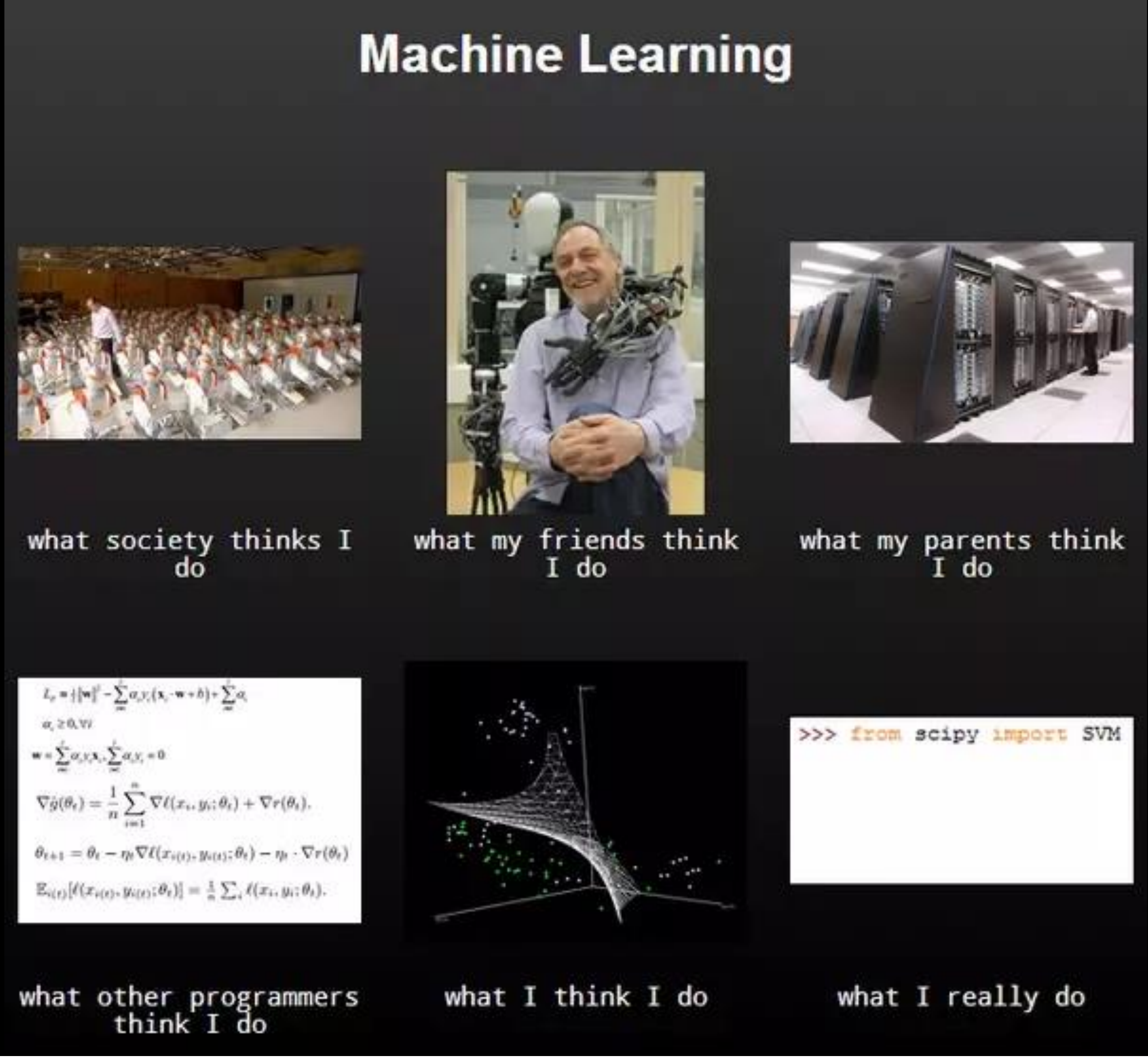


INFO 251: Applied Machine Learning

Intro to Machine Learning

For today’s: Once class starts, please turn your video on if you are able, I appreciate it!

Course Outline

- Causal Inference and Research Design
  - Experimental methods
  - Non-experiment methods
- **Machine Learning**
  - **Design of Machine Learning Experiments**
  - Linear Models and Gradient Descent
  - Non-linear models
  - Fairness and Bias in ML
  - Neural models
  - Deep Learning
  - Practicalities
  - Unsupervised Learning
- Special topics

Today's Outline

- Wrapping up causal inference
- Introduction to Machine Learning
- Supervised vs. Unsupervised Learning
- Key Issues in (Supervised) Machine Learning
- Design of Machine Learning experiments

Regression Discontinuity: Example

- When the discontinuity precisely determines treatment, this is equivalent to quasi-random assignment *in a neighborhood*
- For instance:
  - Everyone older than 75 as of Jan 31 2021 is eligible for a Covid vaccine
    - (Let's assume that compliance is perfect)
  - We might compare rates of illness between people born in January 1946 and February 1946
    - Identifying assumption: Rates of illness in 2021 among people aged born in Jan and Feb 1946 *would have been the same* in the absence of the vaccine

Regression Discontinuity: Estimation

- Quantifying the effect of the discontinuity
  - Instead of estimating: $GotSick_i = \alpha + \beta Vaccine_i + u_i$
  - We estimate: $GotSick_i = \alpha + \beta(Over75_i) + \delta(AgeInDays_i) + u_i$
    - $Over75_i$ is a binary "treatment" variable
    - $AgeInDays_i$ is the individual's age, in days
    - δ is a kernel (but just think of it as a constant, for now)
    - Estimated locally, for people with $s_{\text{min}} < AgeInDays_i < s_{max}$
  - Note the similarity to Instrumental Variables!
    - $\beta(Over75_i)$ is an instrument for treatment status

Regression Discontinuity: Summary

- Advantages
  - Takes advantage of a known rule for determining treatment status, which are common in the real world
  - Yields an unbiased estimate of treatment effect *at the discontinuity*
  - A group of eligible households or individuals need not be excluded from treatment
  - Can be used in other settings
    - Spatial discontinuities
    - Temporal discontinuities (event studies)

Regression Discontinuity: Summary

- Disadvantages
  - Produces *local average treatment effects (LATE)* that may not generalize to groups far away from the discontinuity
  - Effect is estimated at the discontinuity, so often this means there are fewer observations from which effects can be estimated
  - (Specification can be sensitive to functional form, including nonlinear relationships and interactions)

Econometrics: Summary

- Wikipedia says:
  - **Econometrics** is the application of mathematics, statistical methods, and computer science, to economic data and is described as the branch of economics that aims to give empirical content to economic relations
- For the purposes of this class:
  - **Econometrics** is an enormously useful set of quantitative methods for understanding associations and causal relationships in data

Econometrics: What you've learned

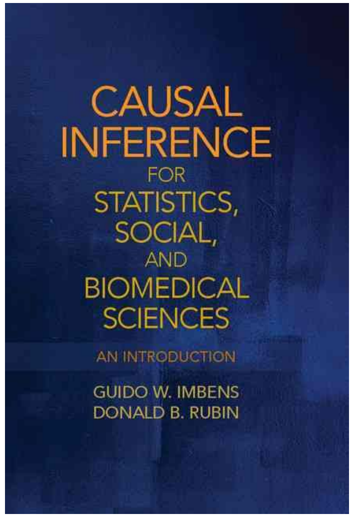
- Experimental methods
  - Design and randomization
  - Simple differences
  - Double differences
  - Regression
  - Fixed effects
- Non-experimental methods
  - All of the above and…
  - Instrumental variables
  - Regression discontinuity

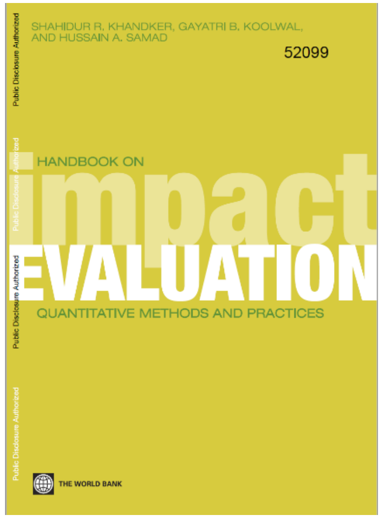
Econometrics: Key lesson

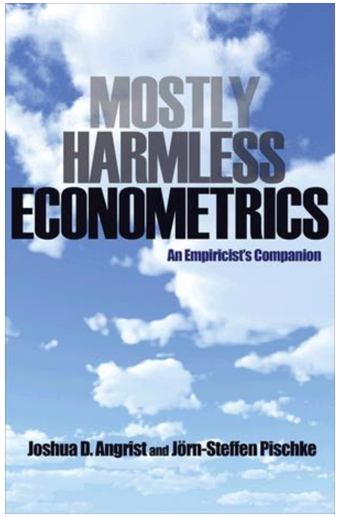
- No single method is “right” or “better”
- Each method requires a different identifying assumption, and implies a different counterfactual
- When deciding which method to use:
  - Determine which methods you *could potentially* use
  - For each candidate, articulate the identifying assumption
  - Brainstorm ways to possibly invalidate that assumption
  - Decide which assumption seems most reasonable, given your context, your data, and your situational knowledge

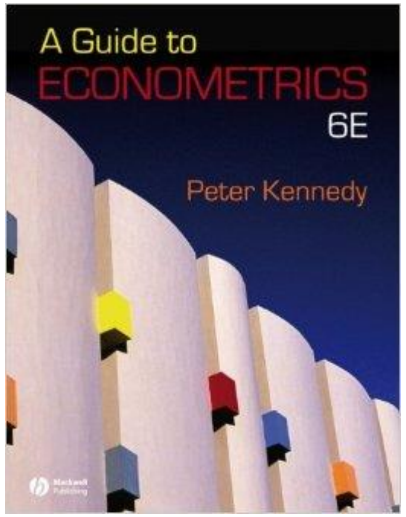
Additional Resources

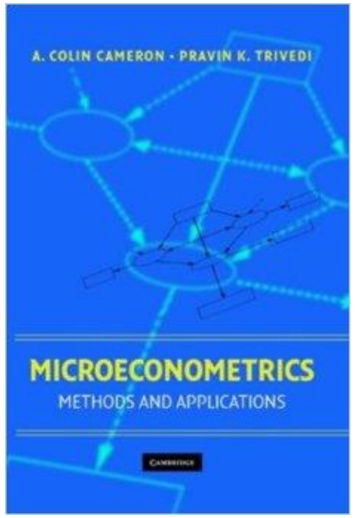
Beginner → Advanced











Key Concepts (IV and RD)

- Conditional exogeneity
- Instrumental variables
- First Stage
- Second Stage
- Reduced Form
- Exclusion restriction
- Instrument relevance
- Regression discontinuity
- Running variables

Today's Outline

- Wrapping up causal inference
- **Introduction to Machine Learning**
- Supervised vs. Unsupervised Learning
- Key Issues in (Supervised) Machine Learning
- Design of Machine Learning experiments

Key Concepts (today's lecture)

- Representation
- Evaluation
- Optimization
- Supervised Learning
- Unsupervised Learning
- The curse of dimensionality
- Feature engineering
- Overfitting
- Generalization

- Cross-validation
- Bootstrap
- Accuracy, ROC, AUC, F-scores
- Baselines
- Error analysis
- Ablative analysis

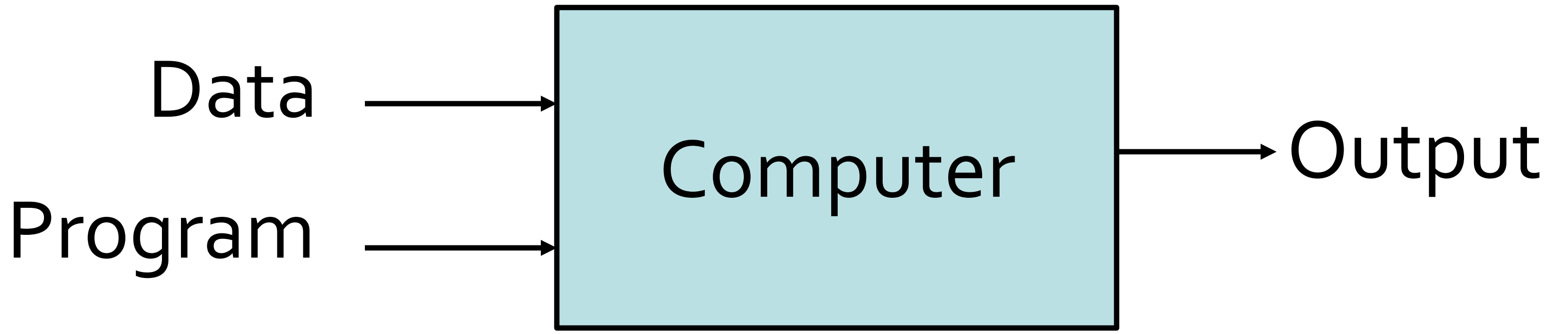
Machine Learning: Introduction

- **What is machine learning?**

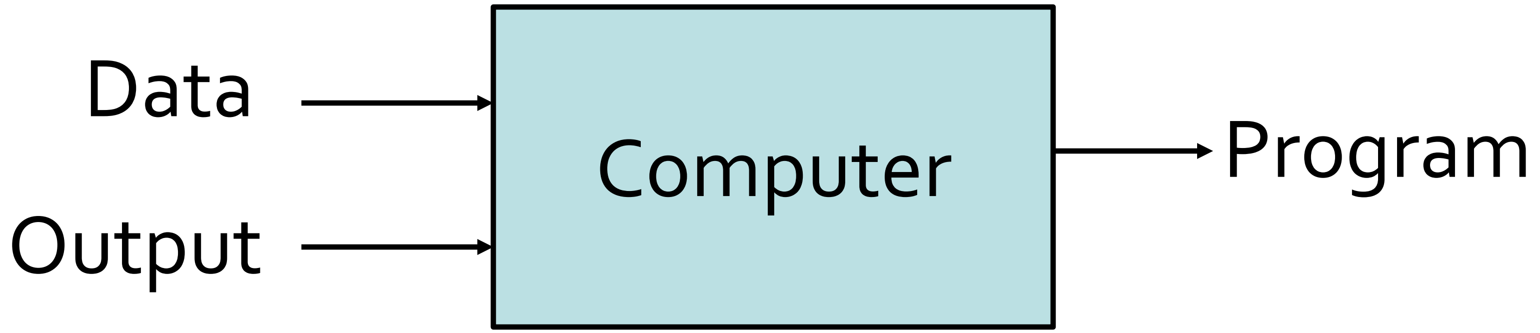
  - Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical **algorithms that can learn from data** and **generalize to unseen data**, and thus perform tasks without explicit instructions

What is Machine Learning?

- Traditional Programming



- Machine Learning



P.Domingos

What is Machine Learning?

- Econometrics: We start with a model $f(\cdot)$ of how the world works, e.g., $Y = \alpha + \beta X + \epsilon$
  - Our focus is on unbiased (and therefore generalizable) estimation of $\hat{\beta}$
  - How we specify $f(\cdot)$ is critical – the validity of causal inferences about β depend on it
- Machine learning: We start with a model $f(\cdot)$
  - Our focus is on accurate (and generalizable) **predictions** of $\hat{Y}$
  - This opens the door to new families of models that optimize for $\hat{Y}$, often at the expense of interpretability

Two paradigms of regression

1. Statistics / Econometrics
   - Explaining relationships
   - Understanding causality
   - E.g.: Why do customers churn?

2. Machine Learning / Computer Science
   - Predicting the future
   - Extracting generalizable patterns
   - E.g., Which customers will churn?

ML in a Nutshell

- Tens of thousands of ML algorithms exist
- Every ML algorithm has three components:
  1. **Representation** (i.e., the Model)
  2. **Evaluation** (i.e., an objective function)
  3. **Optimization** (e.g. Search)

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Representation / Model

- The choice of a model / representation imposes structure on what can be learned from data. Examples include:
  - Linear regression
  - Nearest neighbors
  - Decision trees
  - Neural networks
  - Probabilistic (graphical) models
  - Model ensembles
  - …
- It's common to fixate on the model representation, but in practice, many other factors are more important

Evaluation

- Is our model effective? Are the predictions accurate?
  - What is the model's "error", i.e., squared error, MAE, RMSE
  - "Coefficient of determination" (R^2)
  - Accuracy, Precision, Recall, F-scores, Area under the Curve
  - (Log-) Likelihood
  - Cost / Utility
  - Entropy, K-L divergence, etc.

Optimization / Search

- How to improve?
  - Combinatorial optimization (discrete)
    - E.g.: Greedy search
  - Convex optimization (continuous)
    - E.g.: Gradient descent
  - Constrained optimization
    - E.g.: Linear programming

Outline

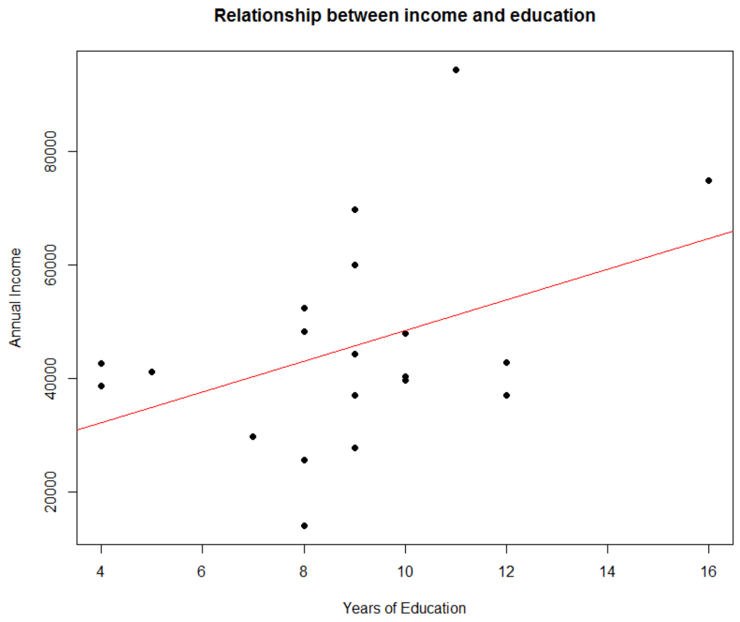
- Introduction to Machine Learning
- **Supervised vs. Unsupervised Learning**
- Key Issues in (Supervised) Machine Learning
- Design of Machine Learning Experiments

Supervised vs. Unsupervised

- Key distinction:
  - Whether or not you know the “right” answer

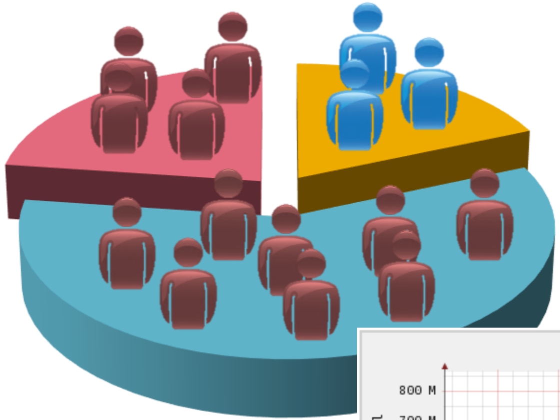
Supervised Learning

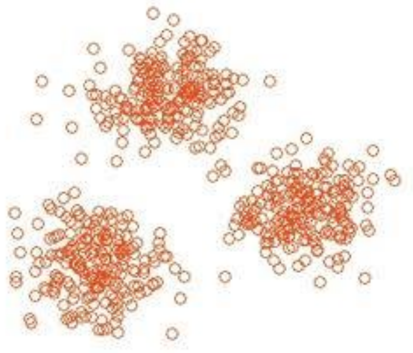
- We know the "right answer" for some values
  - Goal is typically to model relationship between inputs output, where values of output are known
- Examples
  - Disease classification, credit scoring, etc.
- Methods:
  - Linear models (regression, logistic regression, SVM)
  - Decision Trees, random forests
  - Neural Networks
  - Ensemble methods

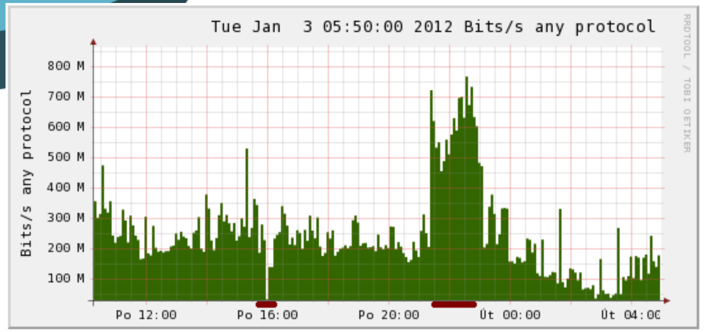


Unsupervised Learning

- We don't know the "right answer", the right groupings, "ground truth"
  - Goal is typically to discover underlying structure in the data
  - Often more exploratory than supervised learning
- Examples
  - Market segmentation, disease classification
  - Visualizing complex data
- Methods:
  - K-means and hierarchical clustering
  - Principal Component analysis
  - SVD, NMA, LDA







Other approaches to ML

- Semi-supervised learning
  - We have some labeled instances
- Reinforcement learning
  - Learning by interacting with an environment
  - Rewards from sequence of actions
- Etc.
  - Fair* Machine Learning
  - Online learning
  - Adversarial learning
  - ...

Outline

- Introduction to Machine Learning
- Supervised vs. Unsupervised Learning
- **Key Issues in Machine Learning**
- Design of Machine Learning Experiments

Key Issues in (Supervised) Machine Learning

- Generalization
  - "The fundamental goal of machine learning is to generalize beyond the examples in the training set. This is because, no matter how much data we have, it is very unlikely that we will see those exact examples again at test time."

Thanks to machine-learning algorithms, the robot apocalypse was short-lived.

Key Issues in (Supervised) Machine Learning

- Feature engineering
  - “Easily the most important factor is the features used.”
  - “This is typically where most of the effort in a machine learning project goes. It is often also one of the most interesting parts, where intuition, creativity and “black art” are as important as the technical stuff.”

Key Issues in (Supervised) Machine Learning

- More Data Matters
  - "As a rule of thumb, a dumb algorithm with lots and lots of data beats a clever one with modest amounts of it.."
- Some models require vast amounts of data and computation

Key Issues in (Supervised) Machine Learning

- Fast vs. exact solutions

Key Issues in (Supervised) Machine Learning

- Ensembles work
  - Bagging: resample the training data to generate multiple data sets, and train classifiers on each one
  - Boosting: Focus on examples that are hard to learn
  - Stacking: Use models to learn from the outputs of other models

Key Issues in (Supervised) Machine Learning

- Interpretability is (usually) important
  - There is beauty in simplicity!
  - Interpretability is hard to measure, but often trumps other measures of performance

Key Issues in (Supervised) Machine Learning

- Summary
  - Generalization and overfitting
  - Feature engineering
  - More data matters
  - Ensembles work
  - Interpretability is important

Outline

- Introduction to Machine Learning
- Supervised vs. Unsupervised Learning
- Key Issues in (Supervised) Machine Learning
- Design of Machine Learning experiments
  - **Motivation**
  - Training, testing, validation, cross-validation and bootstrap
  - Measuring performance
  - Choosing appropriate baselines
  - Error Analysis

What model should you use?

- "All models are wrong but some are useful."

George Box
1919 - 2013

What model should you use?

- Which of the following models is the "right" one?

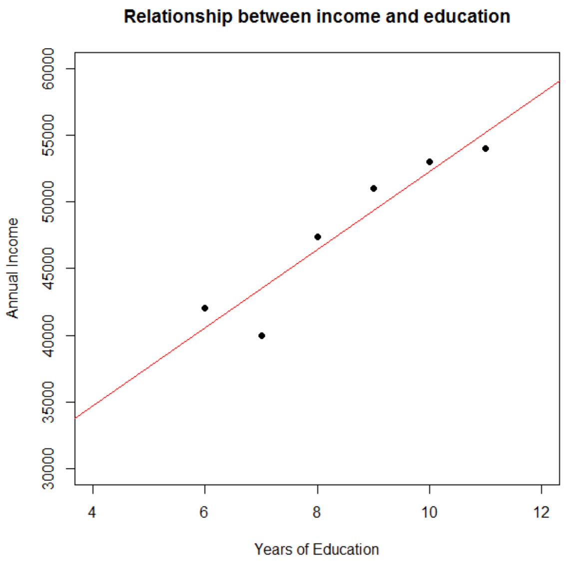
```
sm.ols('income ~ education', data=sl).model.fit().summary()
(Intercept)   education
   24287.71     2518.60
sm.ols('income ~ education + youngkids', data=sl).model.fit().summary()
(Intercept)   education   youngkids
  24590.811    2565.921   -2383.692
sm.ols('income ~ education + youngkids + age', data=sl).model.fit().summary()
(Intercept)   education   youngkids         age
12013.36940  2660.38919    19.25572   274.02269
```

- "But doesn't R-squared tell us the best model?"

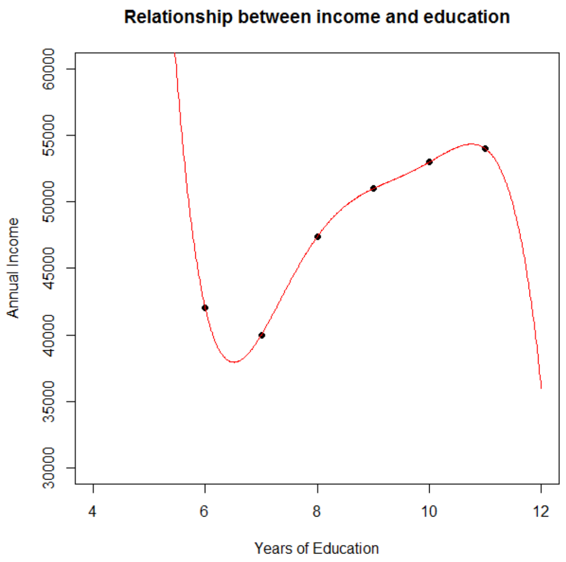
```
sm.ols('income ~ education', data=sl).model.fit().summary().rsquared
0.1066281
sm.ols('income ~ education + youngkids', data=sl).model.fit().summary().rsquared
0.1104819
sm.ols('income ~ education + youngkids + age', data=sl).model.fit().summary().rsquared
0.1214696
sm.ols('income ~ education + youngkids + age + random_noise', data=sl).summary().rsquared
0.1291423
```

Generalization and Overfitting

- Overfitting: When a model fits the training set very well (e.g., high R^2) but fails to generalize to new data



$$wages_i = \alpha + \beta * educ_i + error_i$$



$$wages_i = \alpha + \beta_1 * educ_i + \ldots + \beta_5 * educ_i^5 + error_i$$

Generalization and Overfitting

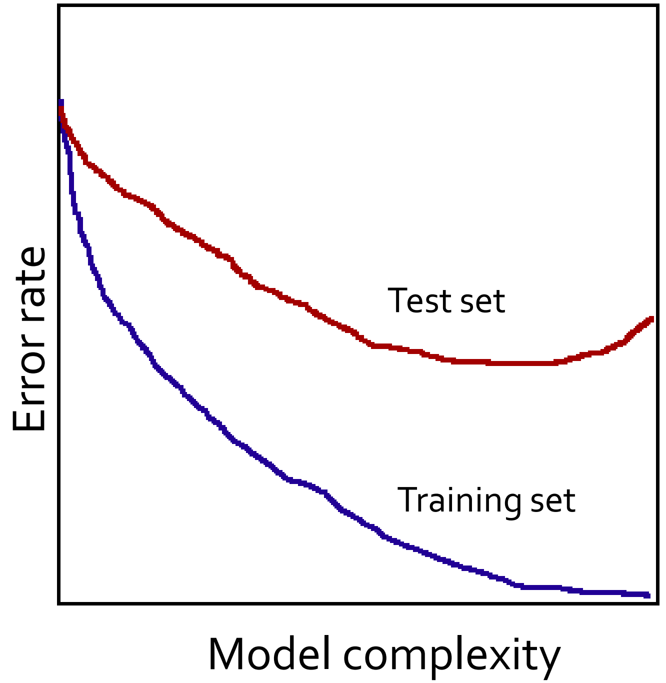
- R^2 does not tell you which model is "right"
- Our R^2 increases as we
  - add complexity
  - iterate on features
  - try different models
  - use different datasets
- **Good fit is not the same as a good model!**

Outline

- Design of Machine Learning experiments
  - Motivation
  - **Training, testing, validation, cross-validation and bootstrap**
  - Measuring performance
  - Choosing appropriate baselines
  - Error Analysis

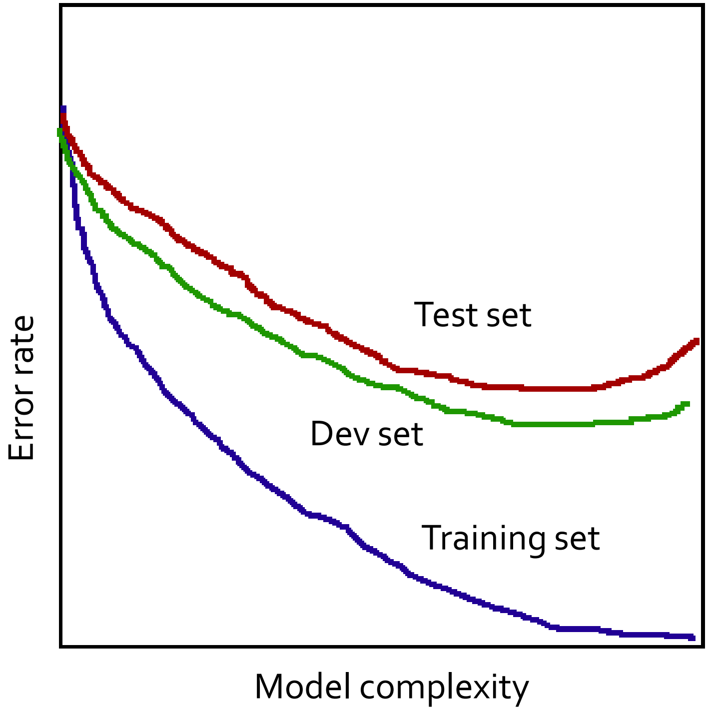
Training and Testing

- ML experiments typically separate data into a **training set** and a **testing set**
  - Model is fit on training set
  - Performance is measured on test set



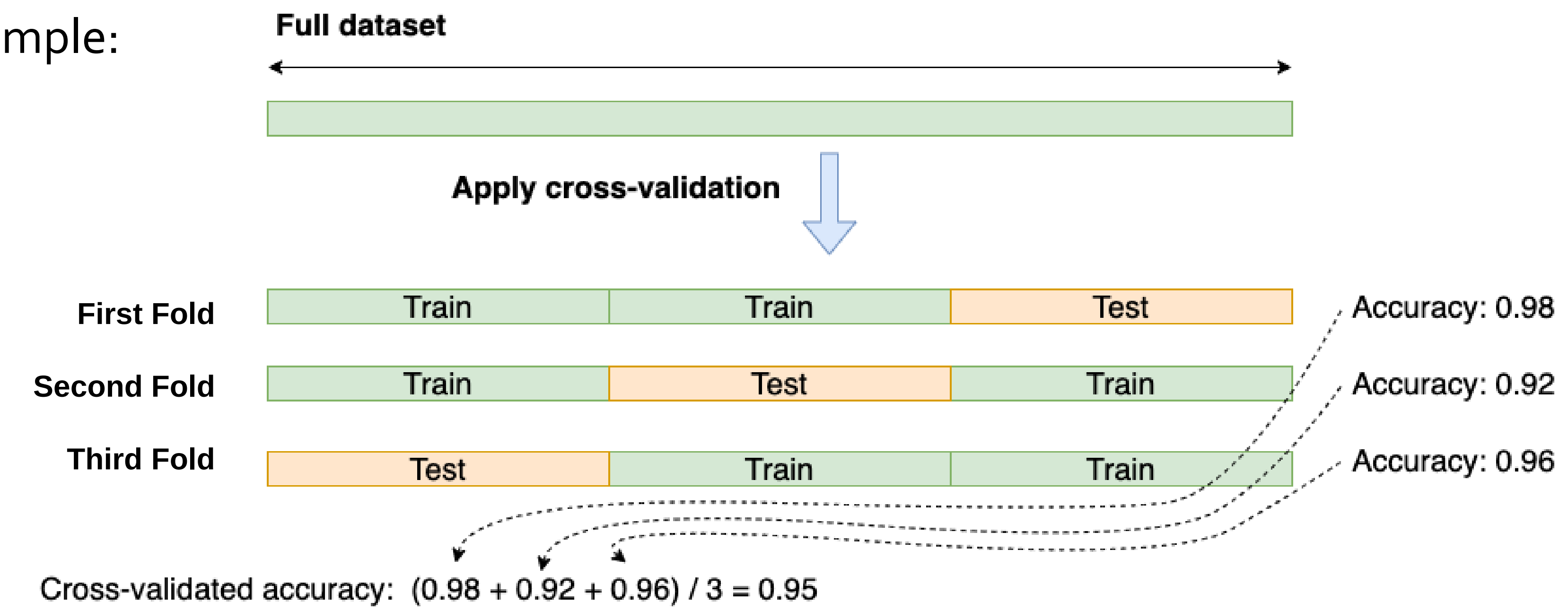
Validation (development) data

- Splitting into training + testing is often not enough
  - Each time you look at the test set, you introduce bias (in yourself!)
  - Hyperparameters must be chosen
  - Model selection, feature selection, etc.
- Validation/development data
  - A third split of the data
  - Used as a pseudo-test set for hyperparameter tuning
- Measure and report performance on test set
  - Don't do this until the very (very!) end



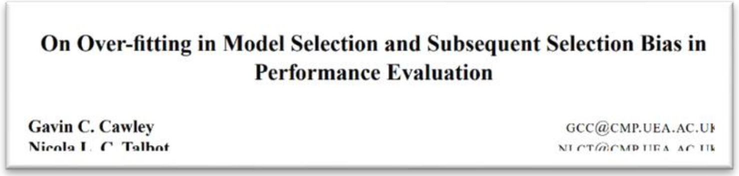
Cross-Validation

- *k*-fold cross-validation
  - When data are limited, cross-validation uses data more efficiently
  - Idea: Randomly partition data into *k* "folds", use each of *k* folds as validation set once
    - Each fold produces a measure of performance on 1/k of the data
    - Average performance across *k* test runs – this is your "CV test performance"
  - 3-fold cross-validation example:



Cross-Validation: A word of caution

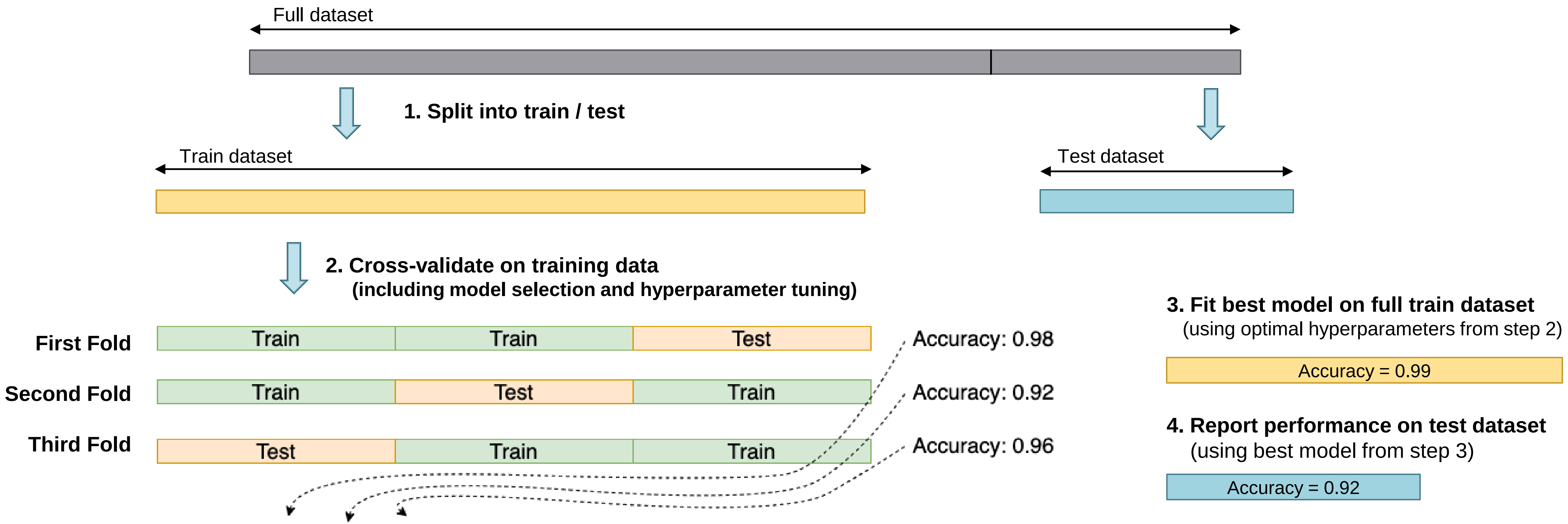
- Cross-validation: Fine if you know your model in advance
    - E.g., if you have a linear model with a few parameters: $Y_i = \alpha + \beta X + \epsilon_i$
  - However, often you want to learn more than just parameter values
    - E.g., hyperparameter tuning: How many predictors should our model have?
      - $wages_i = \alpha + \beta_1 education_i + \epsilon_i$
      - $wages_i = \alpha + \beta_1 education_i + \beta_2 age_i + +\epsilon_i$
      - $wages_i = \alpha + \beta_1 education_i + \beta_2 age_i + \beta_3 zipcode_i + \epsilon_i$
      - In these examples, α and all the β's are **parameters**
      - The decision about how many β's to include in the model is a **hyperparameter**
      - E.g., model selection: regression vs. random forest vs. naïve bayes
      - The choice of the model (e.g., OLS vs. random forests) is also a sort of hyperparameter
- With simple cross-validation, you still risk of overfitting!
    - If you search enough model/hyperparameter combinations, eventually one might look really accurate – even if just by random chance
    - Intuitively, echoes multiple testing concerns


On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation

Gavin C. Cawley GCC@CMP.UEA.AC.UK

Cross-Validation: Preserving your test data

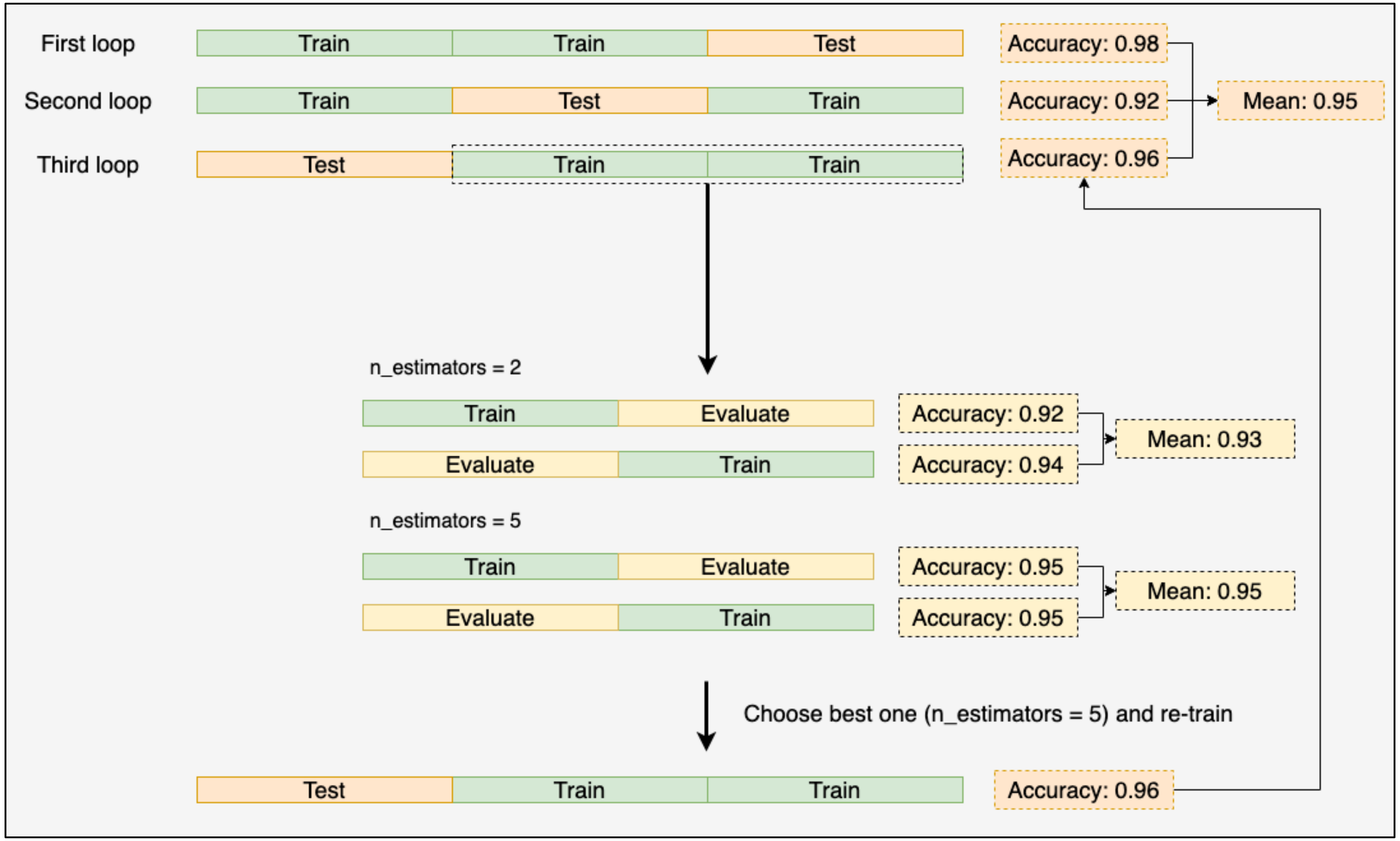
- A simple approach to ensure accurate estimates of out-of-sample performance: preserve a pure test set



Cross-validated accuracy: (0.98 + 0.92 + 0.96) / 3 = 0.95

Better Yet: Nested Cross-Validation

- Idea: Nest hyper-parameter tuning as inner CV loop within outer CV loop



Outer loop: structured as before, will be used to evaluate performance (not select hyperparameters)

Inner loop selects optimal hyperparameters
In this example we are testing random forests with 2 vs. 5 trees (n_estimators)

Given optimal hyperparameters from inner loop, retrain on outer fold, measure "test" performance
In this example, 5 n_estimators was selected

Stratification, Boostrapping

- Basic cross-Validation
  - **Partitions** data into *k* folds, so each instance is used exactly one (either as train or test)
- Stratified cross-validation
  - Ensures some variable(s), often the outcome, are balanced across folds
- Bootstrap
  - Instead of partitioning the data, boostrapping samples *with replacement*
  - Unsampled data become the validation set
  - Training data: 63.2% unique; validation: 36.8%
    - Probability that an instance is not picked = 1-(1/n)
    - $\left(1-\frac{1}{n}\right)^n \cong e^{-1} = 0.368$

Outline

- Design of Machine Learning experiments
  - Motivation
  - Training, testing, validation, cross-validation and bootstrap
  - **Measuring performance**
  - Choosing appropriate baselines
  - Error Analysis