Lecture Start: 2:20PM

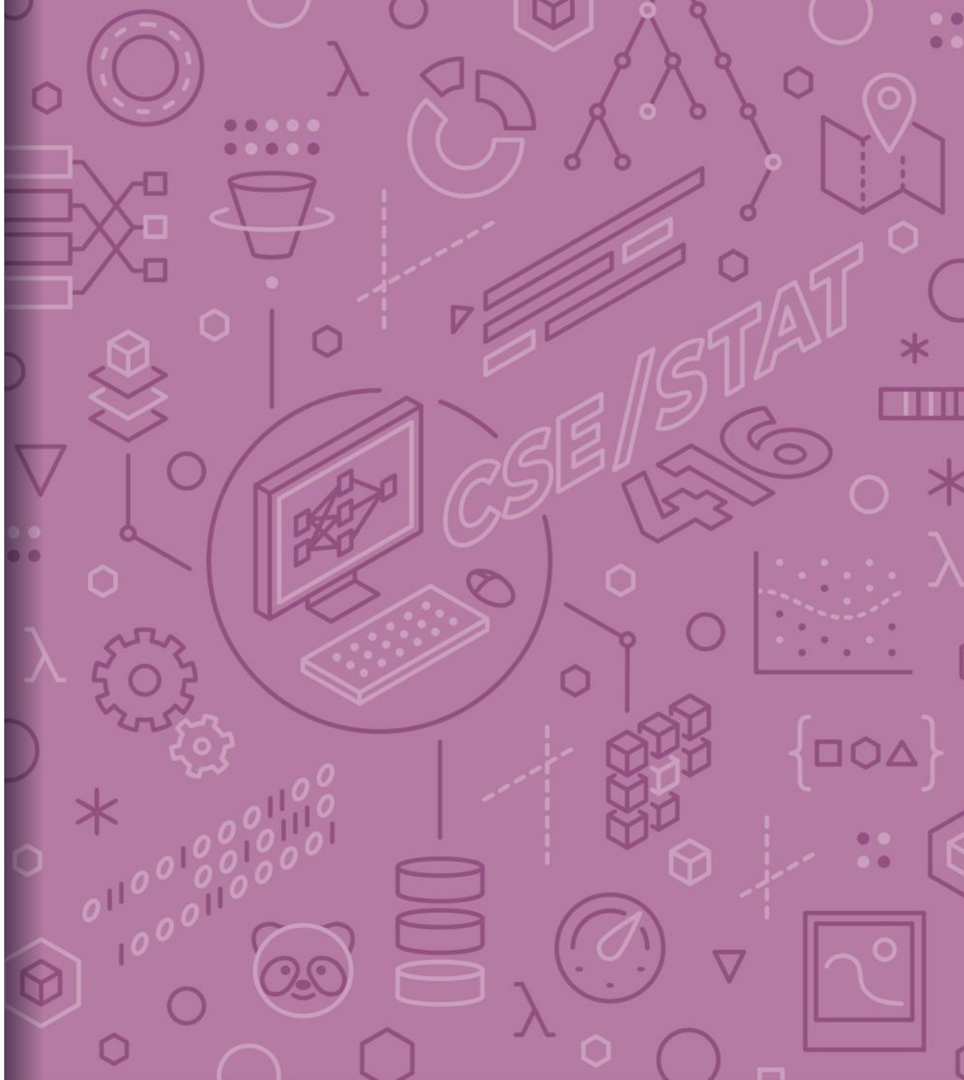
CSE/STAT 416

Introduction + Regression

Amal Nanavati
University of Washington

June 22, 2022

Adapted from Hunter Schafer's Slides



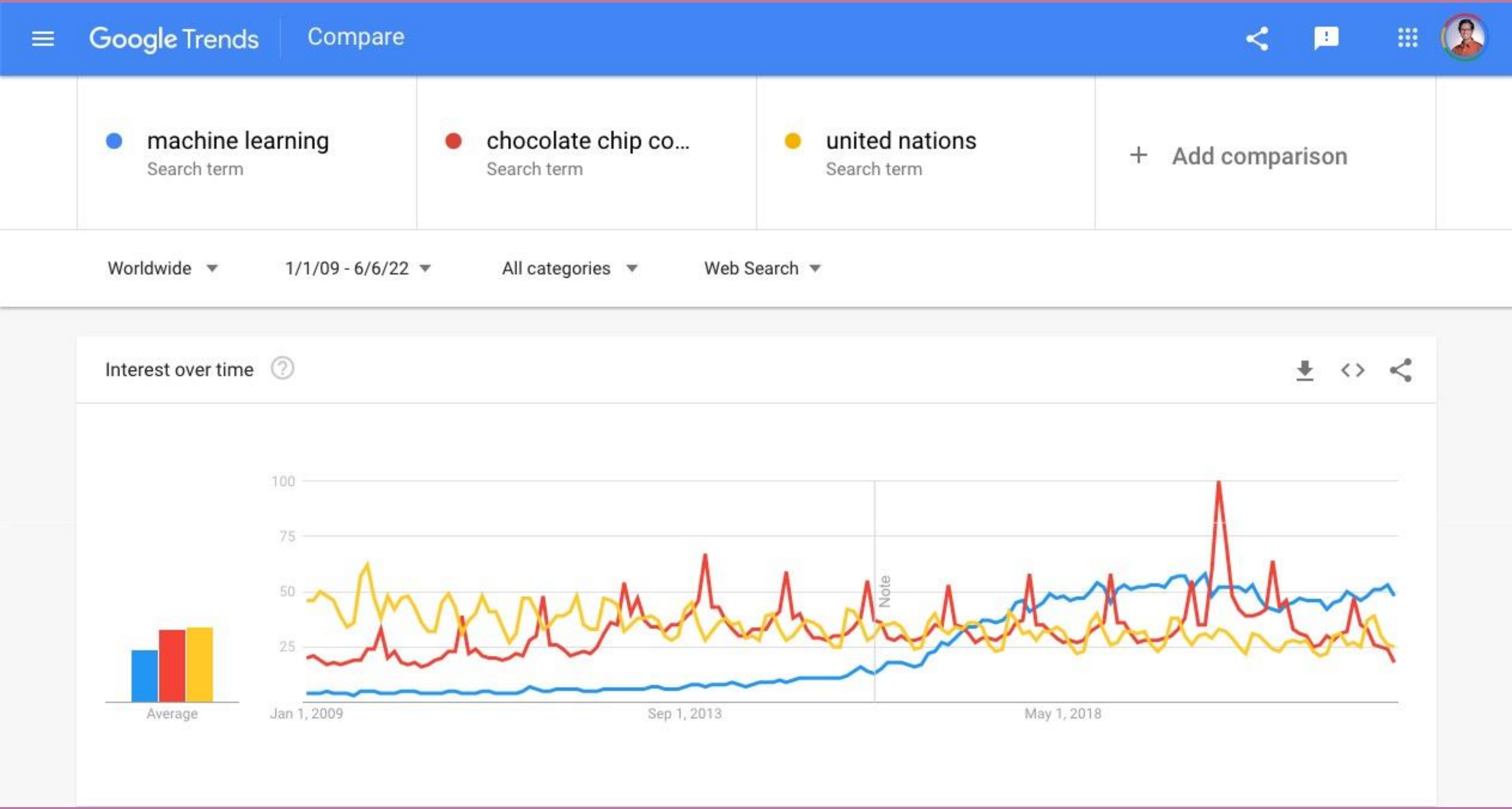
Land Acknowledgement

The University of Washington acknowledges the Coast Salish peoples of this land, the land which touches the shared waters of all tribes and bands within the Duwamish, Puyallup, Suquamish, Tulalip and Muckleshoot nations.

Actions:

- Sign the petition for federal recognition of the Duwamish Tribe: https://www.change.org/p/federal-recognition-for-the-duwamish-tribe
- Visit and support the Duwamish Longhouse & Cultural Center: https://www.duwamishtribe.org/events-1

Machine Learning is changing the world.



Google Trends
Compare
machine learning
Search term
chocolate chip co...
Search term
united nations
Search term
+ Add comparison
Worldwide
1/1/09 - 6/6/22
All categories
Web Search
Interest over time
100
75
50
25
Note
Average
Jan 1, 2009
Sep 1, 2013
May 1, 2018

It's Everywhere!

Retail

PageRank
Search

Coupons

Movie Distribution

Campaigning

Real Estate

Legal Advice

Advertising

Human Resources

Dating

Networking

CRM

Music

Wearables

Disruptive companies differentiated by

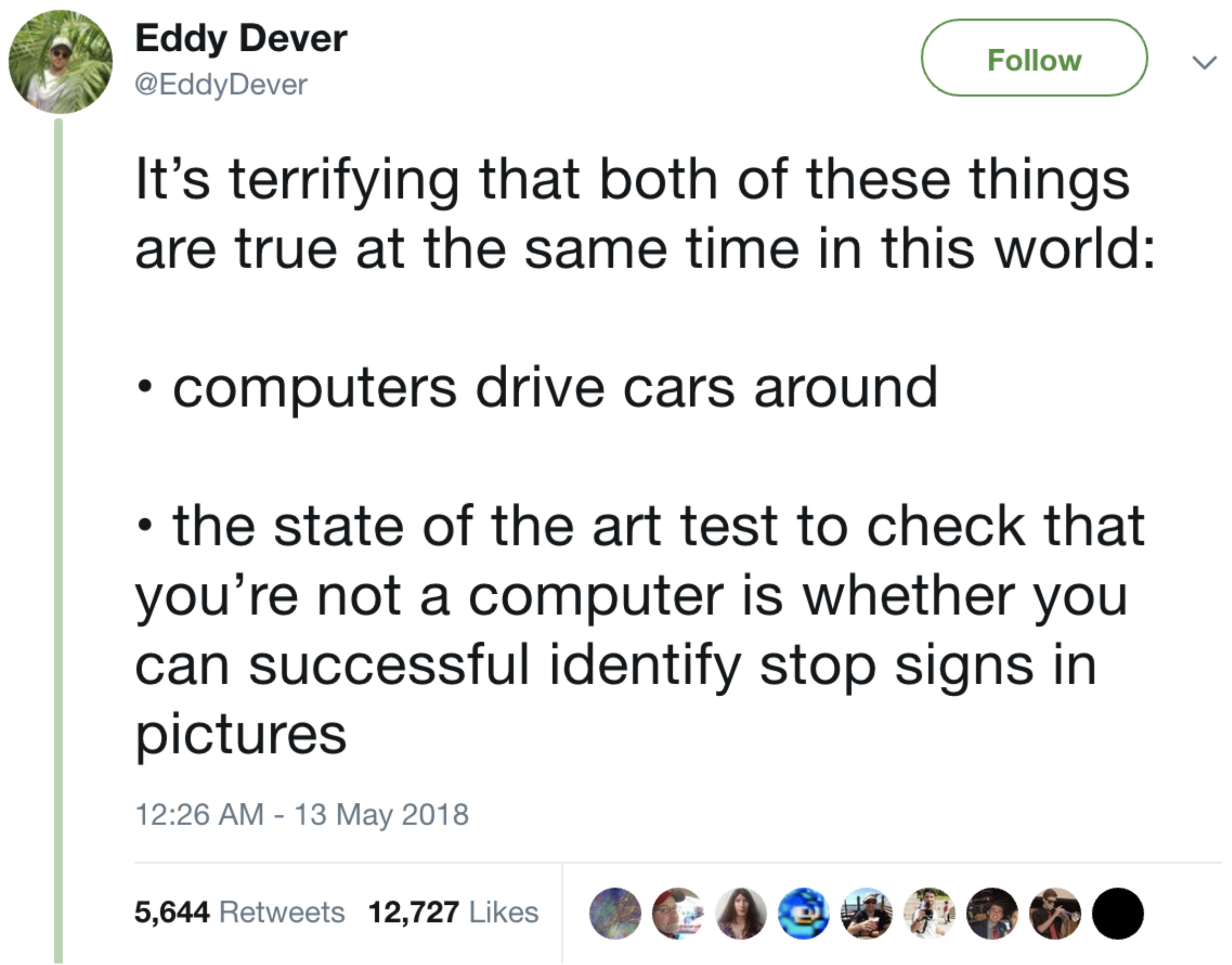
INTELLIGENT APPLICATIONS

using

Machine Learning

It’s Everywhere…

It’s Everywhere...



It’s Everywhere…

Object Detection

It’s Everywhere…

Object Detection

It’s Everywhere…

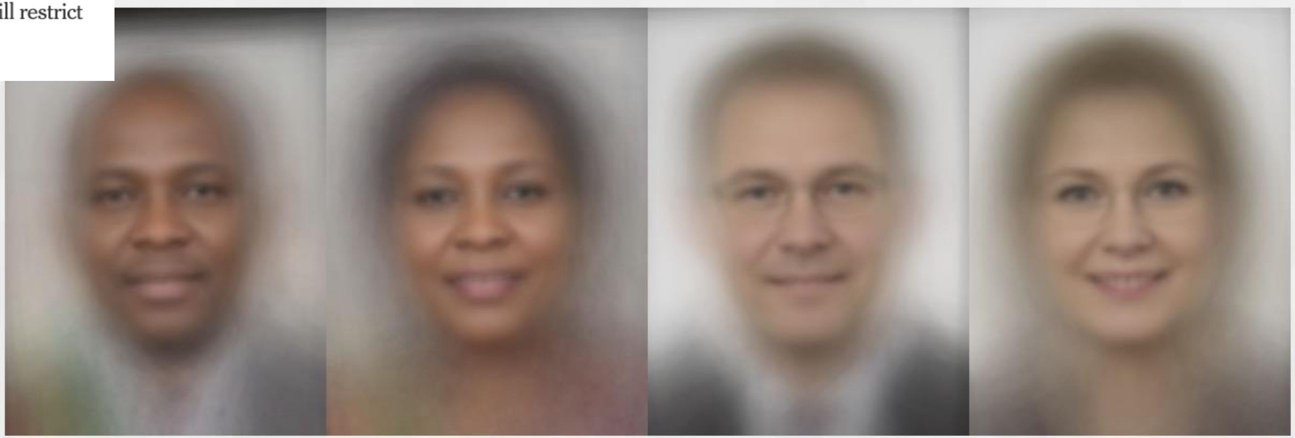
Face Detection

Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%

Microsoft Plans to Eliminate Face Analysis Tools in Push for ‘Responsible A.I.’

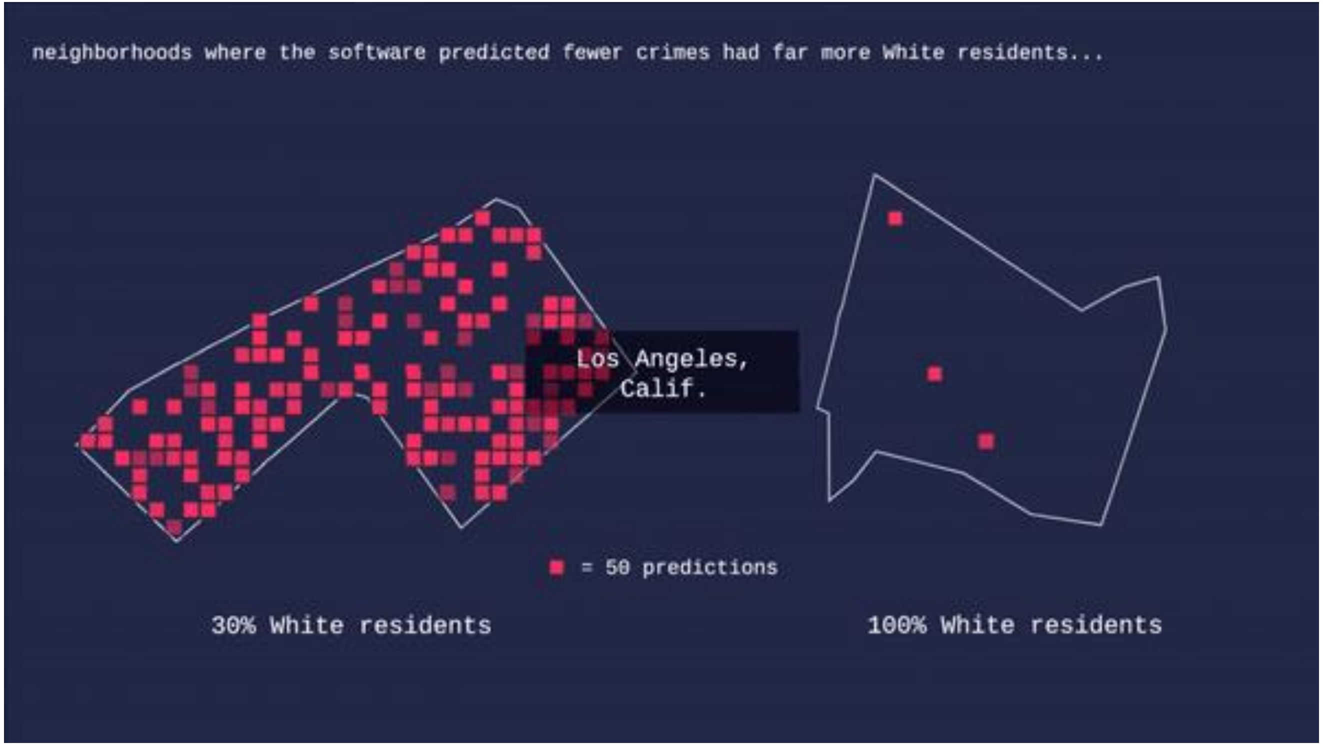
The technology giant will stop offering automated tools that predict a person’s gender, age and emotional state and will restrict the use of its facial recognition tool.

June 21, 2022, 12:02 p.m. ET



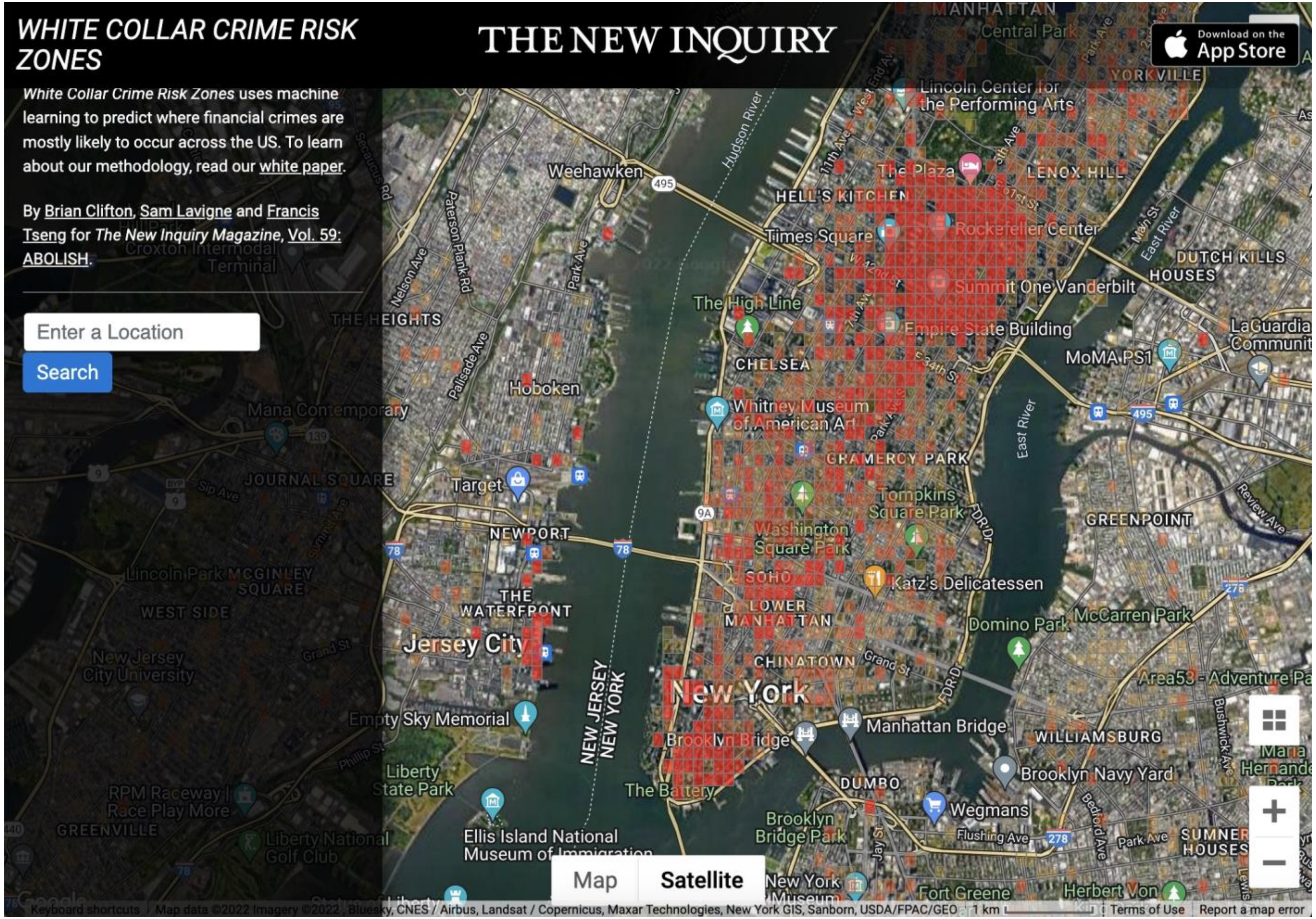
It’s Everywhere…

Predictive Policing



It's Everywhere...

Predictive Policing



What is Machine Learning?

Generically (and vaguely)

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

Tom Mitchell (1998): a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Taxonomy of Machine Learning (Based on tasks)

SUPERVISED LEARNING

UNSUPERVISED LEARNING

REINFORCEMENT LEARNING

Taxonomy of Machine Learning (Based on tasks)

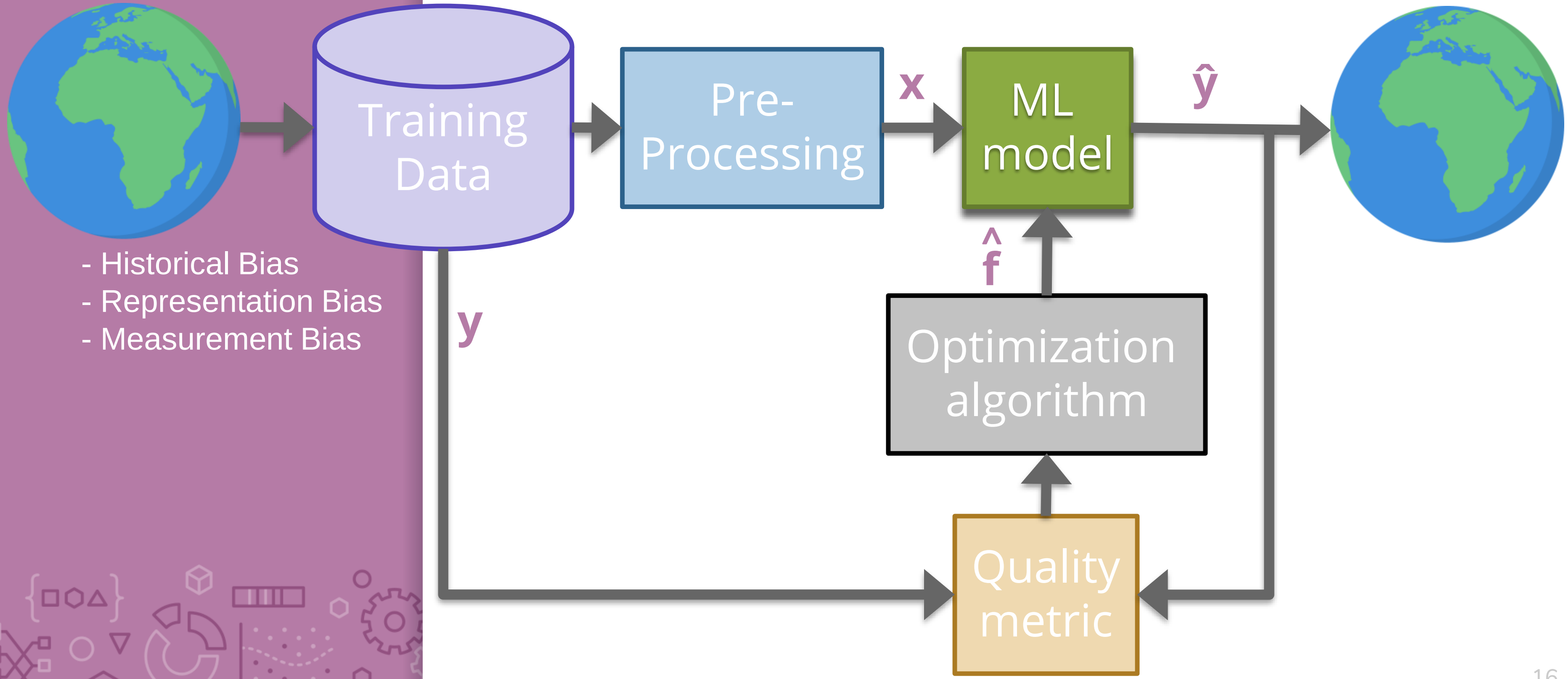
1. **Supervised Learning**
   - Training data is labeled, where inputs are paired with correct outputs
   - Infers a mapping function from the inputs to outputs
   - ***Examples:*** *image classification, stock price predictions*

2. **Unsupervised Learning**
   - Analyze and cluster unlabeled datasets
   - Discover patterns or data categorization without the need for human intervention
   - ***Examples:*** *DNA clustering, anomaly detection*

3. **Reinforcement Learning**
   - Not covered in this class (you can learn this in CSE 415 / 473 (Introduction to Artificial Intelligence)
   - Agents learn the optimal behaviors to obtain maximum reward through interactions with the environment and observations of how they responds.

ML Pipeline

- Deployment Bias



- Historical Bias
- Representation Bias
- Measurement Bias

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. Deep Learning
    - Recognizing objects in images
4. Clustering & Similarity
    - Find similar news articles
5. Recommender Systems
    - Given past purchases, what do we recommend to you?

Course Topics

Models

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

Algorithms

- *Gradient descent*
- Boosting
- K-means

Concepts

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

ML Course Landscape

CSE 446

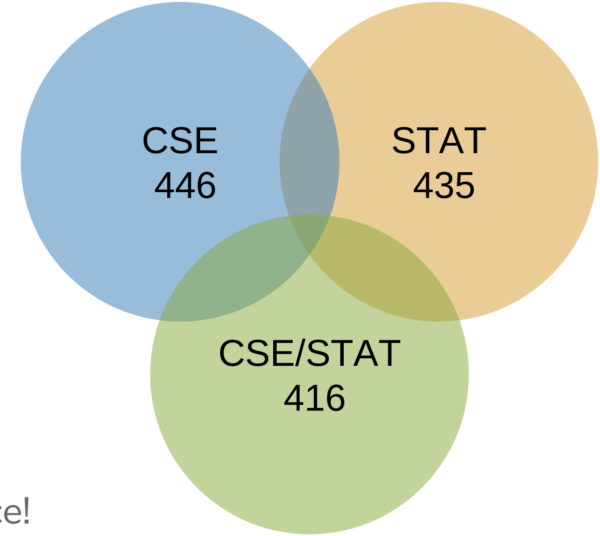
- CSE majors
- Very mathematically demanding course

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…

- Minimizing pre-requisite knowledge
- Allowing you to understand the ML concepts in an intuitive way.
- Focus on important ideas, avoid getting bogged down by math
- Exposed to Python, libraries and infrastructure to program ML problems
- Learn concepts in case studies

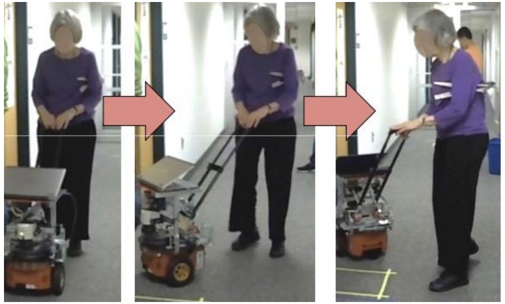
Does not mean course isn't fast paced! There are a lot of concepts to cover!

Course Logistics

Who am I?

Amal Nanavati
Instructor
he/they
amaln@cs

- Background
  - 4th-year PhD student in CSE
  - **Research**: human-robot interactions, assistive technology
  - **Teaching**: CSE 416 Head TA, 7th-12th grade CS & ML
  - **Hobbies**: hiking, biking, board games, cooking.



- Contact
  - Course Content + Logistics: [EdStem]
  - Personal Matters: amaln@cs.washington.edu

Who are the TAs?

Tanmay Shah
Head TA
he/him
tanmay@cs

Josh Gardner
he/him
jpgard@cs

Wuwei Zhang
she/her
wz86@cs

Karman Singh
he/him
shubhs2@cs

Max Bi
he/him
mbi6245@uw

Poll Everywhere

Think

30 seconds

pollev.com/cs416

On your phone / laptop

What is/are your major(s)?

Learning Reflections

Recap broader context of the past week.

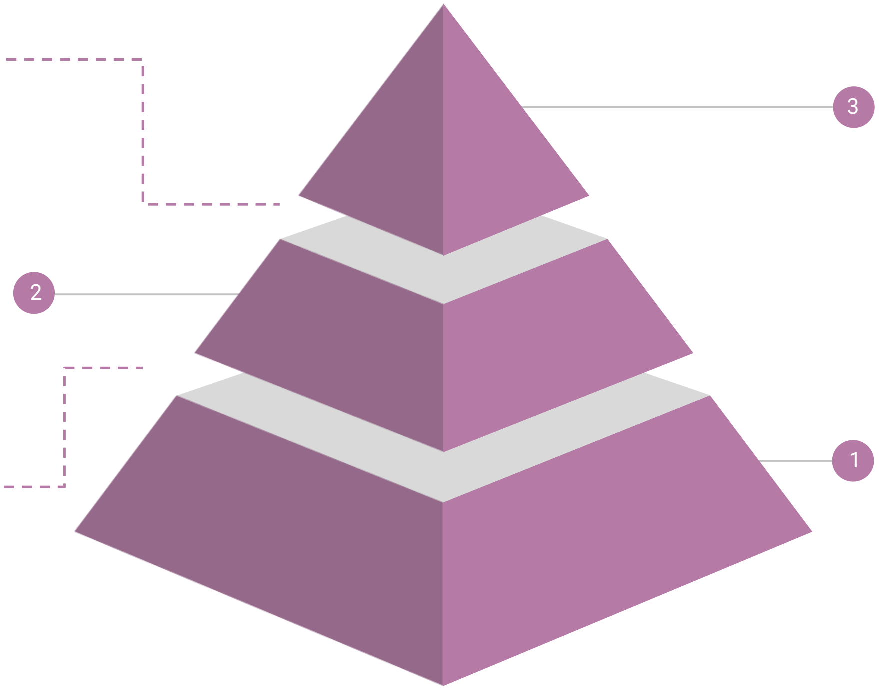
Sections

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

The emphasis is still on you **learning by doing.**

Checkpoints

Test your understanding of the last concept



Homeworks

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.**

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!

Mon	Tue	Wed	Thur	Fri
Lecture		Lecture	Section	
- Previous Checkpoint **Due 1:50pm PST**	- **Homework due 11:59PM PST**	- Previous Checkpoint **Due 1:50PM PST** - **Homework released**		- Learning Refl. **Due**

- We happen to not track attendance, but it is expected that you attend lecture and section
- Panopto for live lecture recordings and weekly section recordings

Assessments

- **Weekly Homework Assignments (65%)** ‼️‼️
  - **Number**: 7 (equally weighted)
  - Each Assignment has two parts that contribute to your grade
    - Programming (40%) – autograded, you receive scores right away
    - Conceptual (25%) – you receive scores after the deadline
- **Checkpoints (10%)**
  - Designed to be doable (30 mins) if you follow each previous lecture
  - **Number:** 14 (each lecture, drop 2)
- **Learning Reflections (10%)**
  - **Number:** 8 (each week, drop 1)
- **Final Exam (15%)**
  - Take-home (intended to take 2 hrs)
  - **Released:** Wed 8/17 9AM
  - **Due:** Thurs 8/18 11:59PM

- Unless otherwise stated, all work must be done individually

Learning Reflections

- Summary: 2-4 sentences
- List of key concepts (≥ 5)
- Uncertainties

Summary

This week we learned about deep learning and neural networks. Unlike most of the machine learning algorithms that we learned about this quarter, deep learning / neural networks is a bit more new and more complicated in a sense that people have been giving it more attention in recent years and there are not yet any definitive rules to how you have to create models to make them perform the best. At a high level, it gets inputs and can have one or more hidden layers where weights are computed, then spits out the outputs at the end.

Concepts

- Neural networks (NN)
  - Each neuron will have weights and values, and the summation of those values (or with an activation function for non-linear) will go into each neuron in the hidden layer.
  - Each neuron in the hidden layer can have a bias term which is subtracted from its input value.
  - Then based on a certain set of rules for the output, its output will go to the neurons in the next layer, which can be another hidden layer or the output.
  - You keep doing this until you get the outputs.
  - There are no set rules yet as to what is better practice for all cases, so there are a lot of hyperparameters like number of hidden layers or hidden neurons, the activation function, learning rate for gradient descent, batch size, epochs to train, etc.
- Convolutional NN: the idea is to reduce the number of weights that needs to be learned in the NN by using kernels and pools to reduce the number of features.
  - Convolution with kernels
    - A kernel can “slide” across the original input data, generally to find sum of element-wise product between the kernel and overlapping part of the image. The output of this is a smaller version of the original image.
    - You can decide the size of kernel, padding size and values, and stride values
  - Pools
    - Similar to a kernel, but instead of using all the values to be part of the output (through summation, products, etc), just pick a similar value like the min, max, average, median, etc. Typical to use the max pool.
  - Then use a combination of kernel convolution and pooling to get a smaller input for the NN.

Uncertainties

It seems neural networks is a less developed field compared to the other machine learning algorithms that we have learned about this quarter. One thing I wonder is, how do you know it is ready to implement in the world for real life applications, especially when it can have high error rates with even the slightest change to the input or model?

CS 416 Learning Reflection- Week 9

Summary: This week's lectures are the introduction to neural networks. The importance and strength of deep learning for different applications are discussed. The convolutional neural networks (CNN) are detailed with help of object recognition in an image. The regression and classification methods using neural networks and hyper-parameter tuning for these networks are explained.

Concepts:

Neural Networks: It is a simulated representation of neurons in brains having a linear expression using the sum of weights multiplied by inputs. It is then followed either by a linear or nonlinear function for activation. A series of layers are formed using these neurons and complex functions are learned. Almost any function could be learned using neural networks but they require sufficient computational resources and can be executed parallelly using GPUs.

Activation Functions: Various activation functions like the sigmoid, hyperbolic tangent (tanh), Rectified linear unit (ReLU) and soft plus are shown with their respective advantages. ReLU being popular has fragile traits during training. Sigmoids are now being used only at outputs for determining class probabilities and are called softmax.

Overfitting: NN's tend to overfit and it can be addressed by regularizing using dropout conditions and sometimes stopping early or using less number of layers when not demanded by the problem.

Backpropagation: It is a popular algorithm to learn coefficients using the predictions obtained from the forward pass. The error using metrics like RSS and cross-entropy loss is used to adjust the weights for better prediction later.

Hyperparameters: They are the most important after forming an architecture for a machine learning problem, especially NN's. The number of epochs, activation functions, layers, batch size, and learning rate are all important in their own aspect. They have to be carefully chosen and later optimized.

CNN: Primarily being used for images, convolutions help to reduce the number of inputs by combing information about local pixels. Typically a predefined kernel is swept over the input using a stride length and the values of the weights are learned in later epochs. Many times a pooling layer is used to downsample channels separately. Only a final fully connected layer is used to form a neural network just before the output. Transfer learning these days can help in faster initialization and reduce training times for common data available publicly. CNN's are sensitive to transformation or external noise added and can get confused.

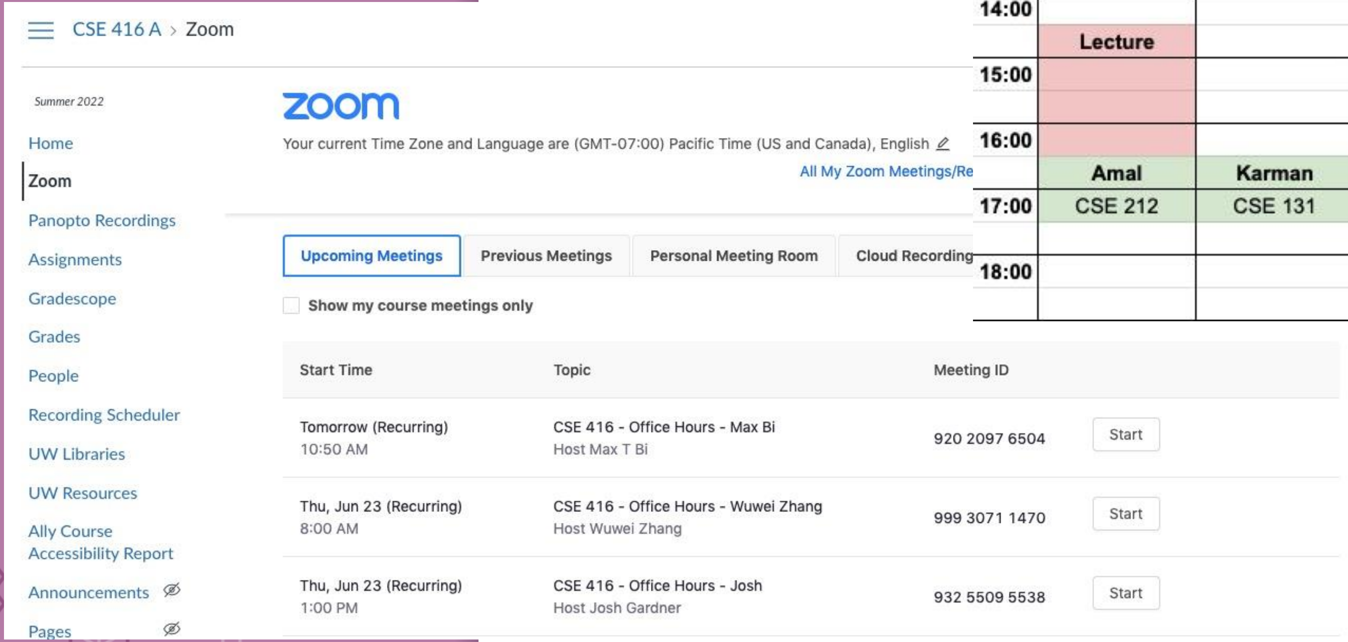
Uncertainties: How to handle various channels of input and does it always have to be an image for CNN?

Homework Logistics

- **Late Days**
  - 6 Late Days for the whole quarter.
  - Can use up to 2 Late Days per homework (except last one!)
  - Each Late Day used after 6 results in a -10% on that assignment
  - No late days on Learning Reflections.
  - Checkpoints can be turned in up to a week later for 50% credit.
- **Collaboration**
  - You are encouraged to discuss assignments and concepts **at a high level**
    - If you are reading off parts of your solution, it's likely not high level
      - Discuss process, not answers!
  - All code and answers submitted must be **<u>your own</u>**
    - We are running a code similarity-checker.
- **Turn In**
  - Homeworks (Conceptual) and Learning Reflections are turned in on **Gradescope**
  - Homeworks (Programming) and Checkpoints are turned in on **EdStem**

Getting Help

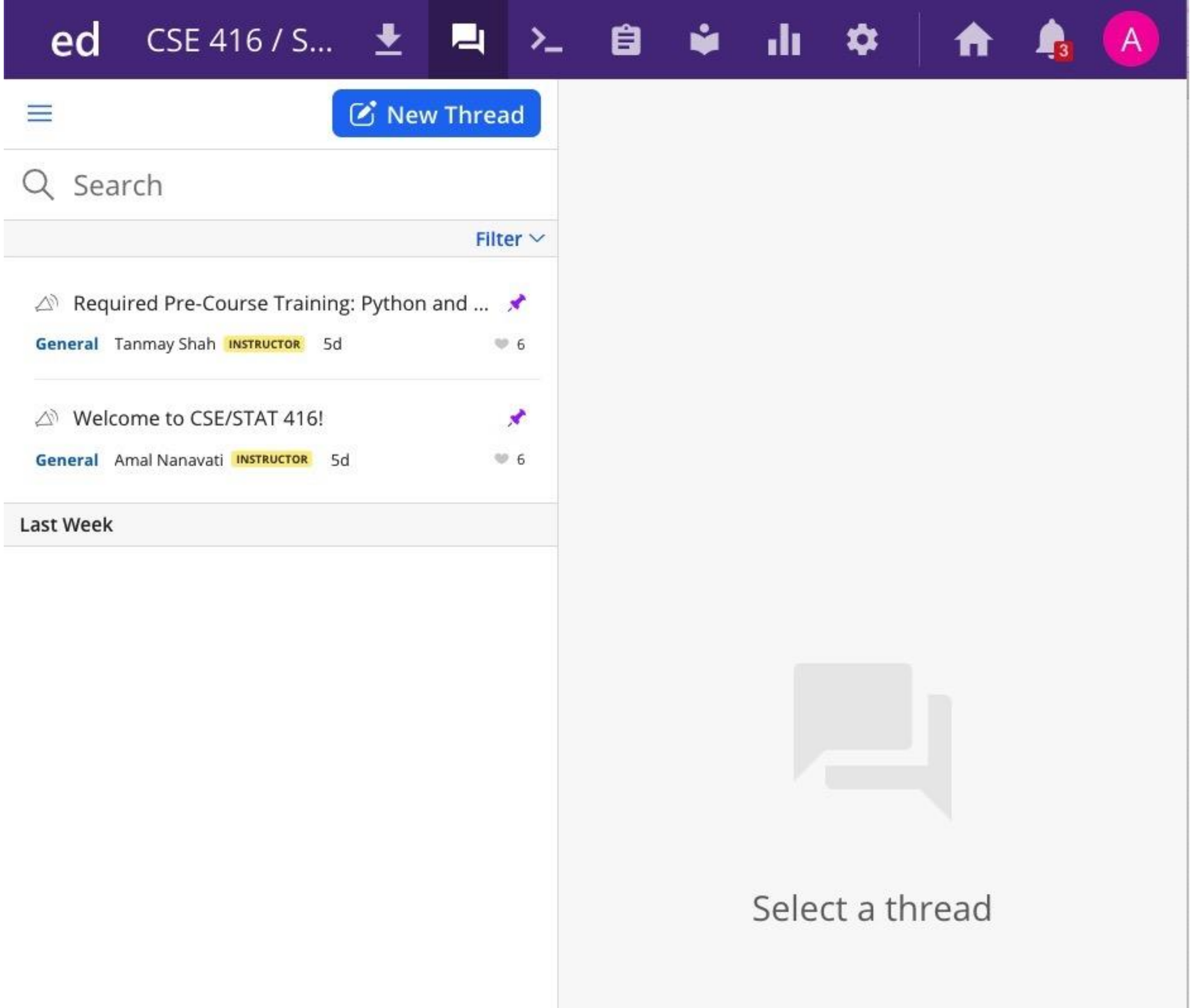
Office Hours



	Mon	Tue	Wed	Thu	Fri
8:00	Wuwei			Wuwei	
	Zoom			Zoom	
9:00		Josh			
		Zoom		Sections	
10:00					Tanmay
					CSE 121
11:00	Max		Max		Tanmay
	Zoom		Zoom		CSE 121
12:00					
13:00				Josh	
				Zoom	
14:00					
	Lecture		Lecture		
15:00					
16:00					
	Amal	Karman	Amal		Karman
17:00	CSE 212	CSE 131	CSE 212		CSE 131
18:00					

Getting Help

EdSTEM



Getting Help

- Office Hours are the best place to get help!

Office Hours (Sample Scenarios)	EdSTEM (Sample Scenarios)
• "My code isn't working, and I don't know why." • "Can you explain this concept to me?" • "I don't understand this homework question." • "This is how I'm thinking about Q#, am I on the right track?"	• "I think there is a typo in this question." • "What does this notation mean?" • "This is how I'm interpreting this (conceptual) homework question, but I think it might be wrong. Can you clarify?"

- Rule-of-thumb: if it requires **back-and-forth interaction**, **Office Hours** is the right place!
- EdSTEM Turnaround: 1 *business day*
- Don't be surprised if a TA responds to your EdSTEM question asking you to come to Office Hours!

Questions?

Brain Break

3:13

Case Study 1

Regression:
Housing Prices

Poll Everywhere

Think

90 seconds

pollev.com/cs416

On your phone / laptop

What factors do you think influence the price of a house?

area (sq ft)

location

materials

year

history of the house

Fitting Data

Goal: Predict how much my house is worth

Have data from my neighborhood

n data pts

(x_i, y_i)

$(x^{(i)}, y^{(i)})$

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

$x_q = 2500$ sq ft

$\hat{y}_q = \$350K$

Assumption:

There is a relationship between $y \in \mathbb{R}$ and $x \in \mathbb{R}^d$ ← # of features (1)

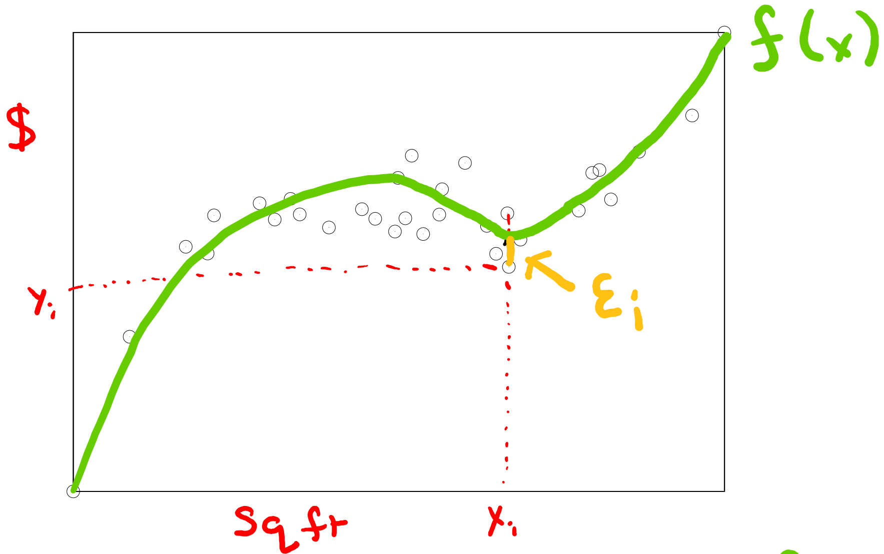
$$y \approx f(x)$$

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:

true function

$$y_i = f(x_i) + \varepsilon_i$$

$$\mathbb{E}[\varepsilon_i] = 0$$

"Essentially, all models are wrong, but some are useful."

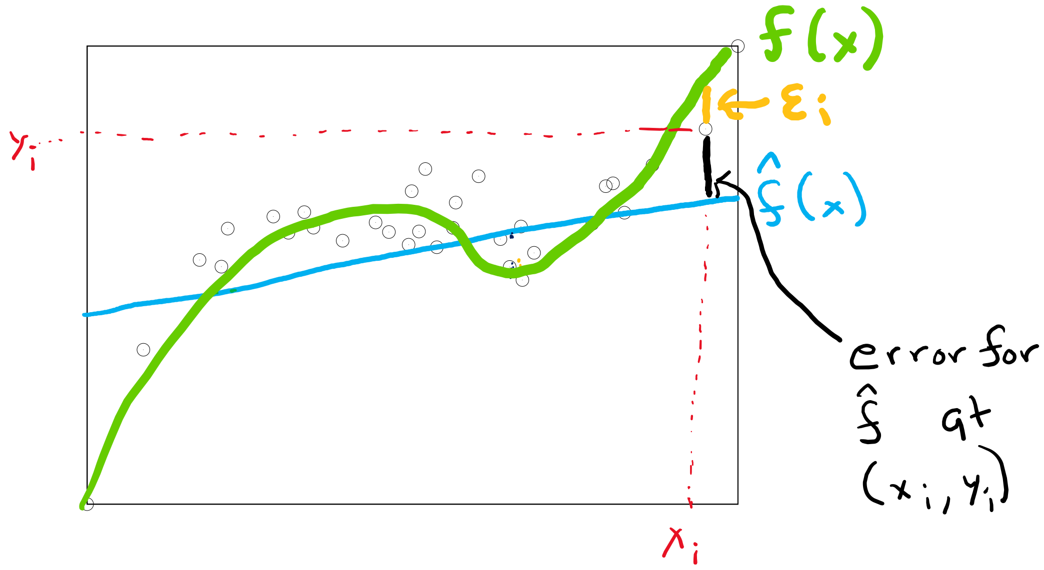
- George Box, 1987

Predictor

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

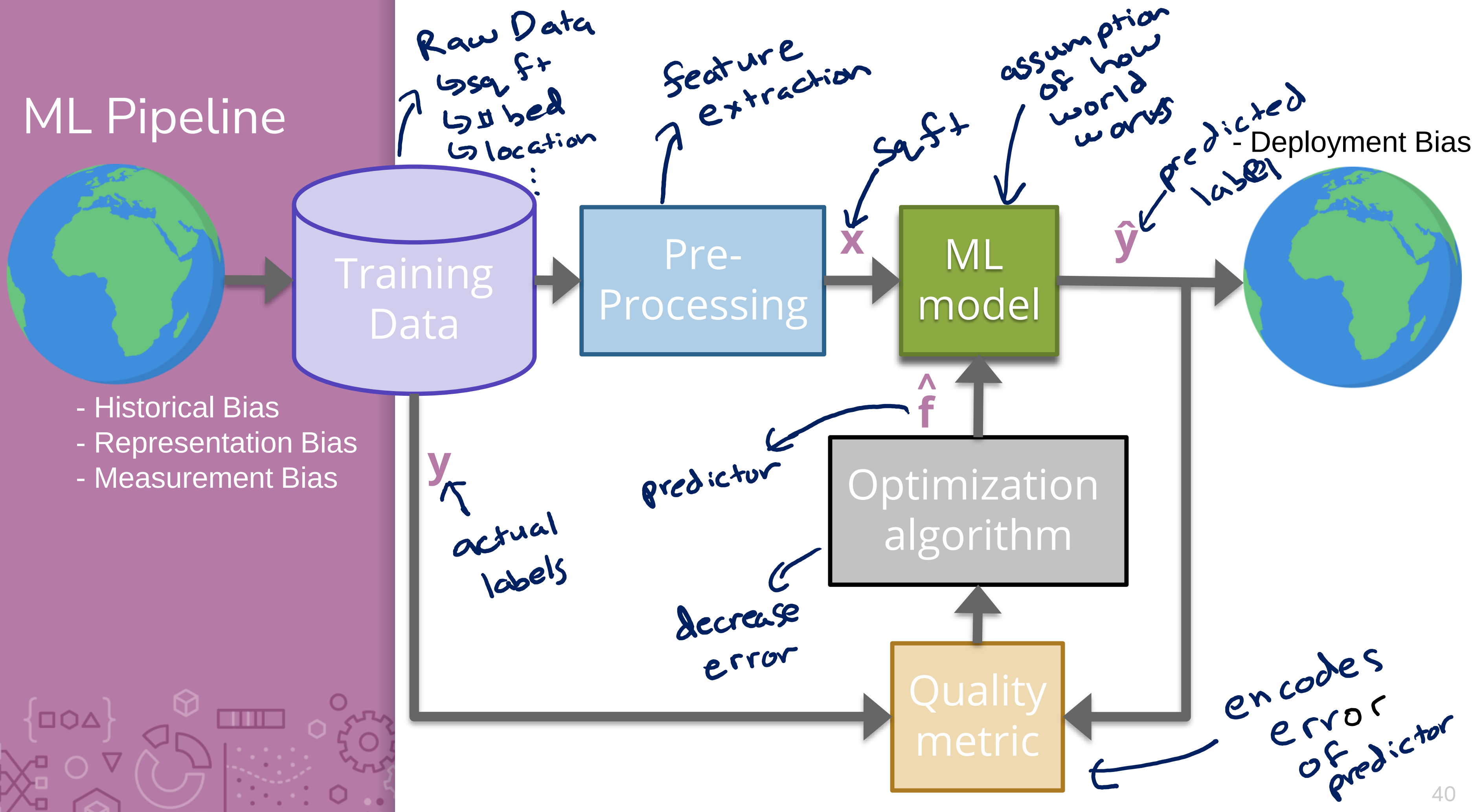
Use machine learning to learn a predictor $\hat{f}$ from the data

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



- Historical Bias
- Representation Bias
- Measurement Bias

Regression

- Is a supervised learning algorithm

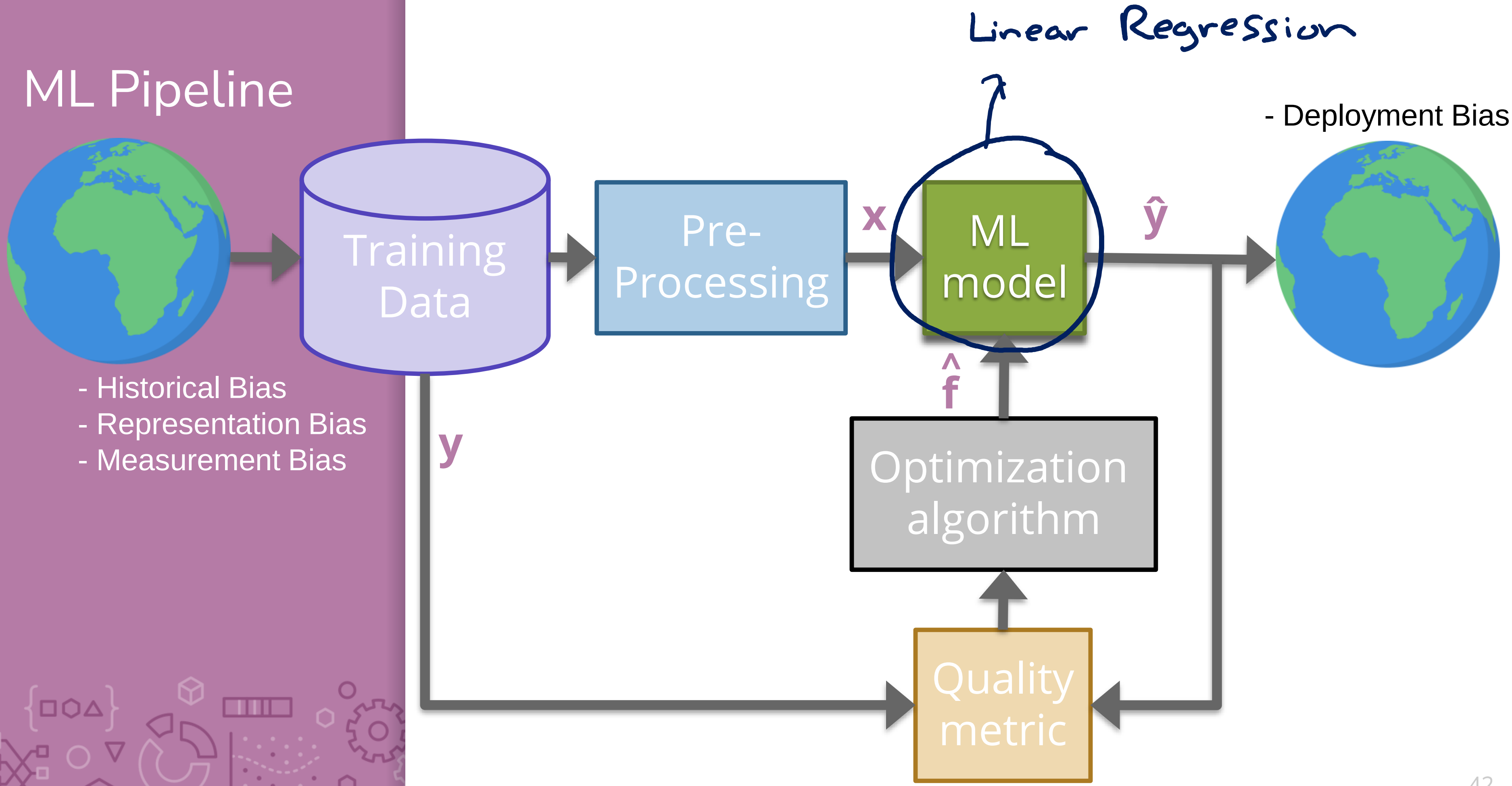
- Given a set of training data examples $X^{(1)}, X^{(2)}, \dots, X^{(n)}$ associated to with set of continuous values $y^{(1)}, y^{(2)}, \dots, y^{(n)}$ we want to build a predictor function that learns how to map $x^{(i)}$ to $y^{(i)}$.

$f: x \Rightarrow y$ $\qquad$ $f: sqft \rightarrow \$\$$

- Each example $x^{(i)}$ can have from 1 to many features $X_1^{(i)}, X_2^{(i)}, \dots, X_d^{(i)}$. We want to establish the relationships between different features of our data in order to make a good prediction.

- A typical regression problem is house price prediction.

ML Pipeline


Linear Regression
- Deployment Bias
Training Data
Pre-Processing
x
ML model
ŷ
- Historical Bias
- Representation Bias
- Measurement Bias
y
f̂
Optimization algorithm
Quality metric

Linear Regression Model

Assume we have a simple model with ***one feature***, where we establish a linear relationship between ***the area of a house*** i and ***its price***:

$$y_i = f(x_i) + \varepsilon_i$$

$$y_i = \overbrace{w_0 + w_1 x_i}^{f(x_i)} + \epsilon_i$$

$$b + mx$$

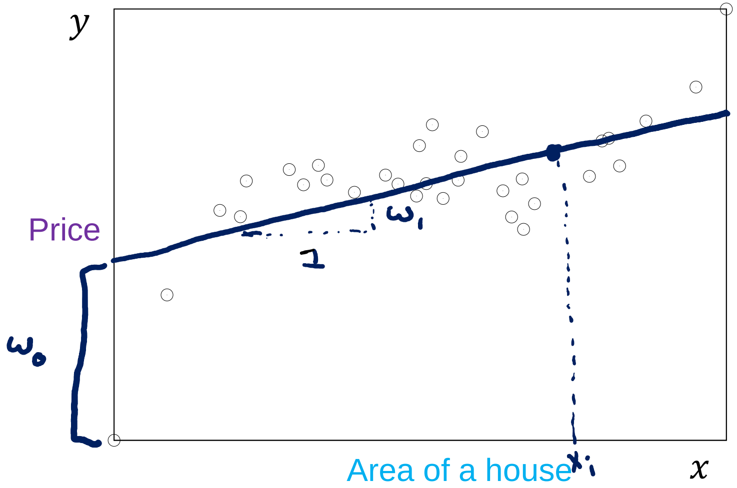
w_0, w_1 are the **parameters** of our model that need to be learned

- w_0 is the intercept / **bias**, representing the starting price of a house
- w_1 is the slope / **weight** associated with **feature** ”area of a house”

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

$$\hat{y} = \overbrace{\widehat{w}_1 x + \widehat{w}_0}^{\hat{f}(x)}$$

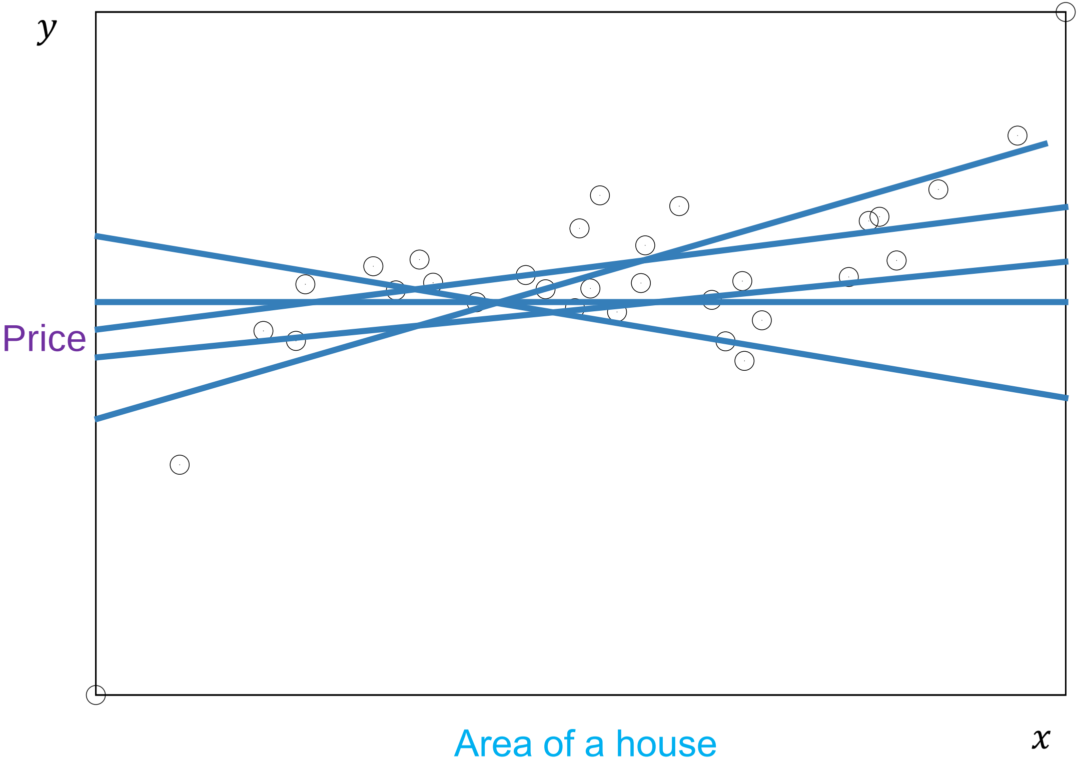
Why don’t we add ϵ?



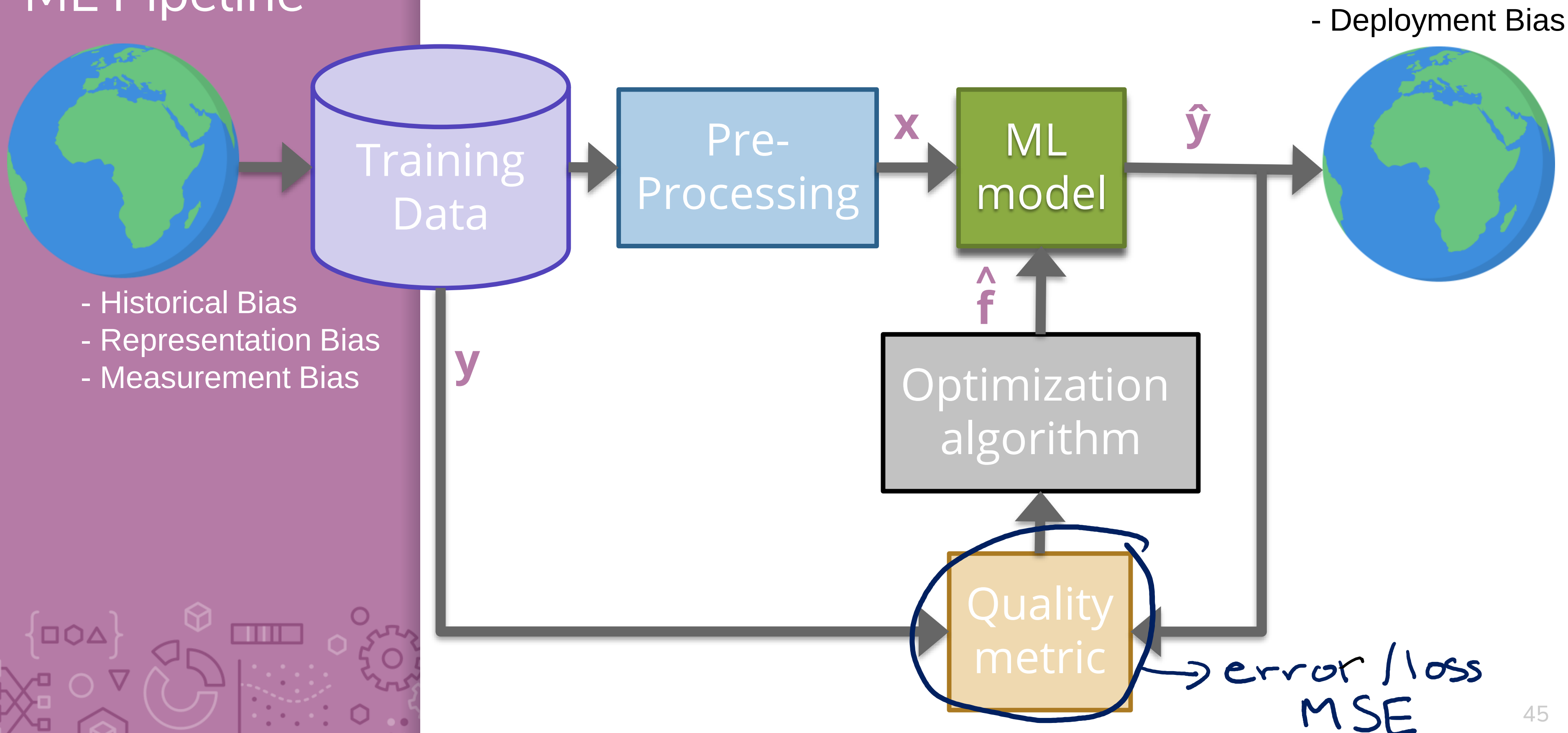
Basic Idea

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

What does best even mean here?



ML Pipeline



- Historical Bias
- Representation Bias
- Measurement Bias
Training Data
Pre-Processing
x
ML model
ŷ
- Deployment Bias
f̂
y
Optimization algorithm
Quality metric
error / loss
MSE

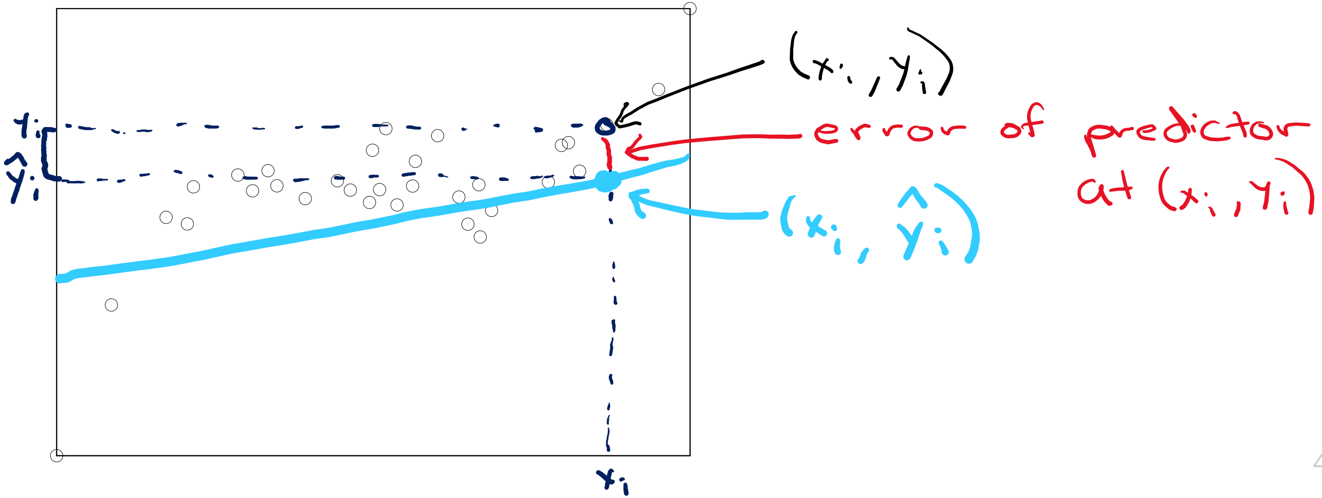
Cost / Loss of predictor

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 predictor (hopefully)**

$$MSE = \frac{1}{n}\left((y_0 - \hat{y}_0)^2 + \cdots (y_n - \hat{y}_n)^2\right)$$

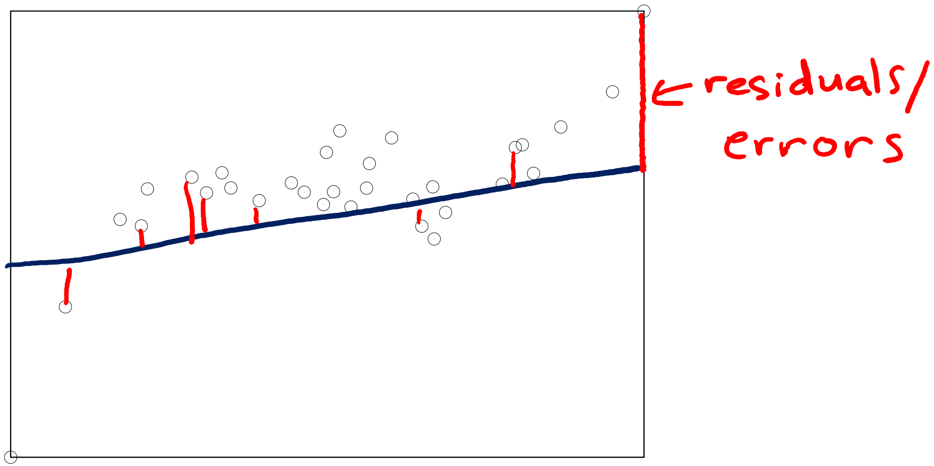
mean squared error



Mean Squared Error (MSE)

How to define error? **Mean squared error (MSE)**

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



Poll Everywhere

pollev.com/cs416

- **Goal**: Get you actively participating in your learning
- Typical Activity
  - Question is posted
  - **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
  - Not worth any points, just here to help you learn!

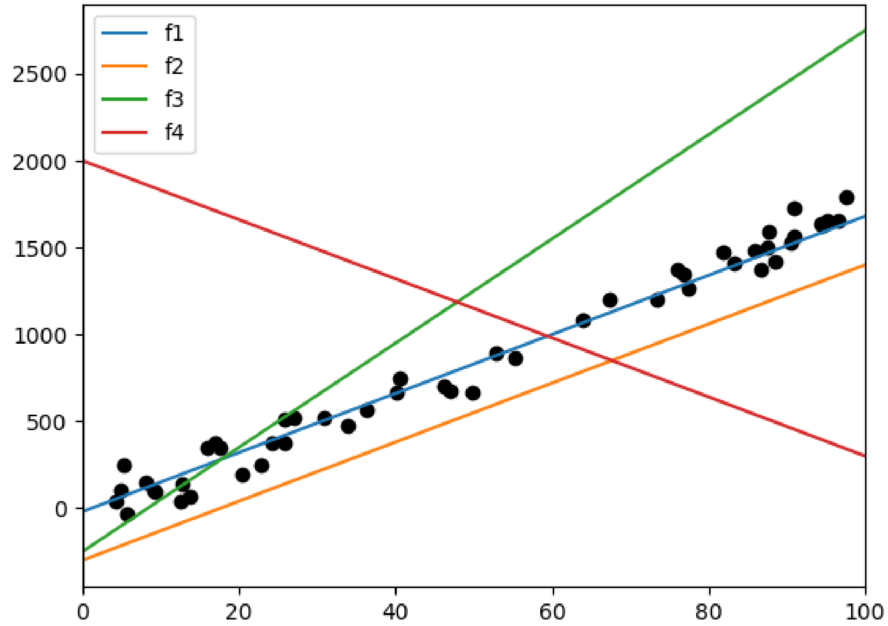
Poll Everywhere

Think

1 min

pollev.com/cs416

Sort the following lines by their MSE (mean-squared errors) on the data, from smallest to largest. (estimate, don't actually compute)



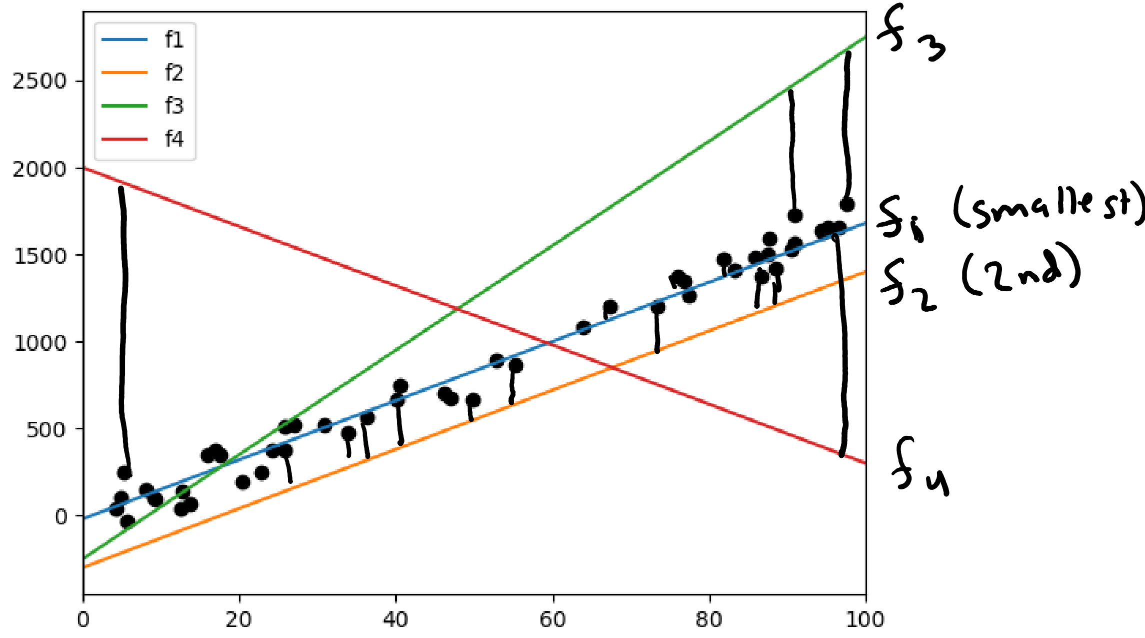
Poll Everywhere

Group

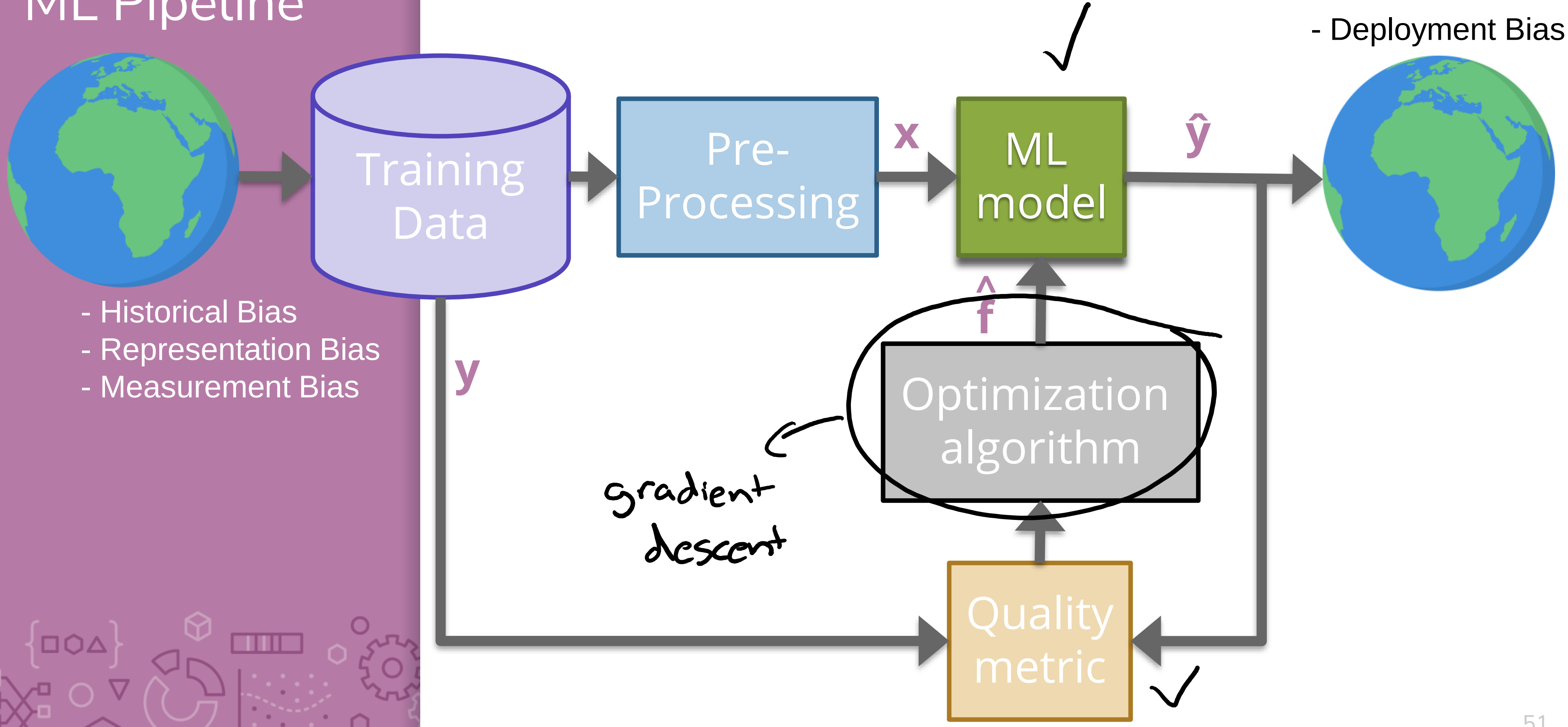
2 min

pollev.com/cs416

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



ML Pipeline

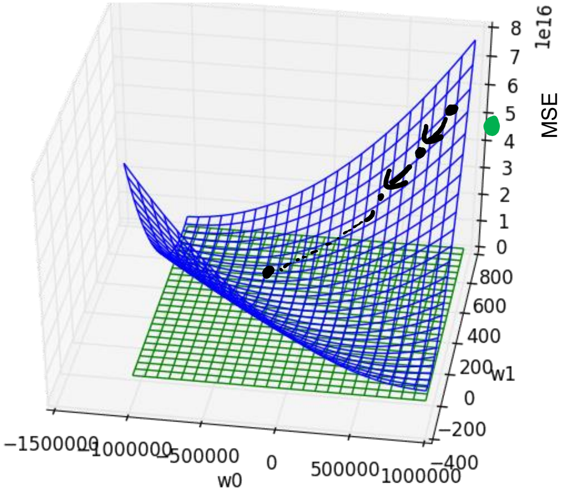


- Deployment Bias
Training Data
Pre-Processing
x
ML model
ŷ
f̂
y
Optimization algorithm
gradient descent
Quality metric
- Historical Bias
- Representation Bias
- Measurement Bias

Minimizing Cost

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

find the w_0, w_1 that minimize $MSE(w_0, w_1)$

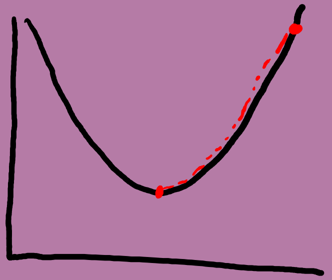


$$\hat{w}_0, \hat{w}_1 = \underset{w_0, w_1}{\operatorname{argmin}} MSE(w_0, w_1)$$

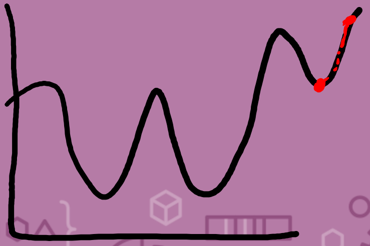
$$= \underset{w_0, w_1}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (w_0 + w_1 x_i)\right)^2$$

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

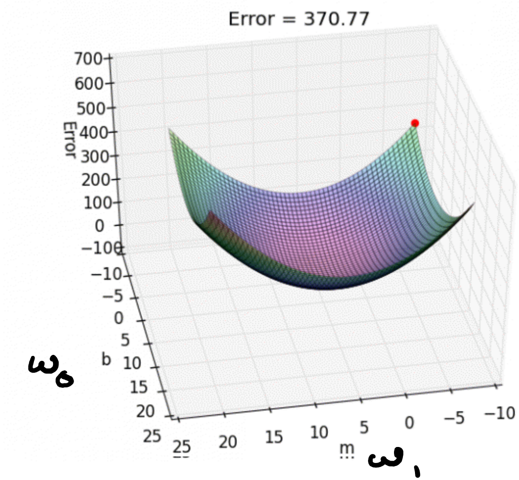
Gradient Descent

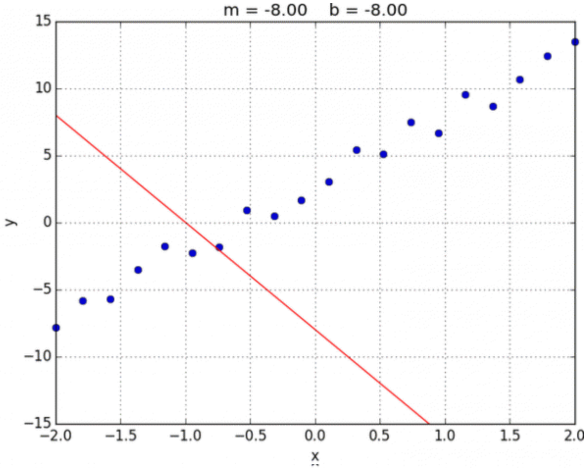


convex :)



non-convex :(





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) weights w

While we haven’t converged:

$$w \mathrel{-}= \alpha \boxed{\nabla L(w)}$$

gradient = direction of maximum ascent

- α: learning rate
- $\nabla L(w)$: the gradients of loss function L on a set of weights w

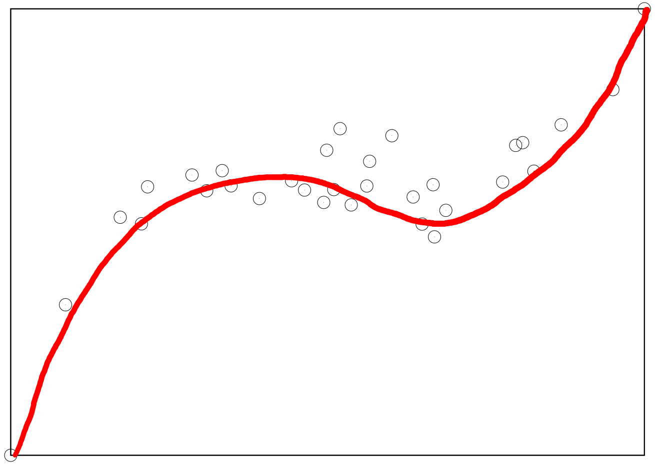
Brain Break

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!

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



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

Polynomial Regression

Model

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

To capture a non-linear relationship in the model, we can transform the original features into more features!

Polynomial Regression

Linear Regression

Feature	Value	Parameter
0	1 (constant)	w_0
1	x	w_1
2	x^2	w_2
...	…	…
p	x^d	w_d

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

Features

Features are the values we select or compute from the data inputs to put into our model. **Feature extraction** is the process of reducing the number of features in a dataset by creating new features from the existing ones.

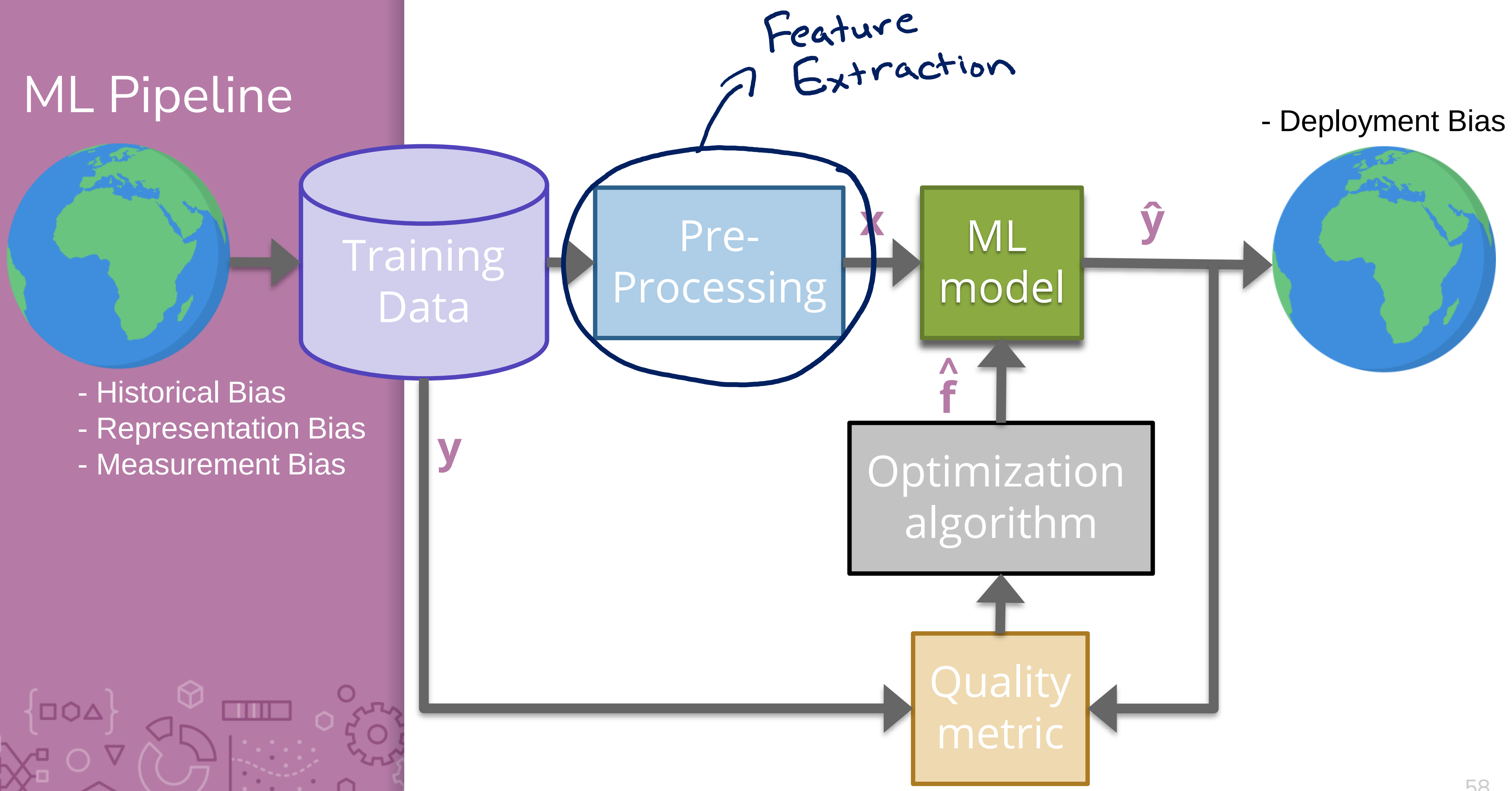
Model

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

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

Feature	Value	Parameter
0	$h_0(x)$ often 1 (constant)	w_0
1	$h_1(x)$ $= x$	w_1
2	$h_2(x)$ $= x^2$	w_2
…	… $= \log(x)+7$	…
d	$h_d(x)$ $= e^x$	w_d

ML Pipeline



- Historical Bias
- Representation Bias
- Measurement Bias

Adding Other Features

Generally, we are given a data table of values we might look at that includes more than one feature per house.

- Each row is a data point.
- Each column represents a feature
- One of the columns contains the actual output values

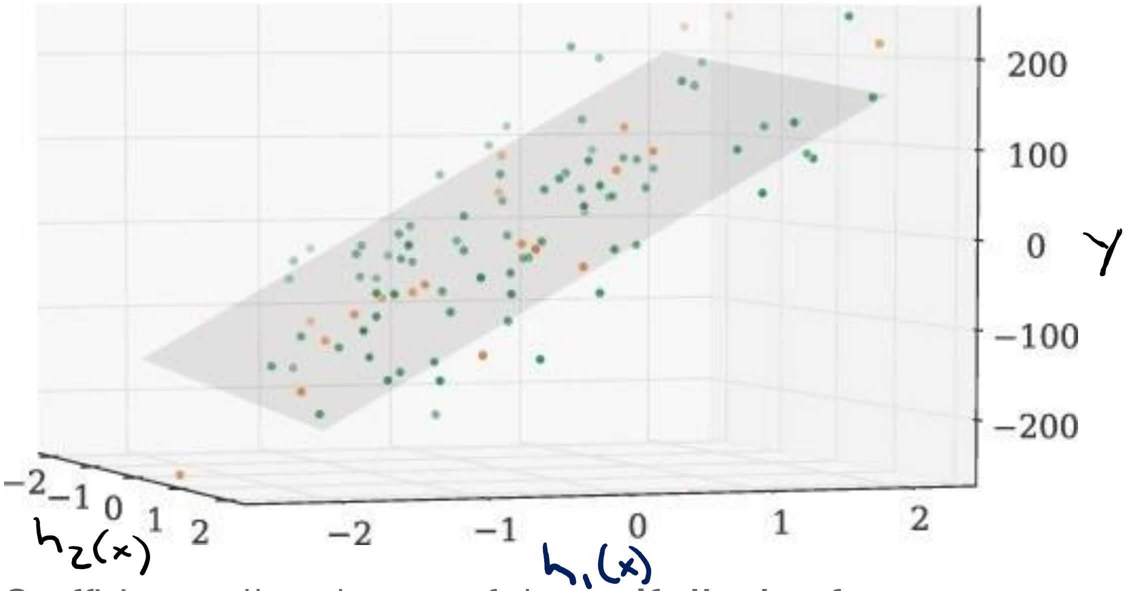
sq. ft.	# bathrooms	owner's age	…	price
1400	3	47	…	70,800
700	3	19	…	65,000
…	…	…	…	…
1250	2	36	…	100,000

- Sometimes we want to extract new features from existing features (e.g., #bath/#bed)

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**

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

Term recap

- **Supervised learning**: The machine learning task of learning a function that maps an input to an output based on example input-output pairs.
- **Regression**: A supervised learning task where the outputs are continuous values.
- **Feature**:
  - An attribute that we're selecting for our model
  - Can come from the original dataset, or through some transformations (***feature extraction***)
- **Parameter**: The weight or bias associated with a feature. The goal of machine learning is to adjust the weights to optimize the loss functions on training data.
- **Loss function**: A function that computes the distance between the predicted output from a machine learning model and the actual output.
- **Machine learning model**: An algorithm that combs through an amount of data to find patterns, make predictions, or generate insights
- **Optimization algorithm**: An algorithm used to minimize the loss during training. The most common one is ***Gradient Descent***.

Linear Regression Recap

Dataset

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

Feature Extraction

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

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

Regression Model

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

sometimes omit ε

Quality Metric / Loss function

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

Predictor

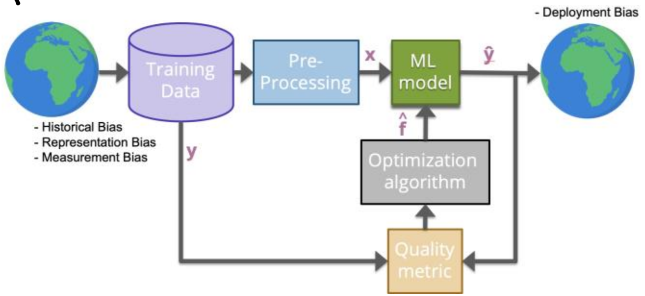
$$\hat{w} = \underset{w}{\operatorname{argmin}}\, MSE(w)$$

Optimization Algorithm

Optimized using Gradient Descent

Prediction

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



Deadlines & Other Logistics

- **Complete Pre-Course Training!**
- **Attend section tomorrow!**
  - Section AA/BA: 9:40-10:40AM, CMU 203
  - Section AB/BB: 10:50-11:50AM, SAV 138
  - Section AC/BC: 12:00-1:00PM, SMI 115
- Homework 0: (weight: 0%)
  - Aim to test your readiness for the course
  - Coding portion on EdStem, at the Assessments tab
  - Conceptual portion on Gradescope **(due Tues 11:59 pm)**
- Learning Reflection 1: **Due Friday 11:59 pm**
- Checkpoint 1: **Due Monday 1:50 pm**
- Please check EdStem regularly for the latest updates on course logistics.