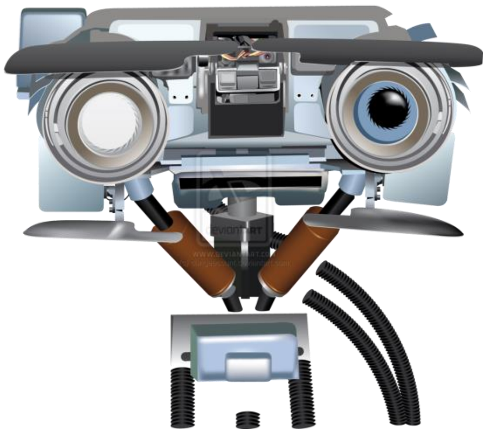


Principles of Machine Learning

CS 542 Course staff

Instructors:

Prof. Kate Saenko

https://ai.bu.edu/

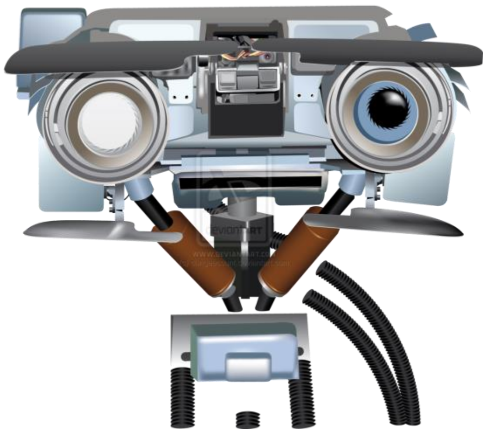
Prof. Boqing Gong

http://boqinggong.info/

Teaching fellows: Piotr Teterwak, Aoming Liu

Today

- Why do we need machine learning?
- What is machine learning?
- Supervised learning intro
- Course logistics

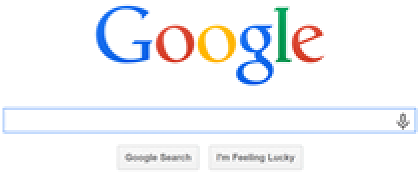


Why Do We Need Machine Learning?

Machine Learning:
Why do we need it?

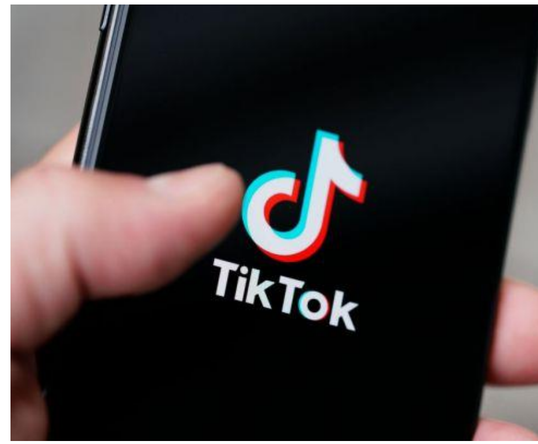
- Help automate boring, hard tasks
- Hard to program computer directly to do the task
- Instead, program a computer to **learn** from examples
- Often use “big data” examples

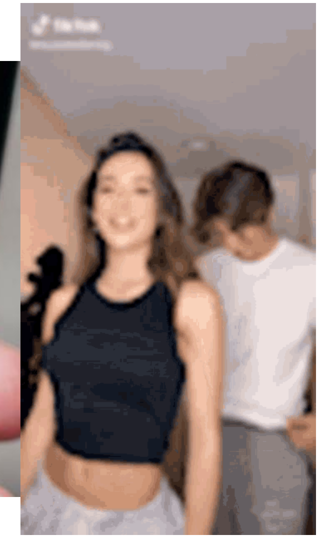
Machine Learning:
name three ways you used it today!



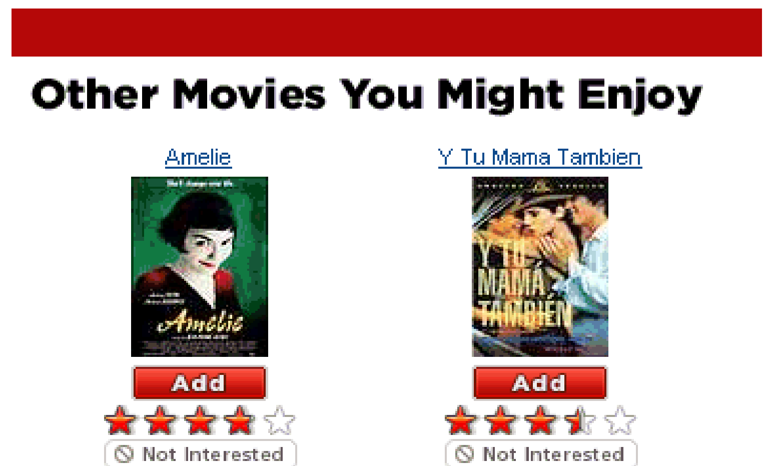
Machine Learning in Real Life:
social media & entertainment

Movie editing





AI Speakers & assistants

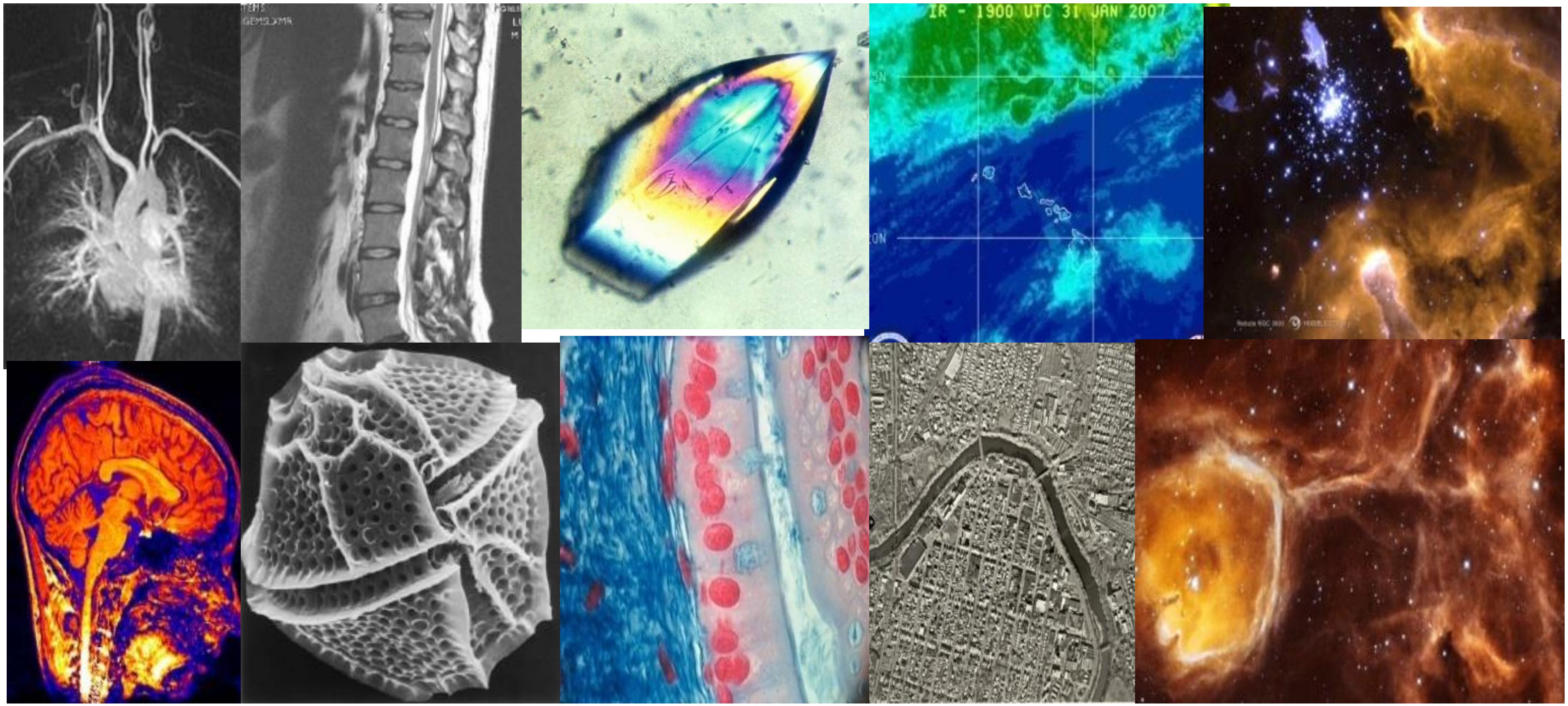


Recommendation algorithms

Machine Learning in Real Life:
Smart Cars

- Stanford/Google one of the first to develop self-driving cars
- Cars “see” using many sensors: radar, laser, cameras

Machine Learning in Real Life:
Medical and Scientific Data



Healthcare, medical image analysis, climate and geography, biology, astronomy…

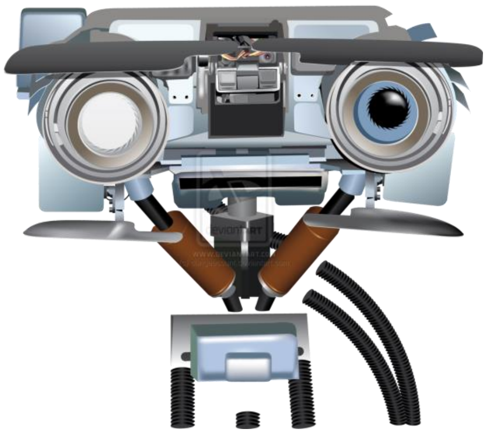
Machine Learning in Real Life:
Robotics

Vierreck, Ten Pas, Saenko, Platt. Learning a Visuomotor Controller for Real World Robotic Grasping… CORL17

Machine Learning in Real Life:

Computational Finance

machine learning models that predict stock price, cash flows, financing programs, optimize pricing for merchants, fraud detection, etc.



Introduction: What is Machine Learning?

Machine Learning

- Branch of Artificial Intelligence

“creating machine algorithms that can learn from data”

- Closely related to
  - Pattern recognition
  - Data Mining
  - Big Data
  - Deep learning

Types of learning

Supervised

Unsupervised

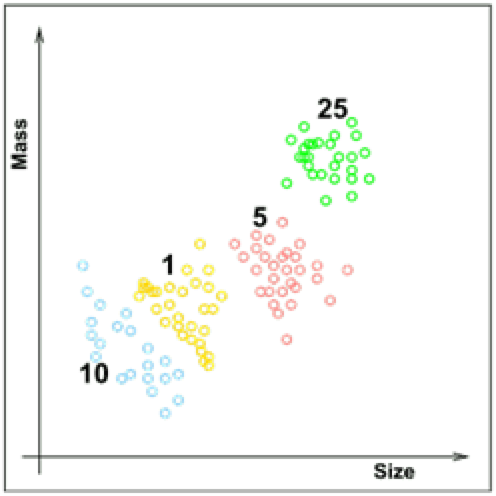
Reinforcement

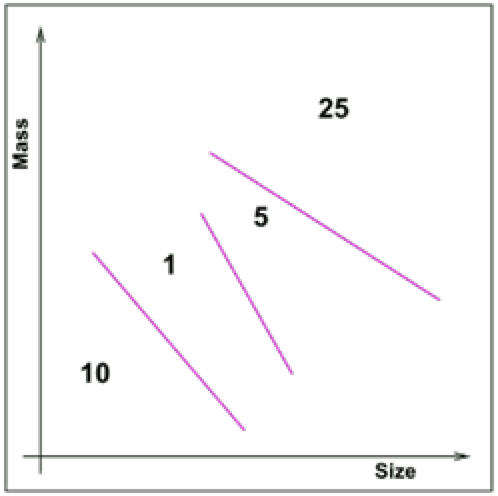
Supervised Learning

- Given a training set consisting of inputs and outputs, learn to map novel inputs to outputs
- The novel inputs are called a test set
- Outputs can be
  - Categorical (classification)
  - Continuous (regression)

Example of Supervised Learning

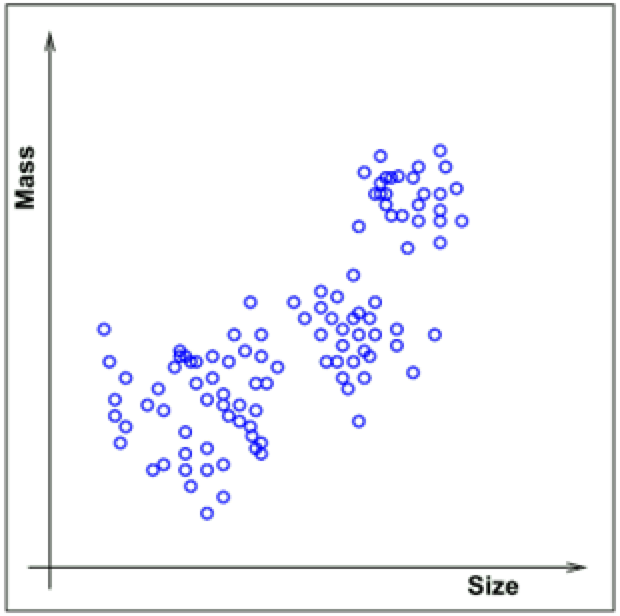
recognize coins





- Given training set consisting of coin denomination (penny, nickel, dime, quarter), mass and size
- Learn to predict denomination
- What is input? Output?

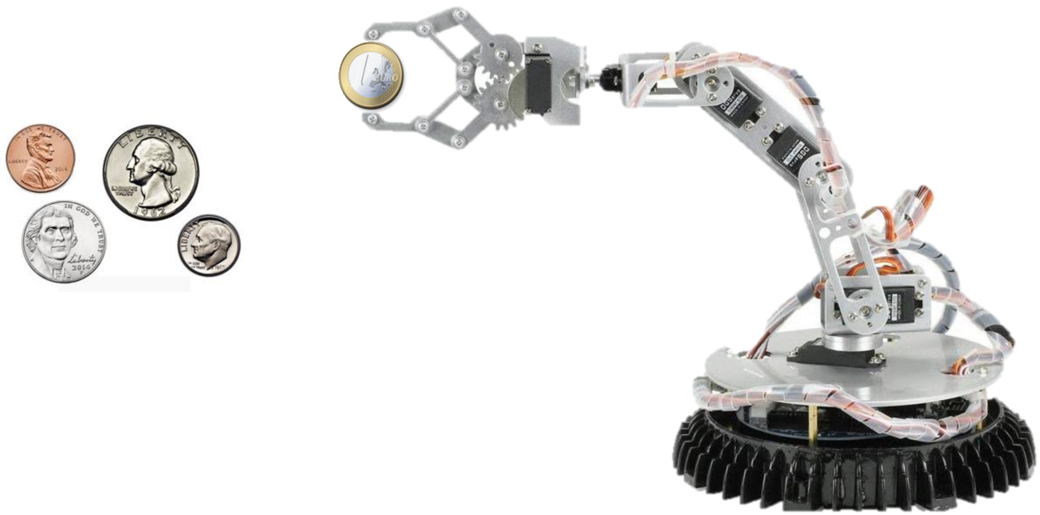
Unsupervised Learning



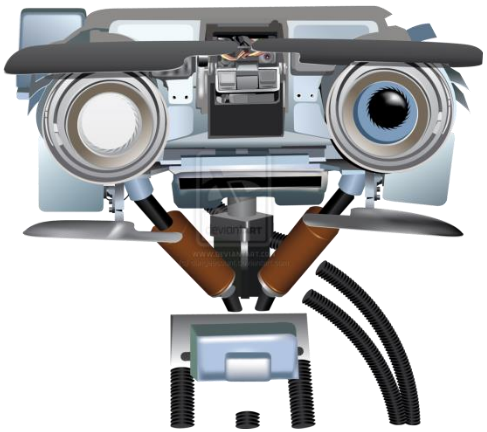
- Given training set consisting of ~~coin denomination (penny, nickel, dime, quarter)~~ mass and size
- Learn… something?

Reinforcement Learning

learn to pick up coins



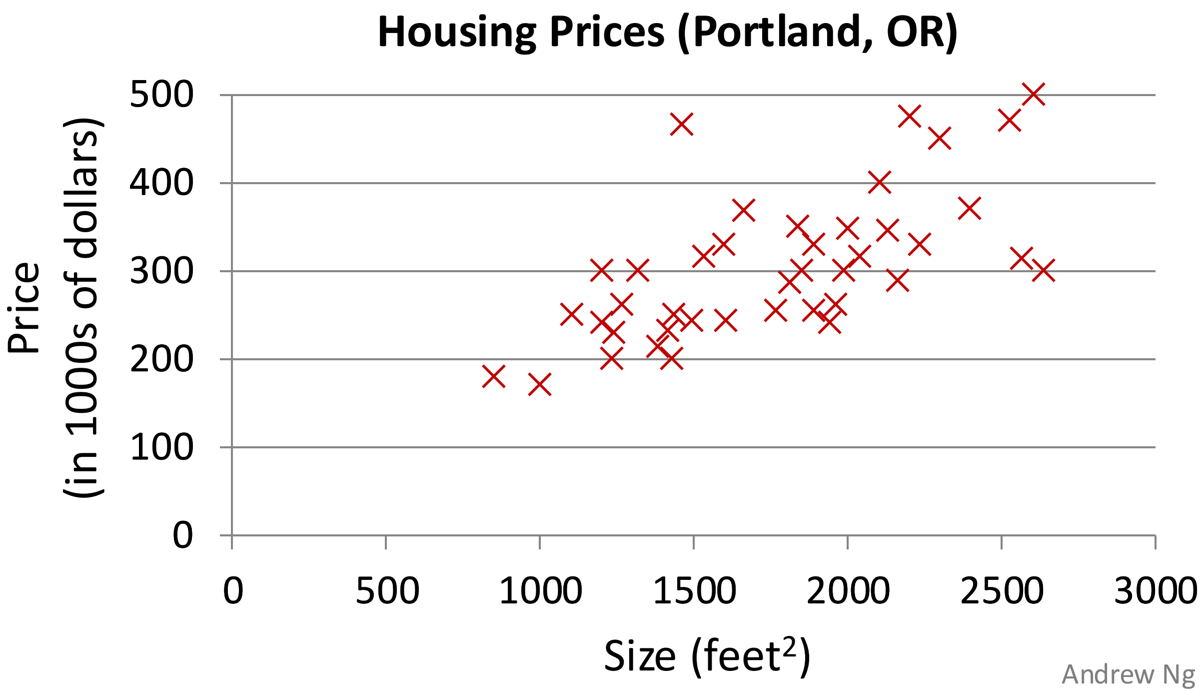
- Given only input, but can take action
- Predict output (action), get a reward for it



Introduction: Supervised Learning

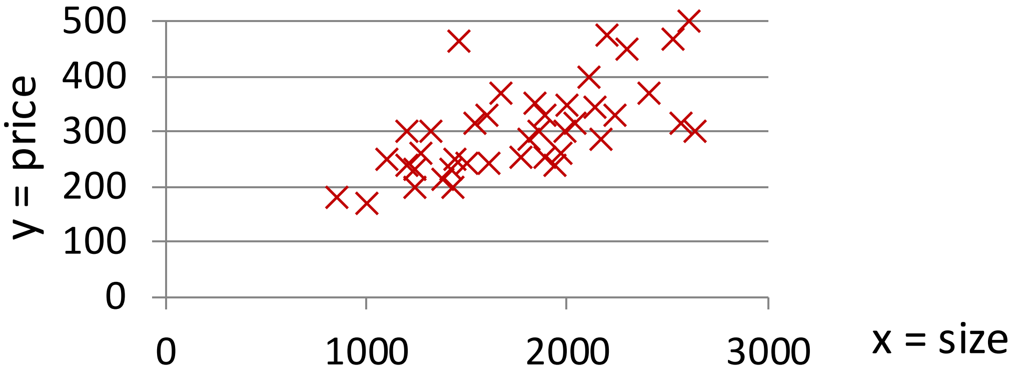
Cost functions

Example: house price prediction



Suppose x = size, y = price

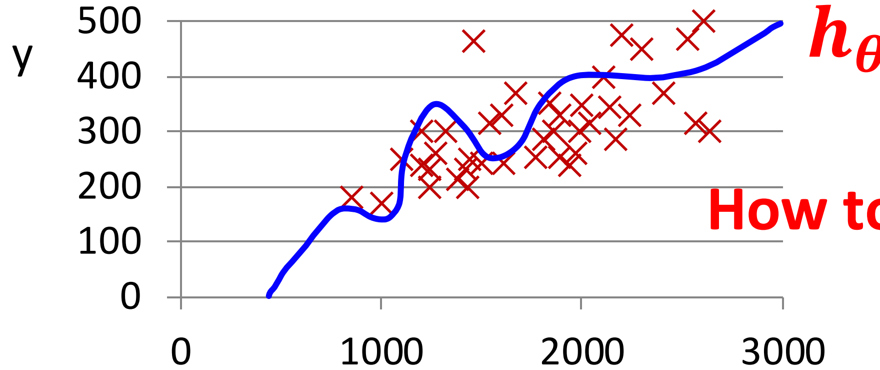
The above is a training set consisting of pairs $\{x^i, y^i\}$



Given: Training Set $\{x^i, y^i\}$

What should the learner be?

Want: input x $\longrightarrow$ **?** $\longrightarrow$ output y



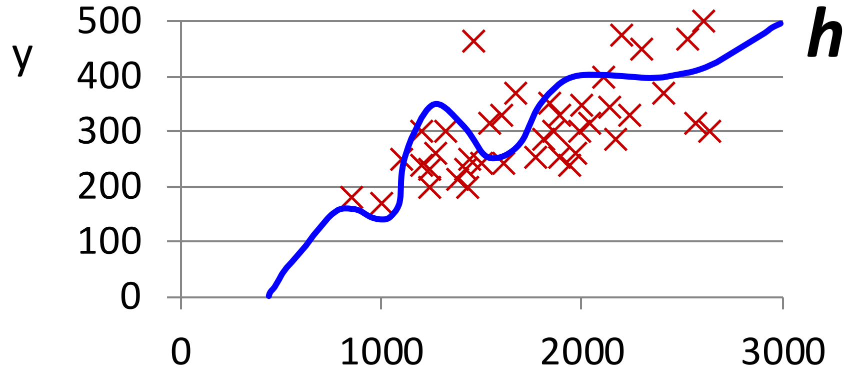
How to get correct θ?

Given: Training Set $\{x^i, y^i\}$

$y = h_\theta(x)$: a function parametrized by θ

"hypothesis"

Want: input x $\longrightarrow$ $\boxed{h_\theta}$ $\longrightarrow$ output y



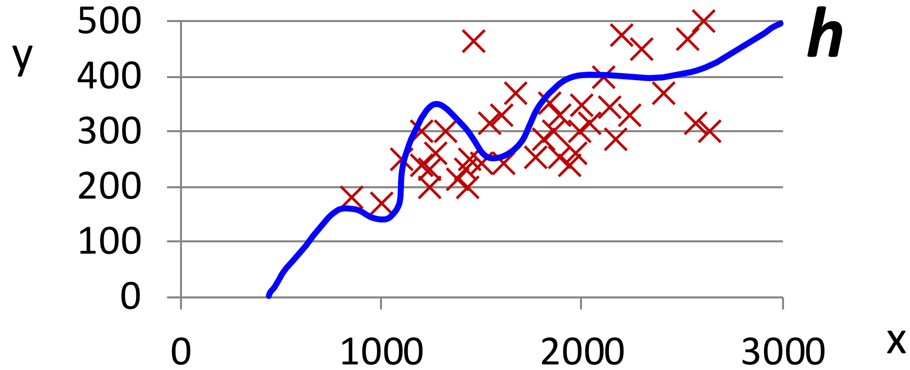
Given: Training Set $\{x^i, y^i\}$

$y = h_\theta(x)$: a function parametrized by θ

Cost function $\boldsymbol{Cost(h_\theta(x^i), y^i)}$

learning == minimizing cost

Want: input x → h_θ → output y



Given: Training Set $\{x^i, y^i\}$

$y = h_\theta(x)$: a function parametrized by θ

Cost function $Cost(h_\theta(x^i), y^i)$

Learn θ*: $\boldsymbol{\min_\theta Cost(h_\theta(x^i), y^i) \quad \forall i}$

Want: input $x \longrightarrow \boxed{h_\theta} \longrightarrow$ output y

Supervised learning

learning == minimizing cost

Given: Training Set $\{x^i, y^i\}$

$y = h_\theta(x)$: a function parametrized by θ

Cost function $Cost(h_\theta(x^i), y^i)$

Learn θ*: $\min_\theta Cost(h_\theta(x^i), y^i)$

Result: input $x \longrightarrow \boxed{h_{\theta}{}^{*}} \longrightarrow$ output y

Training set

Training set:

Size in feet^2 (x)	**Price (\$) in 1000's** (y)
2104	460
1416	232
1534	315
852	178
…	…

Notation:

m = Number of training examples

$x^{(i)}$ = “input” variable / features

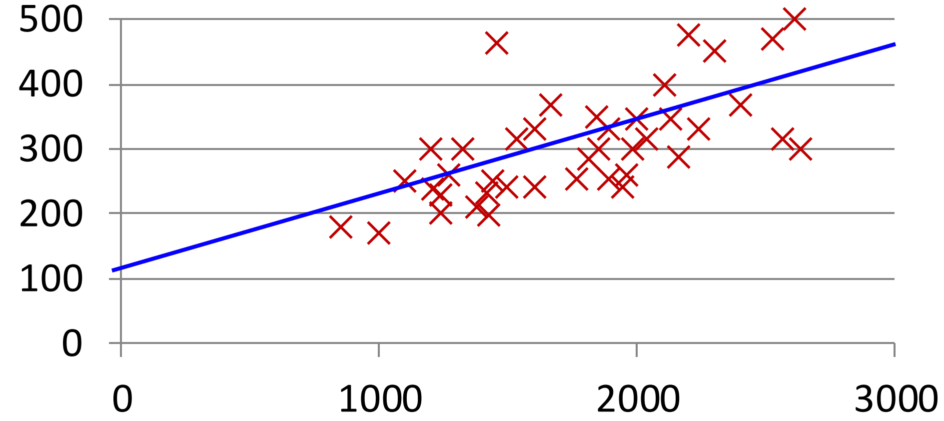
$y^{(i)}$ = “output” variable / “target” variable

What should *h* be?

Linear hypothesis:

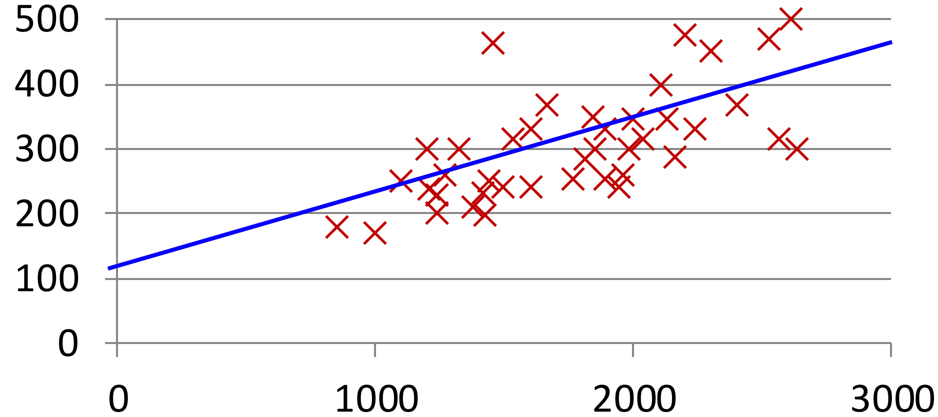
$$h_\theta(x) = \theta_0 + \theta_1 x$$

θ_i: Parameters



$$\min_{\theta} \ \mathrm{Cost}(h_\theta, \{x^i, y^i\})$$

What's a good cost function for this problem?



Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

θ_i: Parameters

Cost Function:

How about "Sum of squared differences" or SSD

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Goal:

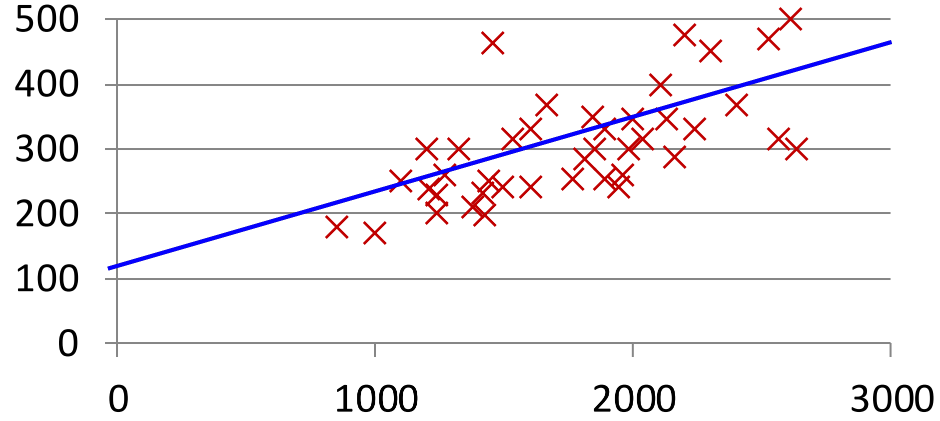
$$\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$$

2-dimensional θ

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

θ_i: Parameters



Cost Function:

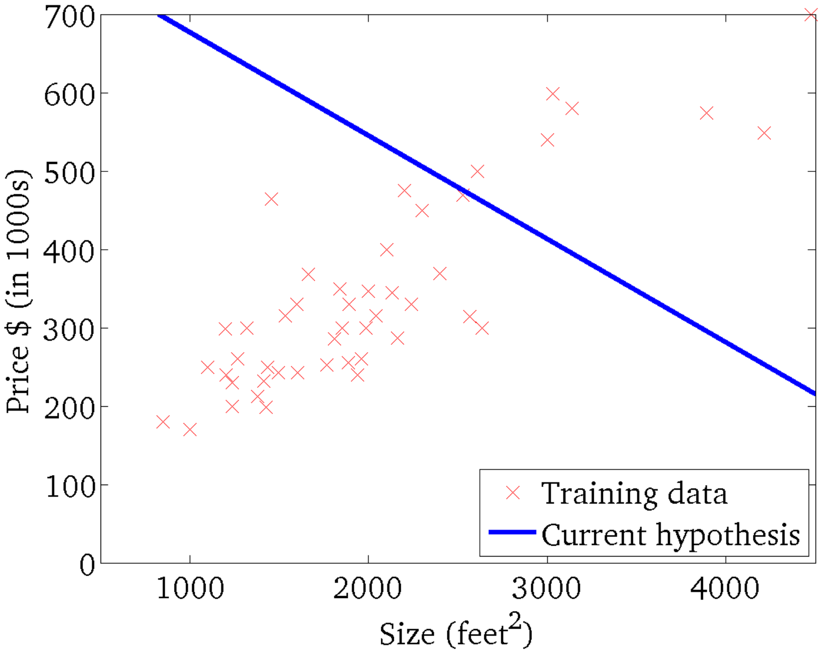
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Can we plot *J* as a function of θ?

Plotting cost for 2-dimensional θ

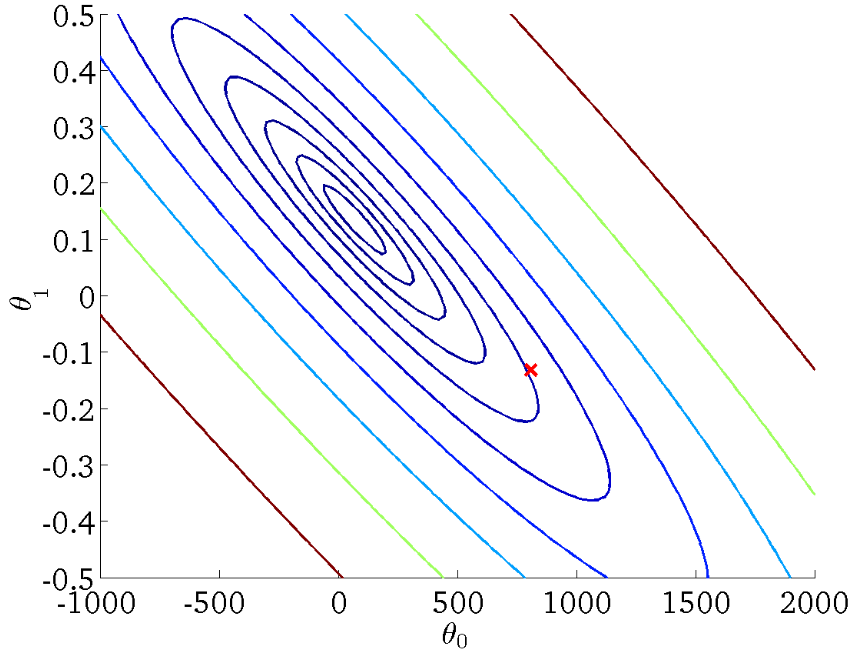
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

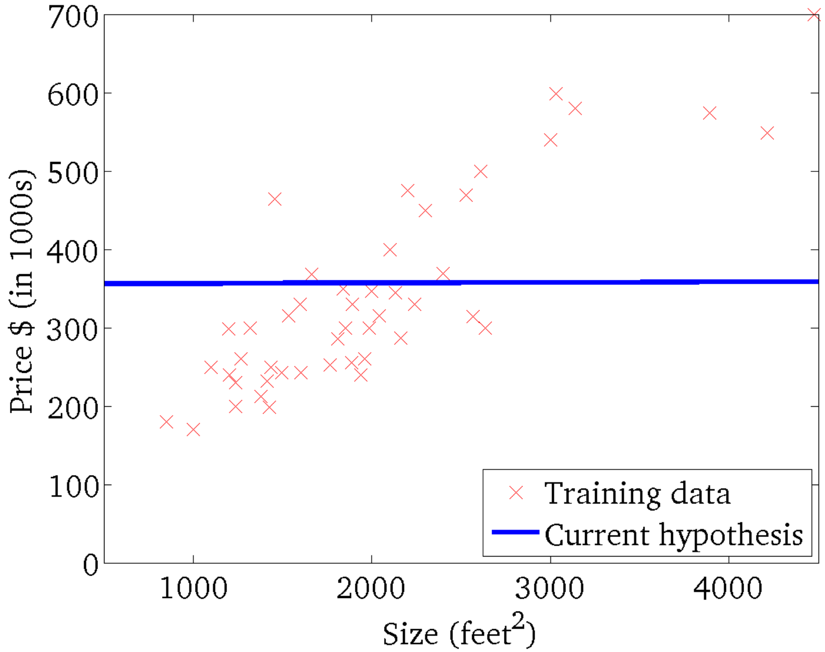
(function of the parameters θ_0, θ_1)



Plotting cost for 2-dimensional θ

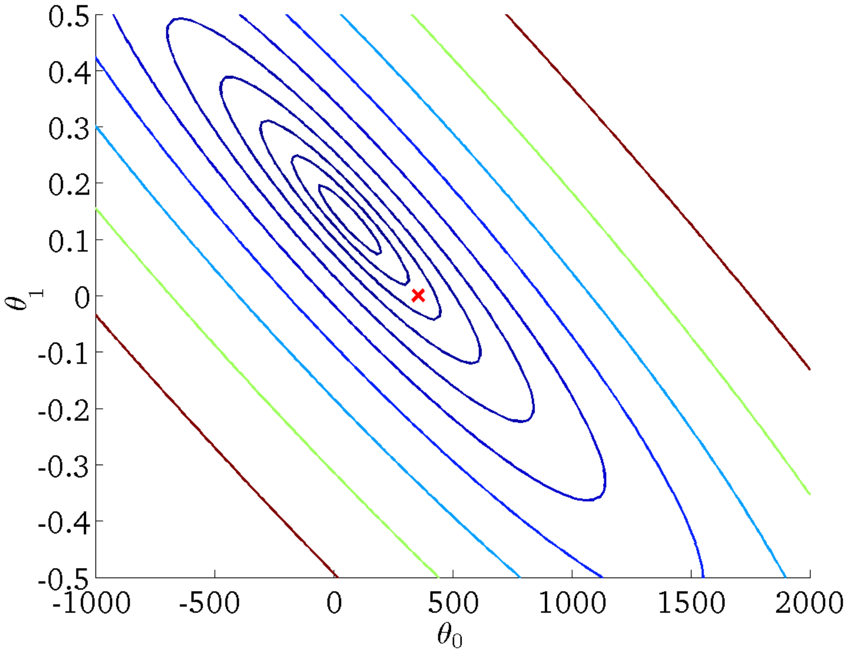
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

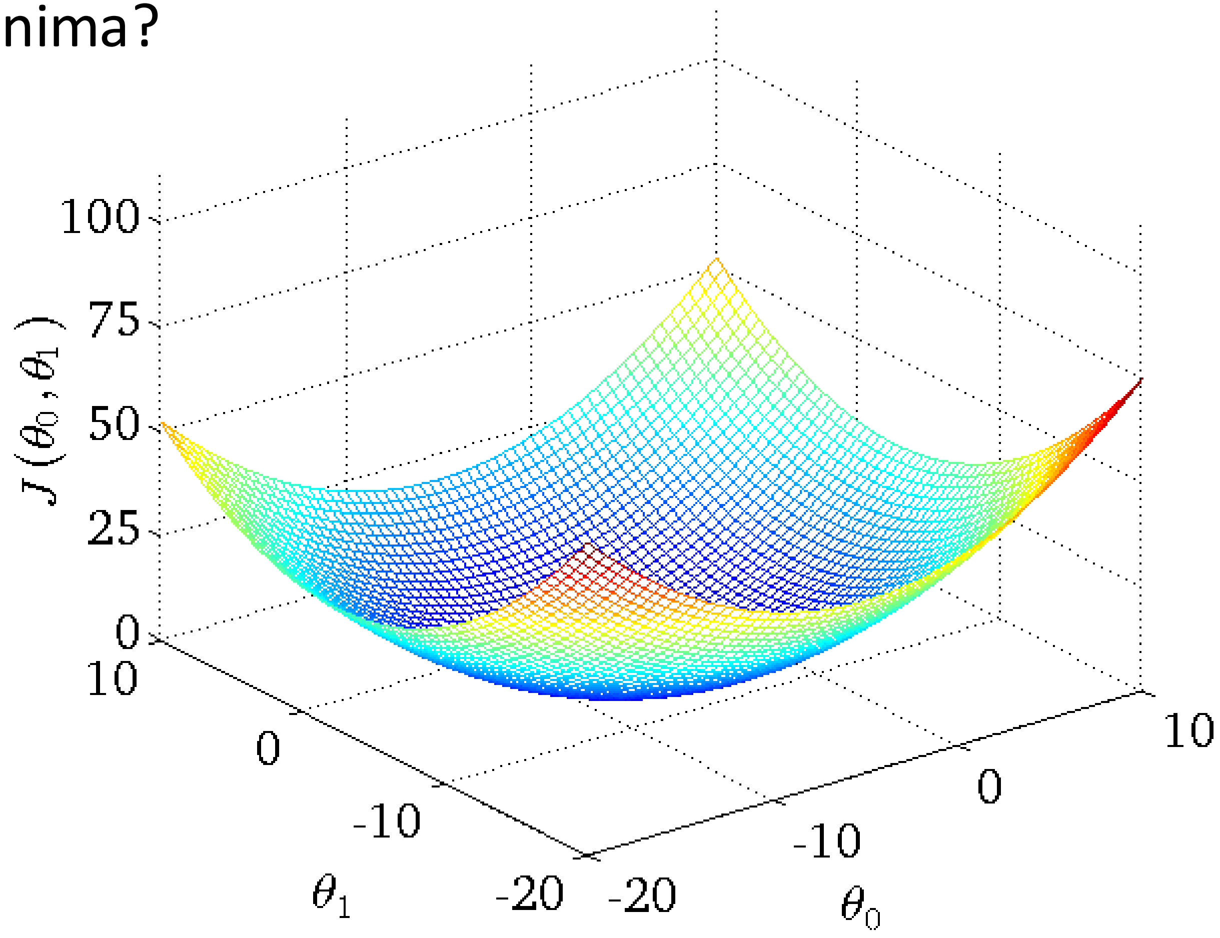
(function of the parameters θ_0, θ_1)



Note, squared loss cost is convex in parameters

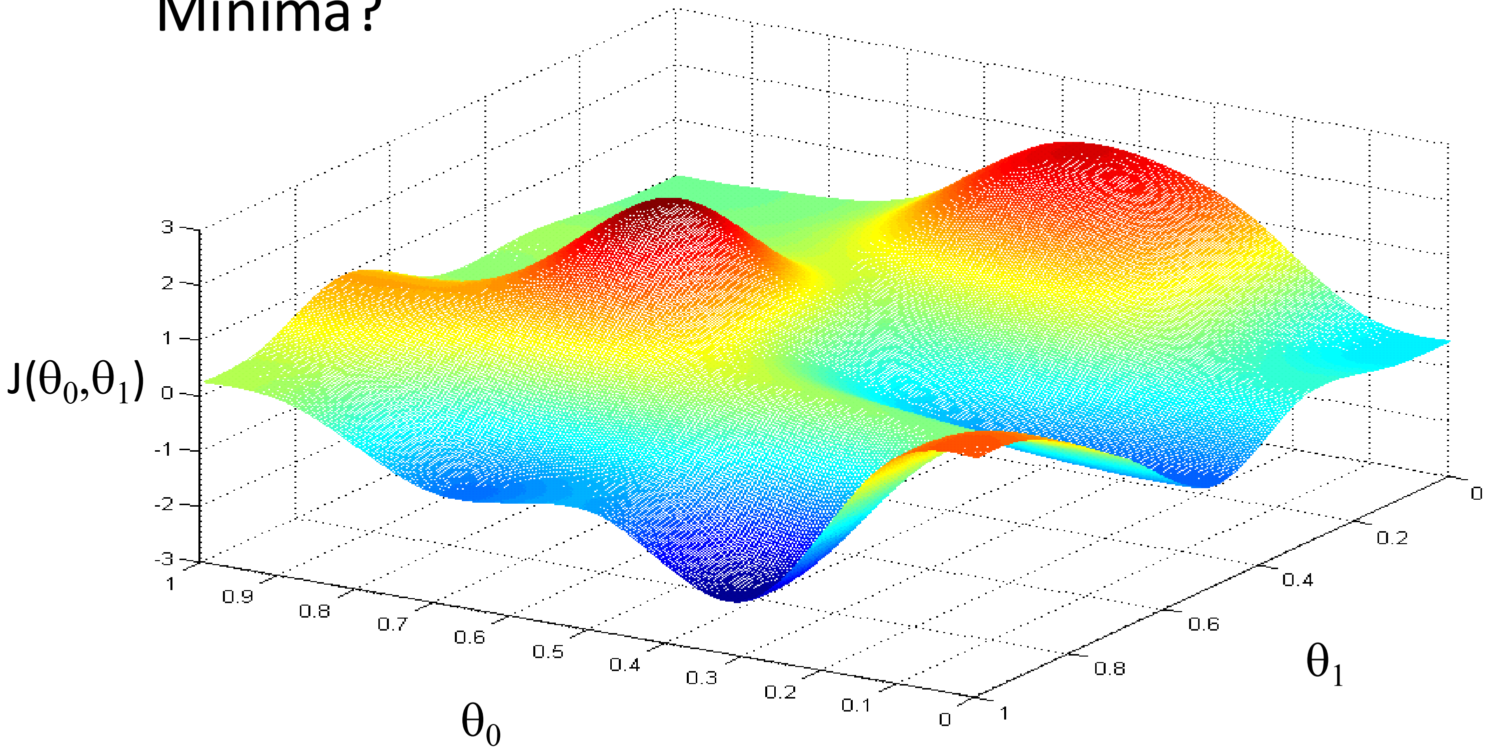
SSD cost function is convex

Minima?



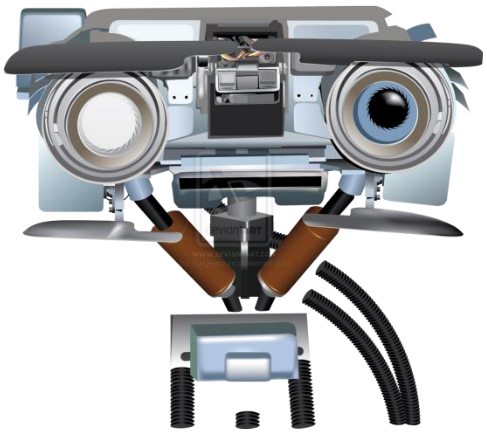
Non-convex cost function

Minima?



Next time…

- How to minimize the SSD cost function
  - Direct solution
  - Indirect solution

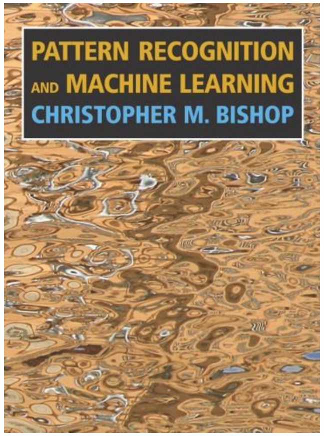


Introduction: Course Overview

Class website

https://sites.google.com/bu.edu/cs542fall2024/

Textbook



- Required textbook

  Bishop, C. M. Pattern Recognition and Machine Learning. Springer. 2007

- Other suggested textbooks

  Duda, R.O., Hart, P.E., and Stork, D.G. Pattern Classification. Wiley-Interscience. 2nd Edition. 2001.

  Marsland, S. Machine Learning: An Algorithmic Perspective. CRC Press. 2009.Theodoridis, S. and Koutroumbas, K. Pattern Recognition. Edition 4. Academic Press, 2008.

  Russell, S. and Norvig, N. Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence. 2003.

  Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press. 1995.

  Hastie, T., Tibshirani, R. and Friedman, J. The Elements of StatisticalLearning. Springer. 2001.

  Koller, D. and Friedman, N. Probabilistic Graphical Models. MIT Press. 2009.

Grading

- Problem Sets (graded on effort) 20%
- Quizzes 20%
- Midterm 25%
- Final 25%
- Participation 10%

Problem Sets

- Bi-weekly problems sets
  - Python coding problems
  - Written math problems
  - Prepare to spend 8+ hours on each
  - Important to prepare you for the tests

- Completion grading
  - Grade = percentage completed by due date
  - Incorrect solution ok, as long as reasonable effort was made
  - Copied solution = 0%

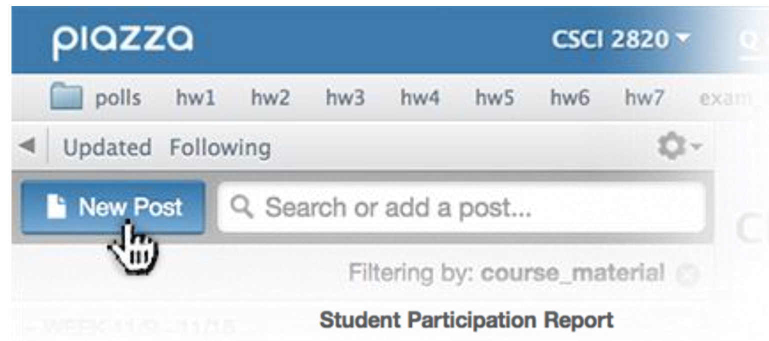
Tests

- Two in-class quizzes
  - Cover only prerequisite math skills (Quiz 1) and problem sets
- Midterm (in class) and final exam
  - Cover all course material

Participation

in class

or online



Name, Email	days online	posts viewed*	contributions**
Rhia J	42	51	66
John K	39	47	58
Jaclyn R	30	39	34

Piazza

https://piazza.com/bu/fall2024/cs542/home

Access code: bishop

Course expectations

- Graduate course
  - 75% grad students, 25% undergrads
- Significant self-learning expected, e.g. by reading the assigned textbook chapters
- Prerequisites
  - Linear algebra
  - Probability & statistics
  - Multivariate calculus
  - Python
  - OR CS 365 (for ugrads)

Is this class for you?

	Yes	No
1. Do you want to do **research** in machine learning or artificial intelligence?	✓	○
2. Are you a graduate student or undergrad interested in grad school?	✓	○
3. Do you want to learn *both* the math and the code behind popular machine learning methods?	✓	○
4. Are you comfortable with the prerequisites?	✓	○

Alternative Machine Learning Courses

- CAS CS 541 Applied Machine Learning covers similar algorithms but focuses more on application, rather than the mathematical principles
- CAS CS 506 Computational Tools for Data Science
- ENG EC 414 Introduction to Machine Learning, designed for undergraduates

Next time…

- review of mathematical skills you need for the course (also in lab this week)
- **Reading**: Bishop 1.2-1.2.4, Appendix B