

# CSCI 567: Machine Learning

Vatsal Sharan  
Spring 2024

Lecture 1, Jan 12



**USC** University of  
Southern California

# Logistics

Course website: <https://vatsalsharan.github.io/spring24.html>

- Logistics, slides, homework etc.

Ed Discussion: <https://edstem.org/>

- Main forum for communication

DEN: <https://courses.uscden.net/d2l/home/27576>

- Recordings

Gradescope: <https://www.gradescope.com/>

- Homework submission

# Prerequisites

**This is a mathematically advanced and intensive class  
(that makes it more interesting!)**

- (1) Undergraduate level training or coursework on linear algebra, (multivariate) calculus, and probability and statistics;
- (2) Programming with Python;
- (3) Undergraduate level training in the analysis of algorithms (e.g. runtime analysis).

Overview of logistics, **go through course website** for details:

**Homeworks:** 4 homeworks (groups of 2), 2 late days per student (max 1 per HW)

**Exams:** **3/1** and **4/26** during lecture hours (1pm-3:20pm)

**Project:** You can choose your topic, groups of 4, more details later

**Note:** Plagiarism and other unacceptable violations

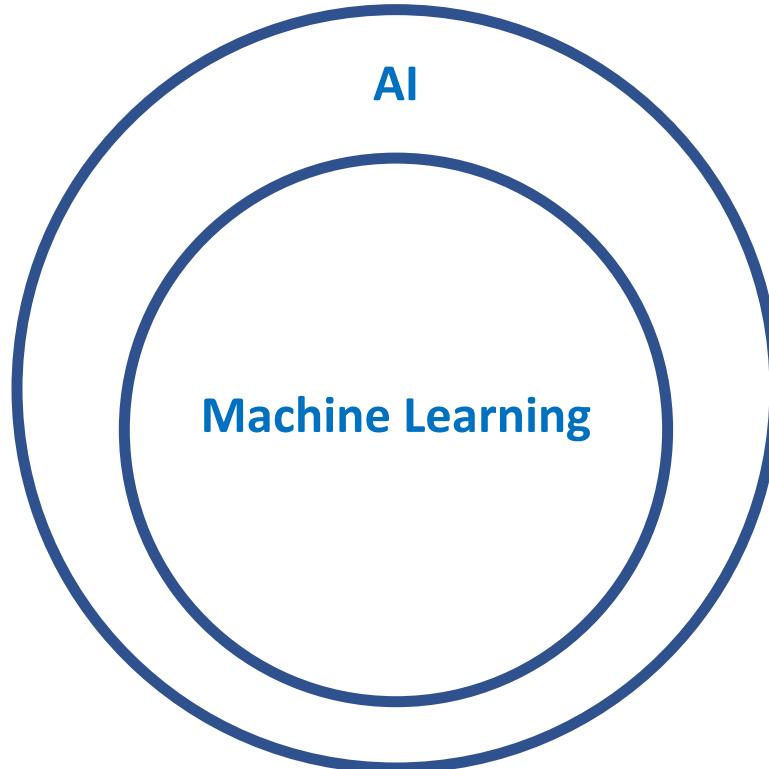
- Neither ethical nor in your self-interest
- Zero-tolerance
- Read collaboration policy on course website

MACHINE LEARNING

Machine Learning

MACHINE LEARNING EVERYWHERE

imgflip.com



ML has been driving the recent advances in AI

# What is ML?

*"Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense. In this paper we regard **learning** as the phenomenon of knowledge acquisition in the absence of explicit programming."*

--- *A Theory of the Learnable*, 1984, Leslie Valiant



# What is ML?

*"Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense. In this paper we regard **learning** as the phenomenon of knowledge acquisition in the absence of explicit programming."*

--- *A Theory of the Learnable*, 1984, Leslie Valiant



*"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  $E$ ."*

--- *Machine Learning*, 1998, Tom Mitchell



# My slides from Fall 2022 motivating ML..

## Enormous advances in recent years

The New York Times

THE SHIFT

### We Need to Talk About How Good A.I. Is Getting

We're in a golden age of progress in artificial intelligence. It's time to start taking its potential and risks seriously.

Give this article    Share    Bookmarks    608



DALL-E 2's output when given  
input "infinite joy"

New York Times, August 24, 2022

# My slides from Fall 2022 motivating ML..

## Text generation: GPT-3

The New York Times

Account ▾

### *Meet GPT-3. It Has Learned to Code (and Blog and Argue).*

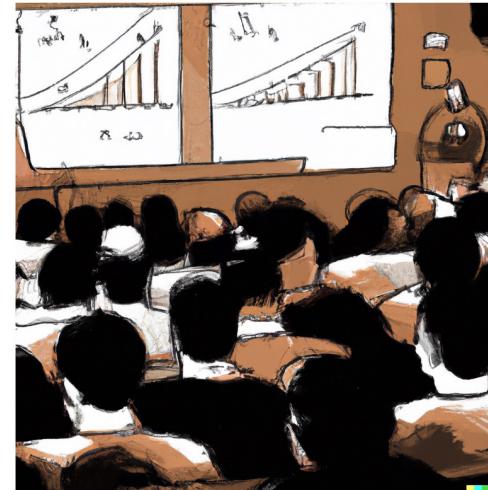
The latest natural-language system generates tweets, pens poetry, summarizes emails, answers trivia questions, translates languages and even writes its own computer programs.



## Image generation: Dall-E 2

I gave the prompt:

*A digital art image of a lecture on statistical machine learning. 200 students are sitting in a classroom, hearing about linear regression.*



# My slides from Fall 2022 motivating ML..

## Text generation: GPT-3

The New York Times

Account ▾



## Image generation: Dall-E 2

I gave the prompt:

*A digital art image of a lecture on statistical machine learning. 200 students are sitting in a classroom, hearing about linear regression.*



# My slides from Fall 2022 motivating ML..

## Text generation: GPT-3

The New York Times

Account ▾



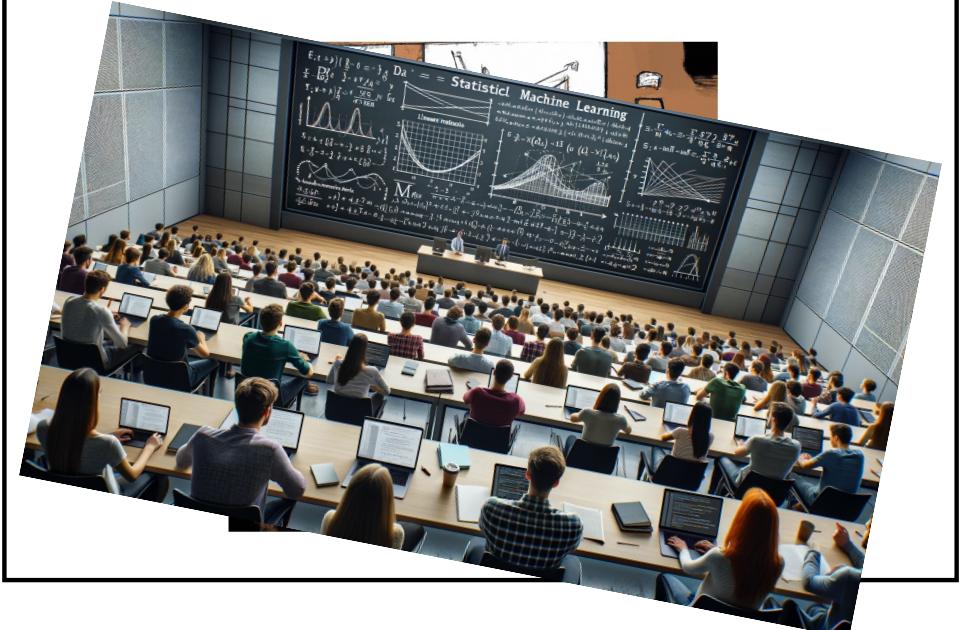
Chat GPT



## Image generation: Dall-E 2

I gave the prompt:

*A digital art image of a lecture on statistical machine learning. 200 students are sitting in a classroom, hearing about linear regression.*



What do you think are the most  
important advances?

# Some other flashy highlights..

## Game playing: AlphaGo



## Protein folding: AlphaFold

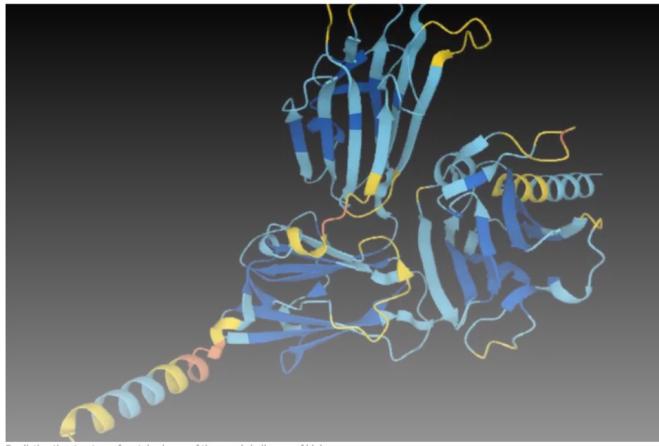
### DeepMind's protein-folding AI cracks biology's biggest problem

Artificial intelligence firm DeepMind has transformed biology by predicting the structure of nearly all proteins known to science in just 18 months, a breakthrough that will speed drug development and revolutionise basic science



TECHNOLOGY 28 July 2022

By Matthew Sparkes



# Exciting time, but a lot needs to be done..

- Require significant computational resources
- Lack of understanding
- Fairness
- Robustness
- Interpretability
- Privacy
- Alignment
- ...

# This class:

- Understand the fundamentals
- Understand when ML works, its limitations, think critically

# This class:

- Understand the fundamentals
- Understand when ML works, its limitations, think critically

In particular,

- Study fundamental statistical ML methods (supervised learning, unsupervised learning, etc.)
- Solidify your knowledge with hands-on programming tasks
- Prepare you for studying advanced machine learning techniques

# A simplistic taxonomy of ML

## **Supervised learning:**

Aim to predict outputs of future datapoints

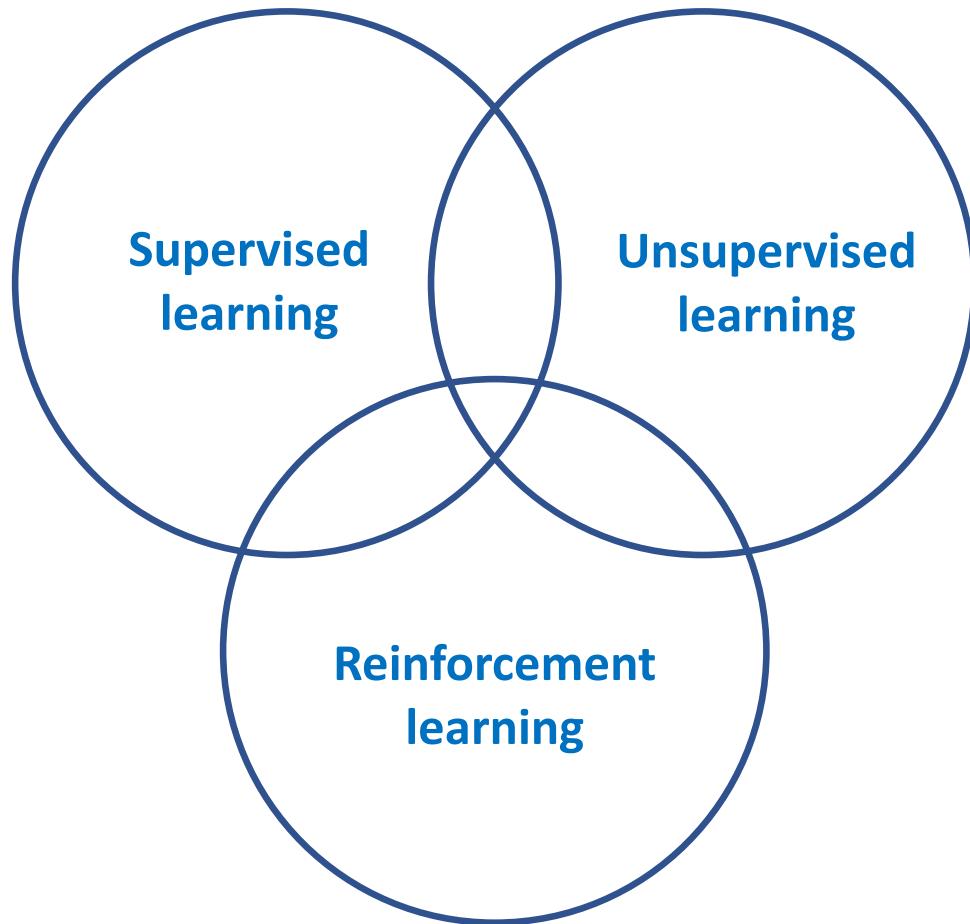
## **Unsupervised learning:**

Aim to discover hidden patterns and explore data

## **Reinforcement learning:**

Aim to make sequential decisions

# A simplistic taxonomy of ML





# Supervised Machine Learning

# Supervised ML: Predict future outcomes using past outcomes

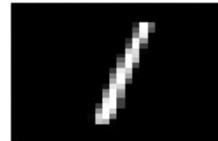
true class = 7



true class = 2



true class = 1



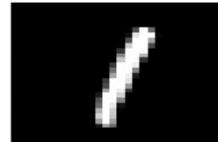
true class = 0



true class = 4



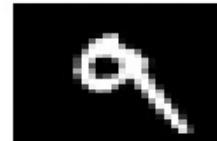
true class = 1



true class = 4



true class = 9



true class = 5



Image classification

English - detected



Hindi

Welcome to our  
machine learning  
class!



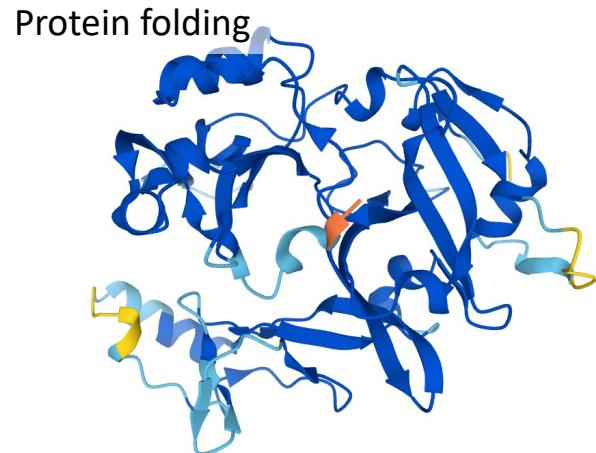
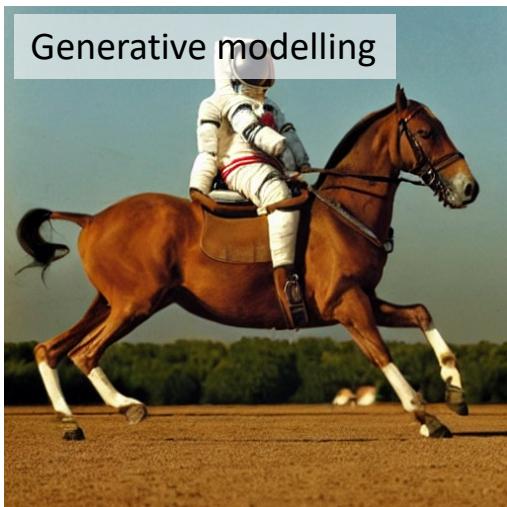
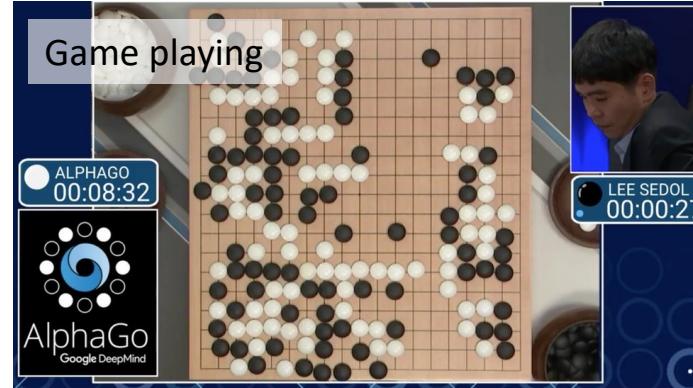
हमारे मशीन लर्निंग क्लास में  
आपका स्वागत है!  
hamaare masheen larning klaas mein  
aapaka svaagat hai!



[Open in Google Translate](#) • [Feedback](#)

Machine translation

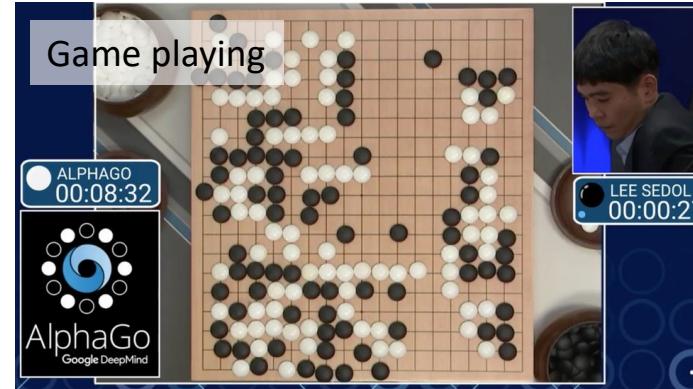
# Supervised ML is at the heart of many AI advances



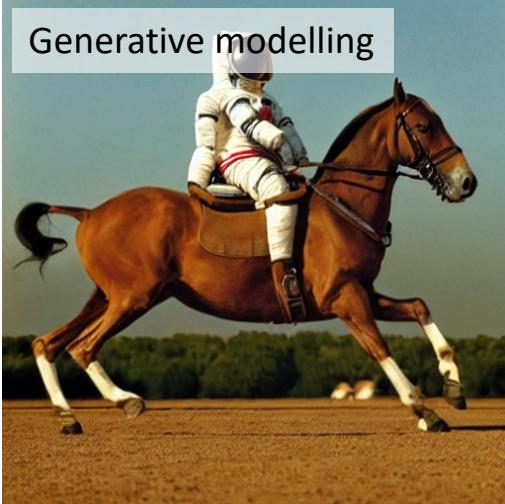
# Supervised ML is at the heart of many AI advances

Language modelling

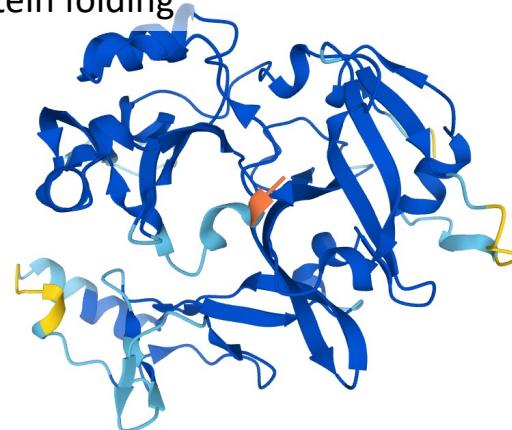
Given previous words ->  
Predict next word



Generative modelling



Protein folding



Medical imaging



# Supervised ML is at the heart of many AI advances

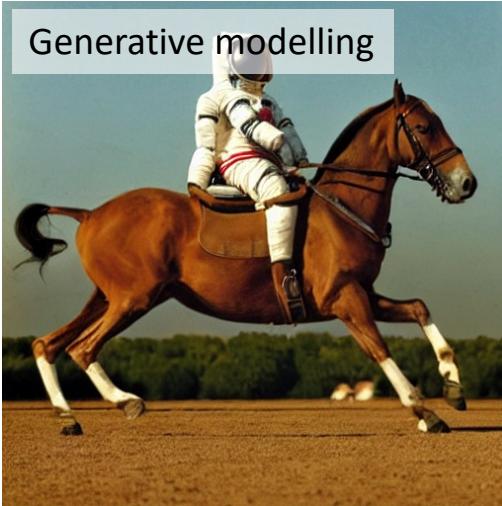
Language modelling

Given previous words ->  
Predict next word

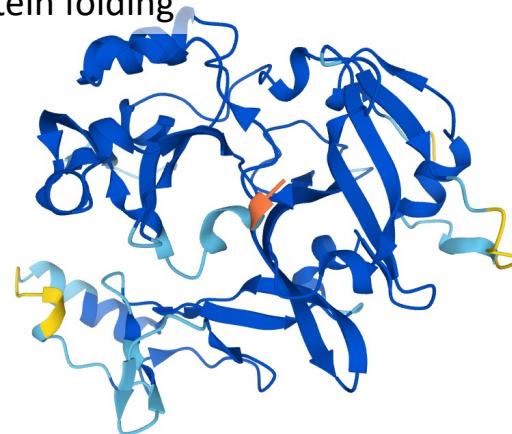
Game playing

Given current board state ->  
Predict probability of winning

Generative modelling



Protein folding



Medical imaging



# Supervised ML is at the heart of many AI advances

Language modelling

Given previous words ->  
Predict next word

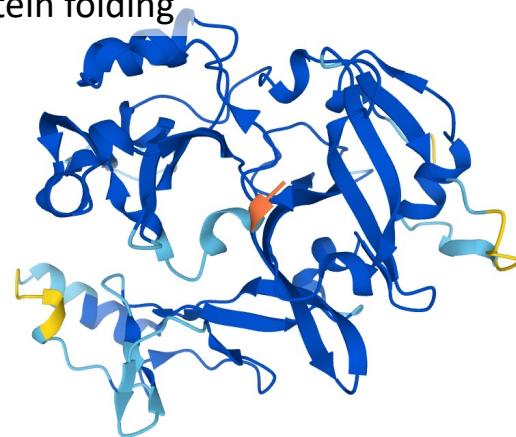
Game playing

Given current board state ->  
Predict probability of winning

Generative modelling

Given noisy image ->  
Predict denoised image

Protein folding



Medical imaging



# Supervised ML is at the heart of many AI advances

Language modelling

Given previous words ->  
Predict next word

Game playing

Given current board state ->  
Predict probability of winning

Generative modelling

Given noisy image ->  
Predict denoised image

Protein folding

Given protein chain ->  
Predict 3D structure



# Supervised ML is at the heart of many AI advances

Language modelling

Given previous words ->  
Predict next word

Game playing

Given current board state ->  
Predict probability of winning

Generative modelling

Given noisy image ->  
Predict denoised image

Protein folding

Given protein chain ->  
Predict 3D structure

Medical imaging

Given image ->  
Predict if there is tumor etc.

# Supervised ML: Predict future outcomes using past outcomes

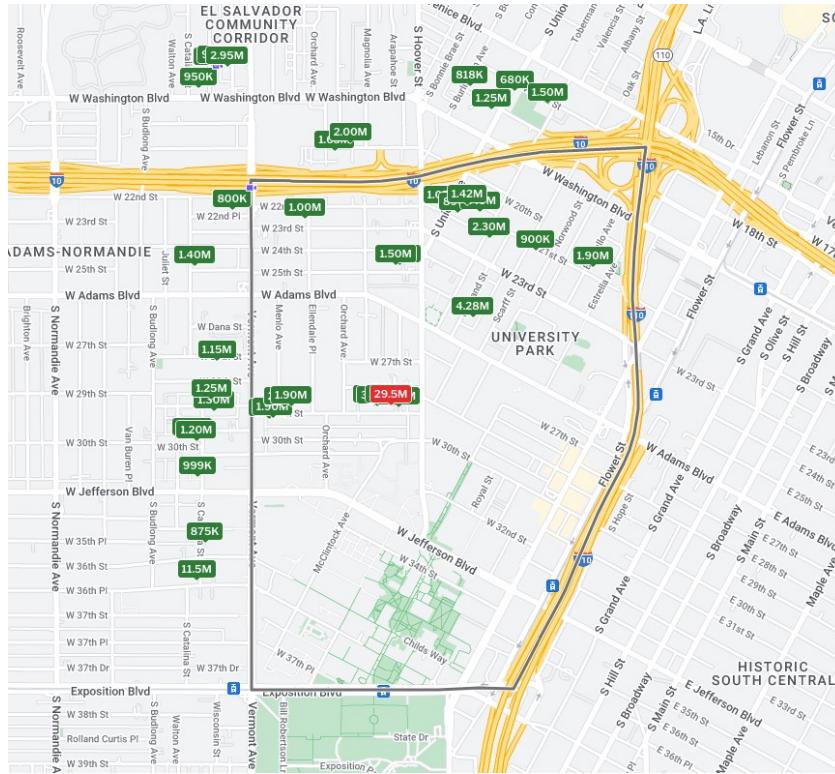
The collage includes:

- A top image showing a bathroom vanity with white cabinets and a double sink.
- A bottom-left image showing a kitchen area with white cabinets, a white refrigerator, and a sink.
- A bottom-right image showing the exterior of a single-story house with a light blue exterior and a small porch.
- To the right of the images is a screenshot of a Zillow listing page for the house at 2640 Monmouth Ave, Los Angeles, CA 90007. The listing shows a price of \$788,800, 5 bedrooms, 2 bathrooms, and 1,944 square feet. It is listed as "For sale by owner" with a Zestimate of \$888,500. An estimated payment of \$4,270 is also shown.
- A red box highlights a pop-up window titled "What's a Zestimate?" which provides information about the Zestimate, its limitations, and how it is calculated.
- Below the listing is a Google Maps interface showing the location of the house on Adams-Normandie Blvd in the University Park neighborhood of Los Angeles.

Predicting sale price of a house

# Simplistic version: Predicting sale price of a house

Retrieve historical sales records (training data):



# Simplistic version: Predicting sale price of a house

## Features used to predict:

**3620 South BUDLONG**  
Los Angeles, CA 90007  
Status: Closed

**\$1,510,000** | **14** Beds | **6** Baths | **4,418 Sq. Ft.**  
Last Sold Price | **14** Beds | **6** Baths | **4,418 Sq. Ft.**  
Built: 1956 | Lot Size: 9,648 Sq. Ft. | Sold On: Jul 26, 2013

Overview Property Details Tour Insights Property History Public Records Activity Schools



1 of 12 

Five unit apartment complex within 2 blocks of USC campus. Gate #6. Great for students (most student leases have parents as guarantors). Most USC students live off campus, so housing units like this are always fully leased. Situated on a gated, corner lot, and across from an elementary school, this complex was recently renovated, and has in-unit laundry hook ups, wall-unit AC, and 12 parking spaces. It is within a DPS (Department of Public Safety) and Campus Cruiser patrolled area. This is a great income generating property, not to be missed!

Property Type: Multi-Family | Style: Two Level, Low Rise  
Community: Downtown Los Angeles | County: Los Angeles  
MLS# 22176741

### Property Details for 3620 South BUDLONG, Los Angeles, CA 90007

Details provided by i-Tech MLS and may not match the public record. [Learn More](#)

#### Interior Features

Kitchen Information	Laundry Information	Heating & Cooling
• Remodeled	• Inside Laundry	• Wall Cooling Unit(s)
• Oven, Range		

#### Multi-Unit Information

Community Features	Unit 2 Information	Unit 5 Information
• Units in Complex (Total): 5	• # of Beds: 3	• # of Beds: 3
Multi-Family Information	• # of Baths: 1	• # of Baths: 2
• # Leased: 5	• Unfurnished	• Unfurnished
• # of Buildings: 1	• Monthly Rent: \$2,250	• Monthly Rent: \$2,325
• Owner Pays Water		
• Tenant Pays Electricity, Tenant Pays Gas	Unit 3 Information	Unit 6 Information
	• Unfurnished	• # of Beds: 3
Unit 1 Information	Unit 4 Information	• # of Baths: 1
• # of Beds: 2	• # of Beds: 3	• Monthly Rent: \$2,250
• # of Baths: 1	• # of Baths: 1	
• Unfurnished	• Unfurnished	
• Monthly Rent: \$1,700		

#### Property / Lot Details

Property Features	Lot Information	Property Information	Financial Information
• Automatic Gate, Card/Code Access	• Lot Size (Sq. Ft.): 9,649	• Updated/Renovated	• Capitalization Rate (%): 6.25
• Lot Information	• Lot Size (Acre): 0.2215	• Square Footage Source: Public Records	• Actual Annual Gross Rent: \$128,331
• Lot Size Source: Public Records			• Gross Rent Multiplier: 11.29

#### Parking / Garage, Exterior Features, Utilities & Financing

Parking Information	Utility Information	Financial Information
• # of Parking Spaces (Total): 12	• Green Certification Rating: 0.00	
• Parking Space	• Green Location: Transportation, Walkability	
• Gated	• Green Walk Score: 0	
Building Information	• Green Year Certified: 0	
• Total Floors: 2		

#### Location Details, Misc. Information & Listing Information

Location Information	Expense Information	Listing Information
• Cross Streets: W 36th Pl	• Operating: \$37,664	• Listing Term: Cash, Cash To Existing Loan
		• Buyer Financing: Cash

# Simplistic version: Predicting sale price of a house

## Features used to predict:

3620 South BUDLONG  
Los Angeles, CA 90007  
Status: Closed

Overview Property Details Tour Insights Property History Public Records Activity Schools

**SOLD**

**\$1,510,000** | **14** Beds | **6** Baths | **4,418 Sq. Ft.** | **\$342 / Sq. Ft.**

Built: 1956 Lot Size: 9,648 Sq. Ft. Build On: Jul 26, 2013

**Interior Features**

- Remodeled
- Oven, Range

**Multi-Unit Information**

- Tenant Pays Electricity, Tenant Pays Gas

**Unit 1 Information**

- # of Beds: 2
- # of Baths: 1
- Unfurnished
- Monthly Rent: \$1,700

**Unit 2 Information**

- # of Beds: 2
- # of Baths: 1
- Unfurnished
- Monthly Rent: \$2,250

**Unit 3 Information**

- Unfurnished

**Unit 4 Information**

- # of Beds: 3
- # of Baths: 1
- Unfurnished

**Unit 5 Information**

- # of Beds: 3
- # of Baths: 2
- Unfurnished
- Monthly Rent: \$2,325

**Unit 6 Information**

- # of Beds: 3
- # of Baths: 1
- Monthly Rent: \$2,250

**Property Features**

- Automatic Gate, Card/Code Access
- Corner Lot, Near Public Transit

**Lot Information**

- Lot Size (Sq. Ft.): 9,649
- Lot Size (Acre): 0.2215
- Lot Size Source: Public Records

**Parking / Garage, Exterior Features, Utilities & Financing**

**Parking Information**

- # of Parking Spaces (Total): 12
- Parking Space
- Gated

**Building Information**

- Total Floors: 2

**Location Details, Misc. Information & Listing Information**

**Location Information**

- Cross Streets: W 36th Pl

**Expense Information**

- Operating: \$37,664

**Financial Information**

- Capitalization Rate (%): 6.25
- Actual Annual Gross Rent: \$128,331
- Gross Rent Multiplier: 11.29

**Listing Information**

- Listing Term: Cash, Cash To Existing Loan
- Buyer Financing: Cash

Numeric data

Free-form text

Categorical data

**Property Details for 3620 South BUDLONG, Los Angeles, CA 90007**

Details provided by i-Tech MLS and may not map to the public record. [Learn More](#)

**Interior Features**

- Remodeled
- Oven, Range

**Laundry Information**

- Inside Laundry

**Heating & Cooling**

- Wall Cooling Unit(s)

**Community Features**

- Units in Complex (Total): 5

**Multi-Family Information**

- # Leased: 5
- # of Buildings: 1

**Unit 1 Information**

- # of Beds: 2
- # of Baths: 1
- Unfurnished
- Monthly Rent: \$1,700

**Unit 2 Information**

- # of Beds: 2
- # of Baths: 1
- Unfurnished
- Monthly Rent: \$2,250

**Unit 3 Information**

- Unfurnished

**Unit 4 Information**

- # of Beds: 3
- # of Baths: 1
- Unfurnished

**Unit 5 Information**

- # of Beds: 3
- # of Baths: 2
- Unfurnished
- Monthly Rent: \$2,325

**Unit 6 Information**

- # of Beds: 3
- # of Baths: 1
- Monthly Rent: \$2,250

**Property Features**

- Automatic Gate, Card/Code Access
- Corner Lot, Near Public Transit

**Lot Information**

- Lot Size (Sq. Ft.): 9,649
- Lot Size (Acre): 0.2215
- Lot Size Source: Public Records

**Parking / Garage, Exterior Features, Utilities & Financing**

**Parking Information**

- # of Parking Spaces (Total): 12
- Parking Space
- Gated

**Building Information**

- Total Floors: 2

**Location Details, Misc. Information & Listing Information**

**Location Information**

- Cross Streets: W 36th Pl

**Expense Information**

- Operating: \$37,664

**Financial Information**

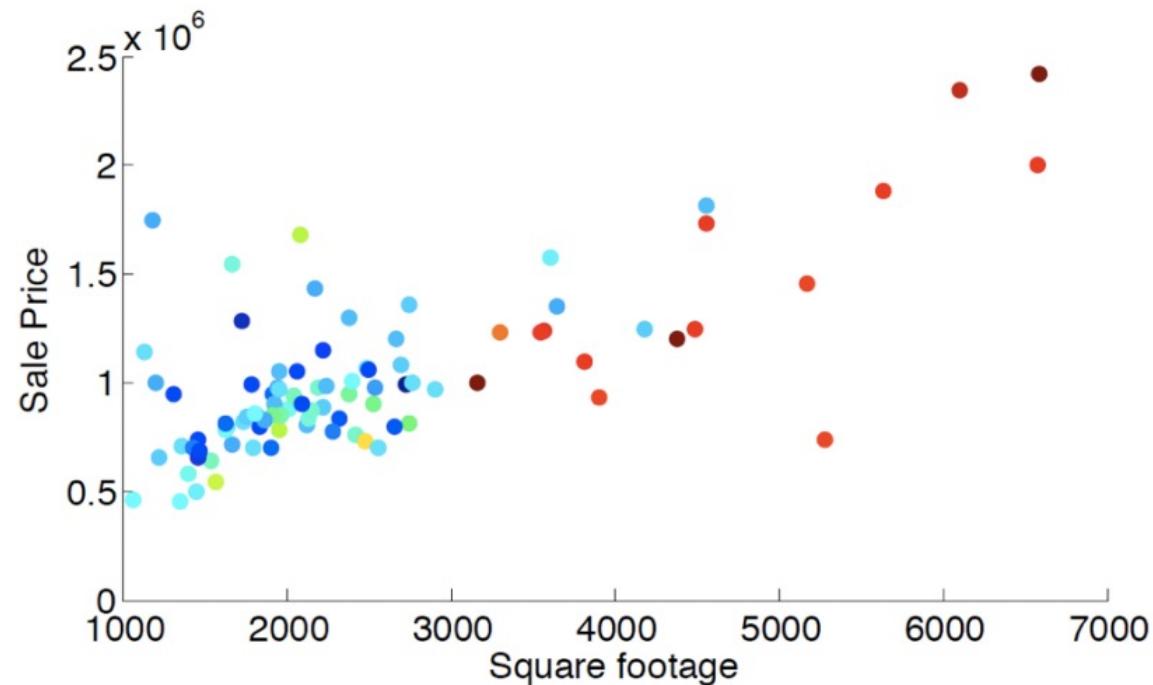
- Capitalization Rate (%): 6.25
- Actual Annual Gross Rent: \$128,331
- Gross Rent Multiplier: 11.29

**Listing Information**

- Listing Term: Cash, Cash To Existing Loan
- Buyer Financing: Cash

## Simplistic version: Predicting sale price of a house

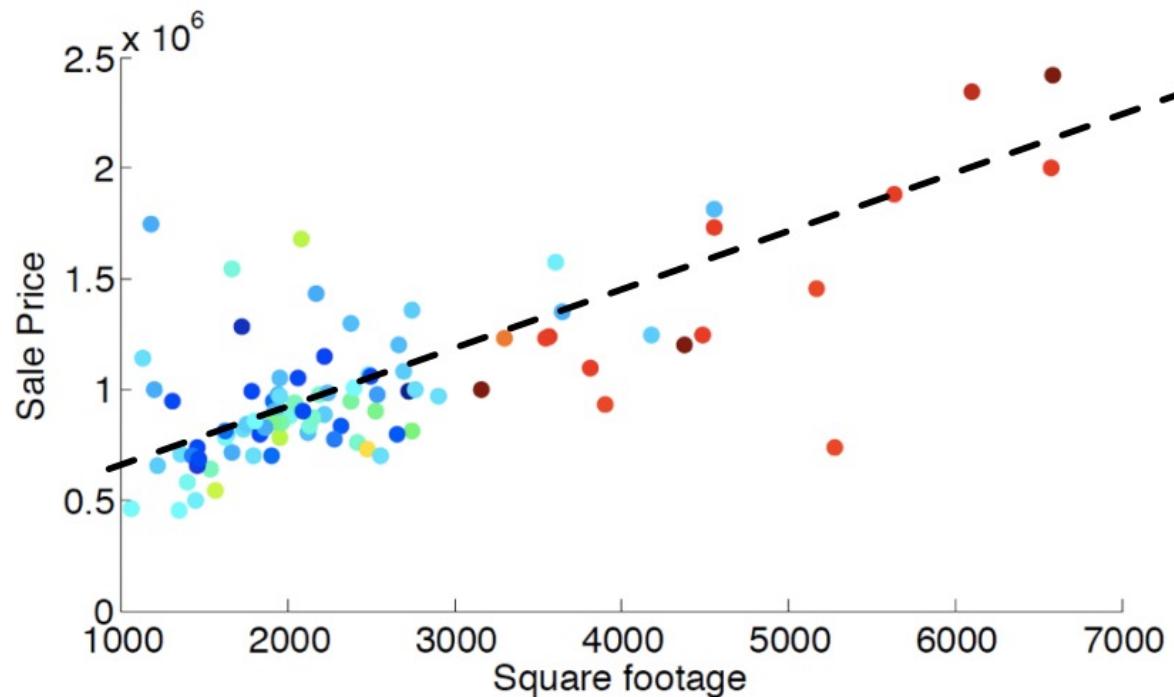
Correlation between square footage and sale price:



## **Simplistic version:** Predicting sale price of a house

## Possibly linear relationship:

Sale price  $\approx$  **price per sqft**  $\times$  square footage + **fixed expense**  
*(slope)* *(intercept)*



## General framework for supervised learning

- An **input space** :  $X \subseteq \mathbb{R}^d$
- \* Datapoints in  $d$  dimensions
  - \* In previous example,  $d=1$
- Feature engineering !

- An **output space** :  $Y$

- \*  $y \in \mathbb{R}$  for sale price prediction

**Goal** : Learn a predictor  $f(x) : X \rightarrow Y$

which predicts output of  $x$

Loss function :  $l(f(x), y)$

e.g. squared loss for  $y = \mathbb{R}$  :  $l(f(x), y) = (f(x) - y)^2$

What to minimize over?

Def : Given a set of labelled data points

$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , the empirical risk for predictor  $f: X \rightarrow Y$  wrt set  $S$  is

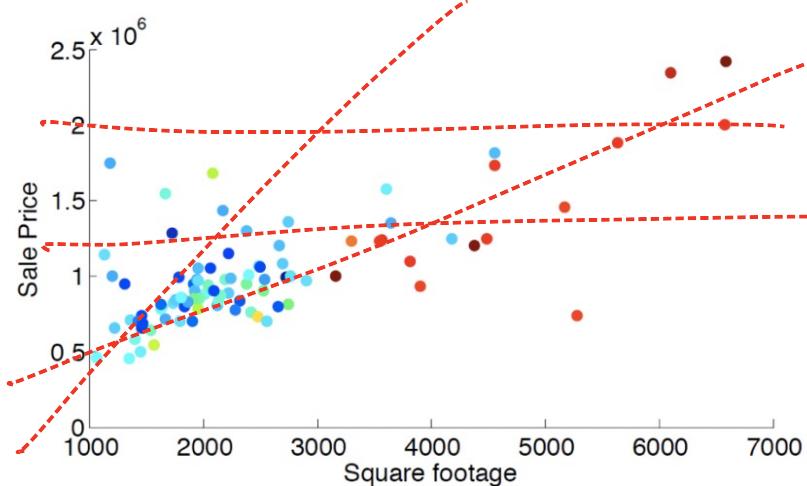
$$\hat{R}_S(f) = \frac{1}{n} \sum_{i=1}^n l(f(x_i), y_i)$$

## Function class

Def : A function class (or hypothesis class) is a collection of functions  $f: X \rightarrow Y$ .

Example :  $X = \mathbb{R}$ ,  $Y = \mathbb{R}$ ,  $\mathcal{F} = \{f: y = wx + c\}$

- Each of these is a linear function.
- The class of all linear functions is a function class.



## Empirical risk minimizer (ERM)

Def: Given a function class  $\hat{F} = \{f: X \rightarrow Y\}$ ,  
empirical risk minimization over a set of  
labelled datapoints  $S$  corresponds to:

$$\min_{f \in \hat{F}} \hat{R}_S(f) = \frac{1}{n} \sum_{i=1}^n l(f(x_i), y_i)$$

Optimization

## Generalization

\* We want predictor to generalize on unseen points.

Def. (Test error): The test error of a predictor  $f$  is the average loss on a "new" set  $S'$  of  $m$  points  $S' = \{(x'_i, y'_i), i \in m\}$

$$\frac{1}{m} \sum_{i=1}^m l(f(x'_i), y'_i)$$

Training / Test paradigm: Assume training set  $S$  & test set  $S'$  are drawn from same distribution.

## Measuring generalization: Training/Test paradigm

Randomly divide data into

Training set : subset of data to train model

Test set : subset used to test model

Generalization gap : Difference b/w test & training set errors

## Generalization: More formally

Minimize loss over distribution of instances

Definition: Risk  $R$  of predictor  $f$

$$\begin{aligned} R(f) &= \mathbb{E}_{(x,y) \sim D} [l(f(x), y)] \\ &= \sum_{x', y'} \text{Prob}_D(x=x', y=y') l(f(x'), y') \end{aligned}$$

How to empirically evaluate this?

The average loss on "test set"  $S'$ :  $S' = \{(x'_i, y'_i)\}, i \in [m]$   
 $((x'_i, y'_i) \sim D)$

$$R(f) \approx \frac{1}{m} \sum_{i=1}^m l(f(x'_i), y'_i)$$

A tautology :

$$R(f) = \hat{R}_S(f) + (R(f) - \hat{R}_S(f))$$

To minimize  $R(F)$

→ First try to minimize  $\hat{R}_S(f)$

→ What's left is  $R(f) - \hat{R}_S(f)$ . This is the generalization gap.

# Supervised learning in one slide

**Loss function:** What is the right loss function for the task?

*Depends on the problem that one is trying to solve, and on the rest...*

# Supervised learning in one slide

**Loss function:** What is the right loss function for the task?

**Representation:** What class of functions should we use?

*Also known as the “inductive bias”.*

*No-free lunch theorem from learning theory tells us that  
no model can do well on every task*

*“All models are wrong, but some are useful”, George Box*

# Supervised learning in one slide

- Loss function:** What is the right loss function for the task?
- Representation:** What class of functions should we use?
- Optimization:** How can we efficiently solve the empirical risk minimization problem?

*Depends on all the above and also...*

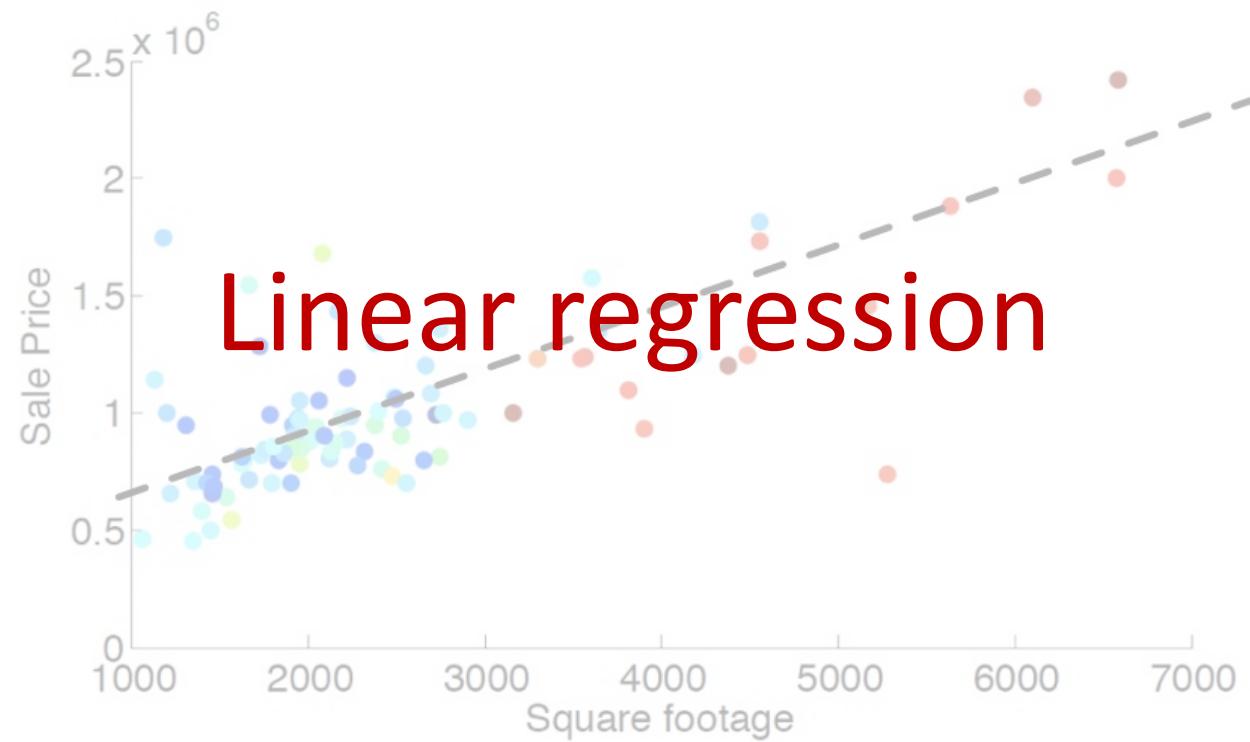
# Supervised learning in one slide

- Loss function:** What is the right loss function for the task?
- Representation:** What class of functions should we use?
- Optimization:** How can we efficiently solve the empirical risk minimization problem?
- Generalization:** Will the predictions of our model transfer gracefully to unseen examples?

# Supervised learning in one slide

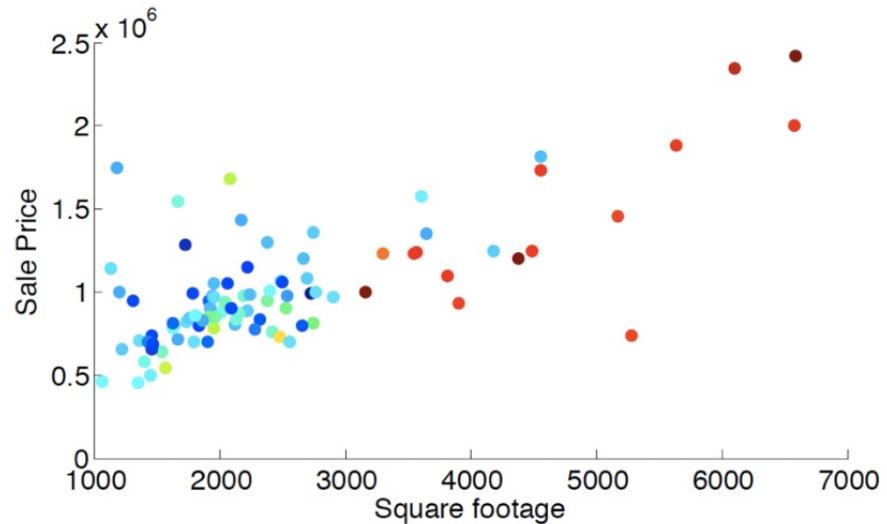
- Loss function:** What is the right loss function for the task?
- Representation:** What class of functions should we use?
- Optimization:** How can we efficiently solve the empirical risk minimization problem?
- Generalization:** Will the predictions of our model transfer gracefully to unseen examples?

*All related! And the fuel which powers everything is **data**.*



# House price prediction: **the loss function**

We're looking at real-valued outputs. Some popular loss functions:

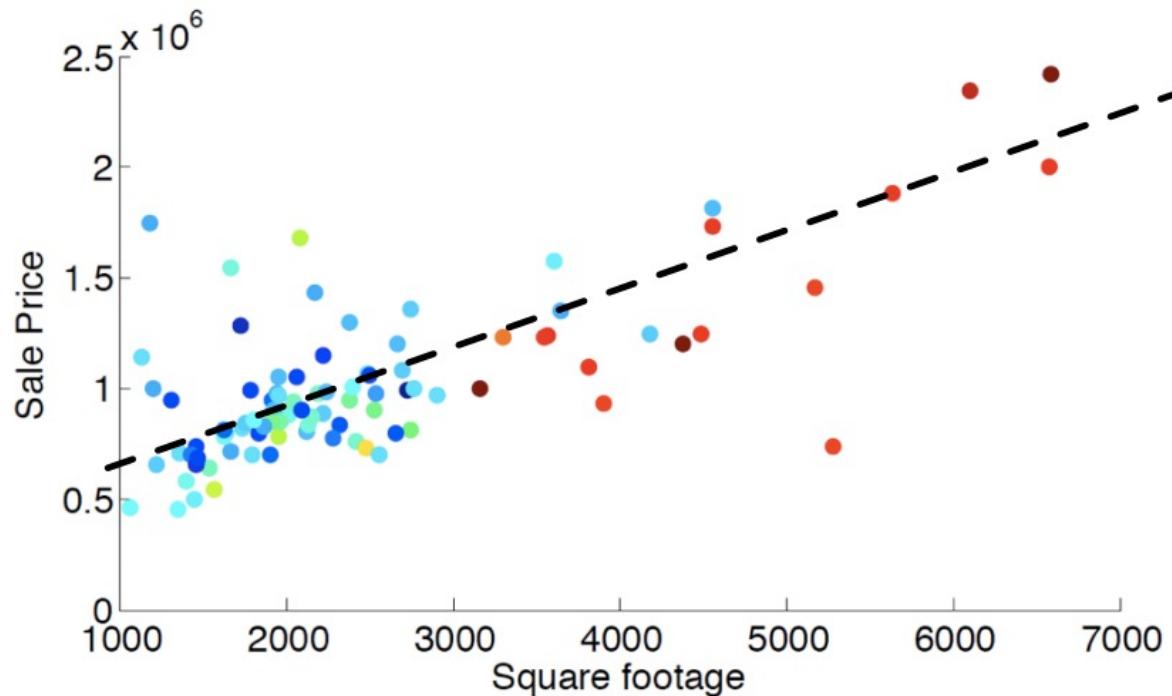


- Squared loss (most common):  $(\text{prediction} - \text{sale price})^2$ .
- Absolute value loss:  $|\text{prediction} - \text{sale price}|$ .

# House price prediction: the function class

Possibly linear relationship:

Sale price  $\approx$  **price per sqft**  $\times$  square footage + **fixed expense**



## Linear regression

Predicted sale price = **price\_per\_sqft** × square footage + **fixed\_expense**

one model:  $\text{price\_per\_sqft} = 0.3K$ ,  $\text{fixed\_expense} = 210K$

sqft	sale price (K)	prediction (K)	squared error
2000	810	810	0
2100	907	840	$67^2$
1100	312	540	$228^2$
5500	2,600	1,860	$740^2$
...	...	...	...
Total			$0 + 67^2 + 228^2 + 740^2 + \dots$

Adjust **price\_per\_sqft** and **fixed\_expense** such that the total squared error is minimized.

# Putting things together: Linear regression

- Input:  $\mathbf{x} \in \mathbb{R}^d$ , Output:  $y \in \mathbb{R}$ .
- Loss for predictor  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  on  $(\mathbf{x}, y)$ :  $(f(\mathbf{x}) - y)^2$ .
- Training data  $S = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$ .
- Linear model  $\{f : f(x) = w_0 + \sum_{j=1}^d w_j x_j = w_0 + \mathbf{w}^\top \mathbf{x}, \mathbf{w} \in \mathbb{R}^d\}$ .
  - $\mathbf{w} = [w_1, \dots, w_d]^\top$  are the weights.
  - $w_0$  is bias.

## Note: For notational convenience

Append 1 to each  $\mathbf{x}$  as first feature:  $\tilde{\mathbf{x}} = [ 1 \ x_1 \ x_2 \ \dots \ x_d ]^T$

Let  $\tilde{\mathbf{w}} = [ w_0, w_1, w_2, \dots, w_d ]^T$  represent all  $d + 1$  parameters

Model becomes  $f(\mathbf{x}) = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$

Sometimes, we'll use  $\mathbf{w}, \mathbf{x}, d$  for  $\tilde{\mathbf{w}}, \tilde{\mathbf{x}}, d + 1$

# Goal

- Goal is to minimize total error (empirical risk minimization):

$$\hat{R}_S(\tilde{\mathbf{w}}) = \frac{1}{n} \sum_{i=1}^n (f(\mathbf{x}_i) - y_i)^2 = \frac{1}{n} \sum_{i=1}^n (\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}} - y_i)^2.$$

- Define Residual Sum of Squares:

$$\text{RSS}(\tilde{\mathbf{w}}) = n\hat{R}_S(\tilde{\mathbf{w}}) = \sum_{i=1}^n (\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}} - y_i)^2.$$

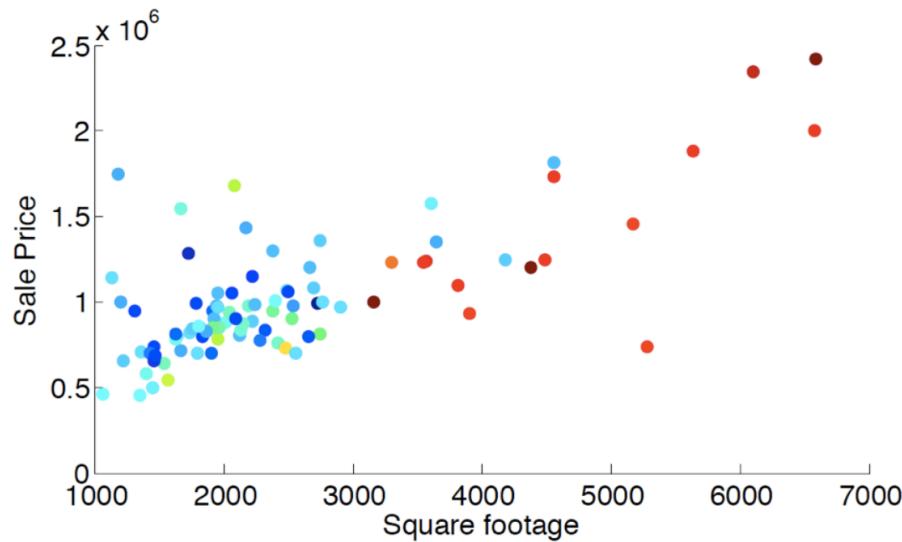
- Goal of empirical risk minimization:

$$\tilde{\mathbf{w}}^* = \underset{\tilde{\mathbf{w}} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \text{RSS}(\tilde{\mathbf{w}})$$

This is known as the **least squares solution**.

## Warmup: $d = 0$

Only one parameter  $w_0$ : constant prediction  $f(x) = w_0$



$f$  is a horizontal line, where should it be?

Warmup:  $d = 0$

$$\begin{aligned} RSS(w_0) &= \sum_{i=1}^n (w_0 - y_i)^2 \\ &= n w_0^2 - 2 \left( \sum_i y_i \right) w_0 + \sum_i y_i^2 \\ &= n \left( w_0 - \frac{1}{n} \sum_{i=1}^n y_i \right)^2 + (\text{constant wrt } w_0) \end{aligned}$$

$$w_0^* = \frac{1}{n} \sum_{i=1}^n y_i \quad (\text{the average})$$

Warmup:  $d = 1$

$$RSS(\tilde{w}) = \sum_i (w_0 + w_1 x_i - y_i)^2$$

General approach: find stationary points i.e. point with zero gradient

$$\frac{\partial RSS(\tilde{w})}{\partial w_0} = 0 \Rightarrow \cancel{\sum_i (w_0 + w_1 x_i - y_i)} = 0$$

$$\frac{\partial RSS(\tilde{w})}{\partial w_1} = 0 \Rightarrow \underline{n w_0 + w_1 \sum_i x_i = \sum_i y_i}$$

$$\cancel{\sum_i (w_0 + w_1 x_i - y_i) x_i} = 0$$

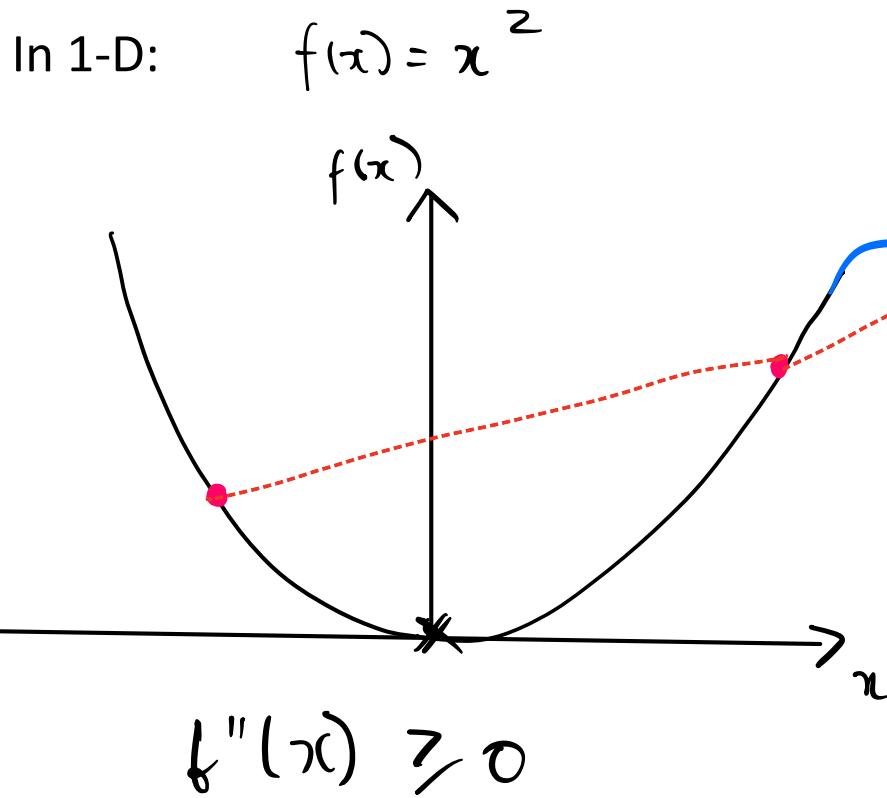
$$\underline{w_0 \sum_i x_i + w_1 \sum_i x_i^2 = \sum_i x_i y_i}$$

Warmup:  $d = 1$

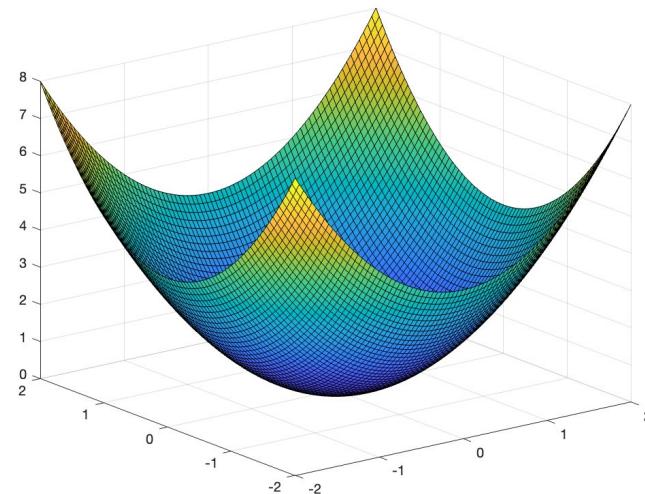
$$\begin{pmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} = \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix}$$
$$\begin{pmatrix} w_0^* \\ w_1^* \end{pmatrix} = \begin{pmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix}$$

# Are stationary points minimizers?

Yes, for **convex** objectives!



In high dimensions, this looks like:



$\nabla^2 F$  is positive-semi-definite

## General least square solution

$$RSS(\tilde{\omega}) = \sum_{i=1}^n (\tilde{x}_i^\top \tilde{\omega} - y_i)^2$$

Set  $\nabla RSS(\tilde{\omega}) = 0$

What is  $\nabla_w F(\omega)$  where  $F(\omega) = (\omega^\top w - y)^2$ ?

$$F(\omega) = \left( \sum_j v_j w_j - y \right)^2$$

$$\frac{\partial F}{\partial w_j} = 2 \left( \sum_j v_j w_j - y \right) v_j$$

$$\begin{aligned}\nabla_w F &= [2 \left( \sum_j (v_j w_j - y) \right) v_1, 2 \left( \sum_j (v_j w_j - y) \right) v_2, \dots] \\ &= 2 (\omega^\top w - y) v\end{aligned}$$

$$\nabla \text{RSS}(\tilde{\omega}) = 2 \sum_i (\tilde{x}_i^T \tilde{\omega} - \tilde{y}_i) \tilde{x}_i = 2 \sum_i \tilde{x}_i (\tilde{x}_i^T \tilde{\omega} - \tilde{y}_i)$$

$$= 2 \left( \sum_i \tilde{x}_i \tilde{x}_i^T \right) \tilde{\omega} - 2 \sum_i \tilde{x}_i \tilde{y}_i$$

$$\tilde{X} = \begin{pmatrix} \tilde{x}_1^T \\ \tilde{x}_2^T \\ \vdots \\ \tilde{x}_n^T \end{pmatrix} \in \mathbb{R}^{n+(d+1)}$$

$$Y = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_n \end{pmatrix} \in \mathbb{R}^n$$

$$\nabla \text{RSS}(\tilde{\omega}) = 2 \left( (\tilde{X}^T \tilde{X}) \tilde{\omega} - \tilde{X}^T Y \right) = 0$$

$$\tilde{\omega}^* = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T Y$$

(assume  $\tilde{X}^T \tilde{X}$  is invertible)

# Covariance matrix and understanding LS

$$\tilde{X}^T \tilde{X} = \begin{pmatrix} | & | & & | \\ \tilde{x}_1 & \tilde{x}_2 & \dots & \tilde{x}_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \text{---} & \tilde{f}_1 & \text{---} \\ \text{---} & \tilde{f}_2 & \text{---} \\ \vdots & & \vdots \\ \text{---} & \tilde{f}_n & \text{---} \end{pmatrix}$$

(Here, we assume all features are centered to be 0 mean, i.e.

$$\sum_{i=1}^m \tilde{x}_{i,j} = 0 \text{ for all features } j,$$

$\tilde{x}_{i,j}$  is the  $(i,j)$  entry of matrix  $\tilde{X}$ )

Suppose  $\tilde{X}^T \tilde{X} = I$ , then  $\tilde{w}^* = \tilde{X}^T \tilde{y}$

Each weight  $w_j$  is just the covariance of the  $j$ th feature with the label

## Another approach

RSS is a **quadratic**, so let's complete the square:

$$\begin{aligned}\text{RSS}(\tilde{\boldsymbol{w}}) &= \sum_i (\tilde{\boldsymbol{w}}^T \tilde{\mathbf{x}}_i - y_i)^2 = \|\tilde{\mathbf{X}}\tilde{\boldsymbol{w}} - \mathbf{y}\|_2^2 \\ &= (\tilde{\mathbf{X}}\tilde{\boldsymbol{w}} - \mathbf{y})^T (\tilde{\mathbf{X}}\tilde{\boldsymbol{w}} - \mathbf{y}) \\ &= \tilde{\boldsymbol{w}}^T \tilde{\mathbf{X}}^T \tilde{\mathbf{X}} \tilde{\boldsymbol{w}} - \mathbf{y}^T \tilde{\mathbf{X}} \tilde{\boldsymbol{w}} - \tilde{\boldsymbol{w}}^T \tilde{\mathbf{X}}^T \mathbf{y} + \text{cnt.} \\ &= \left( \tilde{\boldsymbol{w}} - (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{y} \right)^T (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}}) \left( \tilde{\boldsymbol{w}} - (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{y} \right) + \text{cnt.}\end{aligned}$$

**Note:**  $\mathbf{u}^T (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}}) \mathbf{u} = (\tilde{\mathbf{X}} \mathbf{u})^T \tilde{\mathbf{X}} \mathbf{u} = \|\tilde{\mathbf{X}} \mathbf{u}\|_2^2 \geq 0$  and is 0 if  $\mathbf{u} = 0$ .  
So  $\tilde{\boldsymbol{w}}^* = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{y}$  is the minimizer.

# Computational complexity

Bottleneck of computing

$$\tilde{w}^* = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T y$$

is to invert the matrix  $\tilde{X}^T \tilde{X} \in \mathbb{R}^{(d+1) \times \mathbb{R}^{(d+1)}}$ .

Takes time  $\mathcal{O}(d^3)$

# Optimization methods



## Problem setup

Given: a function  $F(\mathbf{w})$

Goal: minimize  $F(\mathbf{w})$  (approximately)

Two simple yet extremely popular methods

**Gradient Descent (GD):** simple and fundamental

**Stochastic Gradient Descent (SGD):** faster, effective for large-scale problems

Gradient is the *first-order information* of a function.

Therefore, these methods are called *first-order methods*.

# Gradient descent

**GD:** keep moving in the *negative gradient direction*

Start from some  $w^{(0)}$ . For  $t = 0, 1, \dots$

$$w^{(t+1)} = w^{(t)} - \eta \nabla_{w=w^{(t)}} F(w)$$

where  $\eta > 0$  is called step size or learning rate.

- in theory  $\eta$  should be set in terms of some parameters of  $f$
- in practice we just try several small values
- might need to be changing over iterations (think  $f(w) = |w|$ )
- adaptive and automatic step size tuning is an active research area

# An example

Consider squared loss on one datapoint  $(x, y)$  where  $x = (x^{(1)}, x^{(2)})$  for  $\mathbf{w} = (w_1, w_2)$ .

$$F(\mathbf{w}) = (w_1 x^{(1)} + w_2 x^{(2)} - y)^2.$$

Gradient is

$$\frac{\partial F}{\partial w_1} = 2(w_1 x^{(1)} + w_2 x^{(2)} - y) \cdot x^{(1)} \quad \frac{\partial F}{\partial w_2} = 2(w_1 x^{(1)} + w_2 x^{(2)} - y) \cdot x^{(2)}$$

GD:

- Initialize  $w_1^{(0)}$  and  $w_2^{(0)}$  (to be 0 or *randomly*),  $t = 0$
- do

$$w_1^{(t+1)} \leftarrow w_1^{(t)} - \eta \left[ 2(w_1 x^{(1)} + w_2 x^{(2)} - y) \cdot x^{(1)} \right]$$

$$w_2^{(t+1)} \leftarrow w_2^{(t)} - \eta \left[ 2(w_1 x^{(1)} + w_2 x^{(2)} - y) \cdot x^{(2)} \right]$$

$$t \leftarrow t + 1$$

- until  $F(\mathbf{w}^{(t)})$  does not change much or  $t$  reaches a fixed number

# Switch to Colab

optimization.ipynb ☆

File Edit View Insert Runtime Tools Help

+ Code + Text

```
# optimization.ipynb
# This notebook illustrates gradient descent for a two-dimensional objective function.
```

This code defines a gradient descent loop and a plot of the objective function. The plot shows contour lines and a red arrow indicating the direction of the gradient step.

```
    this_theta[1] = last_theta[1] - eta * grad1
    theta.append(this_theta)
    J.append(cost_func(*this_theta))

    # Annotate the objective function plot with coloured points indicating the
    # parameters chosen and red arrows indicating the steps down the gradient.
    for j in range(1,N):
        ax.annotate('', xy=theta[j], xytext=theta[j-1],
                    arrowprops={'arrowstyle': '->', 'color': 'orange', 'lw': 1},
                    va='center', ha='center')
    ax.scatter(*zip(*theta), facecolors='none', edgecolors='r', lw=1.5)

    # Labels, titles and a legend.
    ax.set_xlabel(r' $w_1$ ')
    ax.set_ylabel(r' $w_2$ ')
    ax.set_title('objective function')

plt.show()
```

objective function