Regression:

A machine learning perspective

Emily Fox & Carlos Guestrin

Machine Learning Specialization

University of Washington

Part of a specialization

This course is a part of the Machine Learning Specialization

What is the course about?

What is regression?

From features to predictions

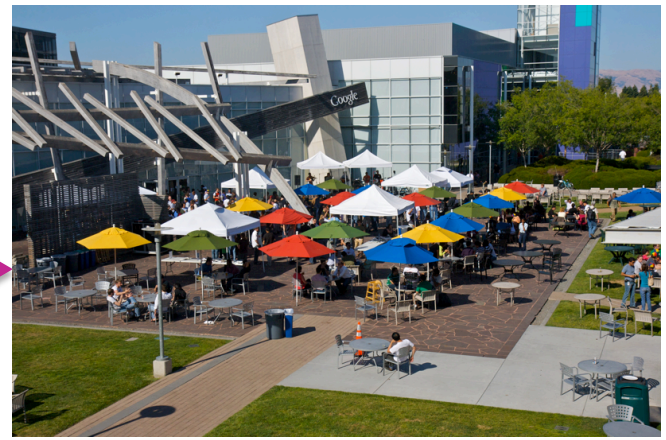
Input x:
features derived from data

Learn x→y relationship

Predict y:
continuous "output" or "response" to input

Salary after ML specialization

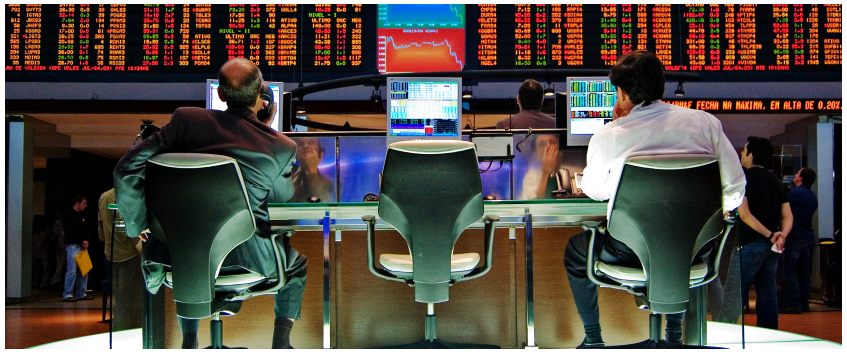
hard work

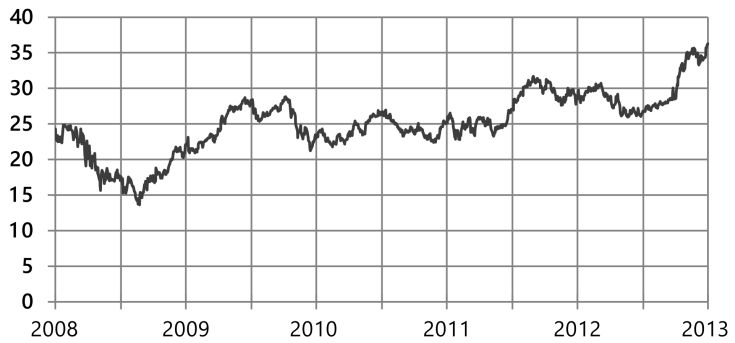


- How much will your salary be? (y = $$)
- Depends on **x** = performance in courses, quality of capstone project, # of forum responses, …

Stock prediction

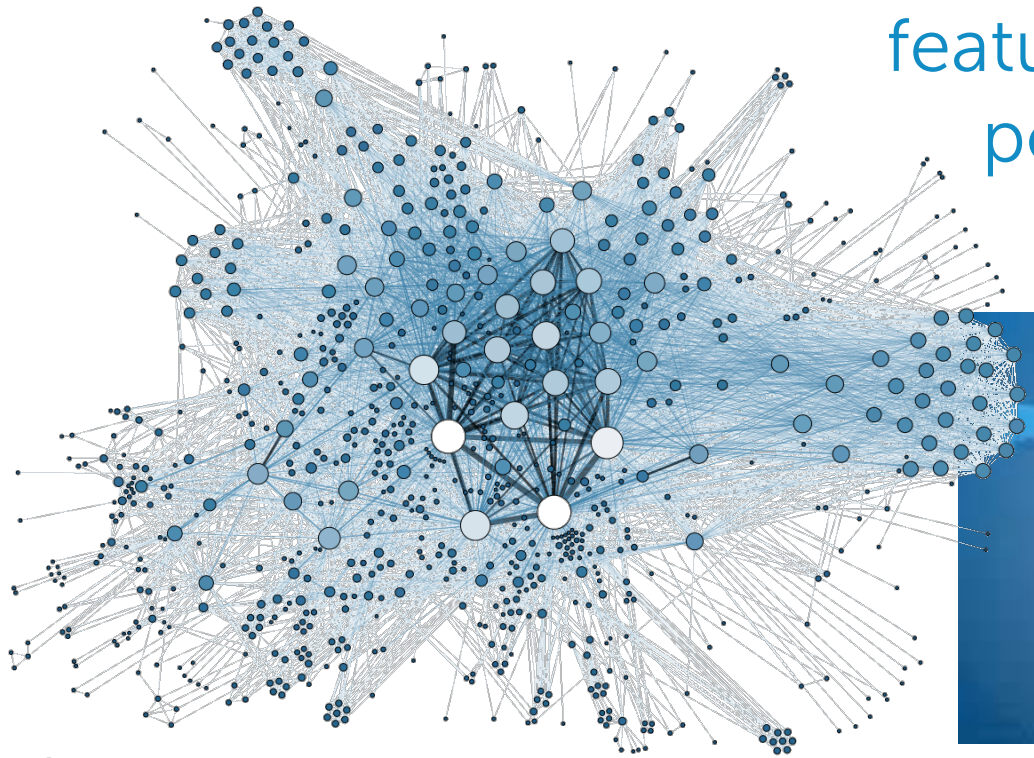
- Predict the price of a stock (y)
- Depends on **x** =
  - Recent history of stock price
  - News events
  - Related commodities



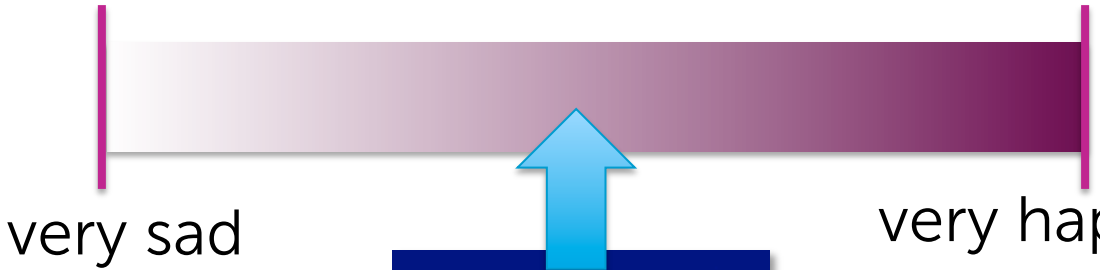


Tweet popularity

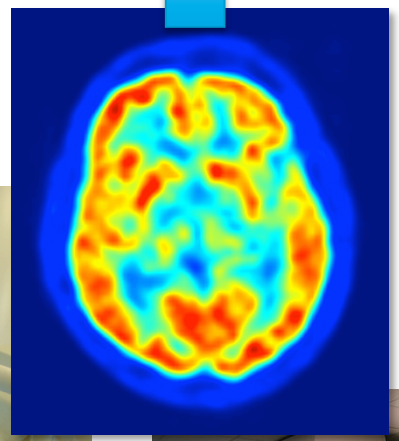
- How many people will retweet your tweet? (y)
- Depends on **x** = # followers,
  # of followers of followers,
  features of text tweeted,
  popularity of hashtag,
  # of past retweets,…

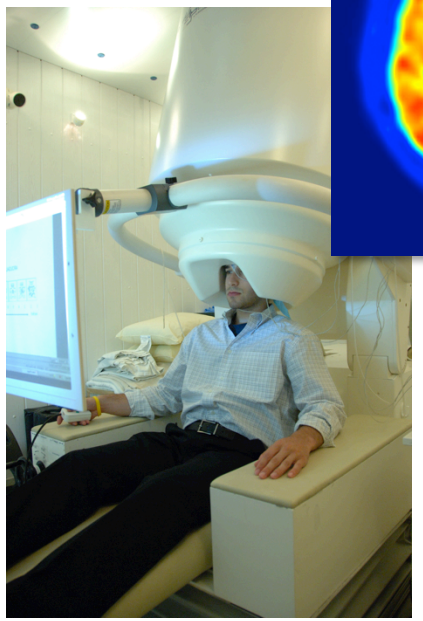


Reading your mind



Output y



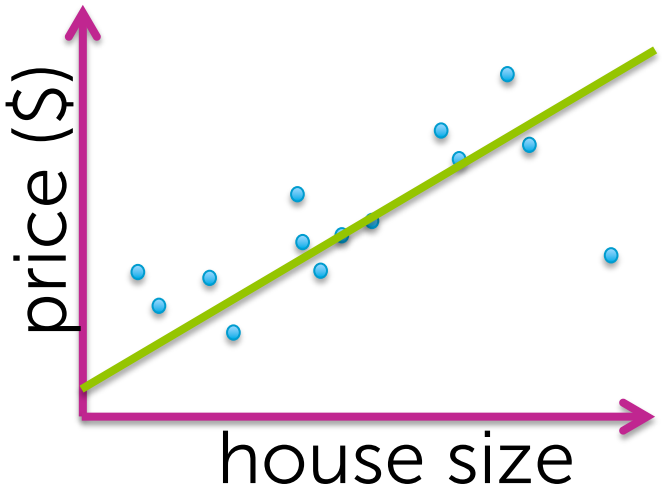


Inputs x are brain region intensities

Case Study: Predicting house prices

(y)

+ house attributes (**x**)



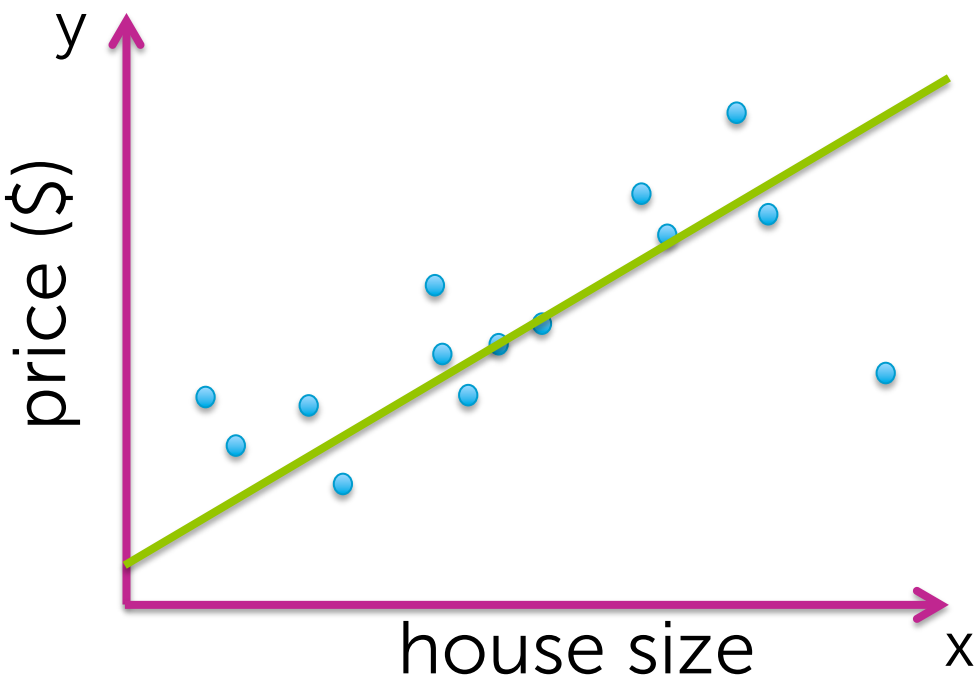
Impact of regression

Course outline

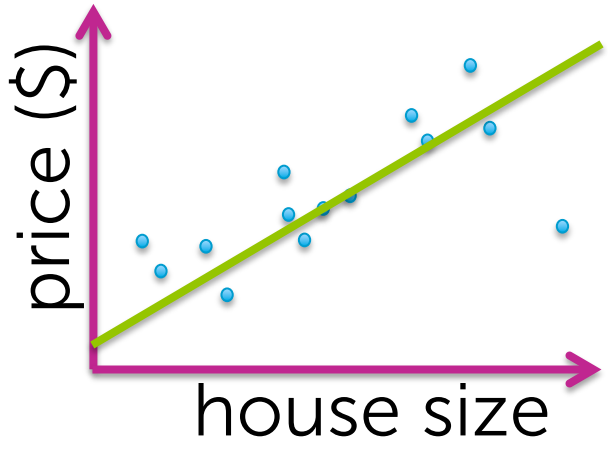
Module 1: Simple Regression

What makes it simple?

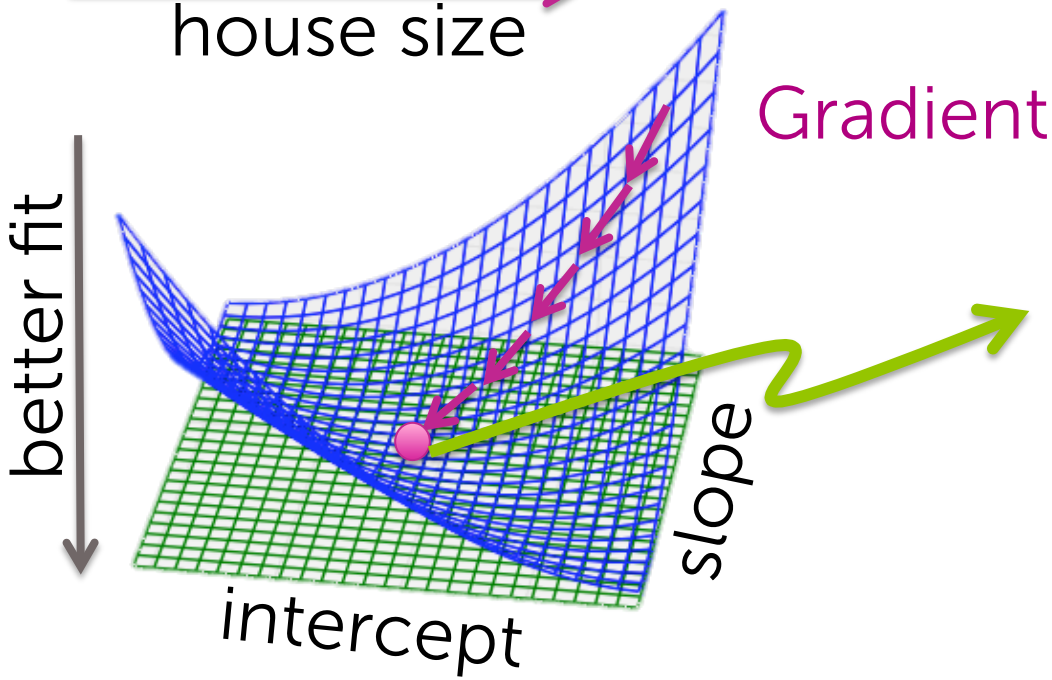
1 input and just fit a line to data



Module 1: Simple Regression



Define goodness-of-fit metric for each possible line

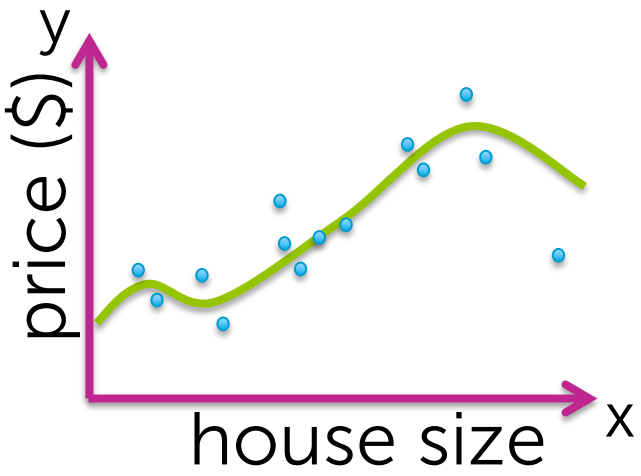


Gradient descent algorithm

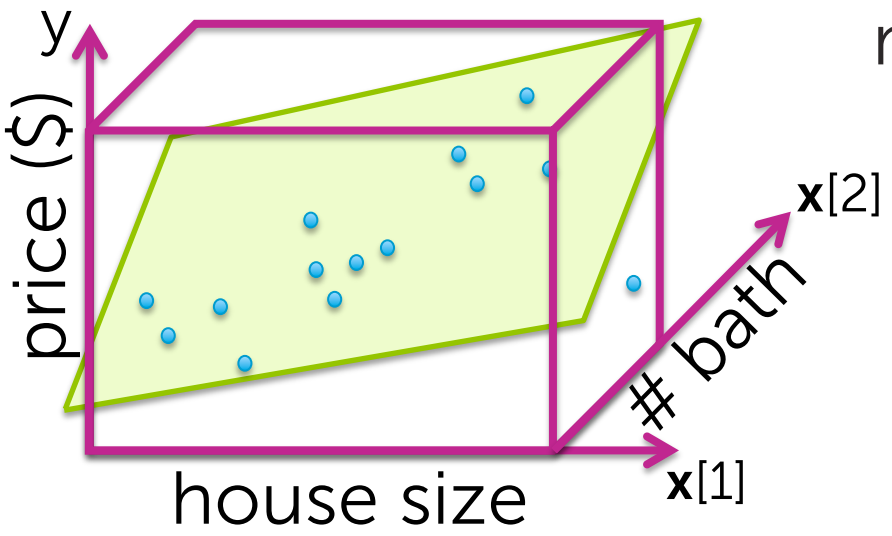
Get estimated parameters

- interpret
- use to form predictions

Module 2: Multiple Regression



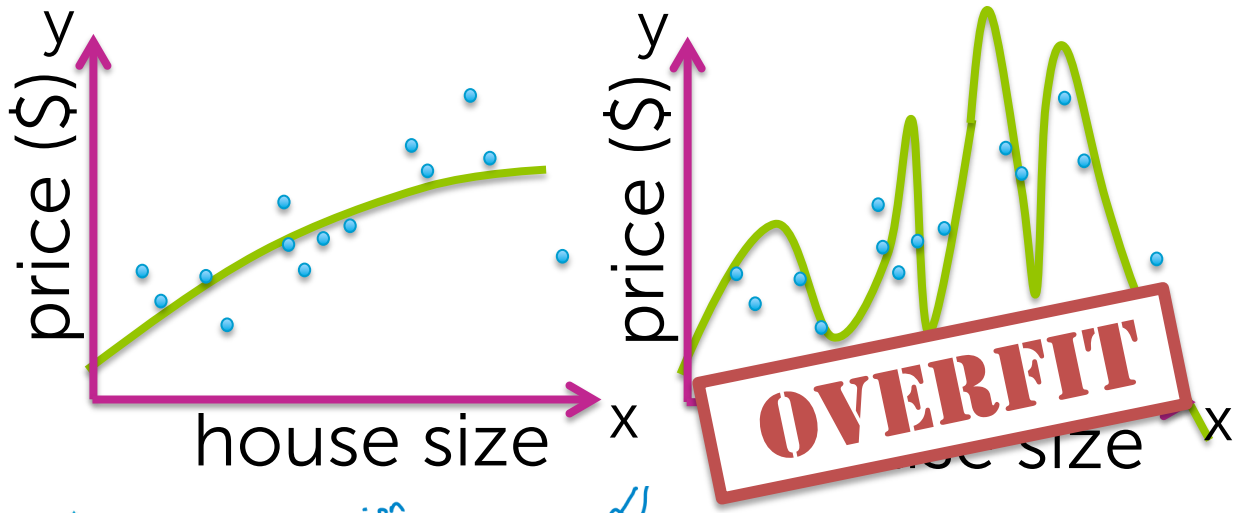
Fit more complex relationships than just a line

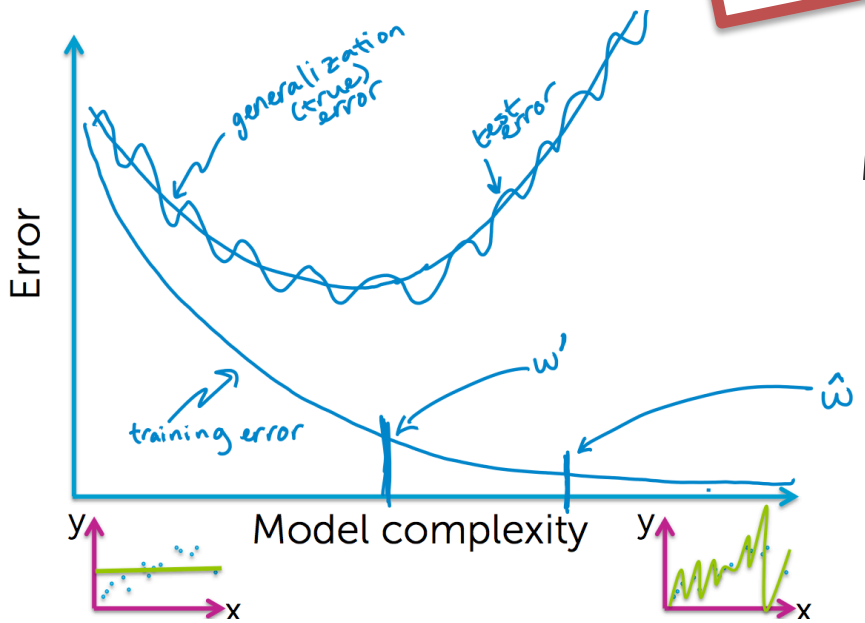


Incorporate more inputs

- Square feet
- # bathrooms
- # bedrooms
- Lot size
- Year built
- …

Module 3: Assessing Performance

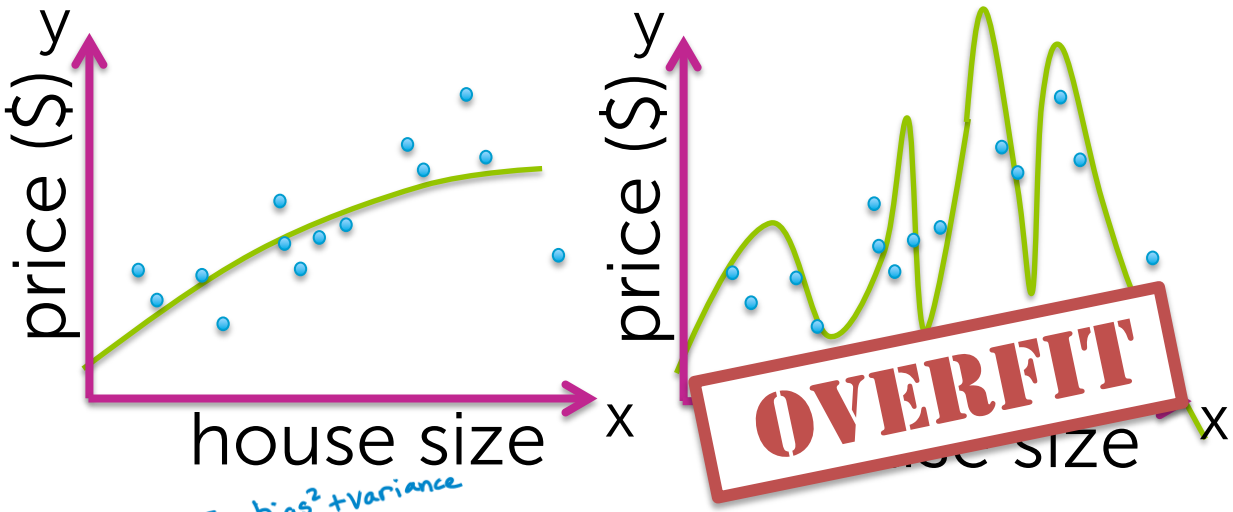


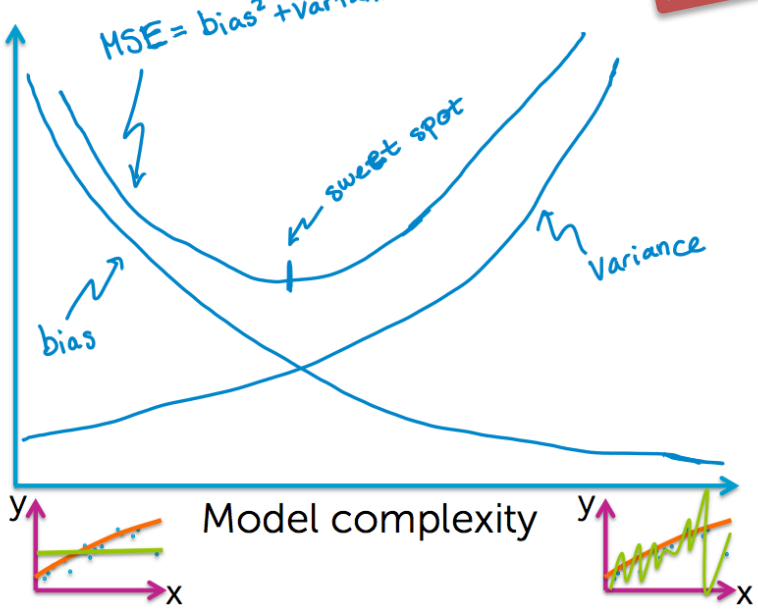


Measures of error:

- Training
- Test
- True (generalization)

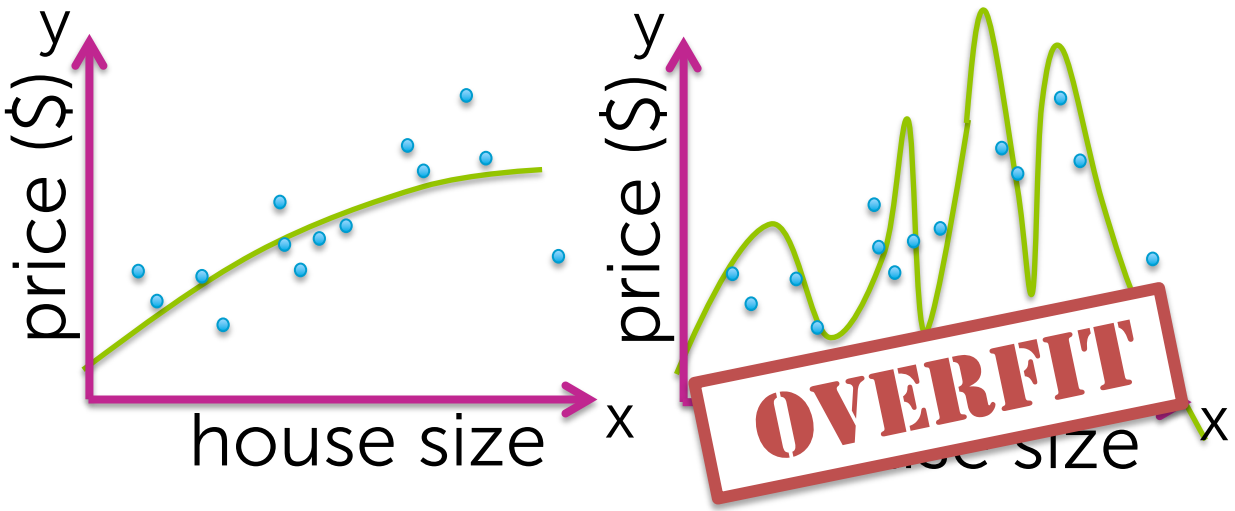
Module 3: Assessing Performance





Bias-variance tradeoff

Module 4: Ridge Regression



Ridge total cost =

measure of fit +

measure of model complexity

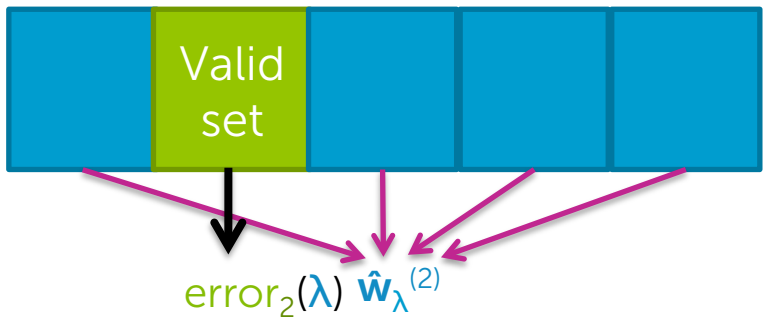
bias-variance tradeoff

Module 4: Ridge Regression

How to choose balance?
(i.e., model complexity)

measure of fit + measure of model complexity

Cross validation



Module 5: Feature Selection & Lasso Regression

Useful for efficiency of predictions and interpretability

Lot size
Single Family
Year built
Last sold price
Last sale price/sqft
Finished sqft
Unfinished sqft
Finished basement sqft
floors
Flooring types
Parking type
Parking amount
Cooling
Heating
Exterior materials
Roof type
Structure style

Dishwasher
Garbage disposal
Microwave
Range / Oven
Refrigerator
Washer
Dryer
Laundry location
Heating type
Jetted Tub
Deck
Fenced Yard
Lawn
Garden
Sprinkler System
⋮

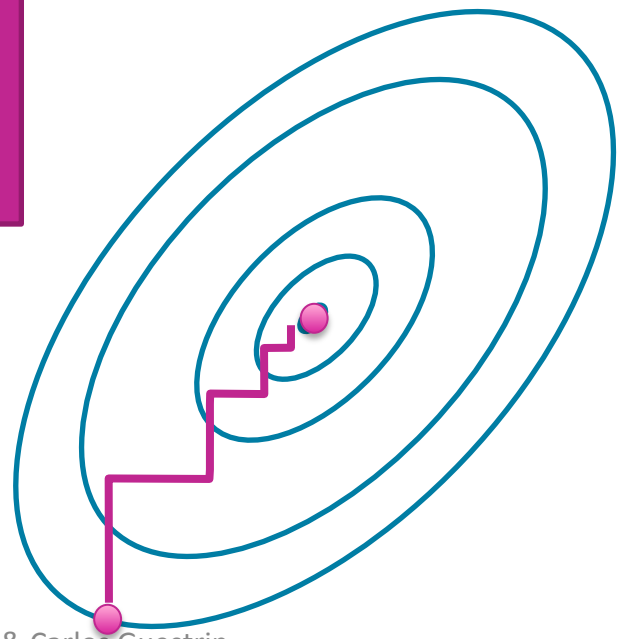
Module 5: Feature Selection & Lasso Regression

Lasso total cost =

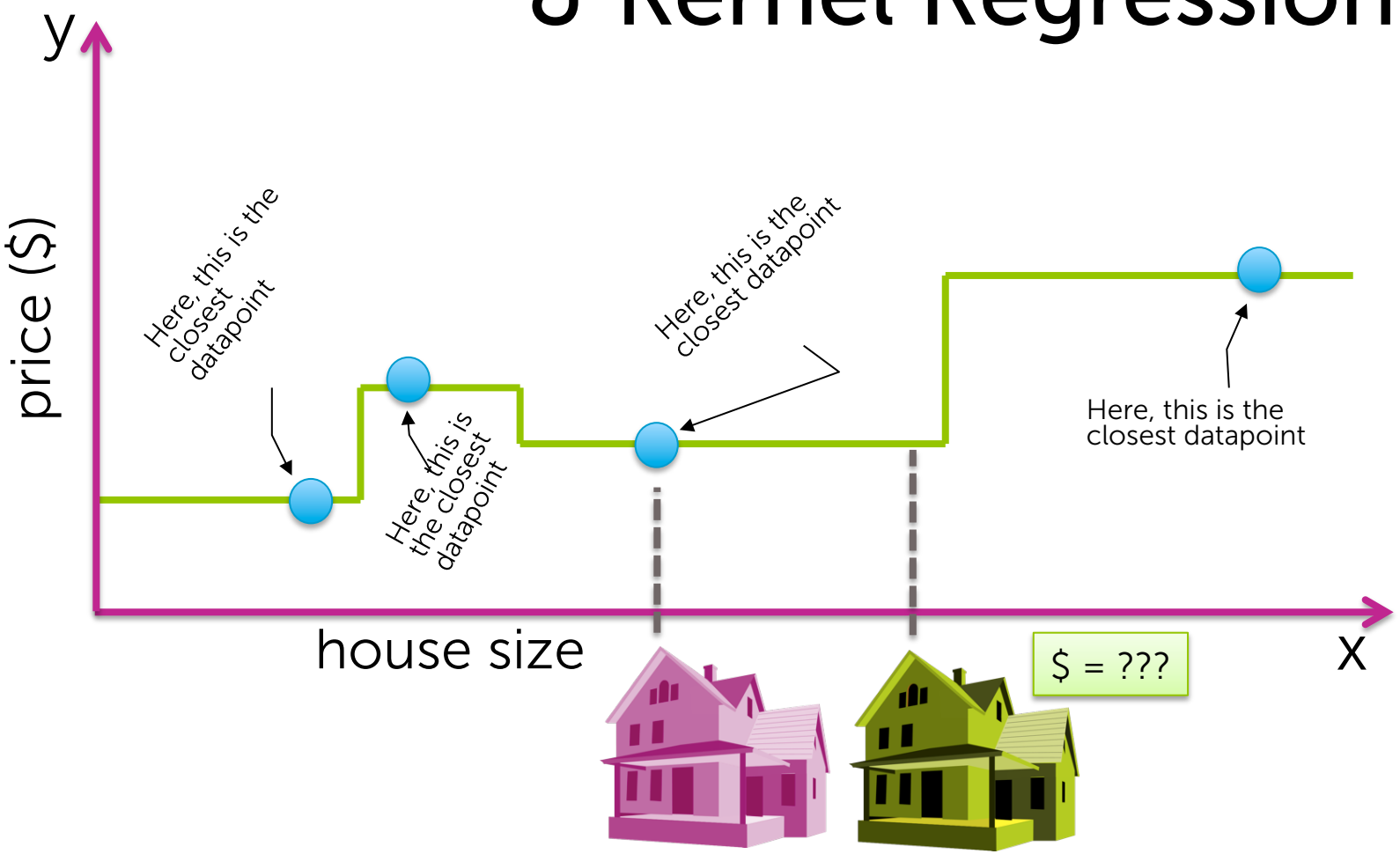
measure of fit + (different) measure of model complexity

knocks out certain features… "sparsity"

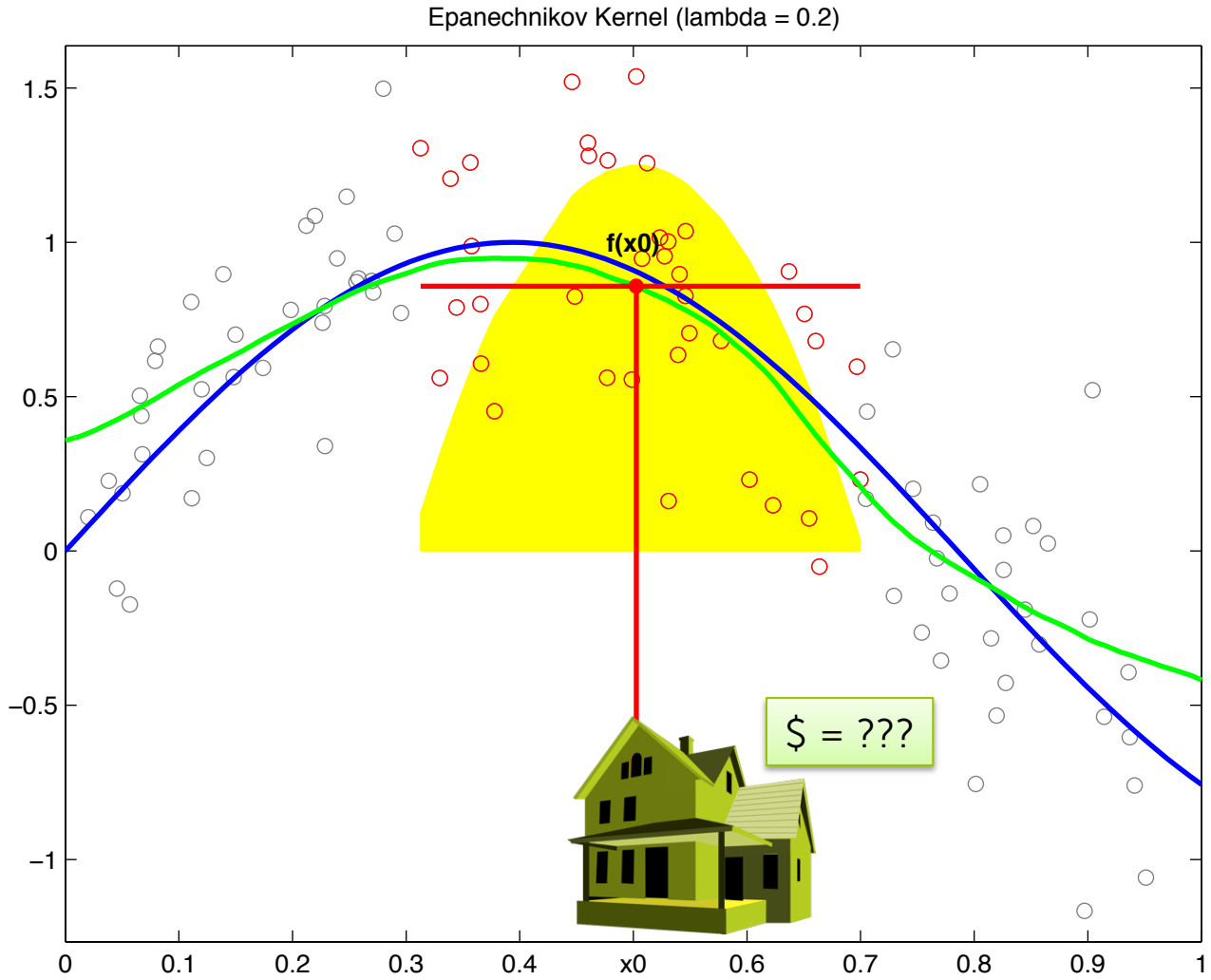
Coordinate descent algorithm



Module 6: Nearest Neighbor & Kernel Regression



Module 6: Nearest Neighbor & Kernel Regression



Summary of what's covered

Models

- Linear regression
- Regularization: Ridge (L2), Lasso (L1)
- Nearest neighbor and kernel regression

Algorithms

- Gradient descent
- Coordinate descent

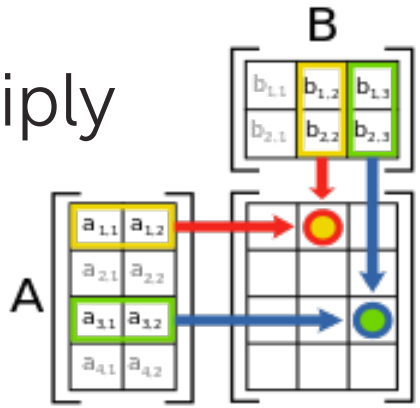
Concepts

- Loss functions, bias-variance tradeoff, cross-validation, sparsity, overfitting, model selection, feature selection

Assumed background

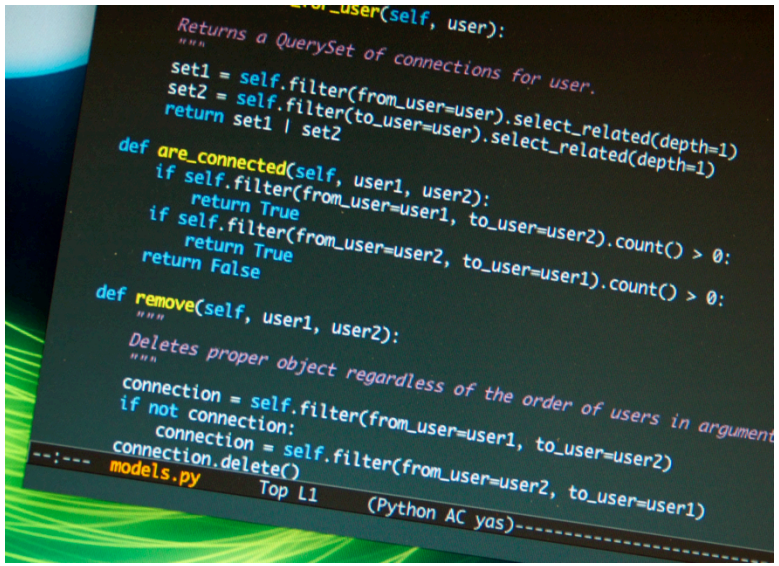
Math background

- Basic calculus
  - Concept of derivatives
- Basic linear algebra
  - Vectors
  - Matrices
  - Matrix multiply



Programming experience

- Basic Python used
  - Can pick up along the way if knowledge of other language



Reliance on GraphLab Create

- SFrames will be used, though not required
  - open source project of Dato (creators of GraphLab Create)
  - can use pandas and numpy instead
- Assignments will:
  1. Use GraphLab Create to explore high-level concepts
  2. Ask you to implement *all* algorithms without GraphLab Create

- Net result:
  - learn how to code methods in Python

Computing needs

- Basic 64-bit desktop or laptop
- Access to internet
- Ability to:
  - Install and run Python (and GraphLab Create)
  - Store a few GB of data

Let’s get started!