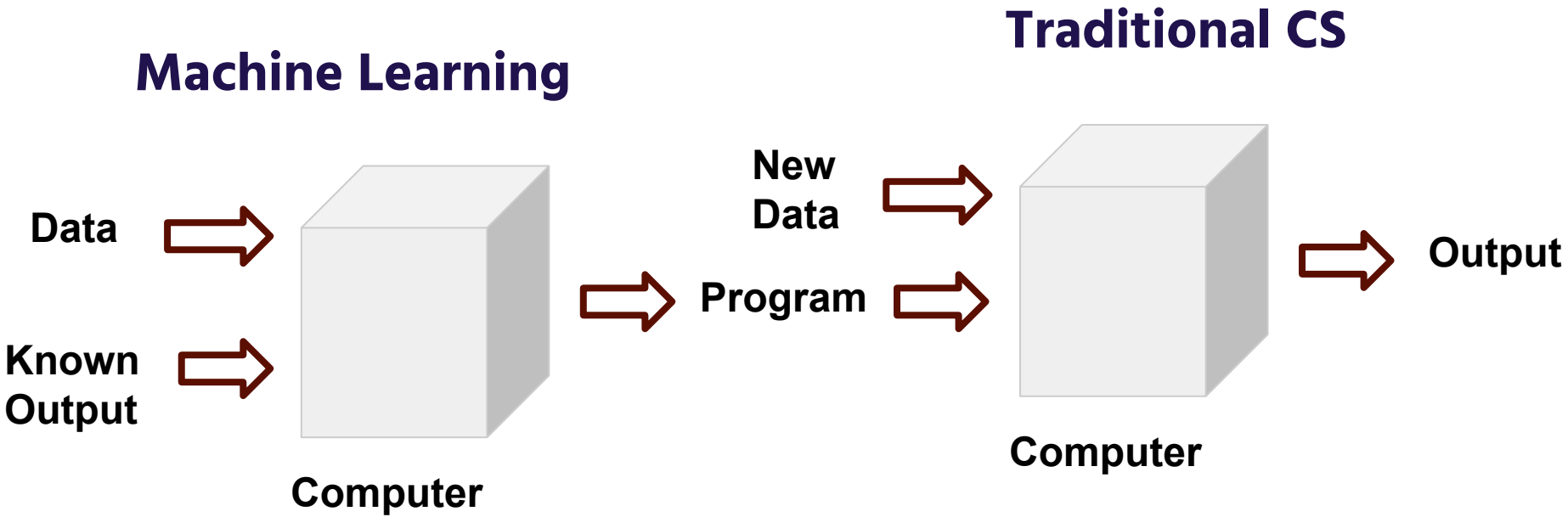


# Traditional Computer Science


# Machine Learning



# Using Machine Learning



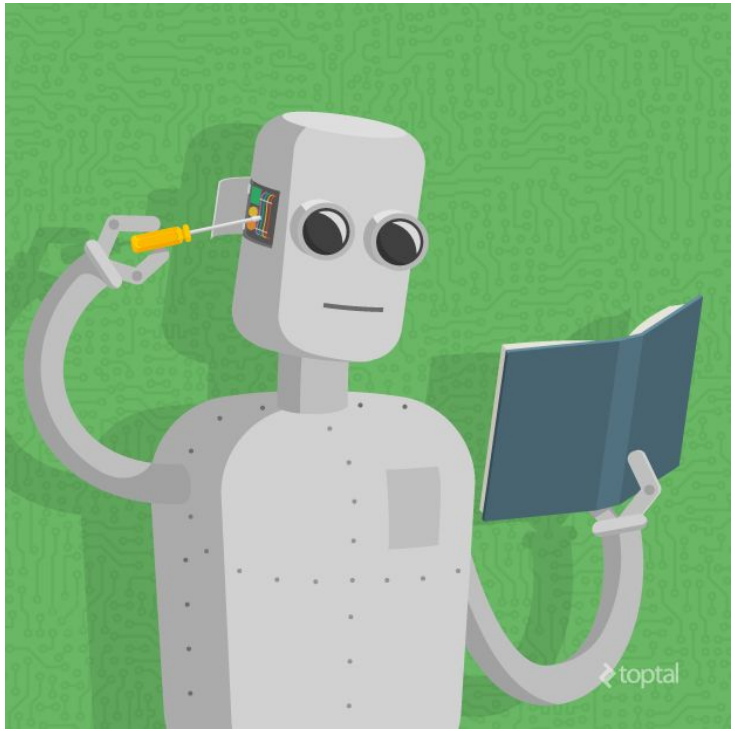
# Physics: A White-Box View of the World

- We have studied the world and crafted **models** *by ourselves* to represent it
- Our models are interpretable
- **White-box algorithms:** The inner workings of the algorithm are transparent


# Black Box Models

- Perhaps we can derive a process to solve the problem
  - Determine what a function would roughly look like
  - Think of relevant inputs
  - Allow the function to *build itself*
- **Black Box Model:** Results may not be interpretable!

# What's Machine Learning? Part 3: what's a model?




# What's a model?

- The output of a machine learning algorithm
- A procedure to produce some outputs when given some inputs
- A relationship between inputs and outputs
- A guess at how inputs and outputs are related
- A set of assumptions we're imposing on the dataset
- A parametrized function we can configure

# Review: Dataset Structure

- Rows are data points
  - AKA samples
- Columns are features
  - A sample is made of lots of features, including the goal

| | Name | Age | Major |
|---|---|---|---|
| 0 | Ann | 19 | Computer Science |
| 1 | Chris | 20 | Sociology |
| 2 | Dylan | 19 | Computer Science |
| 3 | Camilo | NaN | NaN |
| 4 | Tanmay | NaN | NaN |

# A Sample (Regression) Task

| name | city | state | adm_rate | undergrads | cost | compl_4 | median_hh_inc | median_earnings |
|---|---|---|---|---|---|---|---|---|
| Cornell University | Ithaca | NY | 0.1507 | 14226 | 63596 | 0.8639 | 80346.48 | 73600 |
| Washington University in St Louis | Saint Louis | MO | 0.1674 | 7032 | 65887 | 0.8643 | 79298.58 | 66300 |
| Lafayette College | Easton | PA | 0.3025 | 2505 | 61905 | 0.8653 | 85923.51 | 67500 |
| Johns Hopkins University | Baltimore | MD | 0.1412 | 5862 | 63509 | 0.869 | 81539.46 | 69800 |
| Vanderbilt University | Nashville | TN | 0.1168 | 6857 | 62320 | 0.8697 | 76279.78 | 64500 |

# What are some things we can do?

| name | city | state | adm_rate | undergrads | cost | compl_4 | median_hh_inc | median_earnings |
|---|---|---|---|---|---|---|---|---|
| Cornell University | Ithaca | NY | 0.1507 | 14226 | 63596 | 0.8639 | 80346.48 | 73600 |
| Washington University in St Louis | Saint Louis | MO | 0.1674 | 7032 | 65887 | 0.8643 | 79298.58 | 66300 |
| Lafayette College | Easton | PA | 0.3025 | 2505 | 61905 | 0.8653 | 85923.51 | 67500 |
| Johns Hopkins University | Baltimore | MD | 0.1412 | 5862 | 63509 | 0.869 | 81539.46 | 69800 |
| Vanderbilt University | Nashville | TN | 0.1168 | 6857 | 62320 | 0.8697 | 76279.78 | 64500 |

# Predict Median Graduate Earnings

| name | city | state | adm_rate | undergrads | cost | compl_4 | median_hh_inc | median_earnings |
|---|---|---|---|---|---|---|---|---|
| Cornell University | Ithaca | NY | 0.1507 | 14226 | 63596 | 0.8639 | 80346.48 | 73600 |
| Washington University in St Louis | Saint Louis | MO | 0.1674 | 7032 | 65887 | 0.8643 | 79298.58 | 66300 |
| Lafayette College | Easton | PA | 0.3025 | 2505 | 61905 | 0.8653 | 85923.51 | 67500 |
| Johns Hopkins University | Baltimore | MD | 0.1412 | 5862 | 63509 | 0.869 | 81539.46 | 69800 |
| Vanderbilt University | Nashville | TN | 0.1168 | 6857 | 62320 | 0.8697 | 76279.78 | 64500 |

# Pick Some Features

| name | city | state | adm_rate | undergrads | cost | compl_4 | median_hh_inc | median_earnings |
|---|---|---|---|---|---|---|---|---|
| Cornell University | Ithaca | NY | 0.1507 | 14226 | 63596 | 0.8639 | 80346.48 | 73600 |
| Washington University in St Louis | Saint Louis | MO | 0.1674 | 7032 | 65887 | 0.8643 | 79298.58 | 66300 |
| Lafayette College | Easton | PA | 0.3025 | 2505 | 61905 | 0.8653 | 85923.51 | 67500 |
| Johns Hopkins University | Baltimore | MD | 0.1412 | 5862 | 63509 | 0.869 | 81539.46 | 69800 |
| Vanderbilt University | Nashville | TN | 0.1168 | 6857 | 62320 | 0.8697 | 76279.78 | 64500 |

# Our Goal?

| name | city | state | adm_rate | undergrads | cost | compl_4 | median_hh_inc | median_earnings |
|---|---|---|---|---|---|---|---|---|
| Cornell University | Ithaca | NY | 0.1507 | 14226 | 63596 | 0.8639 | 80346.48 | 73600 |
| Washington University in St Louis | Saint Louis | MO | 0.1674 | 7032 | 65887 | 0.8643 | 79298.58 | 66300 |
| Lafayette College | Easton | PA | 0.3025 | 2505 | 61905 | 0.8653 | 85923.51 | 67500 |
| Johns Hopkins University | Baltimore | MD | 0.1412 | 5862 | 63509 | 0.869 | 81539.46 | 69800 |
| Vanderbilt University | Nashville | TN | 0.1168 | 6857 | 62320 | 0.8697 | 76279.78 | 64500 |
| Rutgers University | New Brunswick | NJ | 0.5845 | 35102 | 29076 | 0.5838 | 82669.68 | ? |
| Case Western Reserve University | Cleveland | OH | 0.3627 | 5039 | 59467 | 0.6311 | 69873.4 | ? |

# ML Algorithms

- We pick different kinds of algorithms to accomplish different tasks
- Classification
  - Group Data Into Distinct Classes
- Regression
  - Based on an input, provide a continuous-value output
- **“All Models Make Assumptions”**

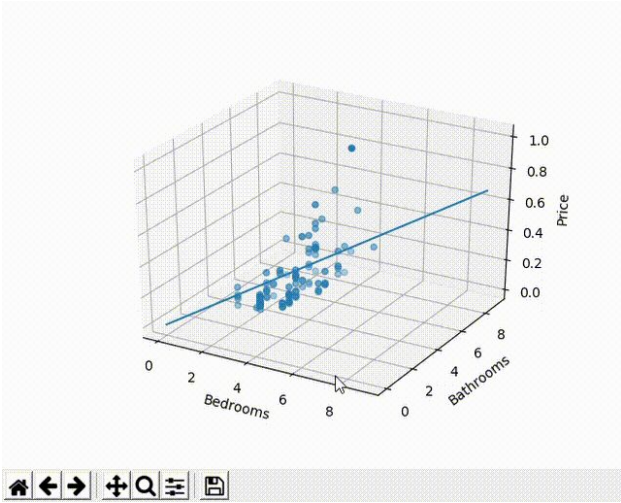
# Linear Regression

*Our first ML model!*

# Linear Regression

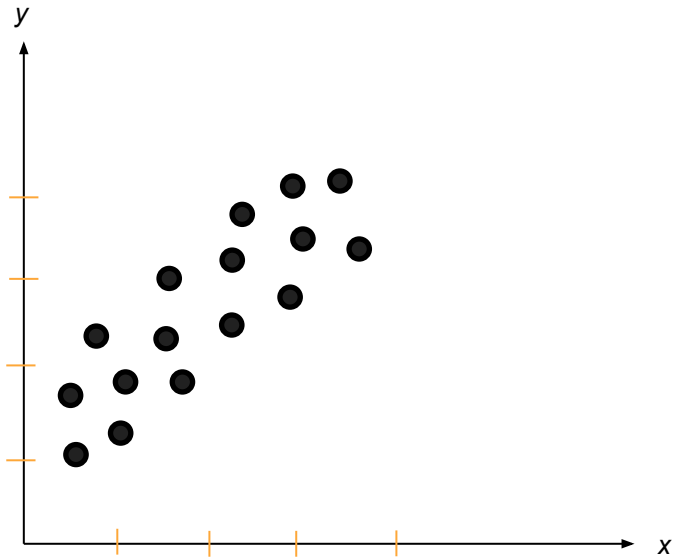
$$y = B_0 + B_1 x_1 + \ldots + B_p x_p + \varepsilon$$

- $x$ is an input; $x_1, x_2, \ldots, x_p$ are the features of $x$
- $y$ is an output (usually a single value)
- $B$'s are "weights"
  - a linear regression equation is defined by its $B$'s
  - "program" produced by ML
- Given a set of $x$'s and $y$'s, find the "$B$"-s that "most closely" satisfies the equation above for training points
- Plug in *new* values to of $x$ to predict respective $y$



# $y = B_0 + B_1 x_1 + \ldots$ is a model

- **A relationship between inputs and outputs**
  $y = B_0 + B_1 x_1 + \ldots$ relates inputs to outputs
- **A guess at how inputs and outputs are related**
  but $y = B_0 + B_1 x_1 + \ldots$ is just a guess/estimate; it's not exactly true
- **A set of assumptions we're imposing on the dataset**
  We're assuming output is linearly related to input features
- **A configurable thing (hyperparameters)**
  Sorry, we don't cover very much linear regression configuration here 😅

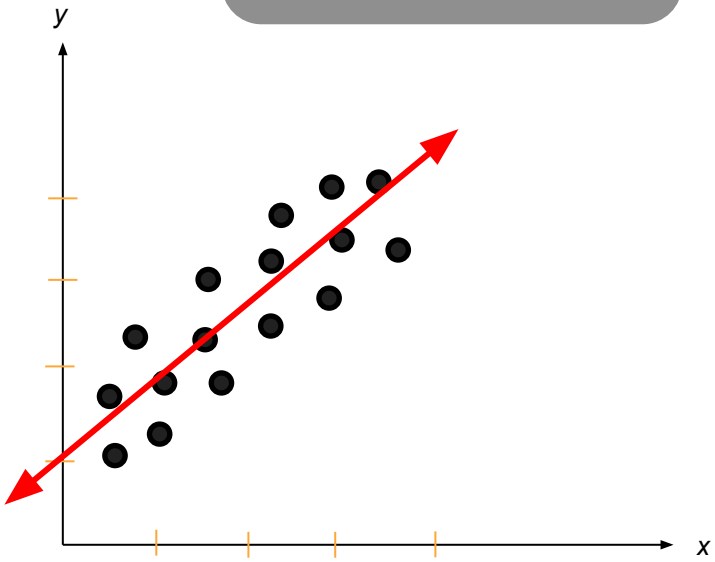



$y = B_0 + B_1 x$

# $y = B_0 + B_1 x_1 + \ldots$ is a model

Use algorithm to "learn" parameters that give us this line of best fit

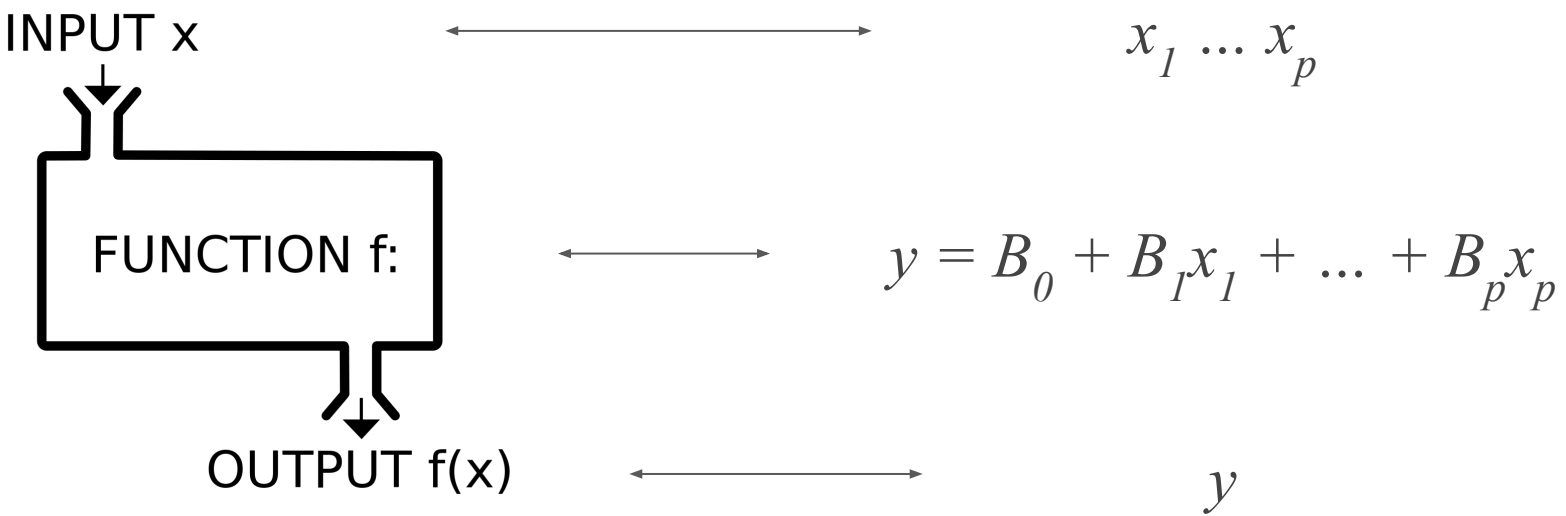
- **A relationship between inputs and outputs**
  $y = B_0 + B_1 x_1 + \ldots$ relates inputs to outputs
- **A guess at how inputs and outputs are related**
  but $y = B_0 + B_1 x_1 + \ldots$ is just a guess/estimate; it's not exactly true
- **A set of assumptions we're imposing on the dataset**
  We're assuming output is linearly related to input features
- **A configurable thing (hyperparameters)**
  Sorry, we don't cover very much linear regression configuration here 😅




$y = B_0 + B_1 x$

# Linear Regression

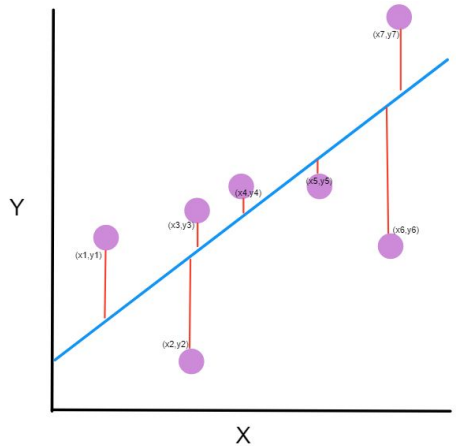
**Function**



# Linear Regression: Ordinary Least Squares

- *Ordinary least squares* method of linear regression calculates the weight vector $B$ by minimizing the **mean-squared error** of the predicted y-values

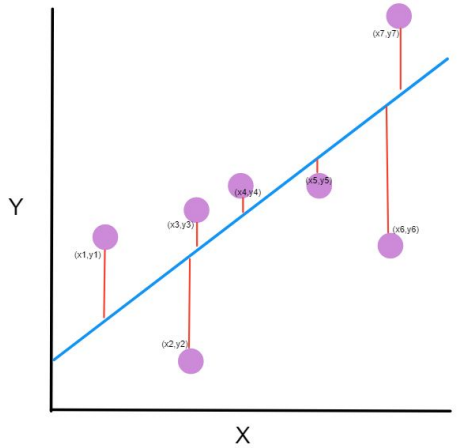
- Other types of linear regression (i.e. ridge regression) use different *loss functions* to calculate the weights




$$\text{MSE} = \boxed{\frac{1}{n}\sum_{i=1}^{n}} \boxed{(Y_i - \hat{Y}_i)} \boxed{^2}$$

Mean — Error — Squared

# “Training” a Model




- Dataset of n training points
- Datapoints: ($X$, $Y_i$) -> (input, output)
- Objective: Minimize MSE

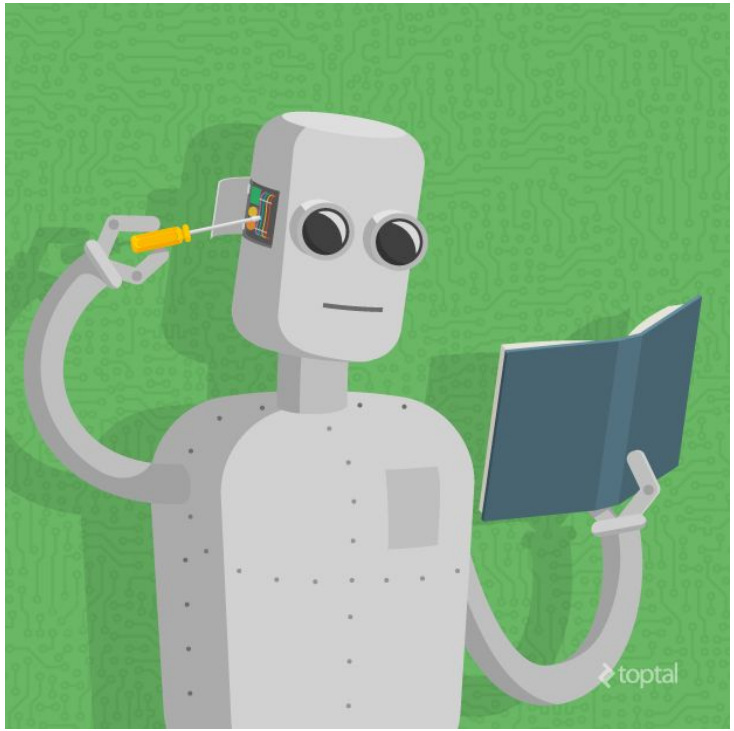
1. Use the $x$ values in our dataset to make a prediction
   a. Note: $x$ is a vector
2. Compare our prediction to the real $Y_i$
3. *Update B to get a better prediction*
   a. Special Algorithm: Gradient Descent
4. Repeat until MSE is as small as possible

$$\hat{Y}_i = B_0 + B_1 x_1 + \ldots + B_p x_p + \varepsilon$$

Mean Error Squared

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

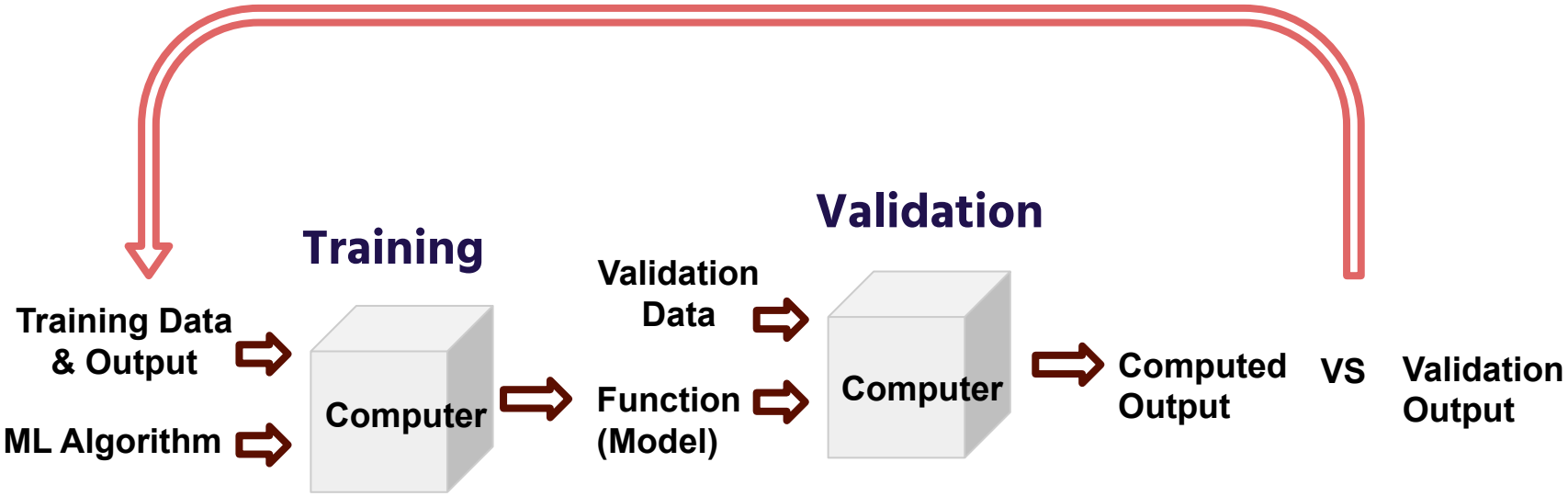
# What's Machine Learning? Part 4: What makes a *good* model?

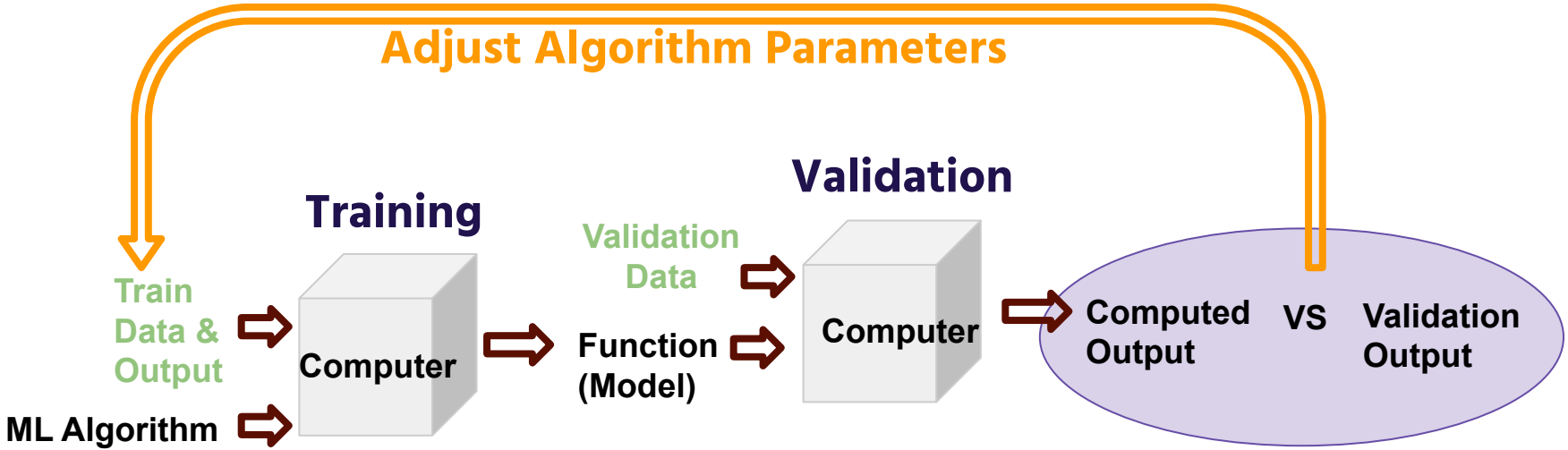


# Terminology: Training and Validating

- Split data into two sets
- Train model on one, validate on the other
- “Model training” = learn a relationship/program
    - e.g. give the linear regression data so it can define the $B$’s
- “Model validation” = see if the learned relationship is accurate on other data

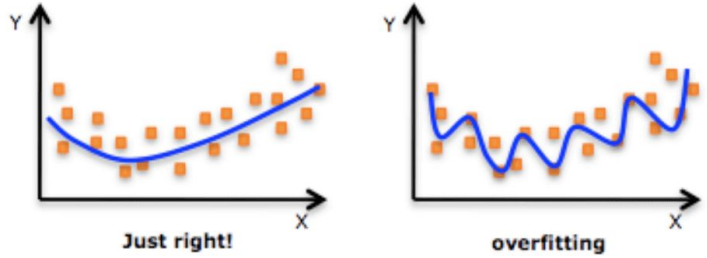
# Our ML Workflow






1. Select data
2. Assess model accuracy
3. Adjust Model

# Pitfall of Training: Overfitting




Model is accurate for **train** data

≠

Model can accurately predict **new** data

- We learned the specific mapping from **train input to train outputs…**
- But, we didn't learn the data's **general patterns** 🥲🥲🥲🥲🥲🥲

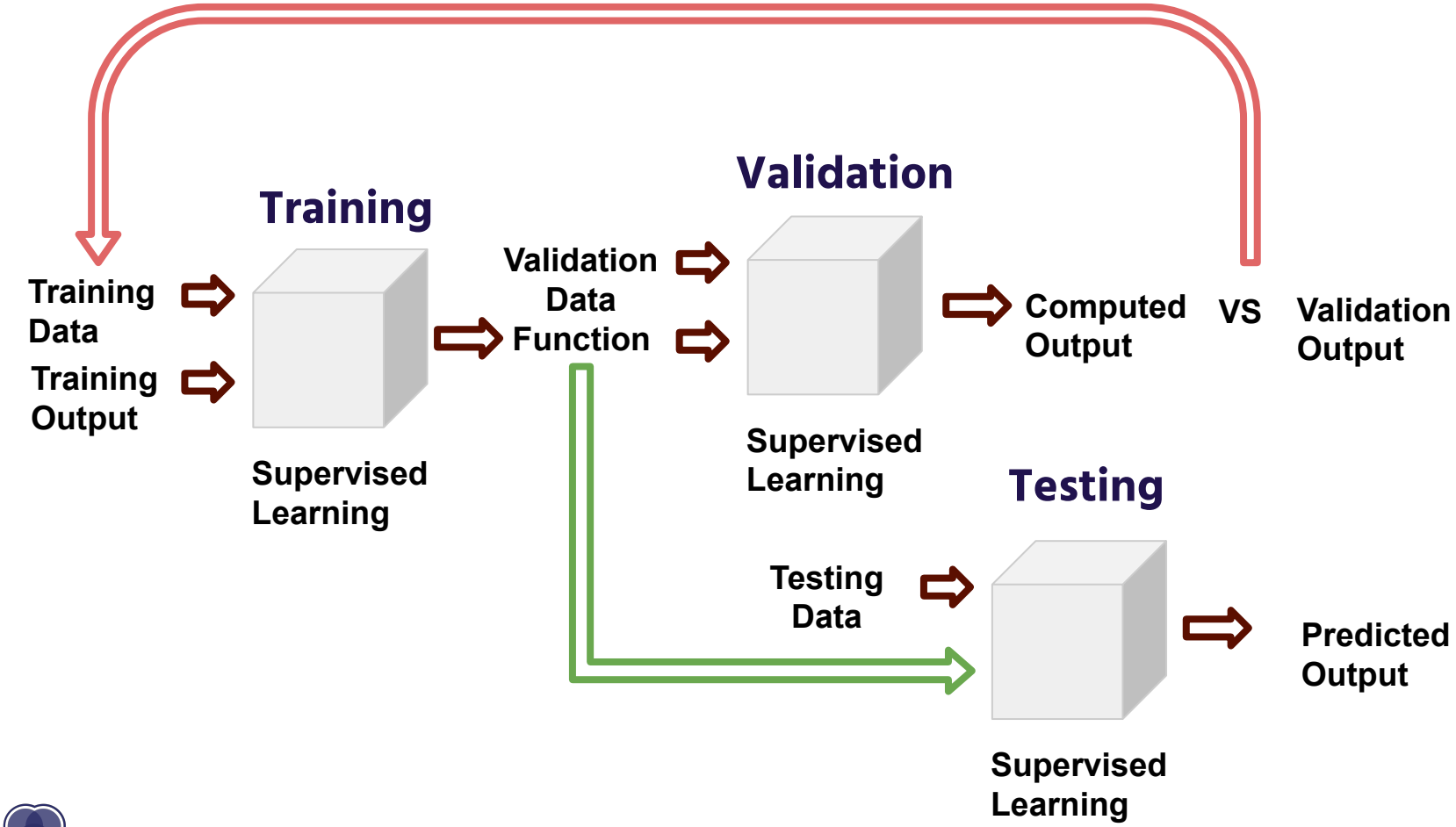
Solution: train on part of data, and check accuracy on a separate part of data (*validation* set)

# Pitfall of Validation: Overfitting

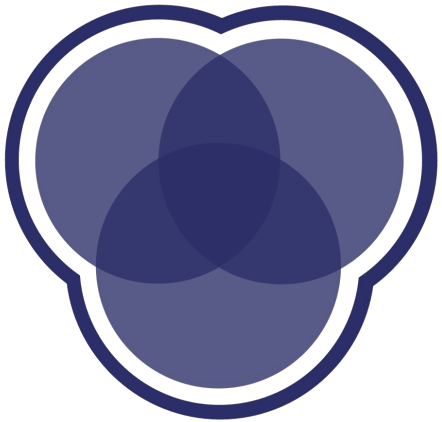
Predicting well on validation set ≠ Predicting well on new data

- We used the validation set to make our adjustments.
  - ⇒ Our model is **biased** to the validation set. 🥲🥲

Solution: keep a separate, rarely-used ***test*** set


Training
Training Data
Training Output
Supervised Learning
Validation
Validation Data Function
Supervised Learning
Computed Output
VS
Validation Output
Testing
Testing Data
Supervised Learning
Predicted Output

# Demo



## Model Goals

When training a model we want our models to:

- Capture the trends of the training data
- Generalize well to other samples of the population
- Be moderately interpretable

The first two are especially difficult to do simultaneously!
The more sensitive the model, the less generalizable and vice versa.

# Things to Keep In Mind

- Linear Regression is just one algorithm — we'll cover many more! 🤠
- The "model" produced by an algorithm is not always a simple equation like in linear regression.
- Validation is *really* important.
  - Overfitting is a huge problem!
  - We'll delve deeper in the next few lectures...

# Coming Up

**Next Lecture**: Assessing Model Accuracy + Fundamentals of ML

(a.k.a. *What's Machine Learning? Part* ∞)

CDS Education