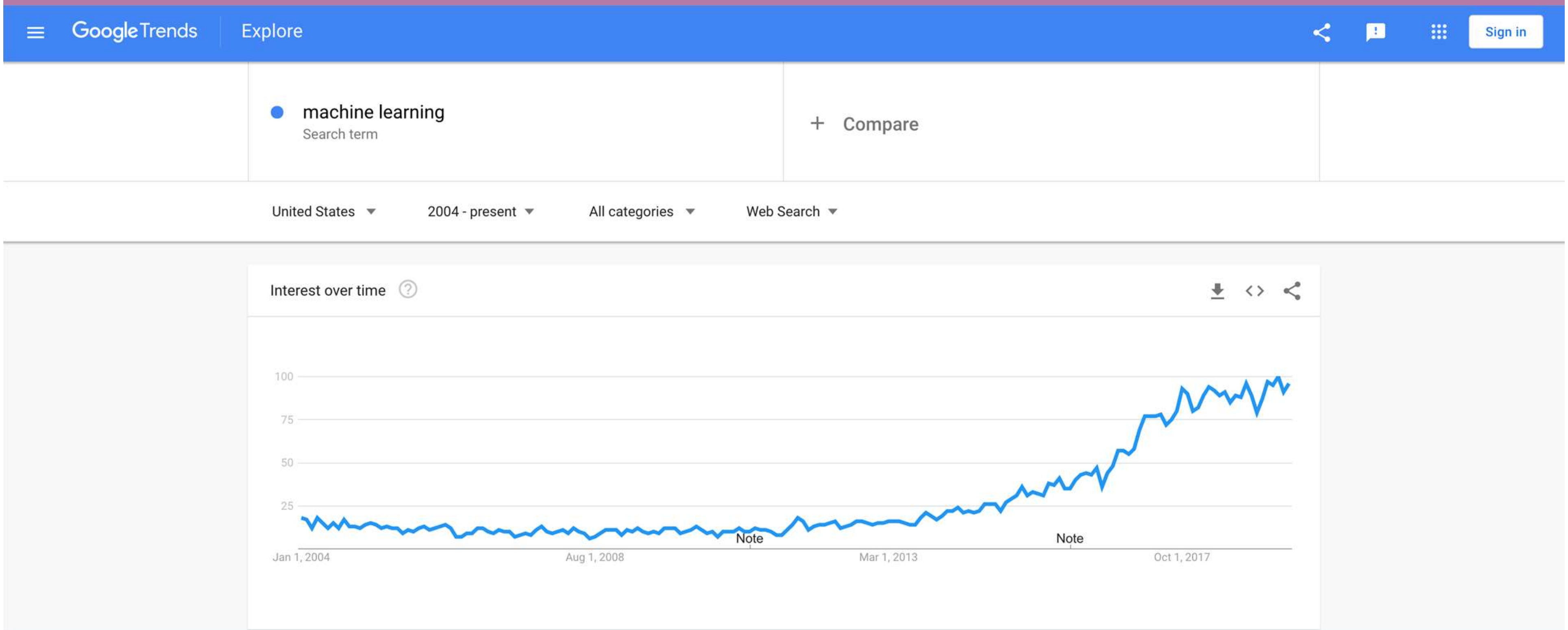
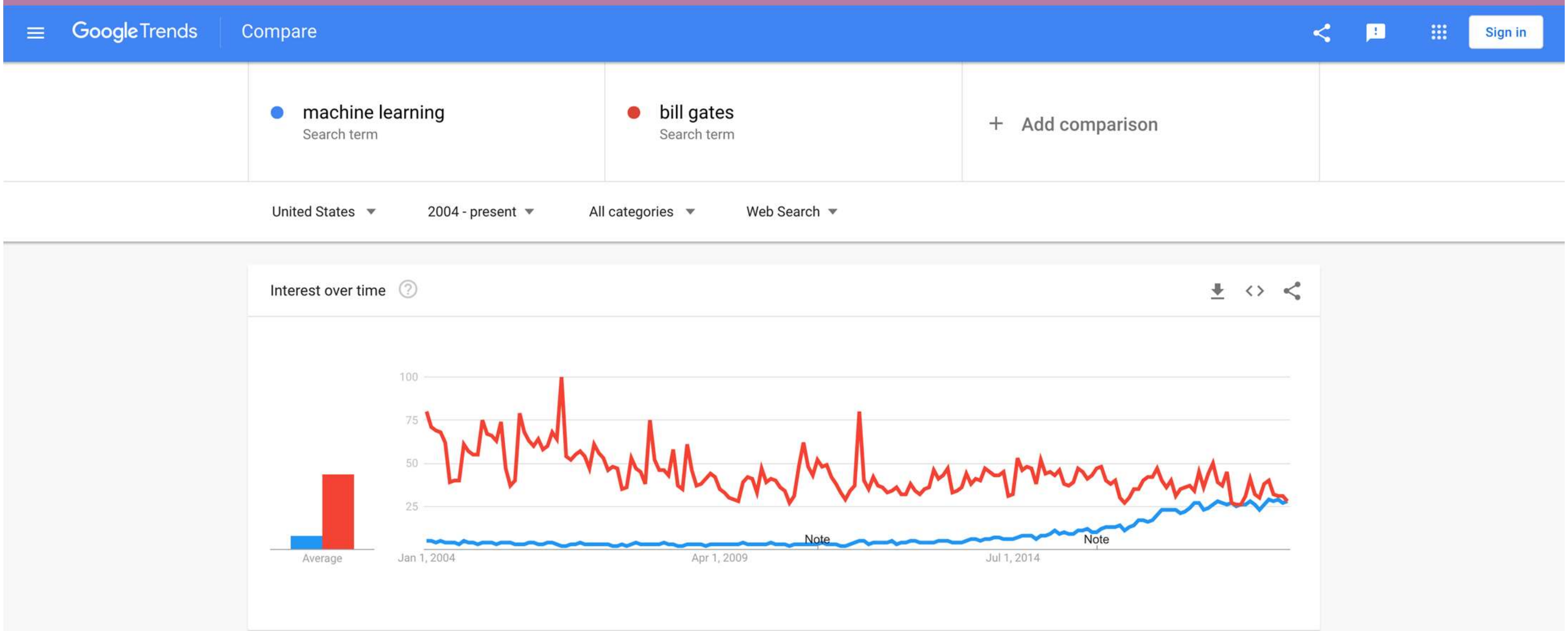
CSE/STAT 416

Introduction + Regression

Hunter Schafer
University of Washington
June 24, 2019

Machine Learning is changing the world.


Google Trends
Explore
Sign in
machine learning
Search term
Compare
United States
2004 - present
All categories
Web Search
Interest over time
100
75
50
25
Note
Note
Jan 1, 2004
Aug 1, 2008
Mar 1, 2013
Oct 1, 2017


Google Trends
Compare
Sign in
machine learning
Search term
bill gates
Search term
Add comparison
United States
2004 - present
All categories
Web Search
Interest over time
100
75
50
25
Average
Jan 1, 2004
Apr 1, 2009
Jul 1, 2014
Note
Note

It's Everywhere!

Retail

PageRank
Search

Coupons

Movie
Distribution

Campaigning

Real Estate

Legal
Advice

Adsense
Advertising

Human
Resources

Dating

Networking

CRM

Music

Wearables

Disruptive companies
differentiated by
INTELLIGENT
APPLICATIONS
using
Machine Learning

It’s Everywhere…

It's Everywhere…

What is Machine Learning?

Generically (and vaguely)

Machine Learning is the study of algorithms that improve their **performance** at some **task** with **experience**

Course Overview

This course is broken up into 5 main case studies to explore ML in various contexts/applications.

1. Regression
   - Predicting housing prices
2. Classification
   - Positive/Negative reviews (Sentiment analysis)
3. Document Retrieval + Clustering
   - Find similar news articles
4. Recommender Systems
   - Given past purchases, what do we recommend to you?
5. Deep Learning
   - Recognizing objects in images

Course Topics

Models

- Linear regression, regularized approaches (ridge, LASSO)
- Linear classifiers: logistic regression
- Non-linear models: decision trees
- Nearest neighbors, clustering
- Recommender systems
- Deep learning

Algorithms

- *Gradient descent*
- Boosting
- K-means

Concepts

- Point estimation, MLE
- Loss functions, bias-variance tradeoff, cross-validation
- Sparsity, overfitting, model selection
- Decision boundaries

ML Course Landscape

CSE 446

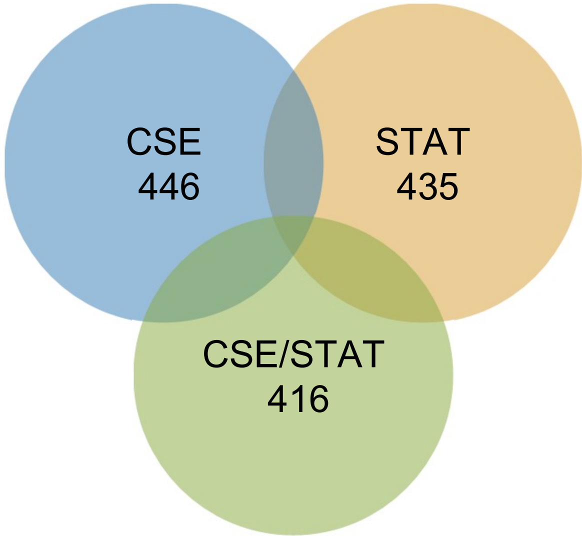
- CSE majors
- Very technical course

STAT 435

- STAT majors
- Very technical course

CSE/STAT 416

- Everyone else!
  - This is a super broad audience!
- Give everyone a strong foundational understanding of ML
  - More breadth than other courses, a little less depth



Level of Course

Our Motto

Everyone should be able to learn machine learning, so our job is to make tough concepts intuitive and applicable.

This means…

- Minimize pre-requisite knowledge
- Focus on important ideas, avoid getting bogged down by math
- Maximize ability to develop and deploy
- Use pre-written libraries to do many tasks
- Learn concepts in case studies

Does not mean course isn't fast paced! There are a lot of concepts to cover!

Course Logistics

Who am I?

- Hunter Schafer
  - Lecturer
  - Paul G. Allen School for Computer Science & Engineering (CSE)
- Office Hours
  - Time: 12:30 pm - 2:30 pm, Mondays
  - Location: CSE 444
- Contact
  - Personal Matters: hschafer@cs.washington.edu
  - Course Content + Logistics: Piazza

Who are the TAs?

Matthew Rockett
rockettm@cs

TA

Josh Ervin
joshue@cs

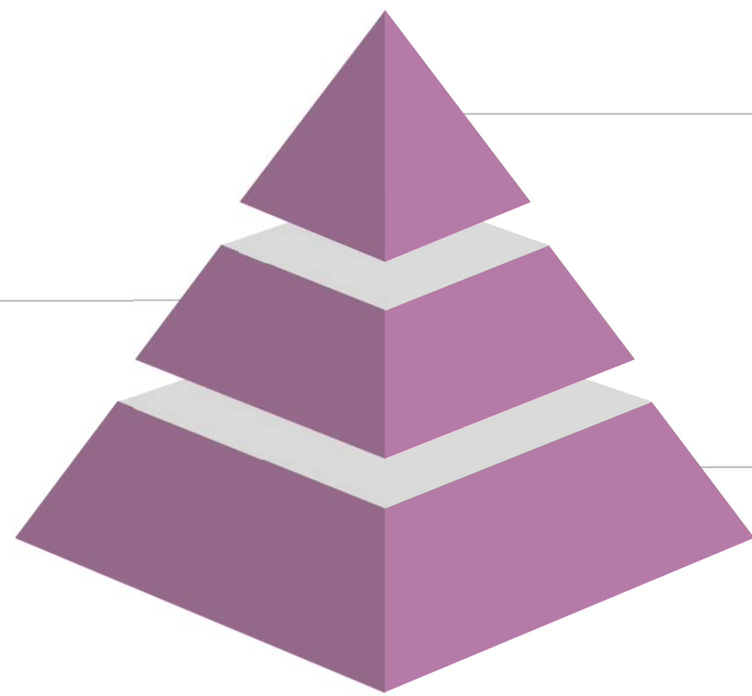
AA (9:40)
AB (10:50)

Rahul Biswas
rbiswas1@uw

AC (12:00)

Mon	Tue	Wed	Thur	Fri
Lecture	Nothing	**Lecture**	**Section**	Nothing
	Previous HW **Due**	Next HW **Released**		

- We happen to not record attendance in lectures and section, but attending these sessions is expected
- Panopto for Lecture (on Canvas)



Homework

With the scaffolding from **1 and 2**, you are probably now capable to tackle the homework. These will be complex and challenging, but you'll continue to **learn by doing.**

1

Sections

Practice material covered in **1** in a context where a TA can help you.

The emphasis is still on you l**earning by doing.**

2

Lectures

Introduced to material for the first time. Mixed with activities and demos to give you a chance to **learn by doing.**

No where near mastery yet!

3

Assessment

- **Weekly Homework Assignments**
  - **Weight**: 70%
  - **Number**: Approximately 8
  - Each Assignment has two parts that contribute to your grade separately:
    - Programming (55%)
    - Conceptual (15%)
- **Final Exam**
  - **Weight**: 30%
  - **Date:** Wednesday, August 21 from 9:40 am – 11:50 pm
  - **Location:** GWN 301

Homework Logistics

- **Late Days**
  - 4 Free Late Days for the whole quarter.
  - Can use up to 2 Late Days on any assignment.
  - Each Late Day used after the 4 Free Late Days results in a -10% on that assignment
- **Collaboration**
  - You are encouraged to discuss assignments and concepts **at a high level**
    - If you have code in front of you in your discussion, probably **NOT** high level
    - Discuss process, not answers
  - All code and answers submitted must be your own
- **Turn In**
  - Everything completed and turned in on Gradescope
  - Multiple “assignments” on Gradescope per assignment

Poll Everywhere

Pair

2 minutes

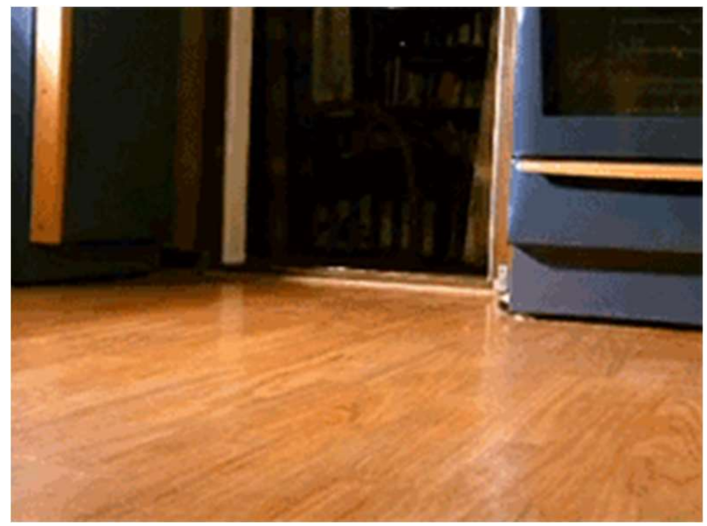
pollev.com/cse416

On your phone

If you could only have one pet, would you rather have a dog or cat?

When you're done

Introduce yourself to your neighbor, and talk about what you did over summer "break"!



Case Study 1

Regression:
Housing Prices

Fitting Data

Goal: Predict how much my house is worth

Have data from my neighborhood

n data points (x_i, y_i)

$$\begin{cases} (x_1, y_1) = (2318\, sq.ft.\,,\ \$\, 315k) \\ (x_2, y_2) = (1985\, sq.ft.\,,\ \$\, 295k) \\ (x_3, y_3) = (2861\, sq.ft.\,,\ \$\, 370k) \\ \quad \vdots \qquad\qquad \vdots \\ (x_n, y_n) = (2055\, sq.ft.\,,\ \$\, 320k) \end{cases}$$

Assumption:

There is a relationship between $y \in \mathbb{R}$ and $x \in \mathbb{R}^d$

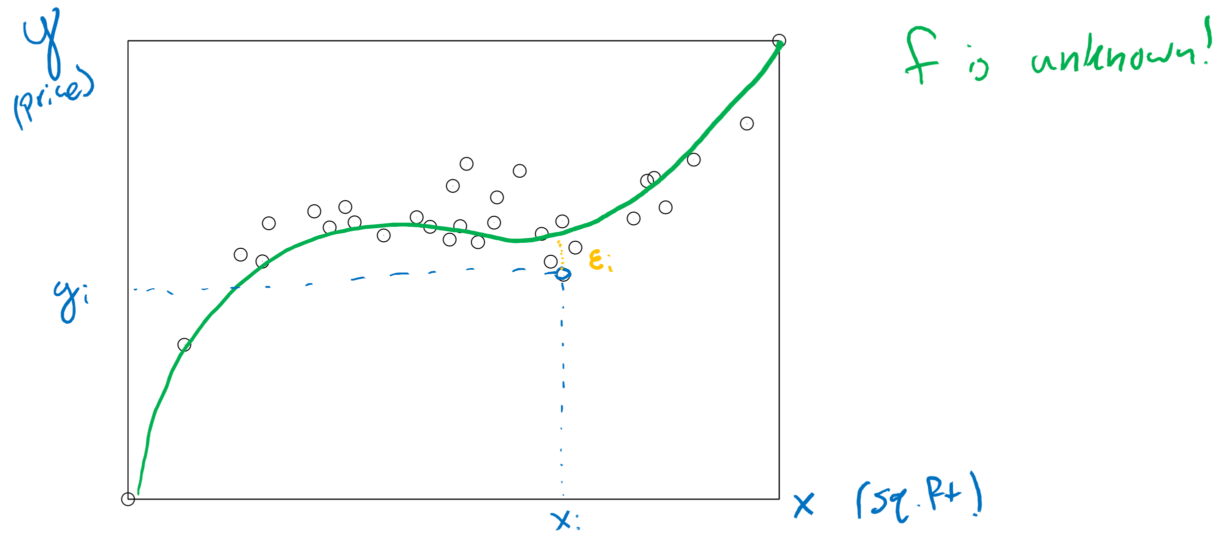
$$y \approx f(x)$$

x is the **input data.** Can potentially have many inputs

y is the **outcome/response/target/label/dependent variable**

Model

A **model** is how we *assume* the world works



Regression model:

$y_i = f(x_i) + \varepsilon_i$

$E[\varepsilon_i] = 0$ ← expectation

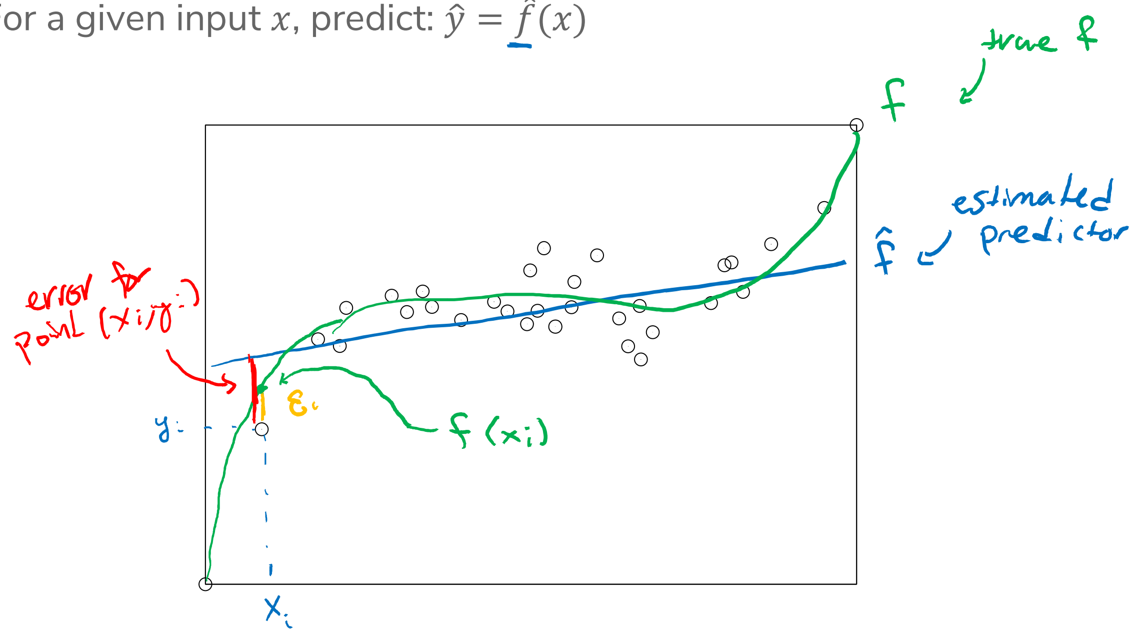
"Essentially, all models are wrong, but some are useful."
- George Box, 1987

Predictor

We don't know f! We need to learn it from the data!

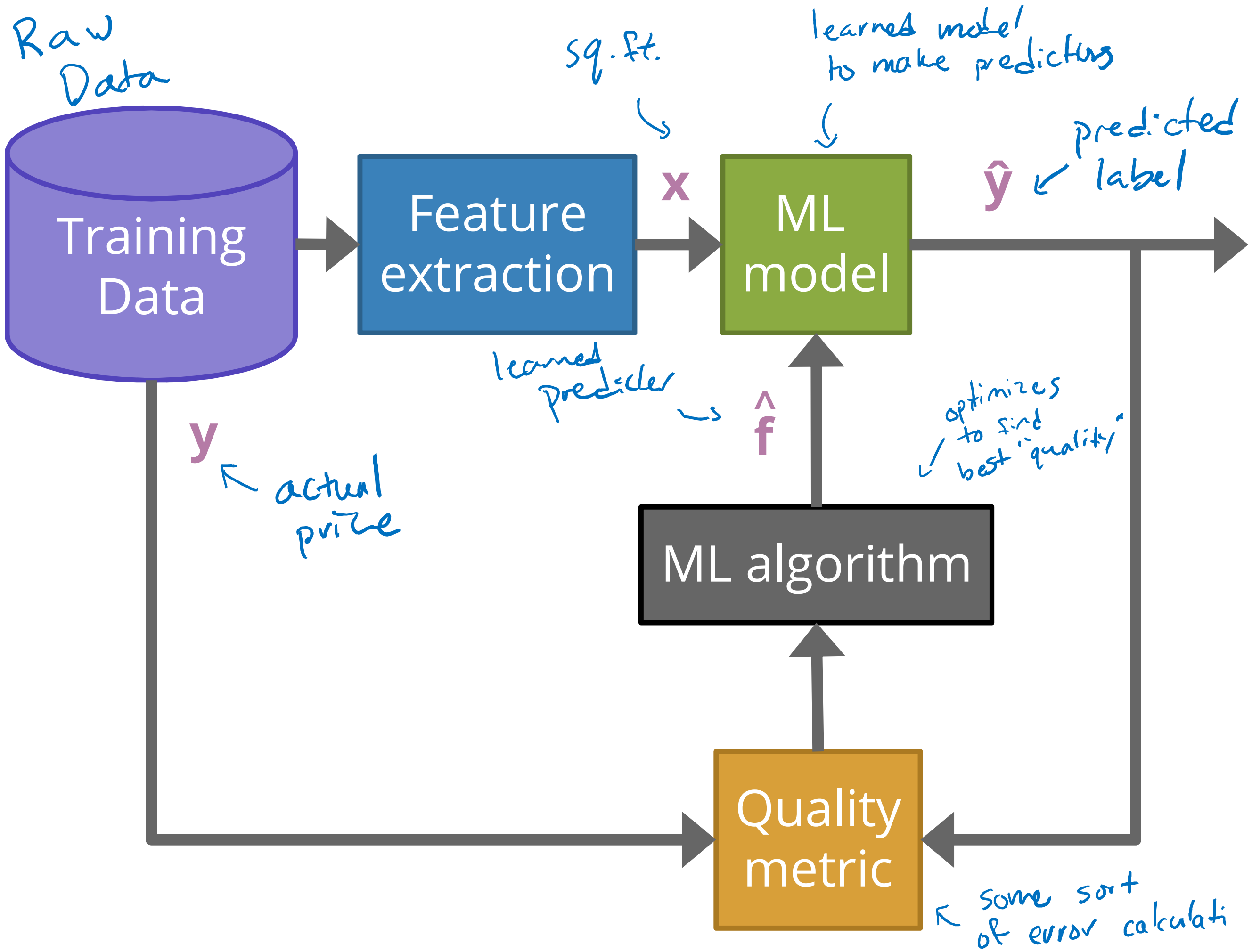
Use machine learning to learn a predictor $\hat{f}$ from the data

For a given input x, predict: $\hat{y} = \hat{f}(x)$

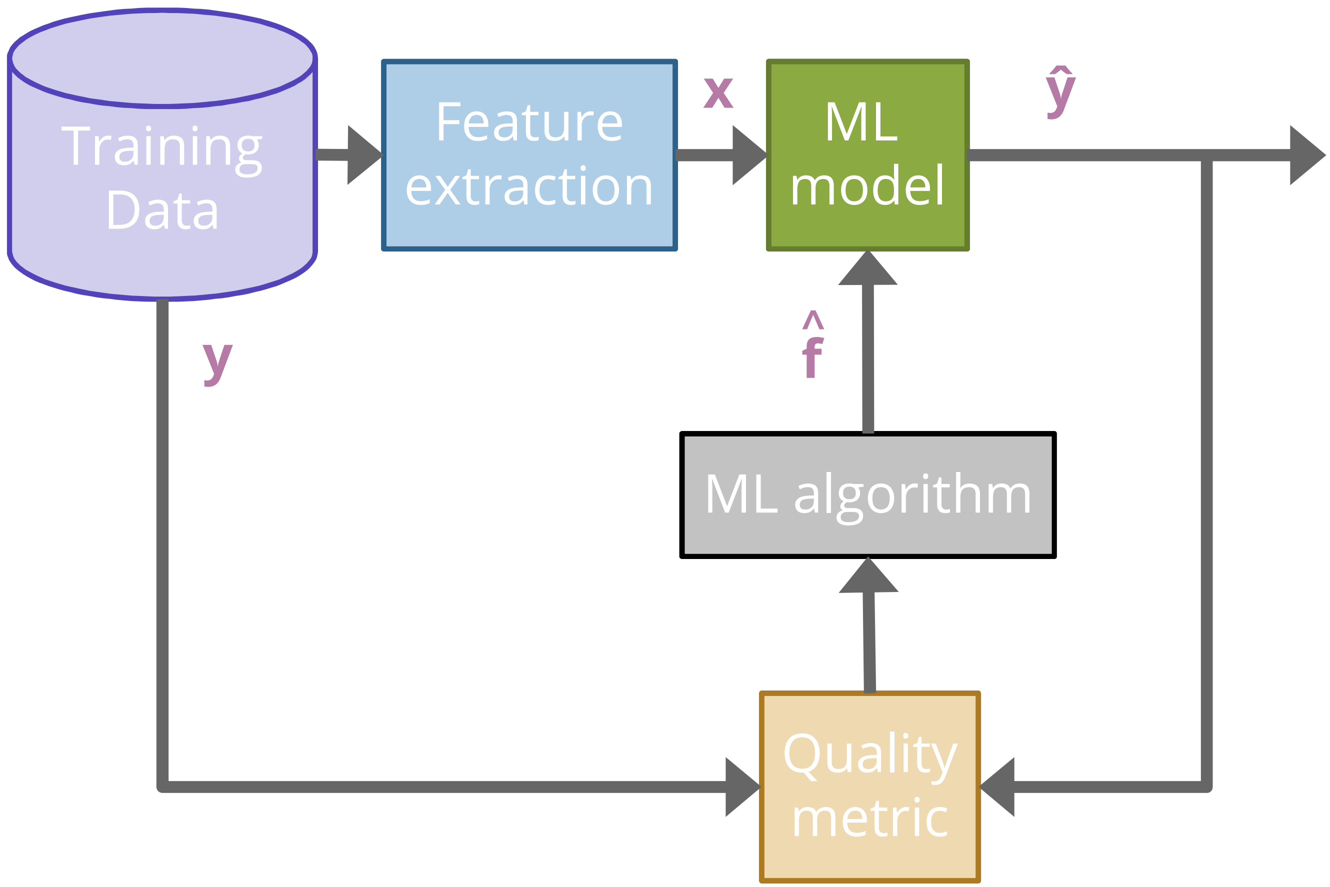


Small error on an example, means we had a good fit for that point

ML Pipeline


Raw Data
Training Data
Feature extraction
sq. ft.
x
learned model to make predictions
ML model
predicted label
ŷ
learned predictor
f̂
y
actual price
optimizes to find best "quality"
ML algorithm
Quality metric
some sort of error calculation

Linear Regression


Training Data
Feature extraction
x
ML model
ŷ
y
f̂
ML algorithm
Quality metric

Linear Regression Model

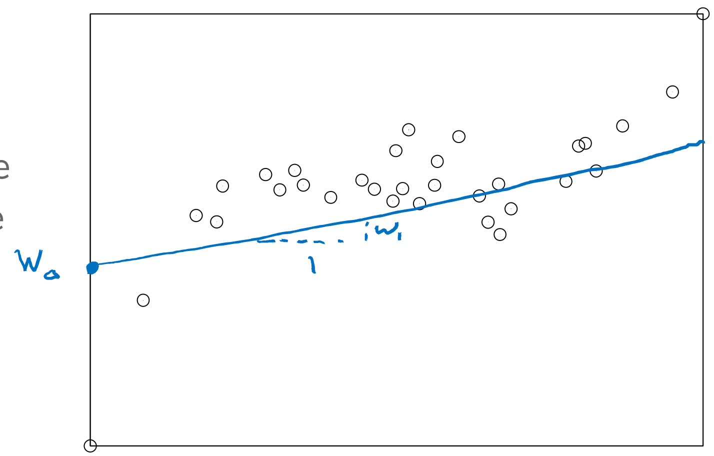
Assume the data is produced by a line.

$$y_i = \underbrace{w_0 + w_1 x_i}_{f(x_i)} + \epsilon_i$$

w_0, w_1 are the **parameters** of our model that need to be learned

- w_0 is the intercept ($ of the land with no house)
- w_1 is the slope ($ increase per increase in sq. ft)

Learn estimates of these parameters $\hat{w}_0, \hat{w}_1$ and use them to predict new value for any input x!



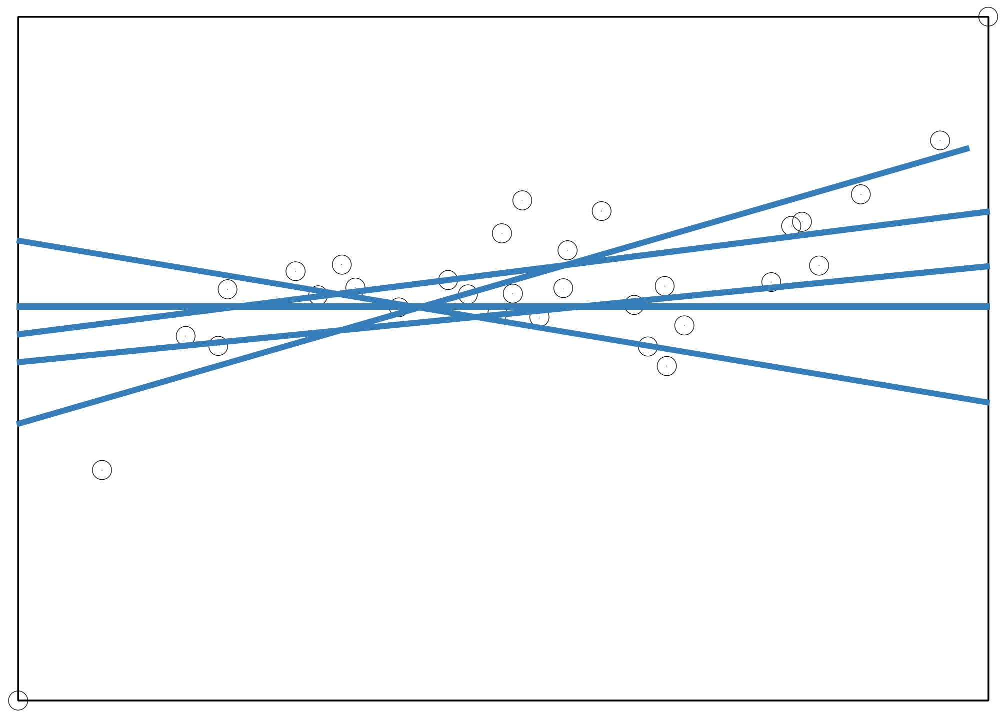
$$\hat{y} = \underbrace{\hat{w}_0 + \hat{w}_1 x}_{\hat{f}(x)}$$

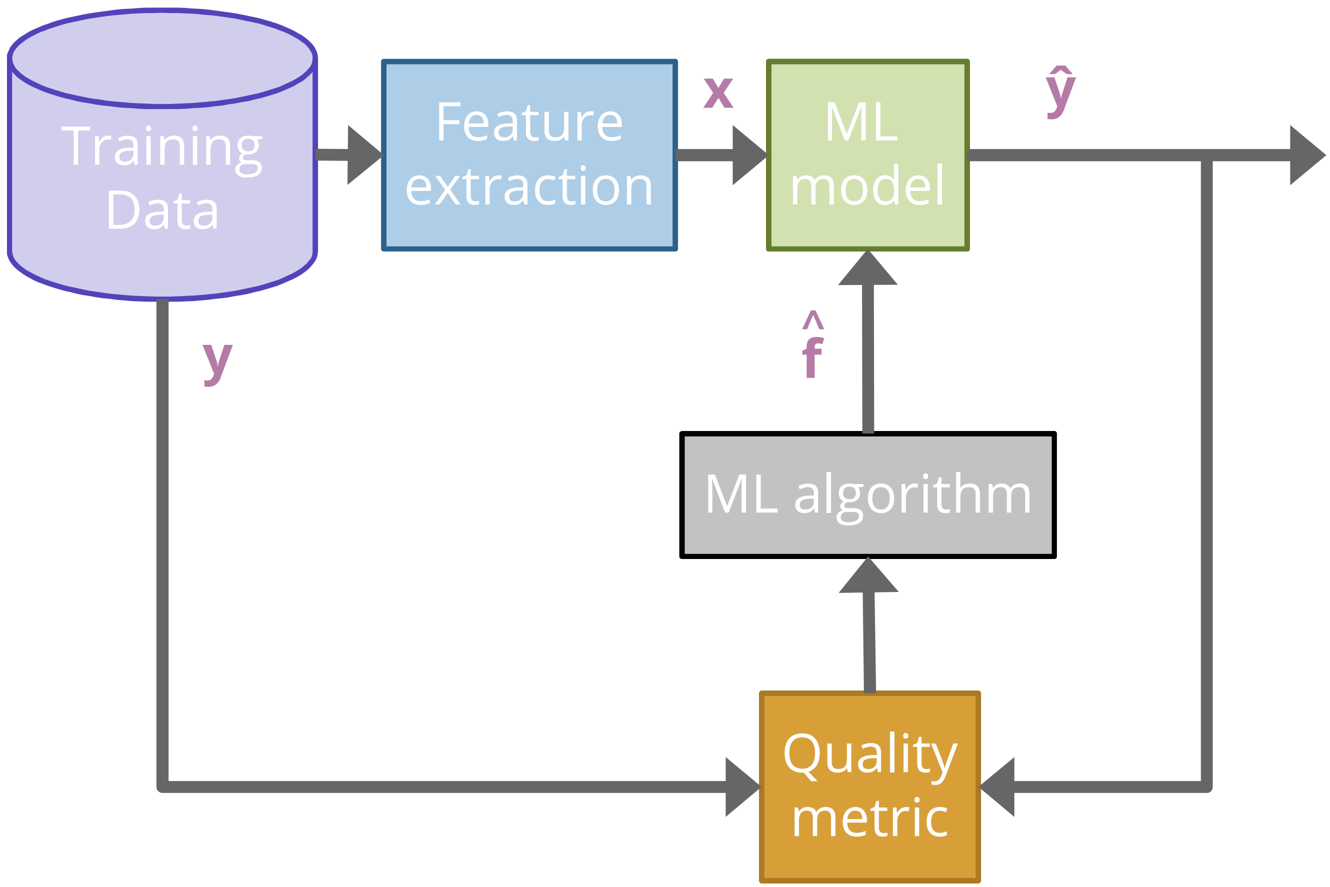
Why don't we add ϵ?

Basic Idea

Try a bunch of different lines and see which one is best!

What does best even mean here?





Training Data
Feature extraction
x
ML model
ŷ
y
f̂
ML algorithm
Quality metric

“Cost” of predictor

Define a “cost” for a particular setting of parameters

- Low cost → Better fit
- Find settings that minimize the cost
- For regression, we will use the error as the cost.
  - Low error = Low cost = Better ~~model~~ predictor (hopefully)

Note: There are other ways we can define cost which will result in different “best” predictors. We will see what these other costs are and how they affect the result.

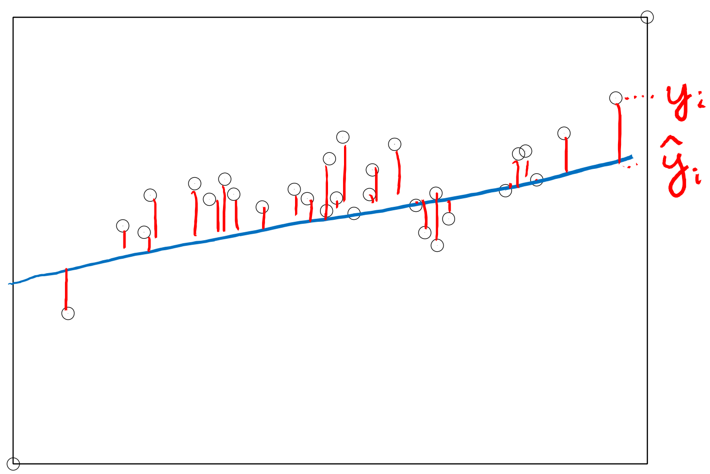
Residual Sum of Squares (RSS)

How to define error? **Residual sum of squares (RSS)**

$$RSS(w_0, w_1) = (y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \dots + (y_n - \hat{y}_n)^2$$

$$= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_i))^2$$



Note

Many people use mean-square error (MSE)

$$MSE(w_0, w_1) = \frac{1}{n} RSS(w_0, w_1)$$

- **Goal**: Get you actively participating in your learning
- Typical Activity
  - Question is posed
  - **Think** (1 min): Think about the question on your own
  - **Pair** (2 min): Talk with your neighbor to discuss question
    - If you arrive at different conclusions, discuss your logic and figure out why you differ!
    - If you arrived at the same conclusion, discuss why the other answers might be wrong!
  - **Share** (1 min): We discuss the conclusions as a class
- During each of the **Think** and **Pair** stages, you will respond to the question via a Poll Everywhere poll
  - The poll will only be open for the last **15** seconds of each of the stage
  - Not worth any points, just here to help you learn!

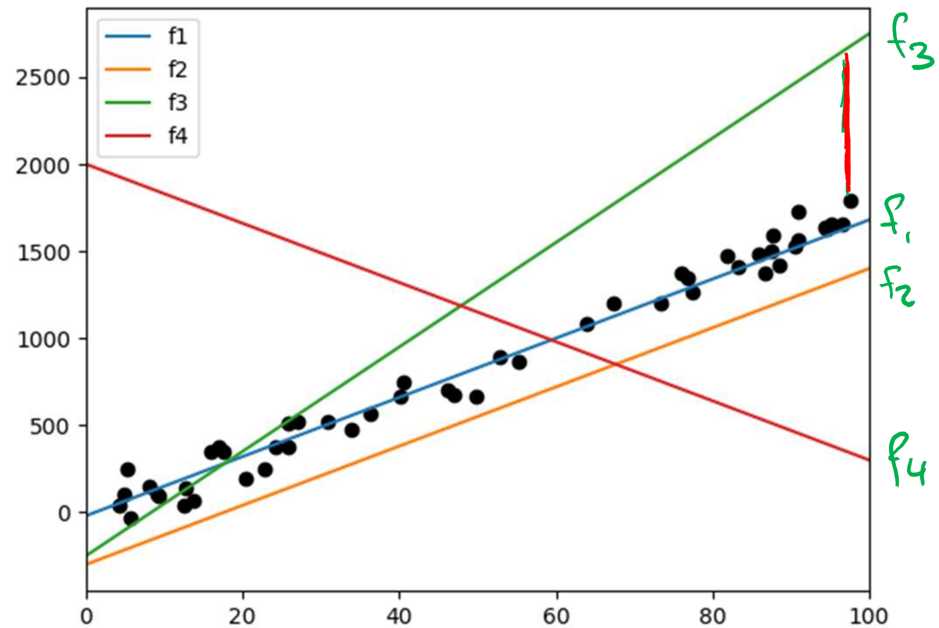
Poll Everywhere

Think

1 min

pollev.com/cse416

Sort the following lines by their RSS on the data, from smallest to largest. (estimate, don't actually compute)



$f_1 < f_2 < f_3 < f_4$

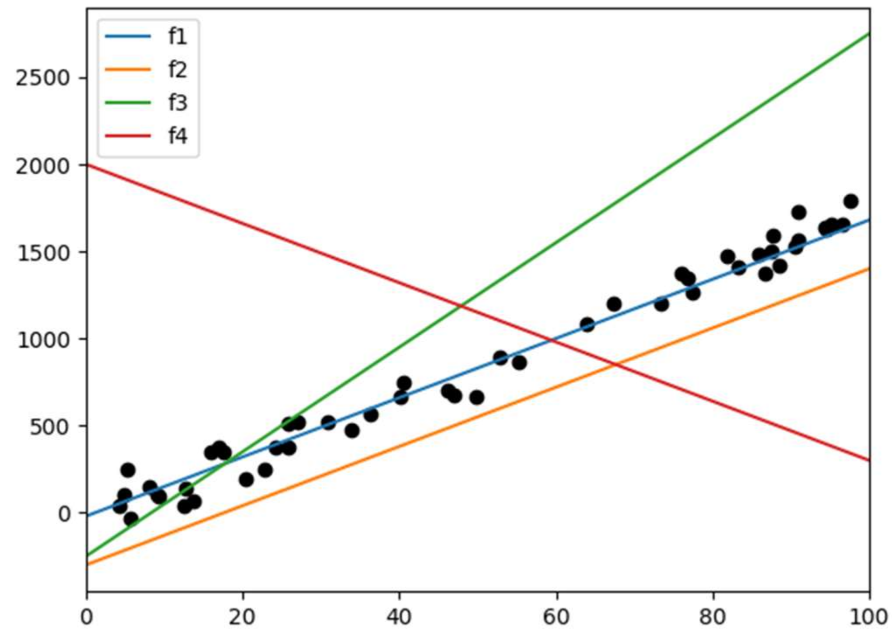
Poll Everywhere

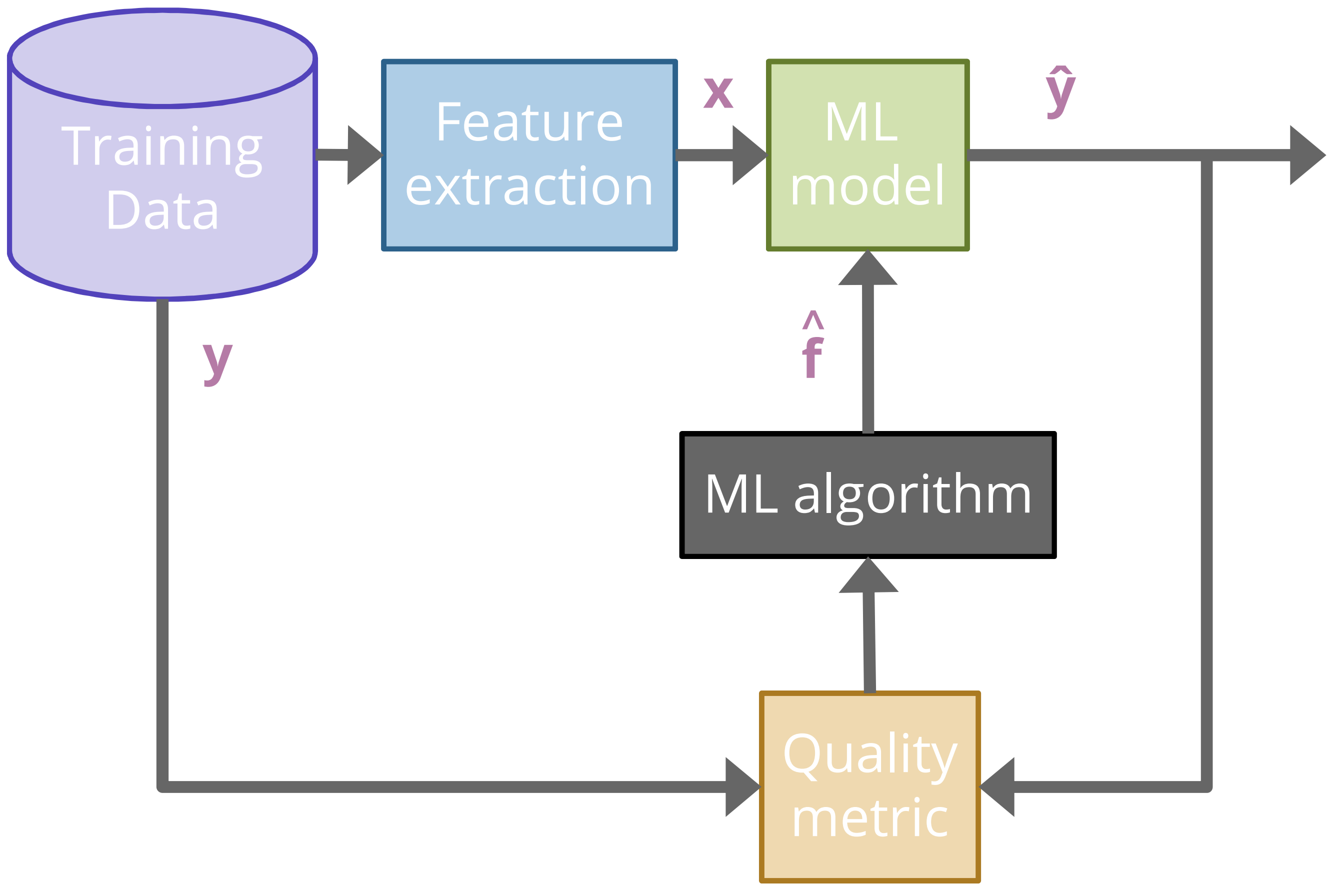
Pair

2 min

pollev.com/cse416

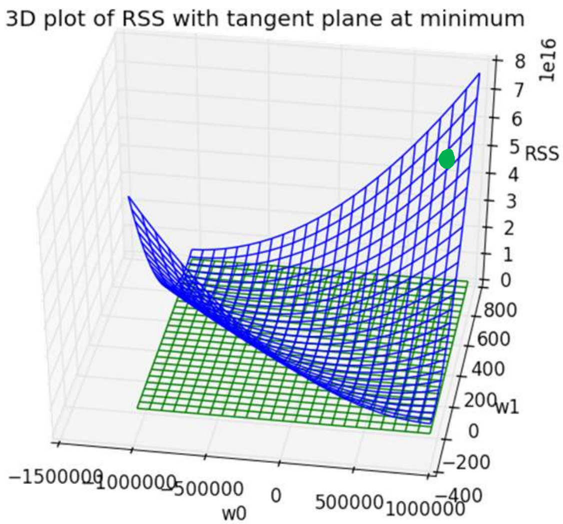
Sort the following lines by their RSS on the data, from smallest to largest. (estimate, don't actually compute)




Training Data
Feature extraction
x
ML model
ŷ
y
f̂
ML algorithm
Quality metric

Minimizing Cost

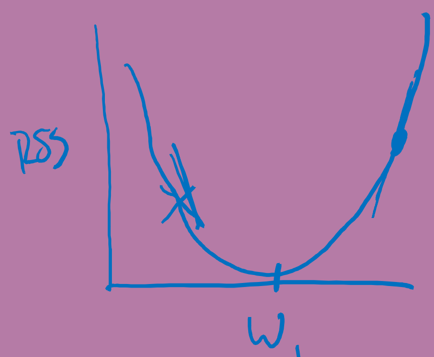
RSS is a function with inputs w_0, w_1, different settings have different RSS for a dataset

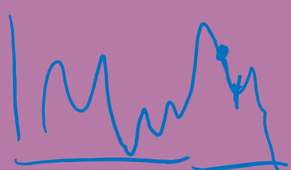


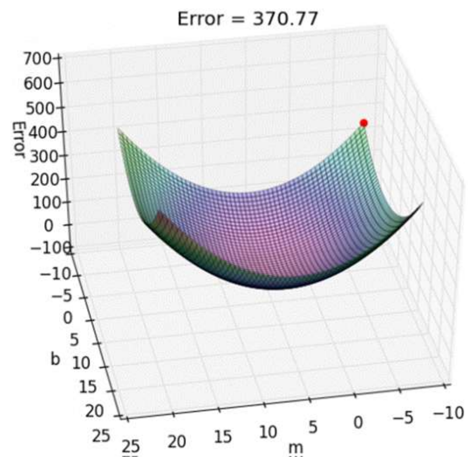
$$\hat{w}_0, \hat{w}_1 = \min_{w_0, w_1} RSS(w_0, w_1)$$
$$= \min_{w_0, w_1} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_i)^2)$$

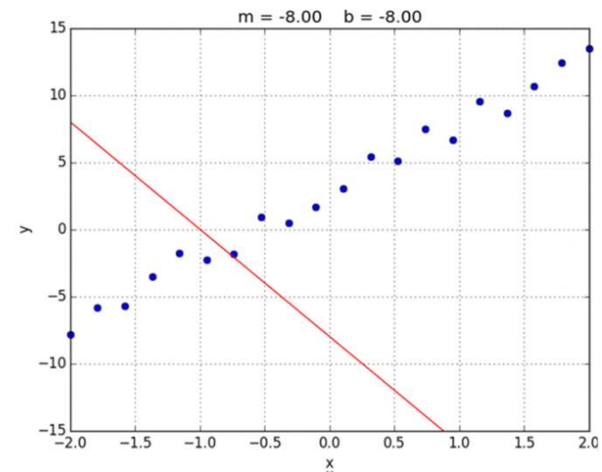
Unfortunately, we can't try it out on all possible settings ☹

Gradient Descent









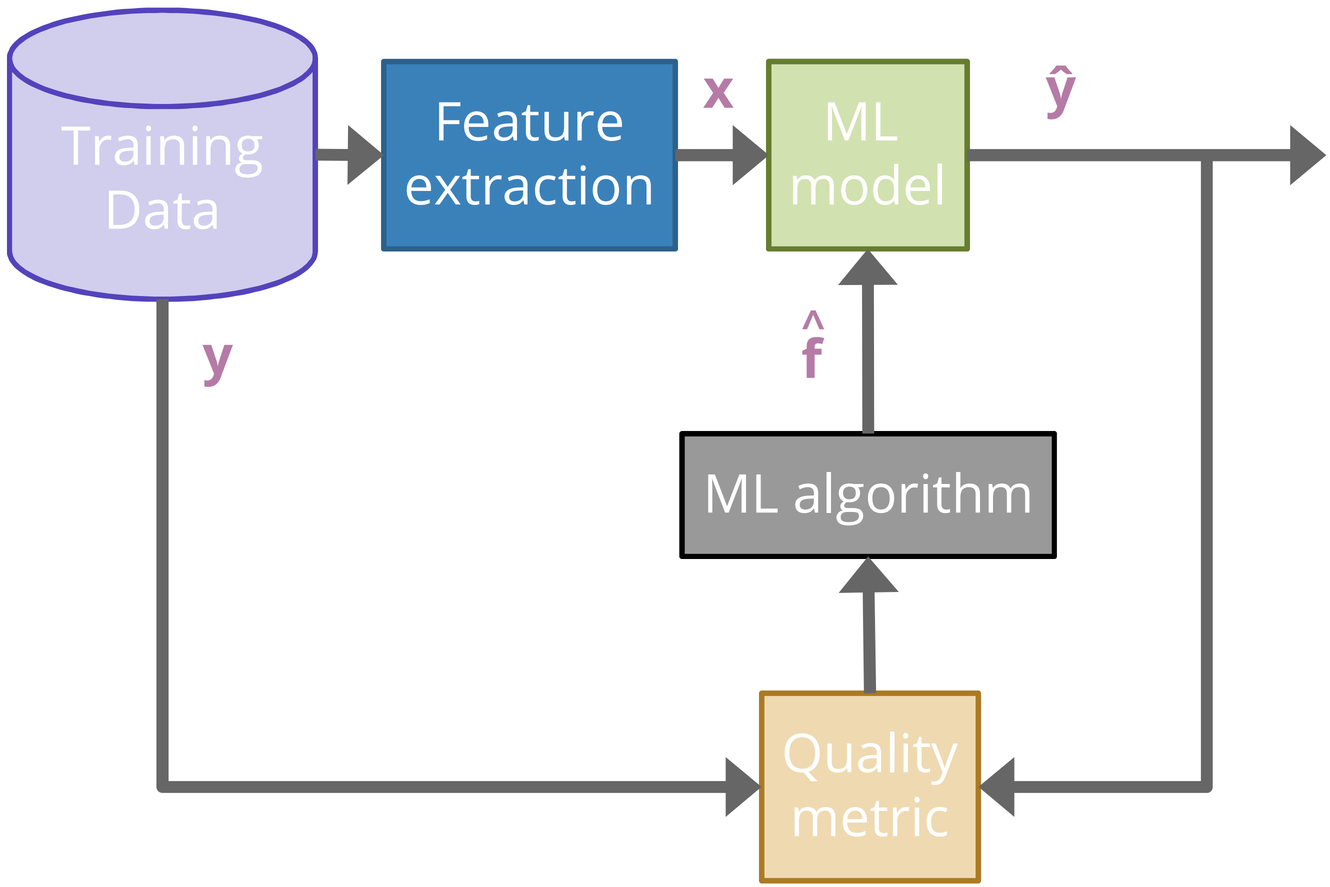
Instead of computing all possible points to find the minimum, just start at one point and "roll" down the hill.
Use the gradient (slope) to determine which direction is down.

start at some (random) point $w^{(0)}$ when $t = 0$

while we haven't converged:

$$w^{(t+1)} = w^{(t)} - \eta \nabla RSS(w^{(t)})$$

Brain Break

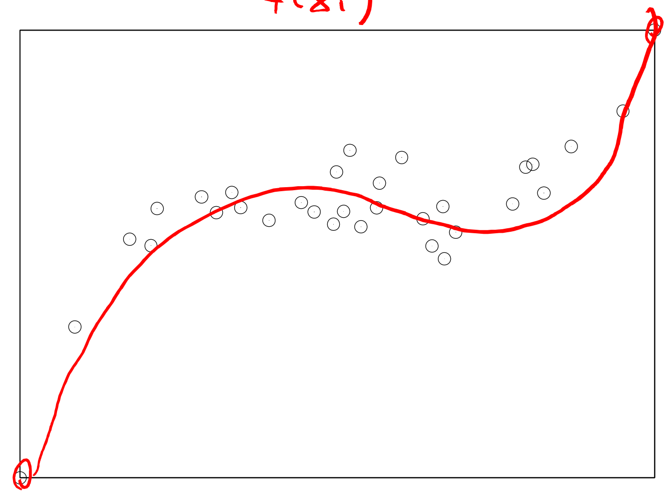

Training
Data
Feature
extraction
x
ML
model
ŷ
y
f̂
ML algorithm
Quality
metric

Higher Order Features

This data doesn't look exactly linear, why are we fitting a line instead of some higher-degree polynomial?

We can! We just have to use a slightly different model!

$$y_i = \underbrace{w_0 + w_1 x_i + w_2 x_i^2 + w_3 x_i^3}_{f(x_i)} + \epsilon_i$$



Polynomial Regression

Model

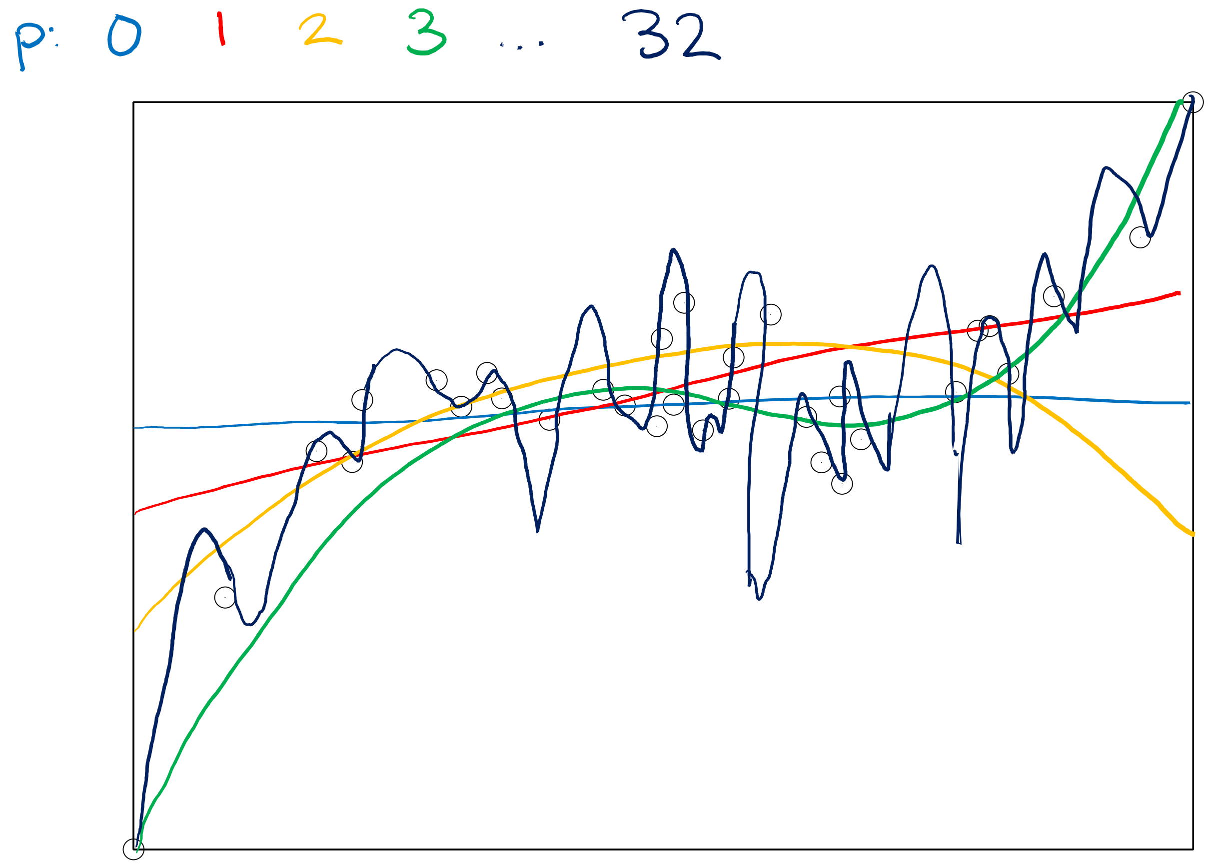
$$y_i = w_0 + w_1 x_i + w_2 x_i + \ldots + w_p x_i^p + \epsilon_i$$

Just like linear regression, but uses more features!

Feature	Value	Parameter
0	1 (constant)	w_0
1	x	w_1
2	x^2	w_2
...	$\ldots$	$\ldots$
p	x^p	w_p

How do you train it? Gradient descent (with more parameters)

Polynomial Regression



How to decide what the right degree? Come back Wednesday!

Features

Features are the values we select or compute from the data inputs to put into our model. **Feature extraction** is the process of turning the data into features.

Model

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \epsilon_i$$

$$= \sum_{j=0}^{D} w_j h_j(x_i) + \epsilon_i$$

Feature	Value	Parameter
0	$h_0(x)$ often 1 (constant)	w_0
1	$h_1(x)$ $x^2 + x$	w_1
2	$h_2(x)$ e^x	w_2
…	…	…
D	$h_D(x)$ $\log(x)$	w_D

Adding Other Inputs

Generally we are given a data table of values we might look at that include more than one value per house.

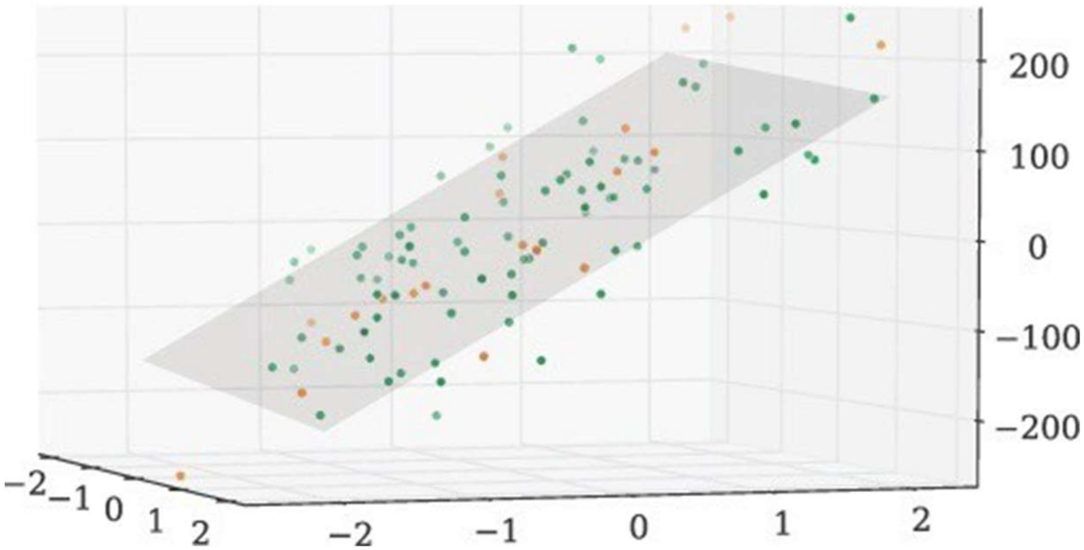
- Each row is a single house.
- Each column (except Value) is a data input.

sq. ft.	# bathrooms	owner's age	…	value
1400	3	47	…	70,800
700	3	19	…	65,000
…	…	…	…	…
1250	2	36	…	100,000

More Inputs - Visually

Adding more features to the model allows for more complex relationships to be learned

$$y_i = w_0 + w_1(sq.ft.) + w_2(\#\ bathrooms) + \epsilon_i$$



Coefficients tell us the rate of change if all other features are constant

Notation

Important: Distinction is the difference between a *data input* and a *feature*.

- Data inputs are columns of the raw data
- Features are the values (possibly transformed) for the model (done after our feature extraction $h(x)$)

Data Input: $x_i = (x_i[1], x_i[2], \dots, x_i[d])$

Output: y_i

- x_i is the i^{th} row
- $x_i[j]$ is the i^{th} row's j^{th} data input
- $h_j(x_i)$ is the j^{th} feature of the i^{th} row

Features

You can use anything you want as features and include as many of them as you want!

Generally, more features means a more complex model. This might not always be a good thing!

Choosing good features is a bit of an art.

Feature	Value	Parameter
0	1 (constant)	w_0
1	$h_1(x)$ … $x[1]$ = sq. ft.	w_1
2	$h_2(x)$ … $x[2]$ = # bath	w_2
…	…	…
D	$h_D(x)$ … like $\log(x[7]) * x[2]$	w_{D}

Linear Regression Recap

Dataset

$\{(x_i, y_i)\}_{i=1}^{n}$ where $x \in \mathbb{R}^d$, $y \in \mathbb{R}$

Feature Extraction

$h(x)\colon \mathbb{R}^d \to \mathbb{R}^D$

$h(x) = (h_0(x), h_1(x), \ldots, h_D(x))$

Regression Model

$$y = f(x) + \epsilon$$
$$= \sum_{j=0}^{D} w_j h_j(x) + \epsilon$$
$$= w^T h(x) + \epsilon$$

Quality Metric

$$RSS(w) = \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Predictor

$$\hat{w} = \min_{w} RSS(w)$$

ML Algorithm

Optimized using Gradient Descent

Prediction

$$\hat{y} = \hat{w}^T h(x)$$