# Effective Programming Practices for Economists

# Data Analysis in Python

## Introduction to Machine Learning

Janoś Gabler, Hans-Martin von Gaudecker, and Tim Mensinger

# The Fundamental difference

**Econometrics**

- Estimate **fundamentally unobservable** parameters and test hypotheses about them
- Cannot test how well it worked
- Focus on justifying assumptions

**(Supervised) Machine learning**

- Predict observable things
- Can check how well it works
- Focus on experimentation, evaluation and finding out what works

# Some implications

- Even though it is tempting, you cannot interpret parameters
- Can be creative in combining simple models into complex ones
- Rapid progress and development of new models
- Programming skills matter more

# Terminology

| Machine Learning | Econometrics |
| --- | --- |
| feature, attribute | x-variable, independent variable |
| target | y-variable, dependent variable |
| model, algorithm | model |
| training procedure | estimation method |
| fitting | running an estimation |
| classification | regression with discrete dependent variable |
| logistic regression | binary or multivariate logit |
| instance | observation |
| classes | possible values of a discrete dependent variables |

# Supervised vs unsupervised learning

- **Supervised learning**
  - Training data contains labeled examples of the task to solve
  - Model generalizes this to unseen data
  - Example: Regression, classification
- **Unsupervised learning**
  - Training with label free data
  - Model finds patterns in data
  - Example: Clustering, dimensionality reduction

# Overfitting

- Estimating large models on small datasets can lead to overfitting
- Overfitting means:
  - Model can explain the concrete dataset well
  - Model would not work on any other dataset
  - Same reason why we need adjusted $R^2$ in econometrics
  - Need to make sure our model evaluation accounts for overfitting!
- Example: Estimate person fixed effects in short panel

# The bias-variance trade-off

- Econometrics: Model is correctly specified, want consistency and unbiasedness
- Very simple models, e.g. just an intercept and a couple of regressors
  - Large bias, low variance, no overfitting
- Very large models, e.g. including squares, interactions, ...
  - Small bias, high variance, danger of overfitting
- ML: Model is a simplification and some amount of bias is ok
- Most ML models have one or more parameters that govern the bias variance trade-off

# Holdout samples

- Split data into training and test dataset
- Fitting and experimentation is only done on training data
- Evaluation is only done on test data
  - Overfitting on training data cannot influence the evaluation
  - Need to avoid leaking any information from test data into model training!
- Typical sizes:
  - 70 to 80 percent for training
  - Rest for validation

# Hyperparameters

- No parameters of the model itself
- Instead: Control behavior of the model algorithm
- E.g., balance bias and variance