# Welcome to CSE446 / 546!

Fall 2025

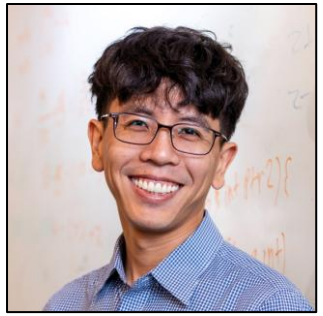
Sewoong Oh & Pang Wei Koh

hello!

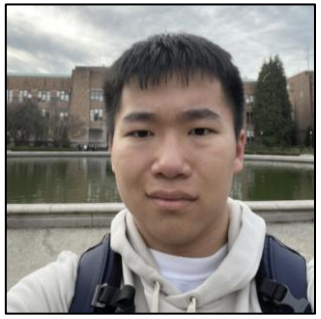
# Course staff

cse446-staff@cs.washington.edu
https://courses.cs.washington.edu/courses/cse446/25au/

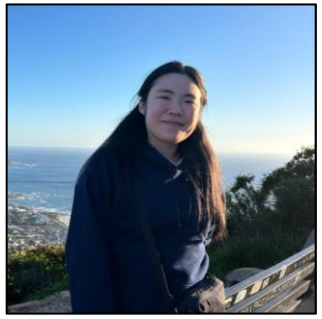

Sewoong


Pang Wei

Donovan


Anthony


Ariel

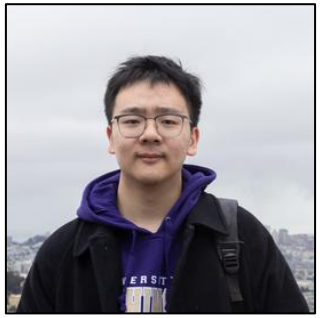

Cynthia

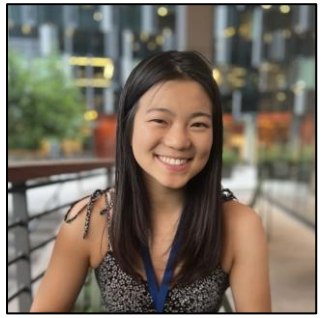

Emmanuel


Leo


Sankar

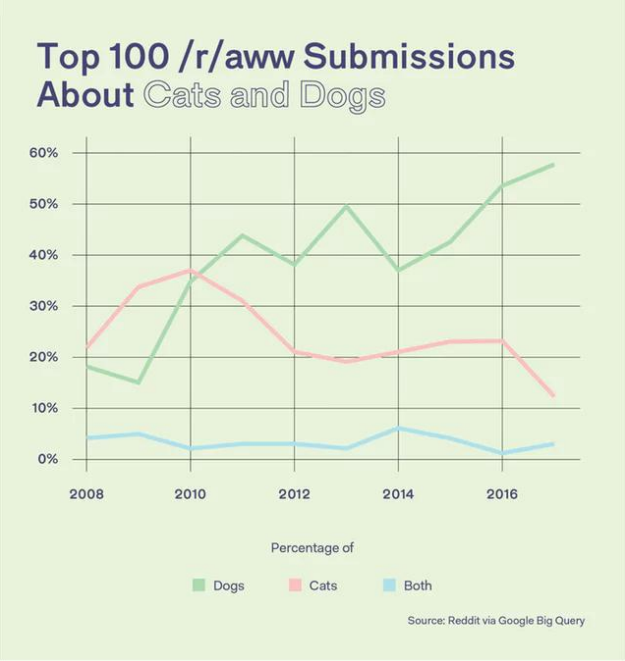

Varun


Yichuan

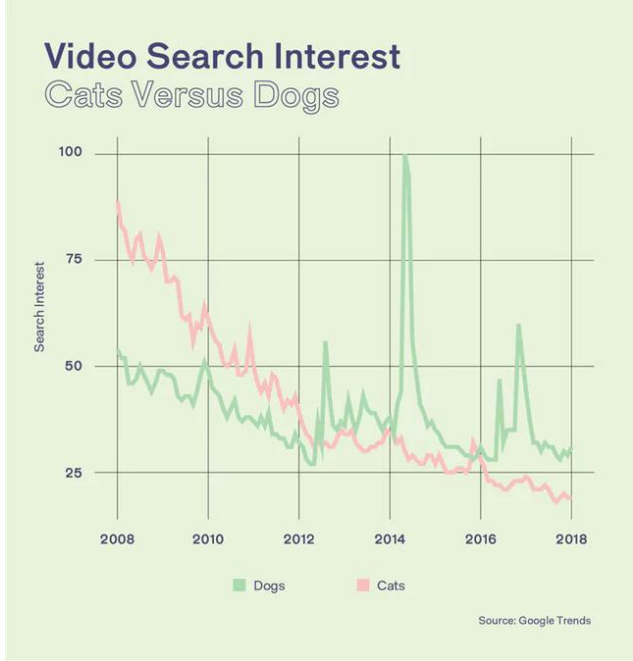

Yufei

# Classifying cats vs. dogs

## Reddit




## Google



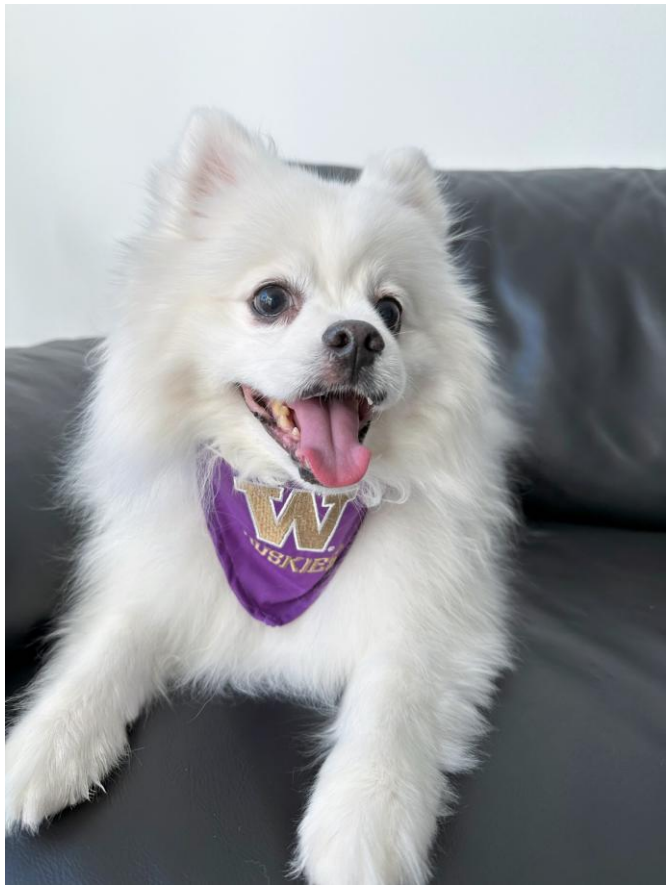

## Twitter?

Write a program that classifies tweets into “cat” or “dog”

```
cats = []
dogs = []
other = []
for tweet in tweets:
    if "cat" in tweet:
        cats.append(tweet)
    elif "dog" in tweet:
        dogs.append(tweet)
    else:
        other.append(tweet)
return cats, dogs, other
```

Graphics courtesy of https://theoutline.com/post/3128/dogs-cats-internet-popularity?zd=1

# Classifying cats vs. dogs



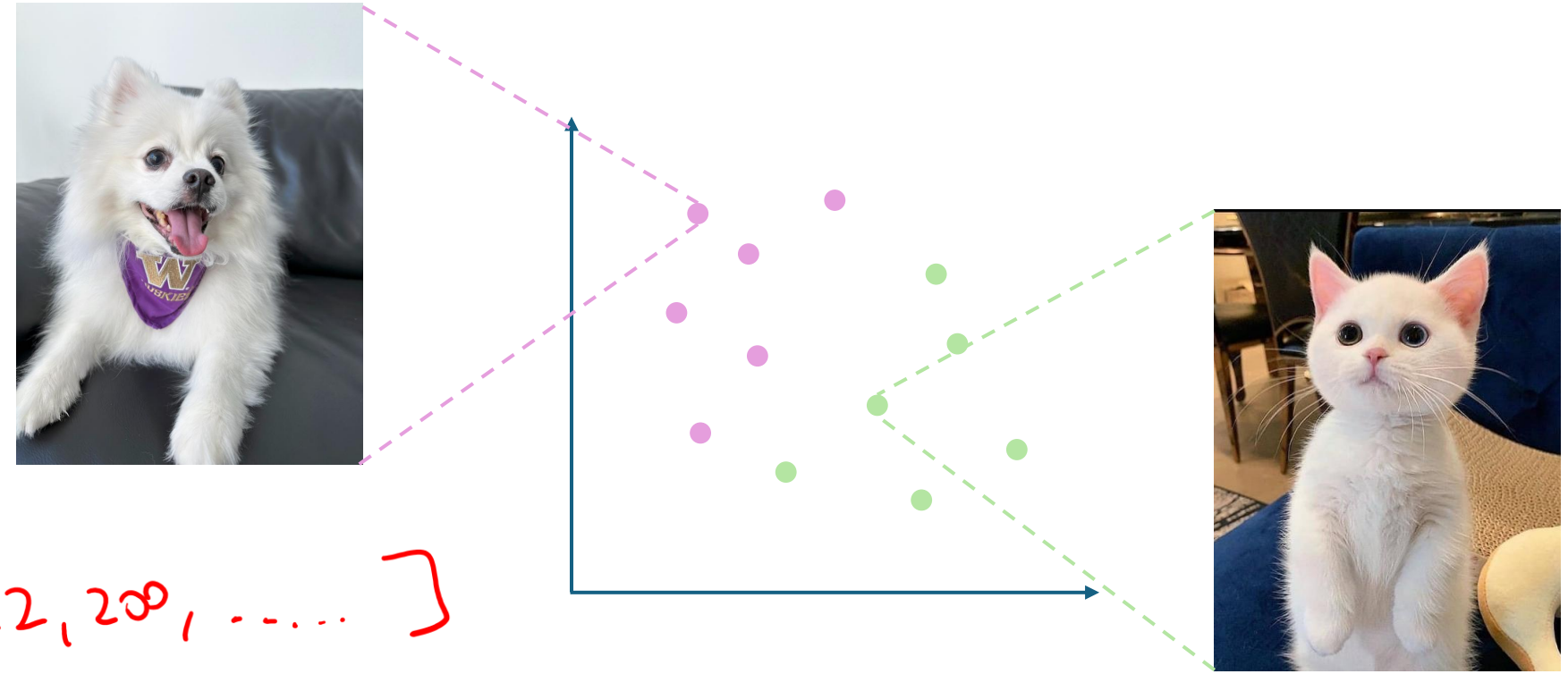
Is this a dog or a cat?

```
if "dog" in image…?
```

https://www.reddit.com/r/aww/comments/odgawj/can_somebody_explain_why_this_cat_look_so_cute/

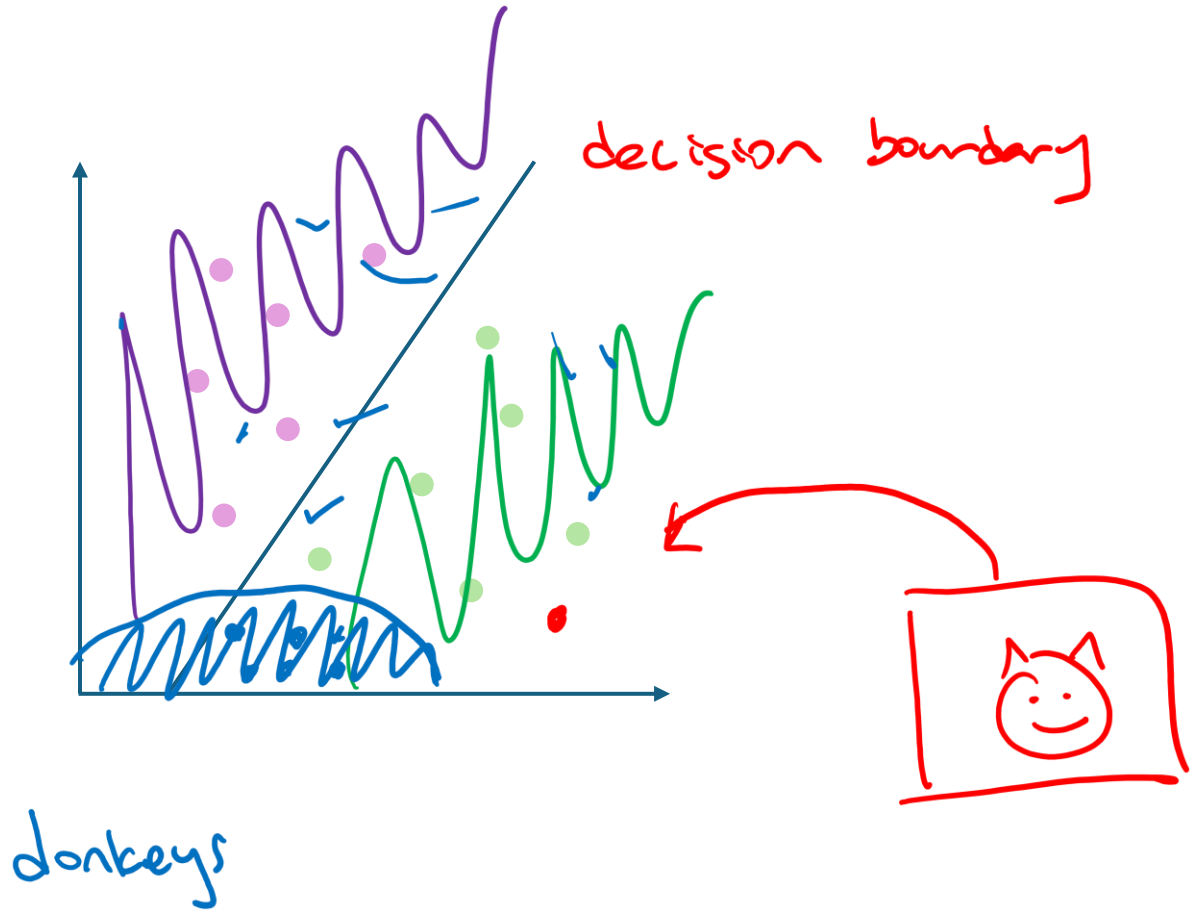
# Classifying cats vs. dogs

Key idea of machine learning: **Learn from data**



# Classifying cats vs. dogs

Key idea of machine learning: **Learn from data**



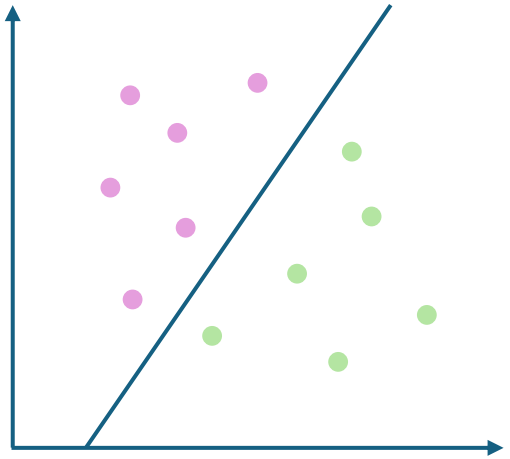
# Machine learning

GOFAI

- Traditional AI: **Rules**-based, hard-coded by experts

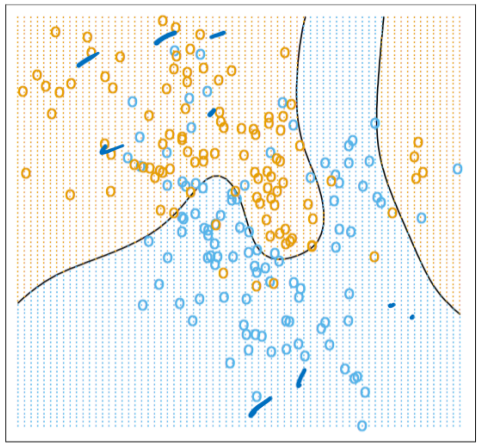
```
if "dog" in tweet
```

- Machine learning: Decisions learned from **data**



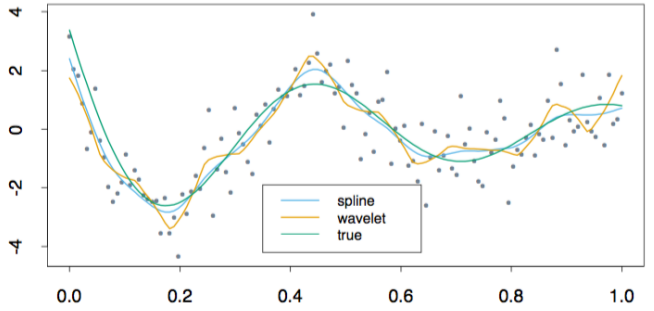
# Machine learning ingredients

- **Data:** Past observations
- **Models:** Devised to capture the patterns in data
- **Predictions:** Apply model to predict future observations



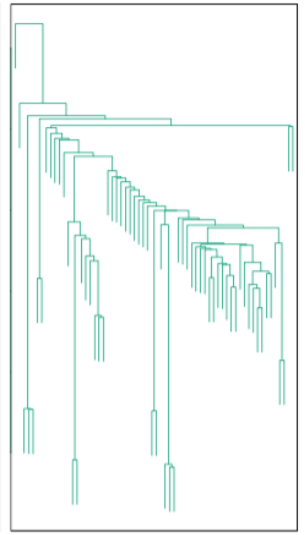
Predict categorical value:
loan or not? spam or not? what disease is this?

classification



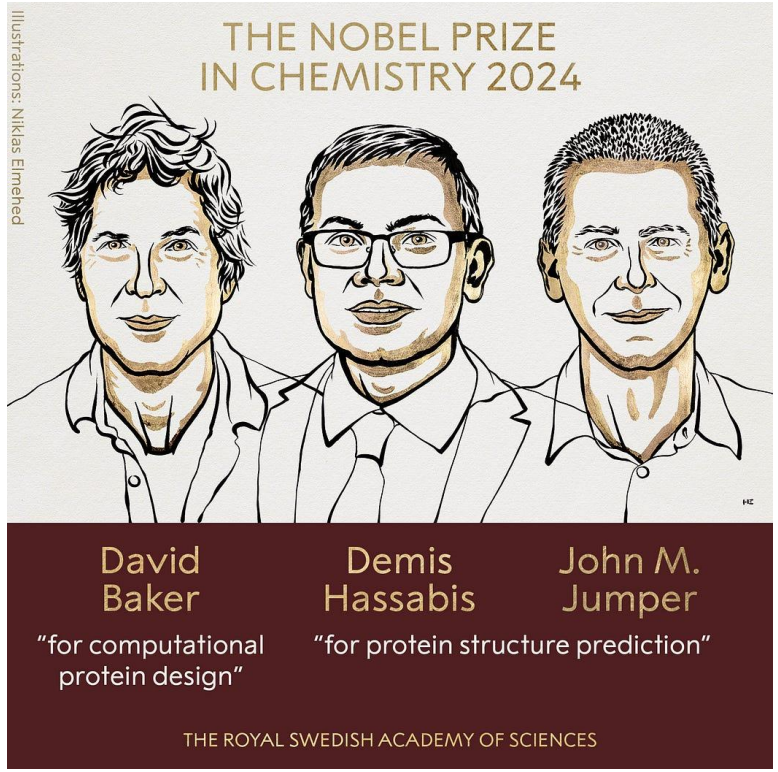

Predict continuous value:
ex: stock market, credit score, temperature, Netflix rating

regression



Predict structure:
tree of life from DNA, find similar images, community detection

# Machine learning is the basis of modern AI

## What this class is about:

- **Fundamentals of ML:** Supervised learning, unsupervised learning, bias/variance tradeoff, overfitting, optimization and computational tradeoffs…
- **Preparation for further learning:** AI is fast-moving; you will learn the foundations of ML to understand the latest results

## What this class is not:

- **A survey course:** Laundry list of algorithms, how to win Kaggle
- **An applied course:** No details of how to implement the latest LLMs, image models, etc.
- **An easy course:** Mathematical maturity and familiarity assumed, homework will be time-consuming

# Prerequisites

- Linear algebra
  - Linear dependence, rank, linear equations
  - Eigenvectors
- Multivariate calculus
- Probability and statistics
  - Distributions, densities, marginalization, moments
- Algorithms
  - Basic data structures, complexity

- Not enforced; Use HW0 to judge readiness
- See assigned reading and website for additional review materials!

# Course registration

- All CSE course registration processes are managed centrally by CSE
- Resources:
  - https://www.cs.washington.edu/academics/ugrad/advising/
  - https://www.cs.washington.edu/academics/undergraduate/non-major-options/
  - https://www.cs.washington.edu/academics/phd/advising/

# Lectures and exams

- Will be recorded on Panopto and uploaded to Canvas
- In-person attendance is encouraged!
- **For exams, in-person attendance is mandatory.**
  - **Midterm:** Tuesday 10/28, 10:00am -- 11:20am.
  - **Final:** Monday 12/8, 10:30am -- 12:20pm.
  - We will work with DRS on accommodations
  - We cannot schedule alternate exams; see course FAQ for more info

# CSE 446 vs. 546

| Course | Section | Homework | Grading |
| --- | --- | --- | --- |
| 446 | Attend the section you are registered for. | A problems only. No credit will be rewarded for completing B problems. | You will be graded/curved against your peers in 446 only (on a 4.0 scale). |
| 546 | Optional; you can attend any section of your choice. | A and B problems. | You will be graded/curved against your peers in 546 only.<br><br>Exams will be the same as 446, but you will be graded against 546 students. |

# Grading

- 5 homeworks (40%)
  - Each contains both theoretical and programming questions.
  - Collaboration okay. You must write and understand your answers and code.
  - Do not Google for answers or ask chatGPT to do it.
  - Read collaboration policy on website
- Exams (60%):
  - Midterm (20%)
  - Final (40%)
- Separate grading curves for 446 and 546

# Homeworks

- HW 0 is out (Due Wednesday 10/8 at 11:59pm)
  - Should be review (but being rusty is expected)
  - Work individually, treat as guide for what to brush up on
- HW 1,2,3,4. They are not easy or short. Start early.
- Assignments due at 11:59pm, submit early (to Gradescope) and often
- Late days:
  - 5 days total over the quarter; no more than 2 per assignment.
  - Exceptions only for extremely extenuating circumstances

1. All code must be written in Python
2. All written work must be typeset (e.g., LaTeX)

See course website for tutorials and references.

# Office hours

- https://courses.cs.washington.edu/courses/cse446/25au/officehours/
- We will post schedule updates to EdStem
- More OH on Mon/Tue/Wed because of homework deadlines
- In general:
  - Go to TA office hours for specific homework help
  - Go to any office hours to discuss lecture concepts
  - Go to faculty office hours to talk about ML research etc.

# Communication channels

- Announcements, questions about class, homework help
  - https://edstem.org/us/courses/84662/discussion
  - “I think there is a typo in the homework?”
  - “What does this notation mean?”
  - “Is this an accurate description of how this works?
- Personal concerns (cse446-staff@cs.washington.edu)
  - “Was in hospital…”, “Laptop was stolen…”
- Office hours
  - “How do I get started on problem 2?”
  - “Am I on the right track?”
  - “I have this problem at work—can you point me in the right direction?”
- Anonymous feedback (https://feedback.cs.washington.edu/)
  - “Your real-world example X lacked nuance. I would like you to…”

Please do not email instructors directly

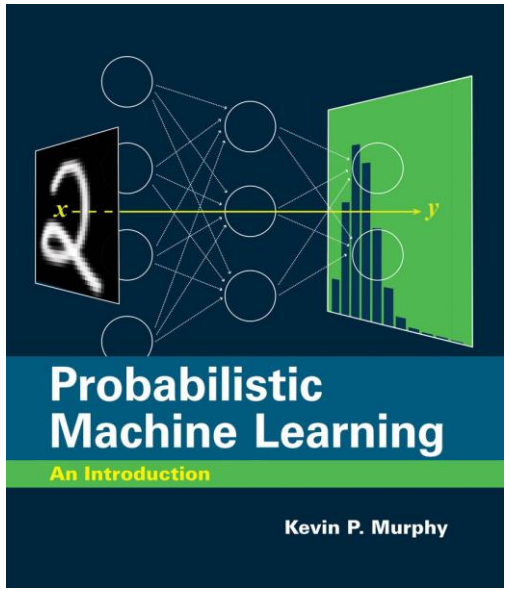
# Other resources

- Free PDF textbook:

  Probabilistic Machine Learning: An Introduction,
  Kevin Murphy

  (also in print)

- Many more resources on the website & web!



# Enjoy!

- ML is:
  - Ubiquitous in science, engineering, and beyond
  - Transforming the world.
- This class will give you a basic foundation for understanding and applying ML.

# Probability Review

# Definitions

- **Random variable**: A variable that takes on different values determined randomly.
  - Example: The height of a person from the US.
- **Distribution**:  The different values a random variable can take on along with the probability of that value.
- We talk about **sampling** from a distribution:
  - “Consider a sample of 100 different heights drawn randomly from the distribution of all heights of people from the US.”

# Random variables and events

- Let $X$ and $Y$ be random variables
  - e.g., $X$ is the outcome of the first roll of a 6-sided die, $Y$ is the outcome of the second roll
- An **event** is a statement about the world that holds or not:
  - $A = \{X \in \{3,4\}\}$
  - $B = \{X = 1\}$
  - $C = \{Y \in \{3,4\}\}$
- Each event is assigned a probability
- Union:

$P(A) = 2/6 = 1/3$

$U, V$ are events

$P(U \cup V) = P(U) + P(V) - P(U \cap V)$

$\leq P(U) + P(V)$ "union bound"

# Independence

- Let $X$ and $Y$ be random variables
  - e.g., $X$ is the outcome of the first roll of a 6-sided die, $Y$ is the outcome of the second roll
- An **event** is a statement about the world that holds or not:
  - $A = \{X \in \{3,4\}\}$
  - $B = \{X = 1\}$
  - $C = \{Y \in \{3,4\}\}$
- Any pair of events $U, V$ are independent if: $P(U \cap V) = P(U) \times P(V)$

A & B indp?

$P(A \cap B) = 0$

$P(A) = 1/3$

$P(B) = 1/6$

A & C indp? Y

B & C indp? Y

# Conditional probability

- Let $X$ and $Y$ be random variables
  - e.g., $X$ is the outcome of the first roll of a 6-sided die, $Y$ is the outcome of the second roll
- An **event** is a statement about the world that holds or not
- Conditional probability of event $U$ given event $V$:

$$P(U|V) = \frac{P(U \wedge V)}{P(V)}$$

if $U$ & $V$ indp:

$$P(U|V) = \frac{P(U \wedge V)}{P(V)} = \frac{P(U)\,P(V)}{P(V)}$$

$$P(V)\,P(U|V) = P(U \wedge V)$$

# Mean, variance, median

**Mean** — The expected value of X; each value is weighted by the probability of seeing it.

$\mathbb{E}[X]$, $E[X]$, $\mu$

$$\mathbb{E}[X] = \sum_x P(X = x)\ x$$

**Variance** — The expected squared deviation of X from its mean.

$\mathrm{Var}(X)$, $\sigma^2$

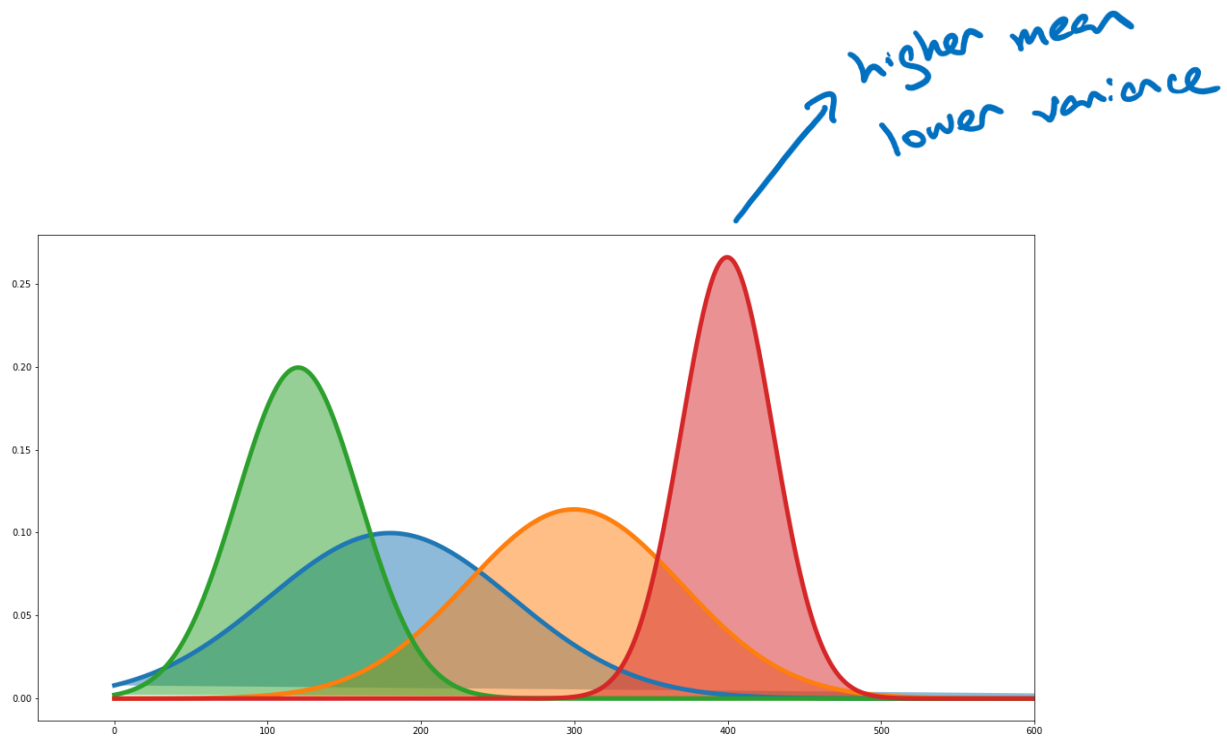
$$\mathrm{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$$
$$= \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

**Median** — The value of X that is separating the higher half of its range from the lower half.

$m$

$$P(X \leq m) = 0.5$$

# Mean, variance, median



# Maximum Likelihood Estimation

# Your first ML job

- *Client:* I have a special coin. If I flip it, what's the probability it'll be heads?
- *You (a machine learner):* I need to collect data…

HH

- *You:* The probability is:

100% H

# Your first ML job

- *Client:* …you sure about that? I just got a tails.
- *You:* I need to collect more data!

HHTTH

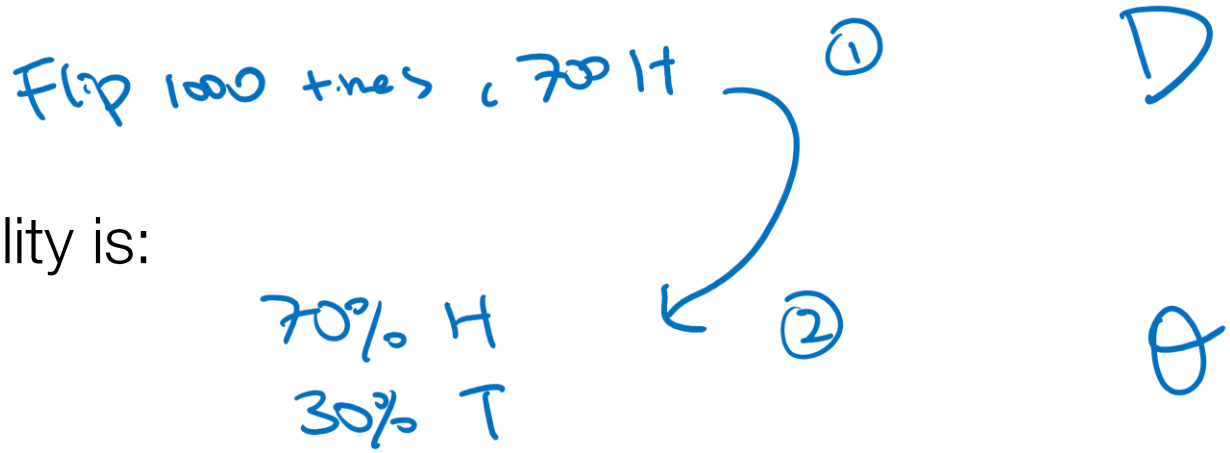
- *You:* The probability is:

60% H

40% T

# Your first ML job

- *Client:* …you sure about that? I just got a tails.
- *You:* I need to collect more data!




- *You:* The probability is:
- *Client:* Why should I believe you?
- *You:* Let's do some math!

# Coin flipping – Bernoulli distribution

- Data: sequence $D = (HHTHT \ldots)$, $k$ heads out of $n$ flips
- Model / hypothesis:
  - Flips are i.i.d.

$P(H) = \theta \in [0, 1]$, $P(T) = 1 - \theta$

Independent

Identically Distributed

$P(H|\theta)\, P(H|\theta, H)\, P(T|\theta, HH) \ldots$

$\Rightarrow P(H|\theta)\, P(H|\theta)\, P(T|\theta) \ldots$

$\Rightarrow \theta^k (1-\theta)^{n-k}$

- Likelihood:

$P(D|\theta) = \theta^k (1-\theta)^{n-k}$

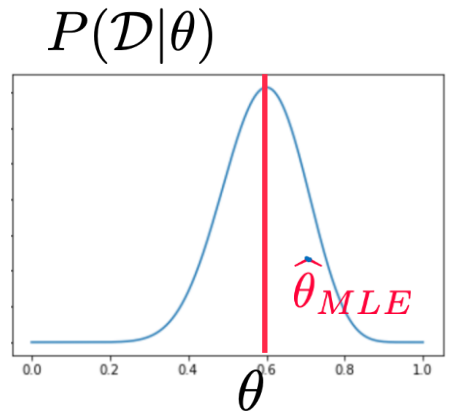
# Maximum likelihood estimation (MLE)

- **Data:** sequence $D = (HHTHT \ldots)$, $k$ heads out of $n$ flips
- **Model / hypothesis:** $P(\mathrm{H})=\theta, P(T) = 1 - \theta$
- **Likelihood:** $P(D|\theta) = \theta^k(1-\theta)^{n-k}$
- **Maximum likelihood estimation:** Choose model ($\theta$) that maximizes the likelihood of the observed data.

$$\hat{\theta}_{MLE} = \arg\max_{\theta} P(D|\theta)$$

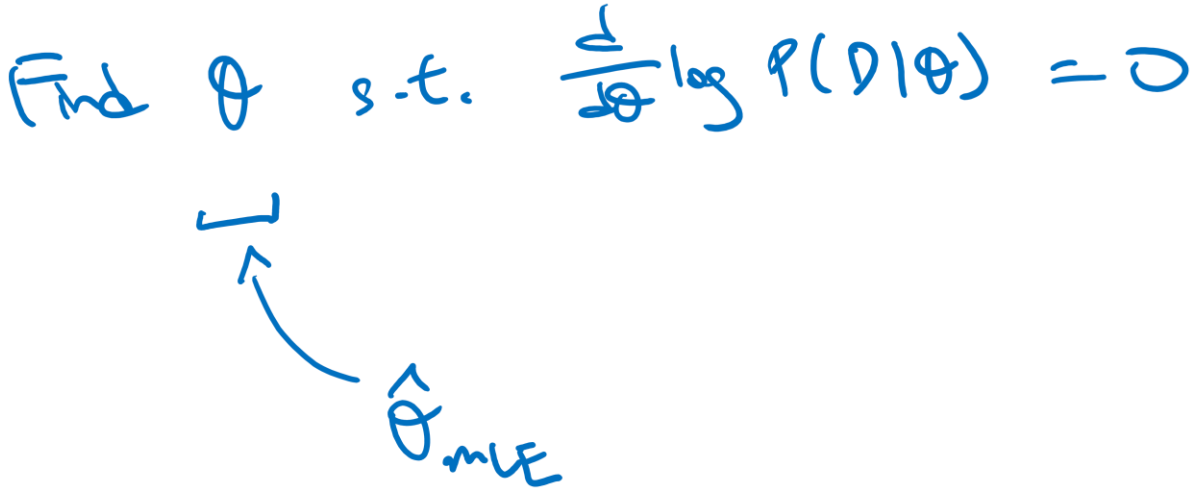
$$= \arg\max_{\theta} \log P(D|\theta)$$

"log-likelihood"



# MLE: Our first learning algorithm

- How do we find $\theta$ that maximizes likelihood?
- Use fact that derivative is 0 at local maxima/minima



# MLE: Our first learning algorithm

- First, manipulate likelihood to make it easy to work with:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \log P(D|\theta)$$

$$= \arg\max_{\theta} \log \left[ \theta^k (1-\theta)^{n-k} \right]$$

$$= \arg\max_{\theta} \underbrace{k \log\theta + (n-k)\log(1-\theta)}_{L(\theta)}$$

- Then set derivative to 0:

$$\frac{d}{d\theta} L(\theta) = \frac{k}{\theta} - \frac{n-k}{1-\theta} = 0$$

$$\Rightarrow k - k\theta = n\theta - k\theta$$

$$\Rightarrow \hat{\theta}_{MLE} = \frac{k}{n}$$

HHTTH 3/5 = 60%

# How good is the MLE?

$p_{\theta^*}$ = true data distribution

$D \sim p_{\theta^*}$

- $\hat{\theta}_{MLE}$ is a random variable
- We assume there is a ground truth parameter $\theta^*$ that generates the data $D = (HHTHT \ldots)$ of fixed size $n$
- What can we say about this random variable $\hat{\theta}_{MLE}$? → $k/n$
- Expectation describes how the estimator behaves on average

$$\text{Bias}(\hat{\theta}_{MLE}) = \mathbb{E}_{D \sim p_{\theta^*}}[\hat{\theta}_{MLE}] - \theta^*$$

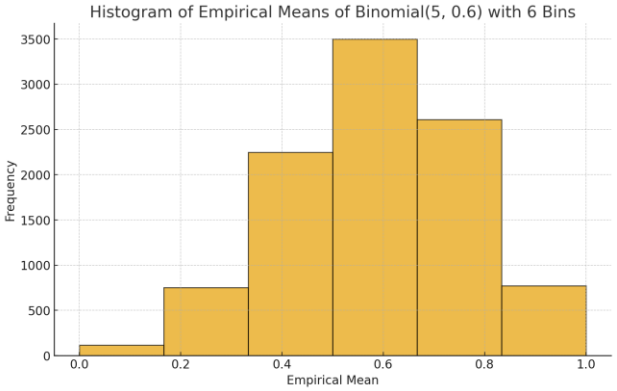
$$= \mathbb{E}_{D \sim p_{\theta^*}}\left[\frac{k}{n}\right] - \theta^* = \theta^* - \theta^* = 0$$

- $\hat{\theta}_{MLE}$ is **unbiased**

# How many flips do I need?

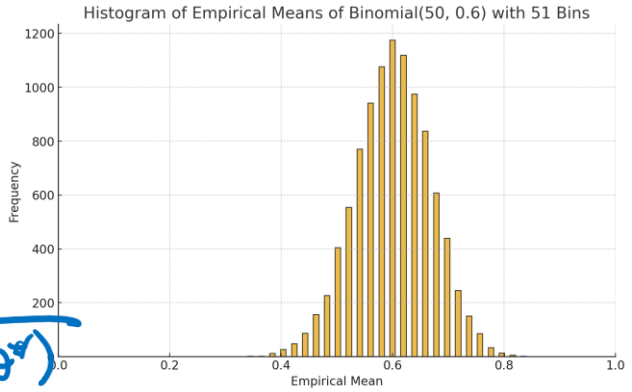
- Consider running many experiments with $\theta^* = \frac{3}{5}$ and observe many instances of $\hat{\theta}_{MLE} = \frac{k}{n}$
- *Client:* I flipped the coin 5 times and got 2 heads.

$\hat{\theta}_{MLE} = 2/5$




- *Client:* I flipped the coin 50 times and got 30 heads.

$\hat{\theta}_{MLE} = 60\%$

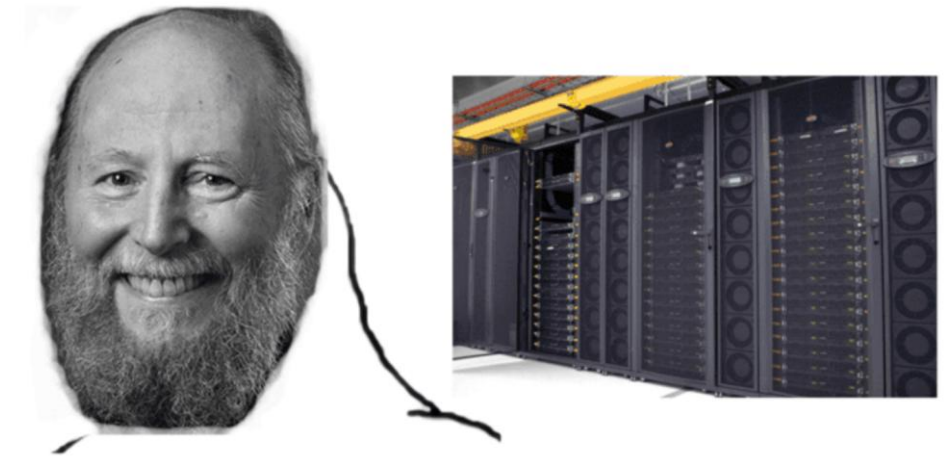



- *Client:* They are both unbiased, which one is right?
- Variance goes down with larger $n$

$Var(\hat{\theta}_{MLE}) = \sqrt{\frac{\theta^*(1-\theta^*)}{n}}$

# Fundamental machine learning truth

- More data -> better performance
  - “The bitter lesson”
  - https://www.cs.utexas.edu/~eunsol/courses/data/bitter_lesson.pdf



# Maximum likelihood estimation

**Observe** $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

**Likelihood function** $L_n(\theta) = \prod_{i=1}^{n} f(X_i;\theta)$

**Log-Likelihood function** $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^{n} \log(f(X_i;\theta))$

**Maximum Likelihood Estimator (MLE)** $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

# Recap: Why is this machine learning?

Learning is:

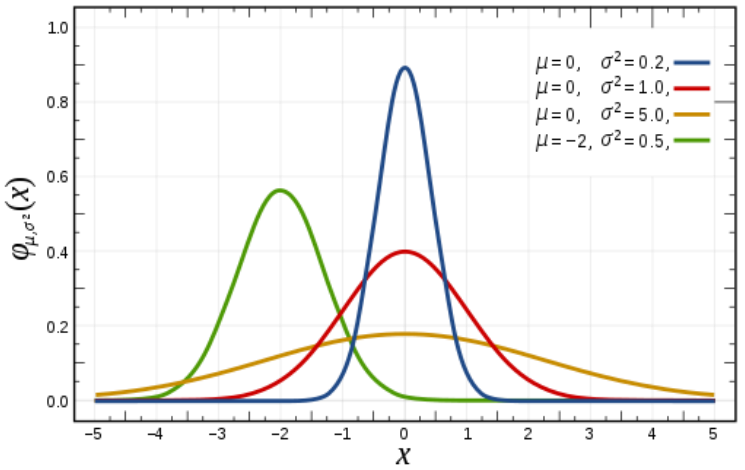
- Collect some data
  - e.g., coin flips
- Choose a model/hypothesis class
  - e.g., Bernoulli
- Choose a loss function
  - e.g., data likelihood
- Choose an optimization procedure
  - e.g., set derivative to zero to obtain MLE

Using the model $\theta$ we can now predict future observations

# What about continuous variables?

- *Client:* What if I am measuring a continuous variable?
- *You:* Let me tell you about Gaussians…

$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$



# Some convenient properties of Gaussians

- Affine transformation (multiplying by scalar and adding a constant)
  - $X \sim N(\mu, \sigma^2)$
  - $Y = aX + b$

$$Y \sim N(a\mu + b, a^2\sigma^2)$$

- Sum of Gaussians
  - $X \sim N(\mu_X, \sigma^2_X)$
  - $Y \sim N(\mu_Y, \sigma^2_Y)$
  - $Z = X+Y$

$$Z \sim N(\mu_X + \mu_Y, \sigma^2_X + \sigma^2_Y)$$

# MLE for Gaussians

- **Data:** i.i.d. samples $D = \{x_1, x_2, \ldots, x_n\}$ (e.g., temperature)

$$P(D \mid \mu, \sigma) = P(x_1, \ldots, x_n \mid \mu, \sigma)$$

$$= \prod_{i=1}^{n} P(x_i \mid \mu, \sigma)$$

$$= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \prod_{i=1}^{n} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

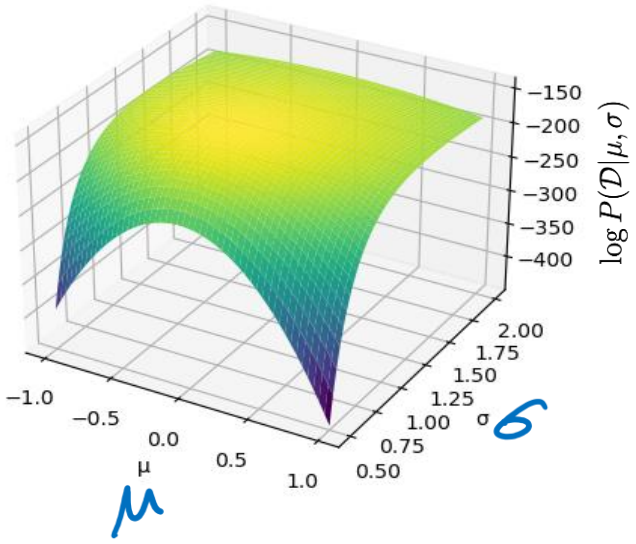
log-likelihood

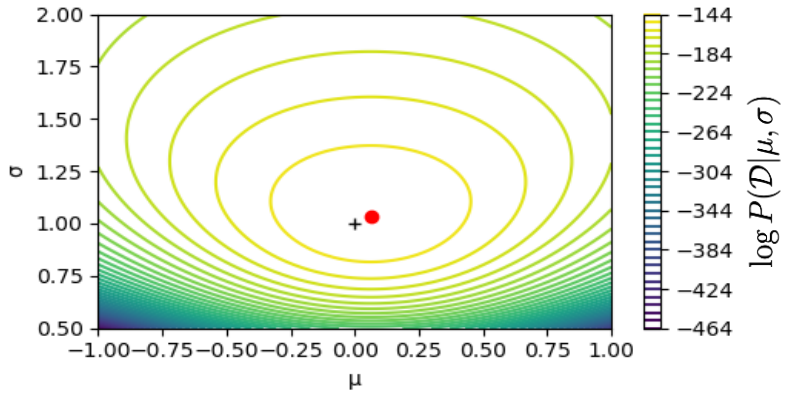
$$\log\left(P(D \mid \mu, \sigma)\right) = -n\log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2}$$

# MLE for Gaussians

$$\text{Generate } \mathcal{D} = \{x_1, ..., x_n\}, \text{ where}$$
$$n = 100$$
$$x_i \sim \mathcal{N}(\mu, \sigma^2)$$
$$\mu = 0$$
$$\sigma^2 = 1$$

$$\log P(\mathcal{D}|\mu, \sigma) = -n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2}$$




+ $(\mu_{True}, \sigma_{True})$

● $(\hat{\mu}_{MLE}, \hat{\sigma}_{MLE})$

# MLE for mean of a Gaussian

- Set partial derivative to 0:

$$\frac{\partial}{\partial \mu} \log P(\mathcal{D} \mid \mu, \sigma) = \frac{\partial}{\partial \mu} \left[ -n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2} \right]$$

# MLE for variance of a Gaussian

- Again, set partial derivative to 0:

$$\frac{\partial}{\partial \sigma} \log P(\mathcal{D} \mid \mu, \sigma) = \frac{\partial}{\partial \sigma} \left[ -n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2} \right]$$

# Learning Gaussian parameters

$$\widehat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\widehat{\sigma^2}_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{\mu}_{MLE})^2$$

$$\mathbb{E}[\widehat{\sigma^2}_{MLE}] \neq \sigma^2$$

# Maximum likelihood estimation

**Observe** $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

**Likelihood function** $L_n(\theta) = \prod_{i=1}^{n} f(X_i;\theta)$

**Log-Likelihood function** $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^{n} \log(f(X_i;\theta))$

**Maximum Likelihood Estimator (MLE)** $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

The MLE is a "recipe" that begins with a model for data $f(x;\theta)$

Under benign assumptions, as $n \to \infty$, we have $\hat{\theta}_{MLE} \to \theta^*$