QTM 347: Machine Learning

Lecture 0: Course Logistics and Introduction

Ruoxuan Xiong

Lecture plan

- Course structure
  - What is this class about?
  - Expectations
  - Course logistics
  - Evaluation
  - Connection to other courses in QTM
- Course outline

What is this course about?

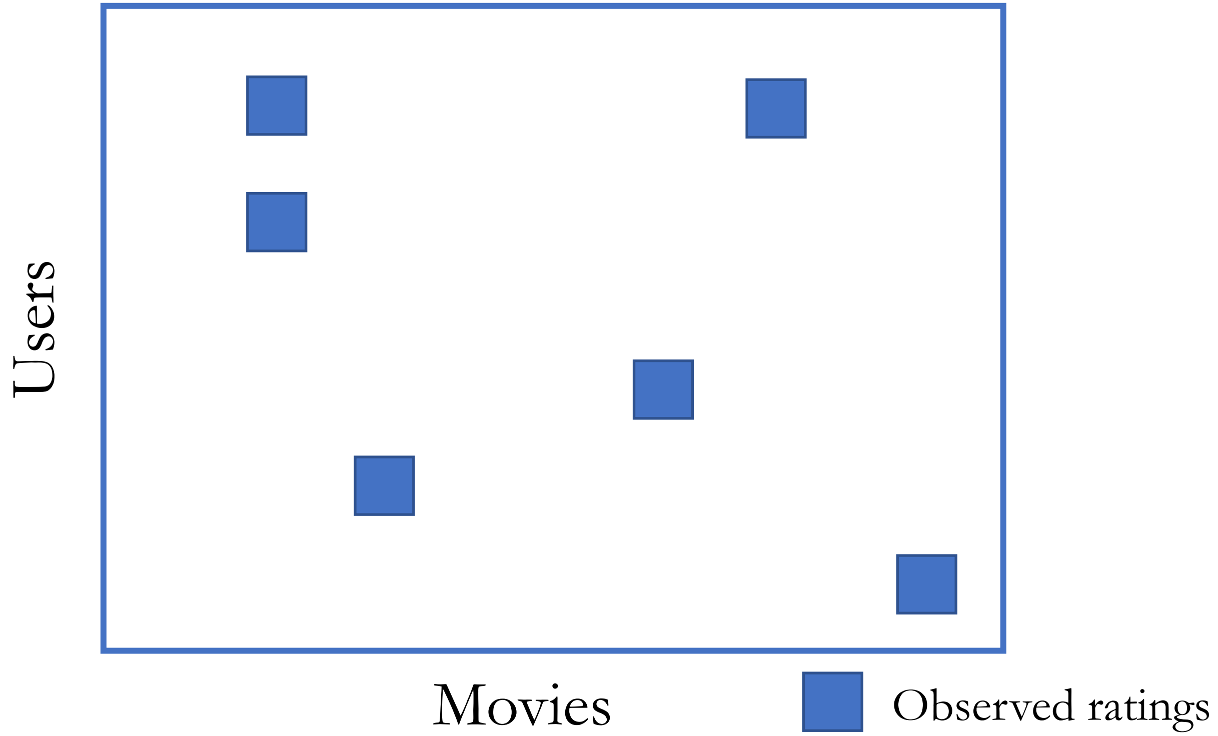
- The study of *computer algorithms* that can *learn from* and *make predictions* or *decisions* based on *data*.
  - An example: Recommender system
  - More examples: Effective web search, speech recognition, self-driving car
  - Applications in social sciences and business, including economics, marketing, finance, …

Recommender system: The Netflix challenge

- Netflix popularized prediction challenges by organizing an open, blind contest to improve its recommendation system.
- Netflix provided ~100M ratings that ~500K users gave to ~18K movies

Recommender system: The Netflix challenge

- Netflix popularized prediction challenges by organizing an open, blind contest to improve its recommendation system.
- Netflix provided ~100M ratings that ~500K users gave to ~18K movies
- Most ratings are missing
  - $500K \times 18K = 9{,}000M \gg 100M$
- Goal is to build a machine learning model to *predict missing ratings*
  - Learn users' preferences
  - Recommend movies to users



Recommender system: The Netflix challenge

- Netflix popularized prediction challenges by organizing an open, blind contest to improve its recommendation system.
- The team whose model with highest accuracy was awarded **$1 million**
- In this course, you will learn
  - How to evaluate model accuracy
  - How does the model look like

Expectations

- You will learn many machine learning methods from this class

- **Lectures**:
  - *Goal*: Understand how these methods work and when to use which method
    - There will be some probability and statistics in this class
    - I will explain concepts and methods with examples

- **Homework**:
  - *Goal*: Practice how to use different methods
    - Most questions are coding questions based on Python
    - There will be some conceptual and theoretical questions

Expectations

- **Course project**:
  - *Goal*: Gain some project experience in machine learning & data mining
    - Gain some hands-on experience in applying machine learning to a real-world problem
    - Learn some frontiers in machine learning
    - Learn how to use GitHub

Course logistics

- Instructor: Ruoxuan Xiong
- Time: Mondays/Wednesdays 11:30 – 12:45 pm in White Hall – 205
- Office hours: Mondays 3:00 – 4:00 pm in my office, 581 PAIS building

- Details in the syllabus on Canvas

- Course website: http://www.ruoxuanxiong.com/QTM347/QTM347.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 in total
  - Group size of up to four. Sign up in the Google spreadsheet by Wednesday 1/22
  - Same group for all homework assignments

- You have a total of three free late days for all homework assignments as a group. You can use at most two late days for one homework assignment

Important dates

- **Homework**
  - Problem set 1: out 1/22, due 2/12
  - Problem set 2: out 2/12, due 3/5
  - Problem set 3: out 3/5, due 4/2

- **Take-home exam**
  - Out Wednesday 4/9 00:00 am, due Saturday 4/12 11:59 pm (no class on 4/9)
  - You can choose any 24 hours in between to complete

Course project

- See instructions in [Google doc]
- We provide a list of *datasets* and *paper venues*
  - Popular data repos, such as UCI ML repos, Kaggle, OpenML
  - Popular image, natural language, network and graph data
  - Publication venues of ML and data mining research (ICML, NeurIPS, ICLR, KDD, etc)
- You have **two options**
  - Pick a dataset, and apply the methods learned this semester to analyze this dataset
  - Replicate a research paper and explore the possible extensions
- The course project is done in the same group as the homework

Important dates

- **Project proposal presentation**: 3/17
  - Five-minute presentation for each group
- **Final project presentation**: 4/23 and 4/28
  - Ten-minute presentation: includes motivation, setup, and results of the project
  - Before the full project presentation, set up a publicly available GitHub repo with detailed documentation about the code and what findings you currently have
  - When each group presents, other groups provide feedback, which will be counted toward the class participation
- **Final project deadline**: 5/7
  - Refine the GitHub repository and the accompanying documentation

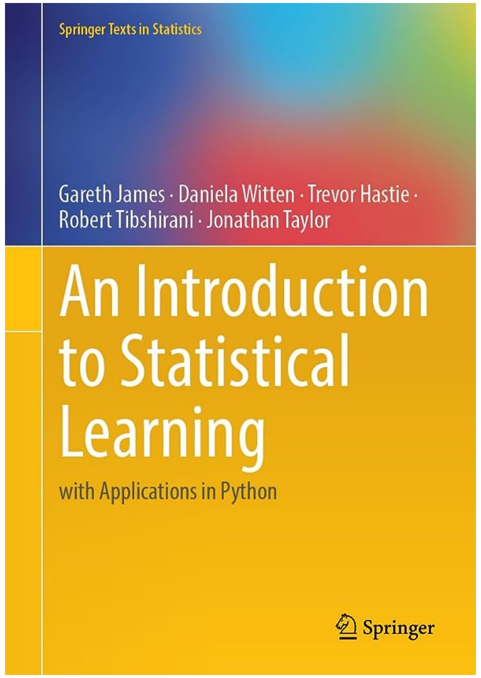
Participation

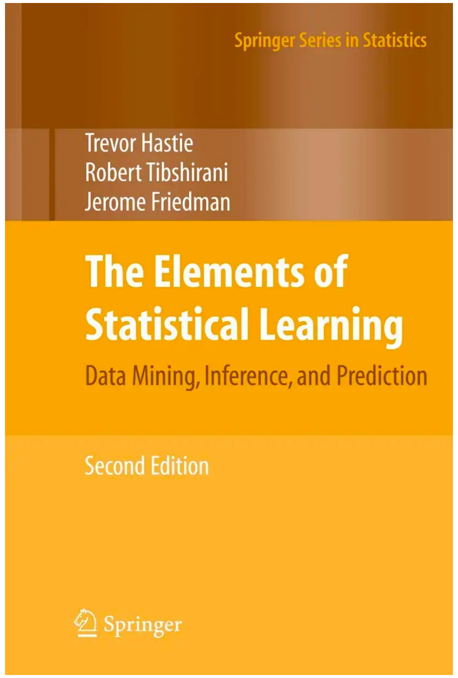
- You can fulfill your class participation through either of the following two ways:

1. Attend the class, ask and answer questions
2. Submit questions for lecture material or feedback for this course through the [Google form]
   - At the beginning of each lecture, there will be a few minutes to review the material of last lecture and answer the questions submitted through 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*





Connection to other courses in QTM

- Prerequisites: QTM 220, 285 or equivalent courses
  - Comfortable with linear algebra, probability, and statistics
  - Comfortable with Python

- Other related courses:
  - QTM 340 Approaches to Data Sci. w/Text (focus on NLP)
  - QTM 447 Machine Learning 2 (second course of the ML sequence)
  - QTM 490 Advanced Seminar: Machine Learning Theory

Lecture plan

- Course structure
  - What is this class about?
  - Expectations
  - Course logistics
  - Evaluation
  - Connection to existing courses in QTM
- Course outline

Supervised and unsupervised machine learning

- **Supervised machine learning** (main focus of this course)
  - **Data**: $(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)$
    - X_i: predictors
    - Y_i: response
  - **Task**: Fit a model that relates response to predictors
    - E.g., linear regression or logistic regression model from your regression analysis class
    - You will learn many more in this course

- **Unsupervised machine learning**
  - **Data**: $X_1, X_2, \cdots, X_n$
  - **Task**: Understand the relationships between variables/observations

Supervised machine learning

- **Illustrative example**: Prediction of housing values in suburbs of Boston

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

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

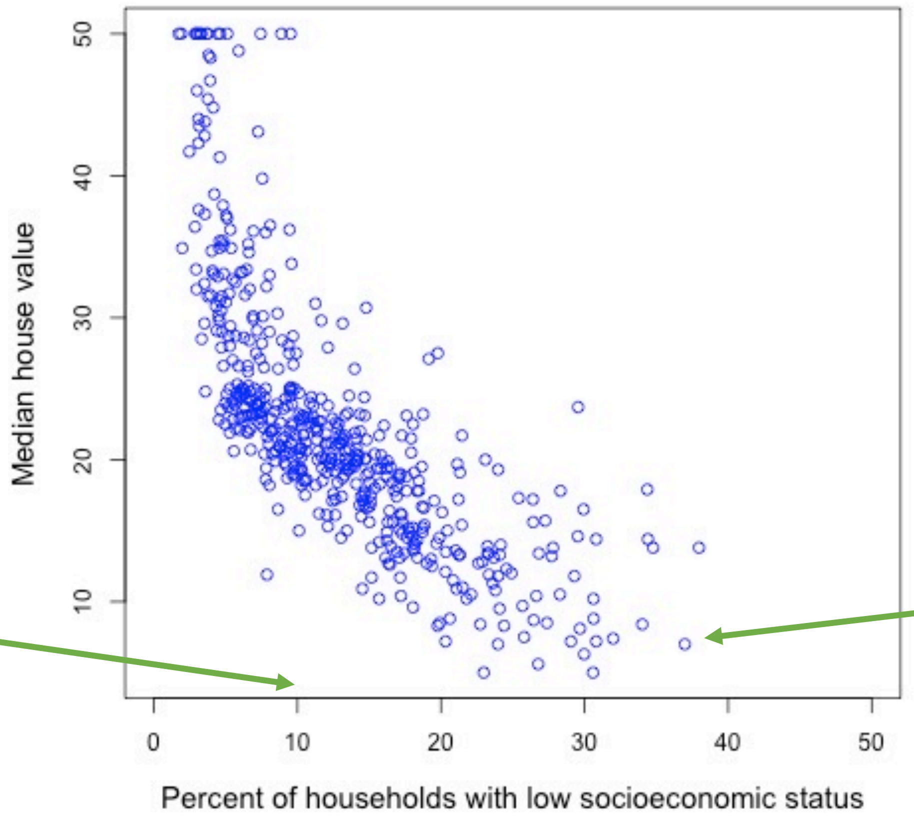
  - X_i is a feature vector
  - Y_i is a label

- **Task**: If a neighborhood has x percent of households with low socioeconomic status, predict the median value of this neighborhood?

Prediction of housing values in suburbs of Boston

- **Predicting housing prices**: A simple feature for predicting the housing price is the median income of the household

$X = 10$
$Y = ?$

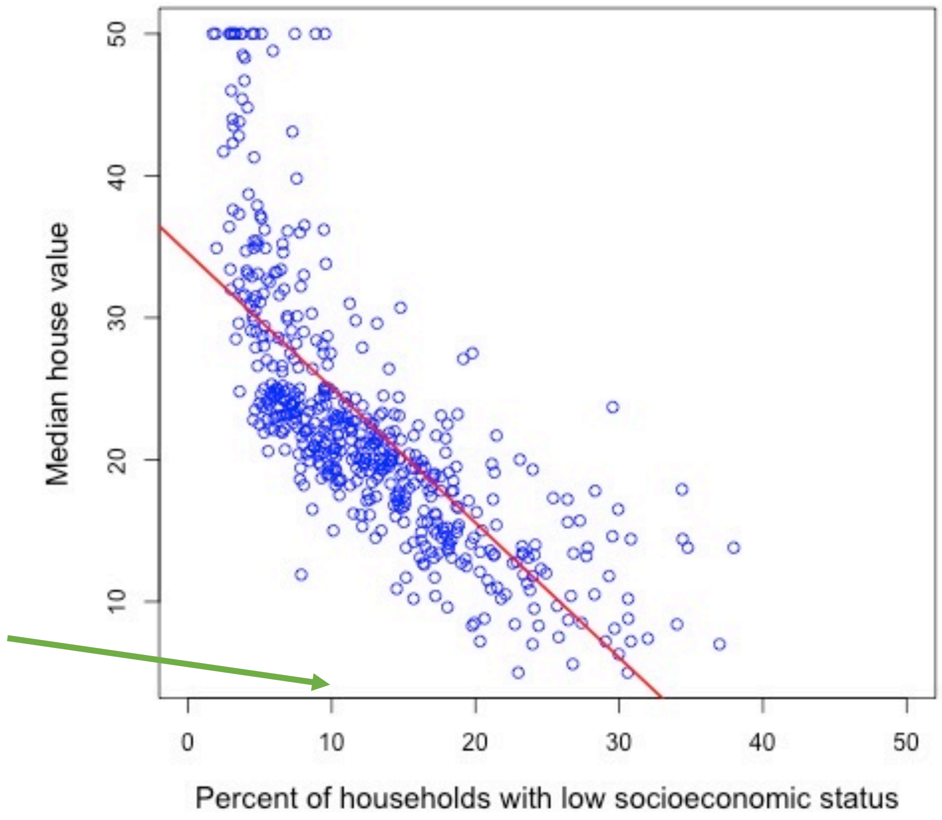


i-th sample
(X_i, Y_i)

Prediction of housing values in suburbs of Boston

- **Predicting housing prices**: A simple feature for predicting the housing price is the median income of the household

$X = 10$
$Y = ?$



Fit a linear model to the data

Prediction of housing values with many features

- If we have **more features**
  - percent of households with low socioeconomic status (lstat)
  - average number of rooms per house (rm)
  - average age of houses (age)
  - per capita crime rate by town (crim)
  - …
- **Predicting housing prices**: Fit a model to predict median house value

features/input

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

label/output

Which model can we use?

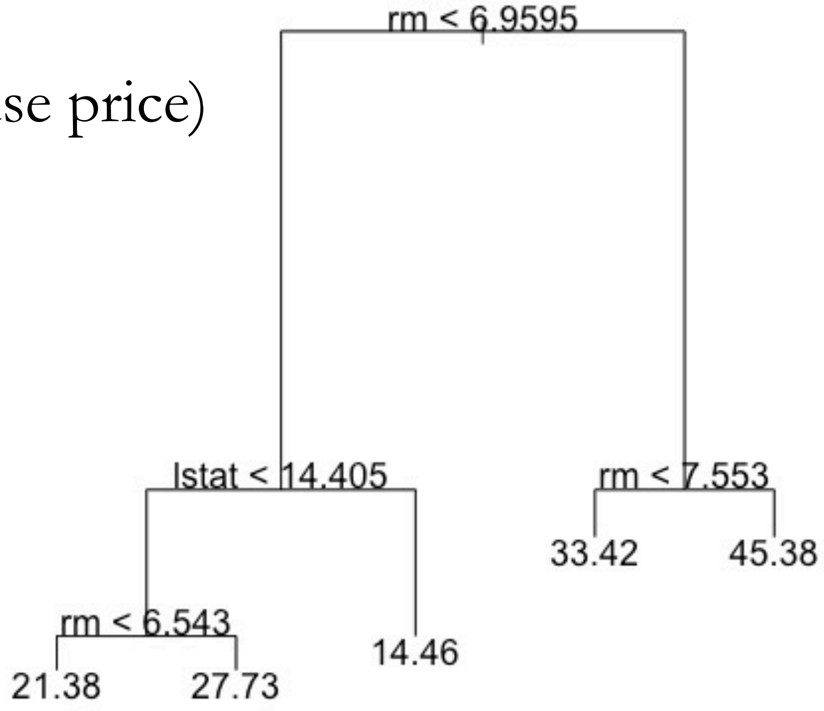
- You can use multiple linear regressions
  - $Y = \beta_0 + \beta_1 \cdot \text{lstat} + \beta_2 \cdot \text{rm} + \beta_3 \cdot \text{age} + \beta_4 \cdot \text{crim} + \cdots + \varepsilon$

- **Some features** may **not** be **useful**
  - ➢Linear model selection and regularization (Lasso, Ridge, principal component regression, …)

- Or we want to use **nonlinear model**…

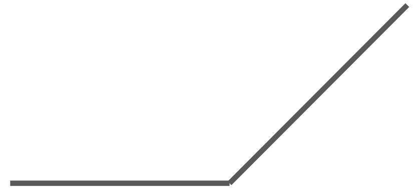
Tree-based methods

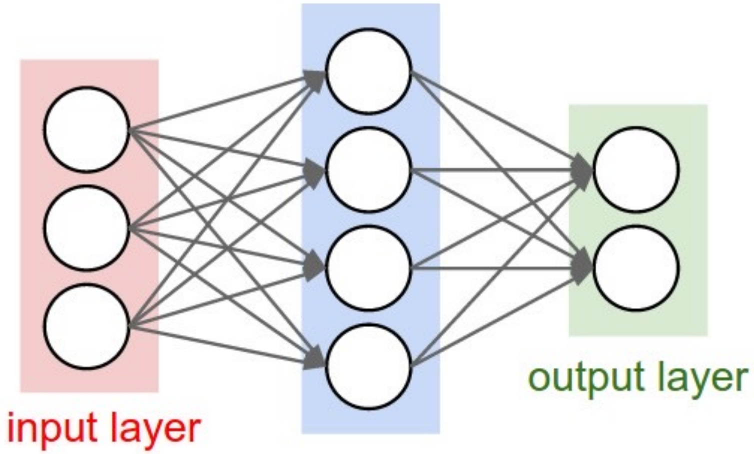
- **Tree-based methods** are nonlinear models
  - **Decision tree**
  - **Random forest** (many decision trees to predict house price)



Neural networks

- **Feedforward neural networks**
  - Input layer, hidden layer, and output layer
  - Nonlinear activation function: $\mathrm{ReLU}(x) = \max(x, 0)$





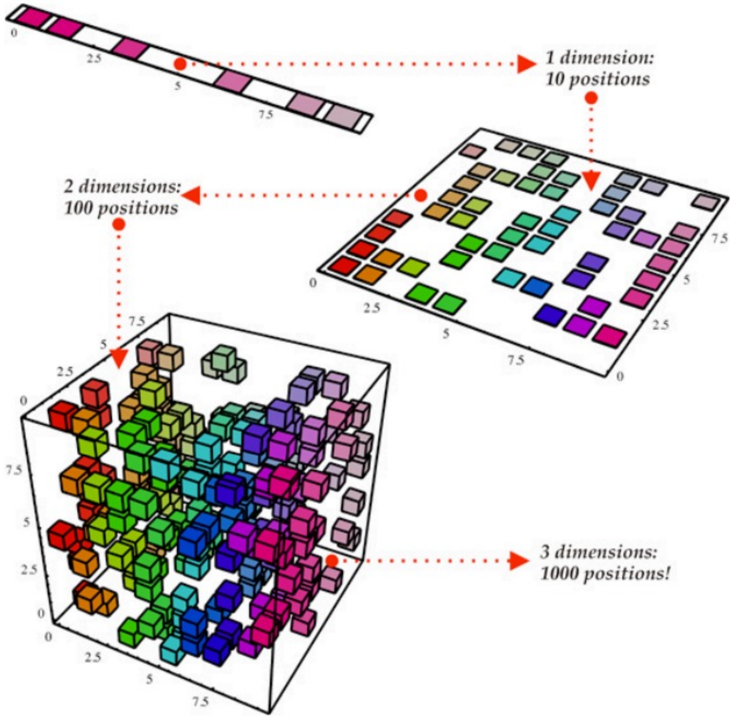
- The first layer maps the input to a feature representation ($z_1 = W_1 x + b_1$)
- The hidden layer uses nonlinear activation function ($a_1 = \max(z_1, 0)$)
- The second layer maps the representation to the output $(z_2 = W_2 a_1 + b_2)$

Which model to choose?

- We have many models. Which model should we choose?
- We will talk about the **tradeoffs** in different models (i.e., bias-variance tradeoffs)
- **Model evaluation**: **Cross-validation**
- **Quantify model uncertainty**: **Bootstrap** to estimate the standard errors (SE) (e.g., SE of estimated coefficient $\hat{\beta}_1$, or SE of predicted value $\hat{Y}_i$)
- Both cross-validation and bootstrap are based on *repeatedly drawing samples* from the original data set (*resampling* methods)

Unsupervised machine learning

- **Illustrative example**: Transform 3-d (lstat, lm, age) into 1-d feature, so that 1-d feature contains meaningful properties of the original data
- For example, reduce (lstat, lm, age) into one-dimensional feature
- **Popular approaches**: **Principal component analysis**, **autoencoder**



Unsupervised machine learning

- **Illustrative example**: Group a set of people by weight and height, such that people in the same group are more similar to each other than to those in other groups
- **Possible approach: Clustering**