DATASCI 347: Machine Learning

Lecture 0: Course Logistics and Introduction

Ruoxuan Xiong

EMORY

Who am I

- Assistant Professor, Data & Decision Sciences
- PhD (Stanford Management Science and Engineering), Postdoc (Stanford Graduate School of Business)
- Research: causal inference, experimental design, machine learning, and econometrics
- Applications: digital platforms, healthcare, and finance

Lecture plan

- Course structure
  - What Is Machine Learning?
  - Expectations
  - Course logistics
  - Evaluation
- Course outline

What Is Machine Learning?

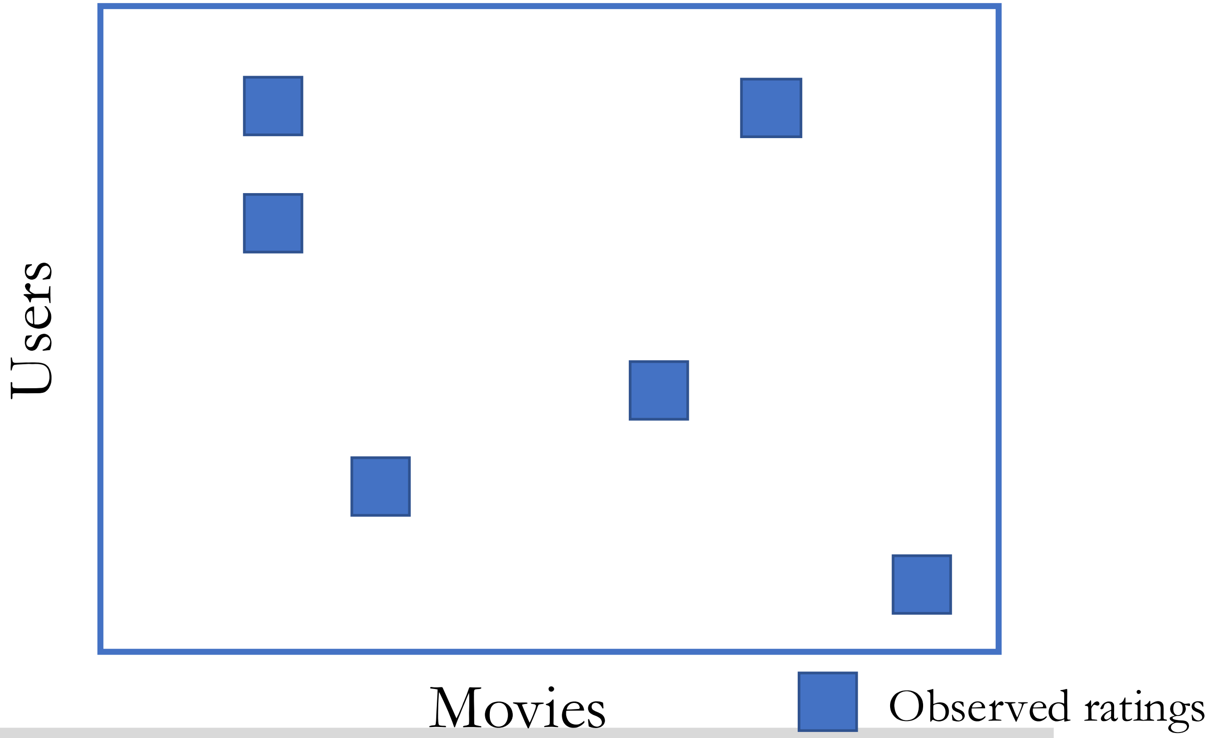
- Algorithms that **learn from data**
- Goal: **prediction or decision-making**
- Core idea: **generalize beyond observed data**
- Examples: recommender systems, search, speech, self-driving cars

Recommender system: The Netflix challenge

- Netflix provided ~100M ratings that ~500K users gave to ~18K movies
- Most ratings are missing
  - $500\text{K} \times 18\text{K} = 9{,}000\text{M} \gg 100\text{M}$
- Goal: *predict missing ratings*
  - Learn users' preferences
  - Recommend movies to users
- The team whose model with highest accuracy was awarded $1 million

Recommender system: The Netflix challenge

- In this course, you will learn
  - How to **build predictive models**
  - How to **evaluate accuracy**
  - How model structure reflects data structure



Expectations

- **Lectures**:
  - Focus: **intuition + when to use what** which method
  - Some probability & statistics
  - Lots of examples

- **Homework**:
  - Mostly **Python coding** to practice how to use various methods
  - Some conceptual/theoretical questions
  - Group-based problem solving

Expectations

- **Course project**:
  - Apply ML to **real data**
  - Learn GitHub
  - Exposure to **research frontiers**

Course logistics

- Instructor: Ruoxuan Xiong
- Class time: Mon/Wed 11:30 – 12:45 pm, PAIS 225
- Office hours: Wed 3:00 – 4:00 pm in my office, PAIS 581

- Details in the syllabus on Canvas

- Course website: http://www.ruoxuanxiong.com/DATASCI347/DATASCI347.html

Evaluation

- Homework 30%
- Take-home exam: 30%
- Course project presentation (proposal and final presentation): 15%
- Project GitHub submission: 20%
- Participation: 5%

Homework

- 3 group homework assignments
  - Groups of **up to 4**; sign up via the Google Sheet by **Wed 1/21**
  - **Same group** for all homework assignments

- Late days (group policy)
  - Each group has **3 total late days** for the semester
  - You may use **at most 2 late days** on any single homework

Important dates

- **Homework**
  - Problem set 1: out 1/21, due 2/11
  - Problem set 2: out 2/11, due 3/4
  - Problem set 3: out 3/4, due 4/1

- **Take-home exam**
  - Out Wed 4/8 00:00 am, due Sun 4/12 11:59 pm
  - No class on 4/8
  - Choose any 24-hour window during the exam period

Course project

- **Instructions:** see the [Google doc]

- Resources we provide
  - Data repos: UCI ML repos, Kaggle, OpenML
  - Domains: image, natural language, network and graph
  - Example research venues: ICML, NeurIPS, ICLR, ACL, EMNLP

- Choose one project path
  - **Applied project:** pick a dataset and apply methods from the course
  - **Replication/extension:** replicate a paper and explore an extension

- **Teams:** same group as homework

Important dates

- **Project proposal presentation**: Wed 3/18
  - **5 minutes** per group

- **Final project presentation**: Wed 4/22 and Mon 4/27
  - **10 minutes** per group (motivation, setup, results)
  - **Before presenting:** create a **public GitHub repo** with code + current findings
  - Peer feedback counts toward **participation**

- **Final project deadline**: Wed 5/6
  - Finalize the GitHub repo and documentation

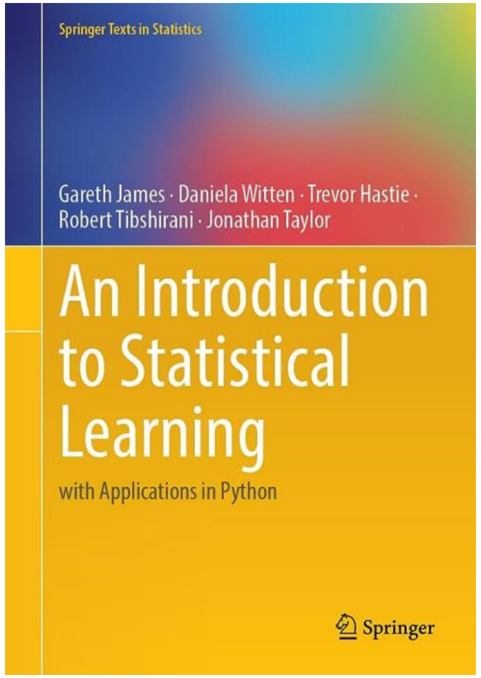
Participation

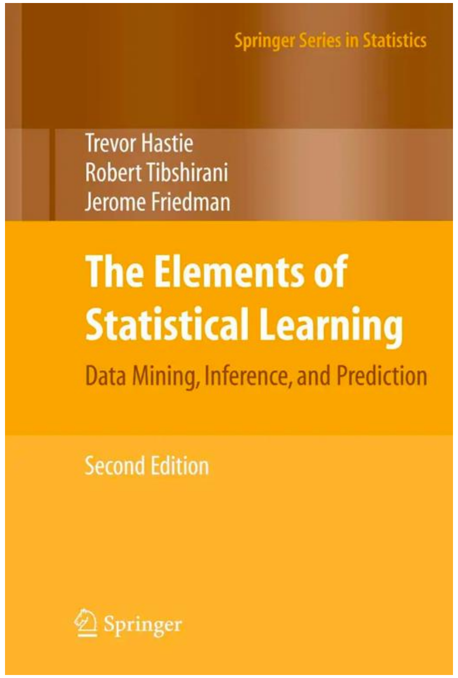
- You can earn participation credit in **either** of these ways:

1. **In class:** attend and **ask/answer questions**
2. **Online:** submit questions or course feedback via the [Google form]
   - We'll spend the **first few minutes of each lecture** reviewing selected questions from the form

Notes and textbooks

- Lecture notes available on course website and Canvas before lecture
- Suggested textbooks (but not required):
  - James, Witten, Hastie, and Tibshirani, *An introduction to statistical learning*
  - Hastie, Tibshirani, and Friedman, *The elements of statistical learning*





Lecture plan

- Course structure
  - What is this class about?
  - Expectations
  - Course logistics
  - Evaluation
- Course outline

Supervised vs unsupervised learning

- **Supervised learning** (main focus of this course)
  - **Data**: $(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)$
    - X_i: features (predictors)
    - Y_i: label/response
  - **Goal:** learn a function f so that $\hat{Y} = f(X)$
  - **Examples:** regression, classification (e.g., linear/logistic regression)

- **Unsupervised learning**
  - **Data**: $X_1, X_2, \cdots, X_n$
  - **Goal:** discover structure/patterns in X (no labels)
  - **Examples:** clustering, dimensionality reduction (PCA)

Supervised machine learning

- **Example**: Predicting housing prices (Boston suburbs)

- **Training data**: given a training dataset that contains n samples

$$(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)$$

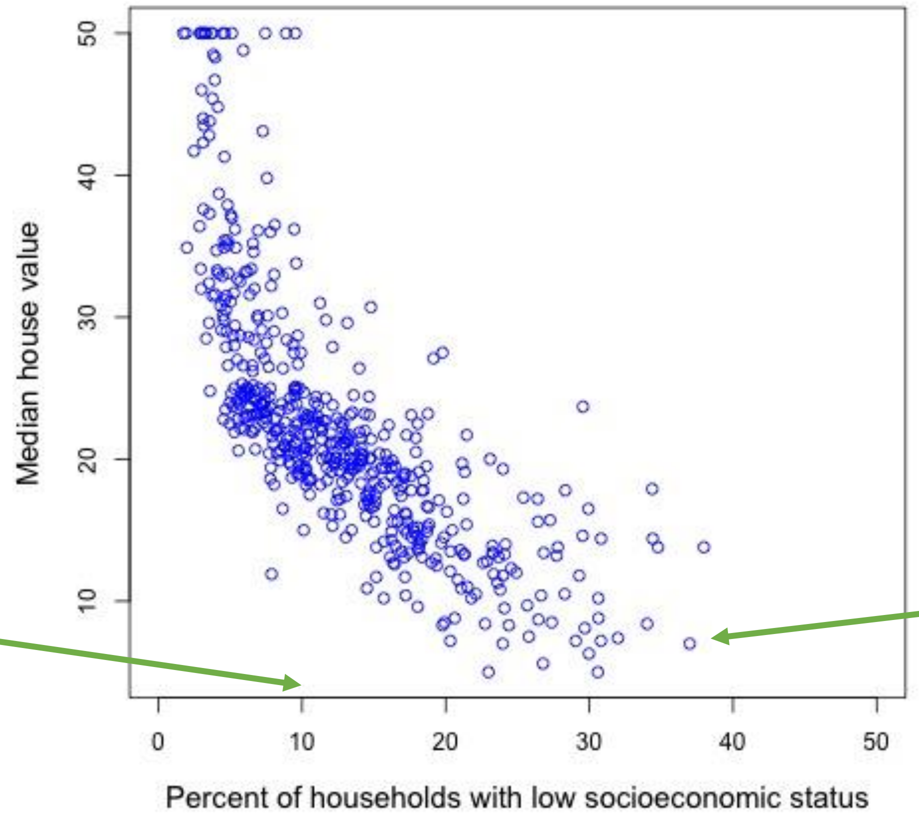
  - X_i: feature vector (predictors)
  - Y_i: target (house value)

- **Task**: If a neighborhood has x % of households with low socioeconomic status, what is the predicted median house value?

Prediction of housing values in suburbs of Boston

- **Goal:** predict **median house value** using a single feature
- Feature X: % of households with low socioeconomic status (**lstat**)

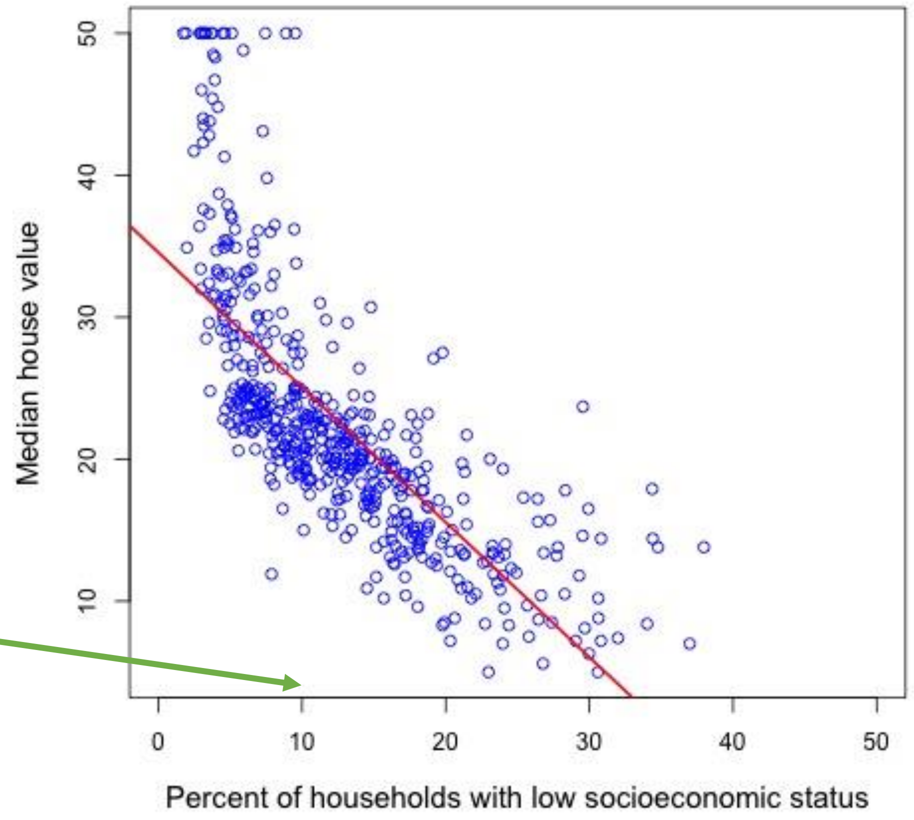
$X = 10$
$Y = ?$



i-th sample
(X_i, Y_i)

Prediction of housing values in suburbs of Boston

- **Goal:** predict **median house value** using a single feature
- Feature X: % of households with low socioeconomic status (**lstat**)



$X = 10$
$Y = ?$

Fit a linear model to the data

Prediction of housing values with many features

- Real problems use **multiple features** (predictors), e.g.
  - % of households with low socioeconomic status (lstat)
  - average number of rooms per house (rm)
  - average age of houses (age)
  - per capita crime rate (crim)
  - …
- **Predicting housing prices**: learn a model f so that $\widehat{Y}_i = f(X_i)$

features/input

$$X_i = \begin{bmatrix} X_{i1} \\ X_{i2} \\ X_{i3} \\ \vdots \\ X_{ik} \end{bmatrix} \begin{array}{l} \text{-- lstat} \\ \text{-- rm} \\ \text{-- age} \\ \text{-- crim} \\ \vdots \end{array} \longrightarrow Y_i \quad \text{--- median house value}$$

label/output

Which model can we use?

- Start simple: **multiple linear regression**
  - $Y = \beta_0 + \beta_1 \cdot \text{lstat} + \beta_2 \cdot \text{rm} + \beta_3 \cdot \text{age} + \beta_4 \cdot \text{crim} + \cdots + \varepsilon$

- **Not all features are useful**: we may need
  - **Feature selection** (Lasso) or **shrinkage** (Ridge)
  - **Dimension reduction** (principal components regression)

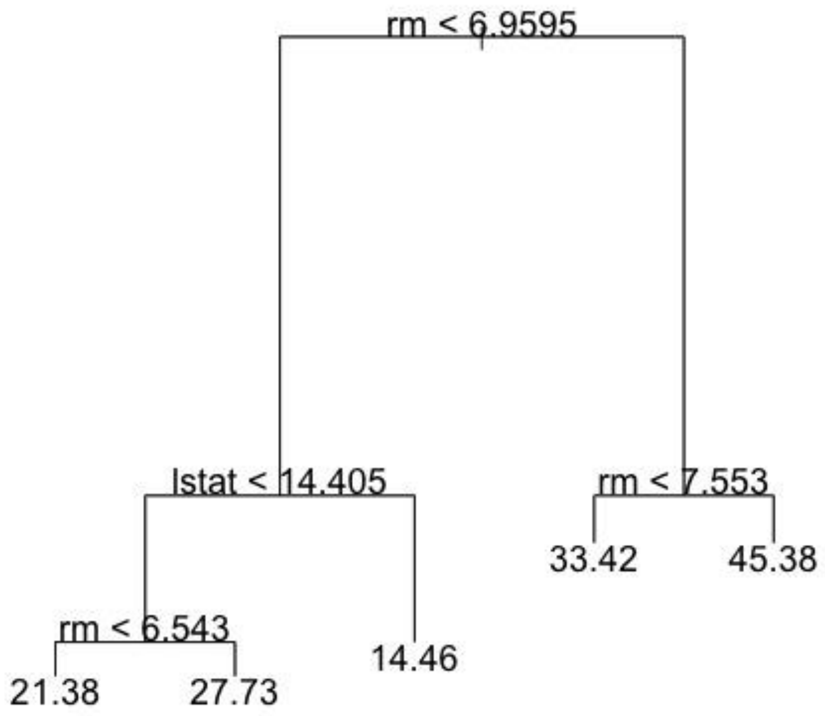
- If the relationship is more complex, we can use **nonlinear models**

Tree-based methods

- **Tree-based models are nonlinear**: they split the feature space into regions

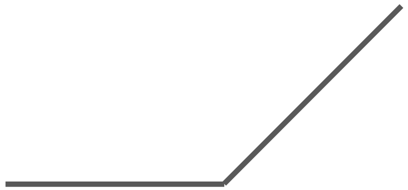
- **Decision tree**
  - A sequence of **if–then rules** (e.g., split on rm, then lstat, …)

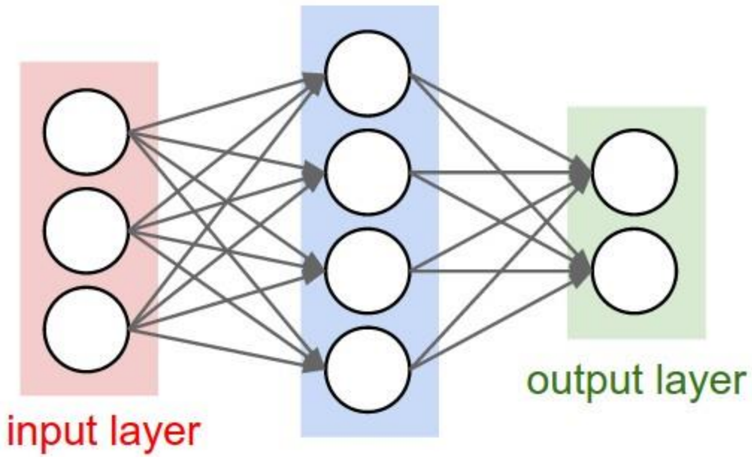
- **Random forest**
  - many trees averaged together (usually better accuracy and stability)



Neural networks

- **Feedforward neural network**
  - Architecture: Input layer → hidden layers → output layer
  - **Key idea:** nonlinear activations make the model flexible
  - Example: $\mathrm{ReLU}(x) = \max(x, 0)$





- **One hidden-layer network**
  - Linear step: $z_1 = W_1 x + b_1$ (maps the input to a feature representation)
  - Nonlinearity: $a_1 = \max(z_1, 0)$
  - Output: $z_2 = W_2 a_1 + b_2$ (then map to $\hat{y}$ depending on task)

Which model should we choose?

- **Which model should we choose?**
  - **There is no single "best" model** — choice depends on goals and data
  - We'll discuss key **tradeoffs** (e.g., **bias–variance**, interpretability vs. flexibility)

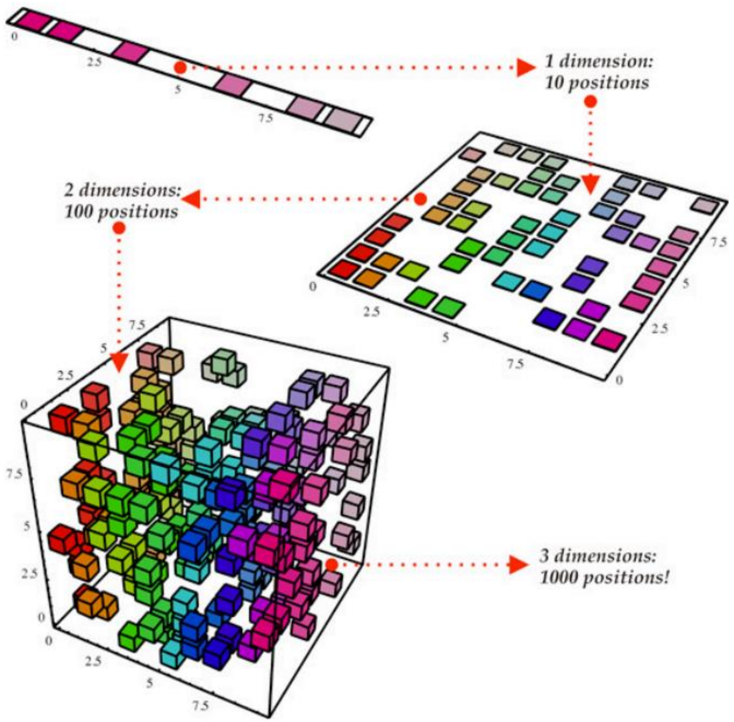
- **Two tools you'll use throughout the course**
  - **Cross-validation:** estimate out-of-sample performance (model selection)
  - **Bootstrap:** quantify uncertainty (e.g., SEs for $\hat{\beta}$ or predictions $\hat{Y}$)

- *Both are **resampling methods**: repeatedly drawing samples* to assess performance or uncertainty

Unsupervised machine learning

- **Goal:** learn structure/representations from **features only** (no Y)

- **Example 1 (dimension reduction):** map 3 features (lstat, lm, age) to a single feature $z \in \mathbb{R}$
  - so z preserves important information in the original data



- **Popular approaches**: **Principal component analysis, autoencoder**

Unsupervised machine learning

- **Example 2 (clustering):** group people using **height** and **weight**

- **Goal:** people in the same group are **more similar** to each other than to those in other groups

- **Method: Clustering** (e.g., *k*-means)