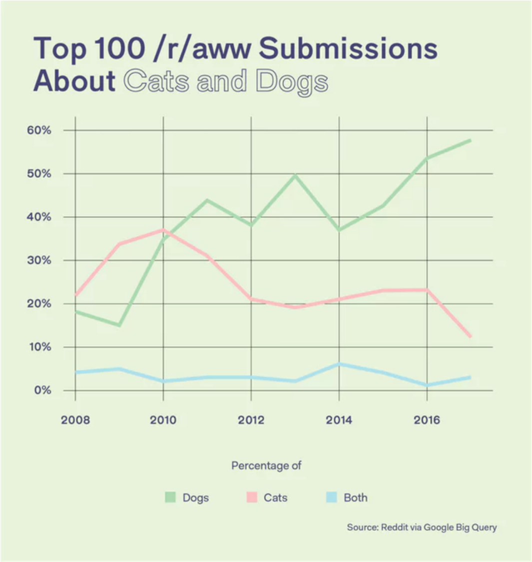
CSE 446: Machine Learning

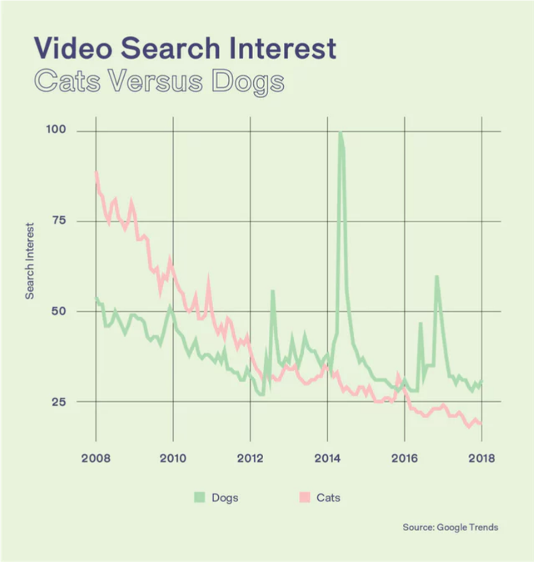
Jamie Morgenstern

Traditional Algorithms

Social media mentions of Cats vs. Dogs

Reddit Google Twitter?



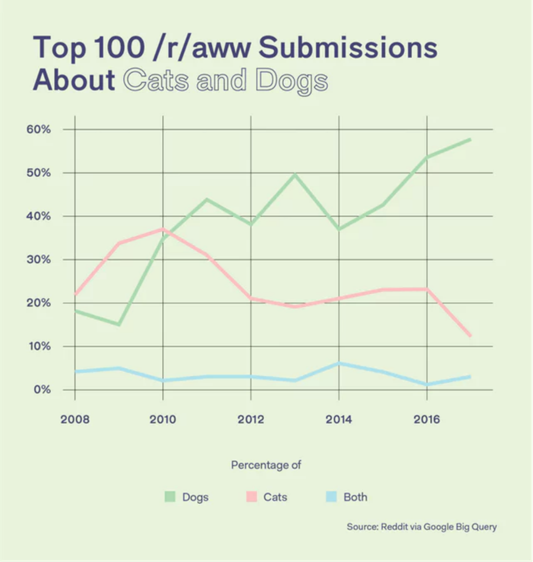


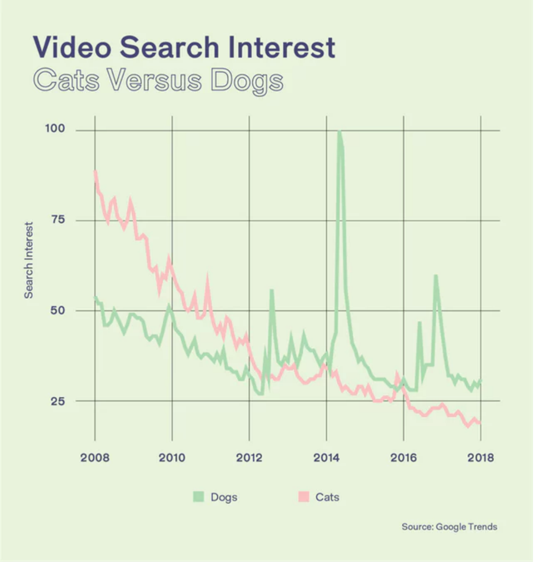
Graphics courtesy of https://theoutline.com/post/3128/dogs-cats-internet-popularity?zd=1

Traditional Algorithms

Social media mentions of Cats vs. Dogs

Reddit Google Twitter?





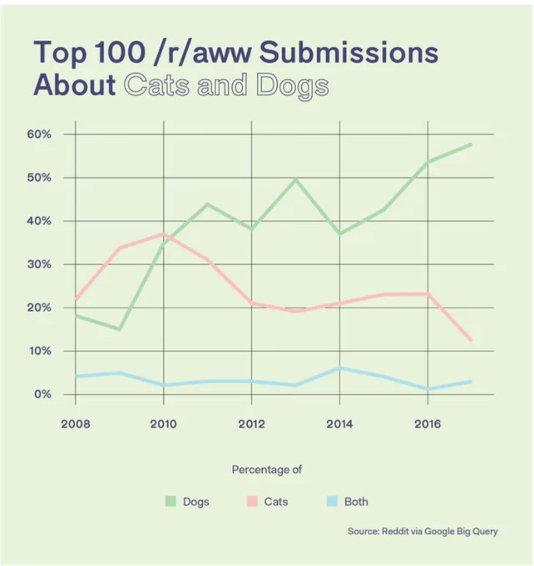
Write a program that sorts tweets into those containing "**cat**", "**dog**", or ***other***

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

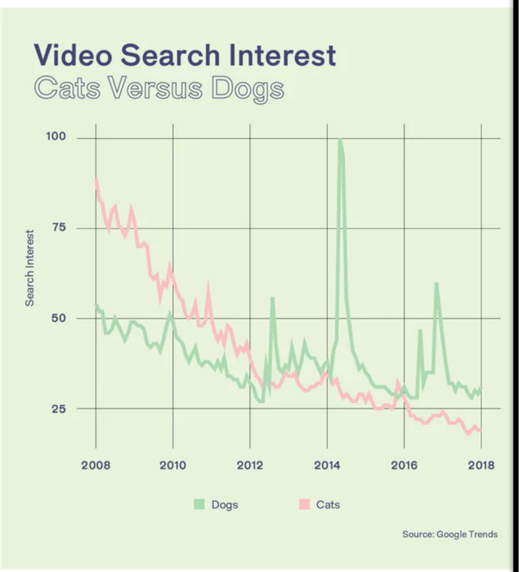
Traditional Algorithms

Social media mentions of Cats vs. Dogs

Reddit



Google



Twitter?

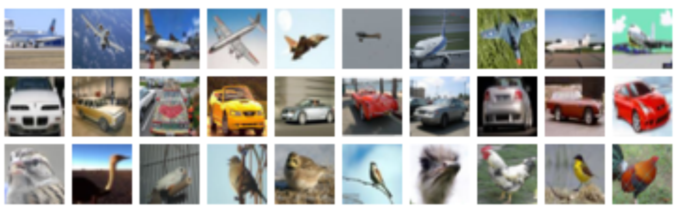
```
cats = []
dogs = []
other = []
for tweet in tweets:
    if "cat" in tweet:
        cats.append(tweet)
    elseif "dog" in tweet:
        dogs.append(tweet)
    else:
        other.append(tweet)
return cats, dogs, other
```

Write a program that sorts tweets into those containing "**cat**", "**dog**", or ***other***

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

Machine Learning algorithms

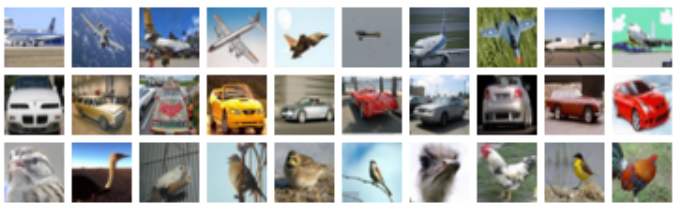
Write a program that sorts images
into those containing "**birds**",
"**airplanes**", or ***other.***



airplane
other
bird

Machine Learning algorithms

Write a program that sorts images
into those containing “**birds**”,
“**airplanes**”, or ***other.***

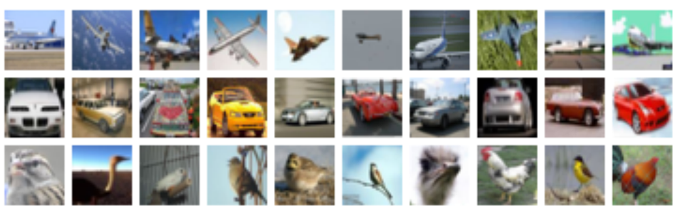


airplane
other
bird

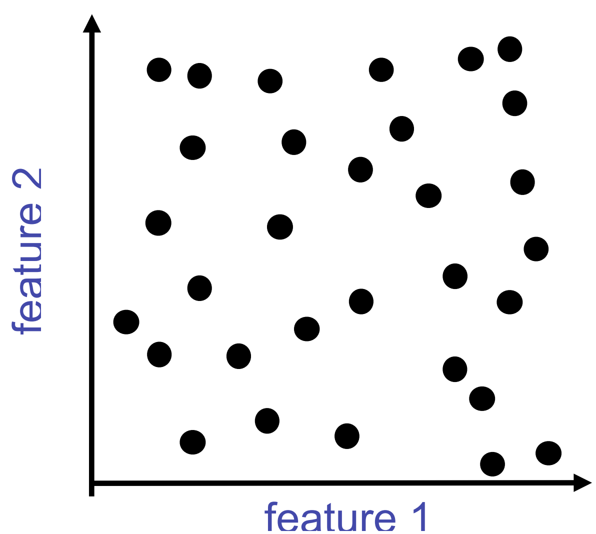
```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elseif plane in image:
        planes.append(image)
    else:
        other.append(tweet)
return birds, planes, other
```

Machine Learning algorithms

Write a program that sorts images
into those containing "**birds**",
"**airplanes**", or ***other.***



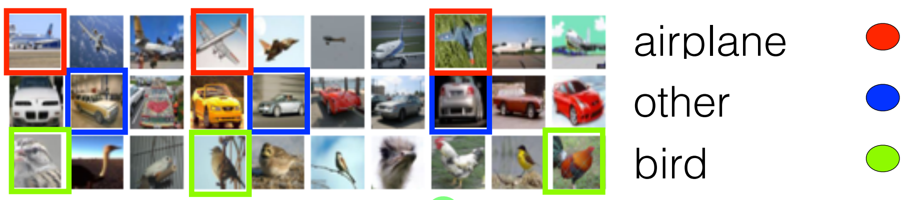
airplane
other
bird

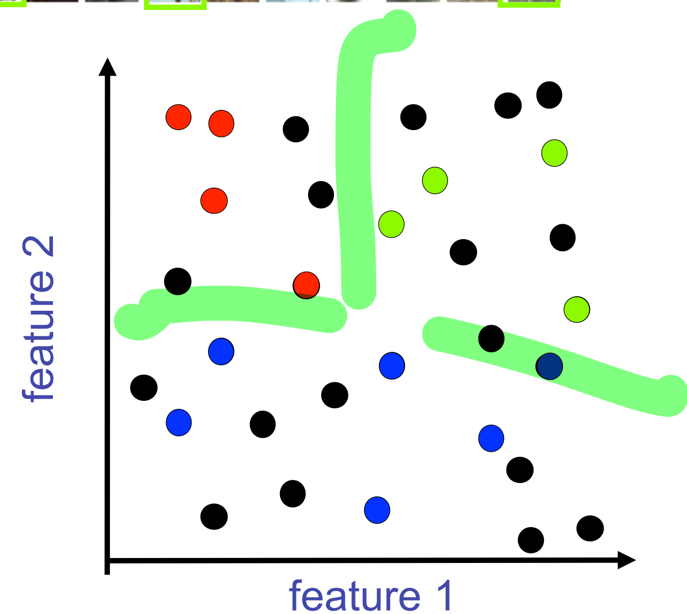



```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elseif plane in image:
        planes.append(image)
    else:
        other.append(tweet)
return birds, planes, other
```

Machine Learning algorithms

Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***

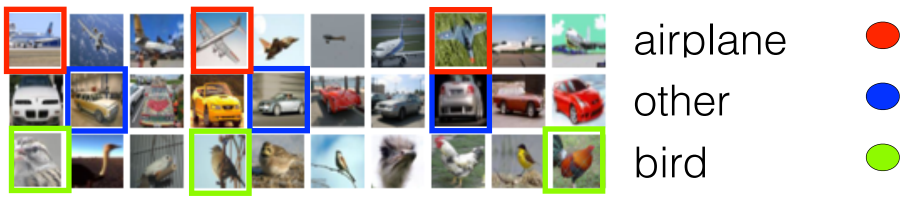


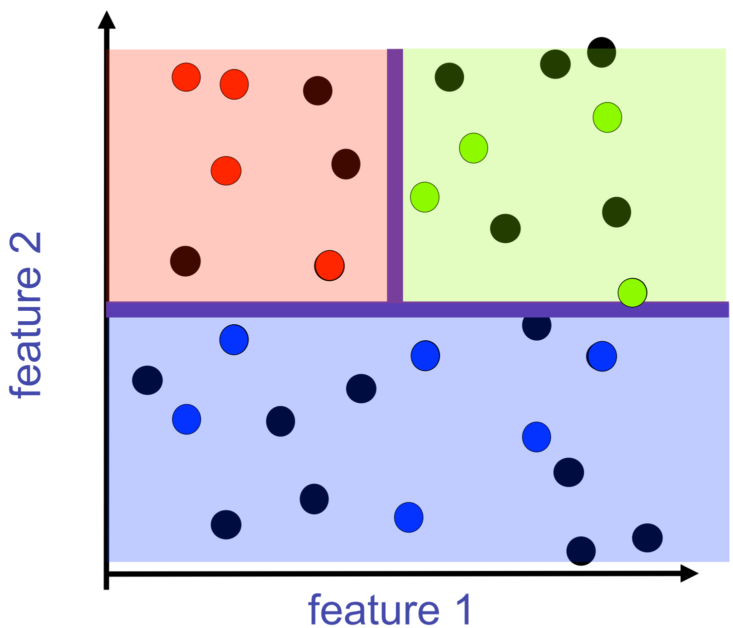



```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elseif plane in image:
        planes.append(image)
    else:
        other.append(tweet)
return birds, planes, other
```

Machine Learning algorithms

Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***





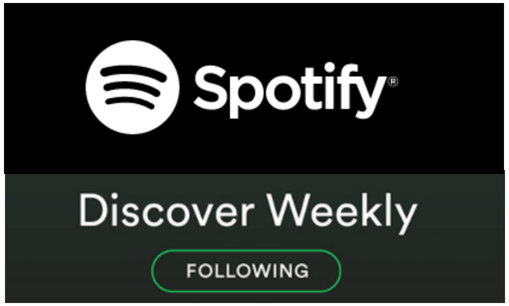
The decision rule of
`if "cat" in tweet:`
is **hard coded by expert.**

The decision rule of
`if bird in image:`
is **LEARNED using DATA**

Machine Learning Ingredients

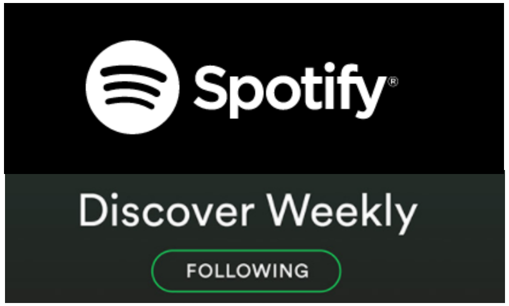
- **Data**: past observations
- **Hypotheses/Models**: devised to capture the patterns in data
- **Prediction**: apply model to forecast future observations

ML uses past data to make personalized predictions

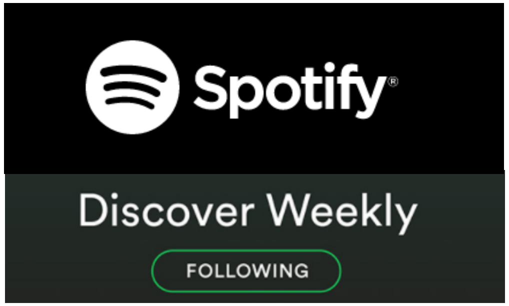


ML uses past data to make personalized predictions

Tick Tock

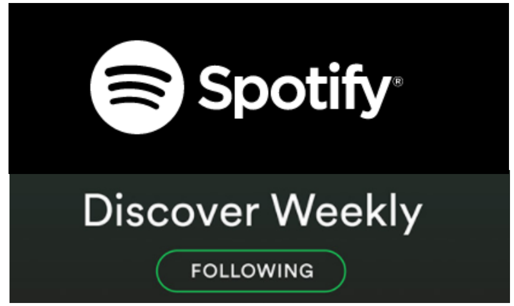


ML uses past data to make personalized predictions



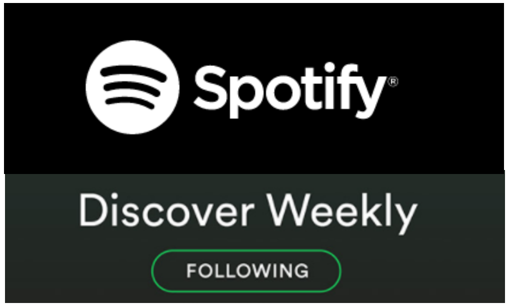
You may also like…

ML uses past data to make personalized predictions



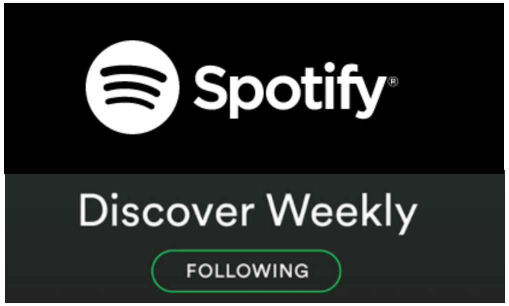
You may also like…

ML uses past data to make personalized predictions



You may also like…

ML uses past data to make personalized predictions



You may also like…

ML uses past data to make personalized predictions

NETFLIX

98% Match 2017 TV

Spotify

Discover Weekly

FOLLOWING

amazon prime

You may also like…

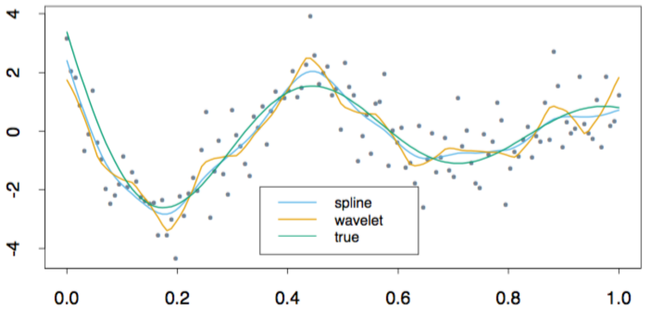
ML uses past data to make personalized predictions

You may also like…

ML uses past data to make personalized predictions

Flavors of ML

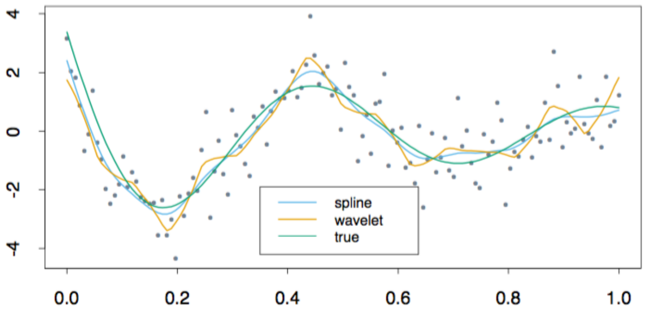
Flavors of ML



Regression

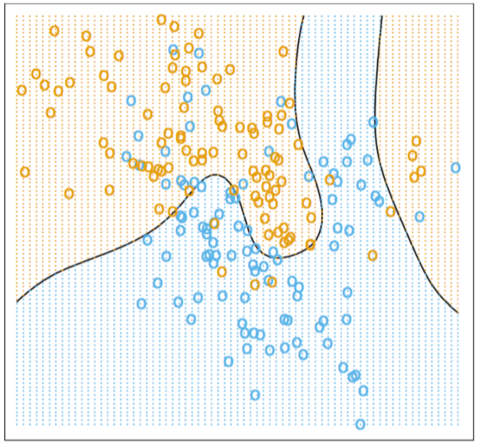
Predict continuous value:
ex: stock market, credit score, temperature, Netflix rating

Flavors of ML



Regression

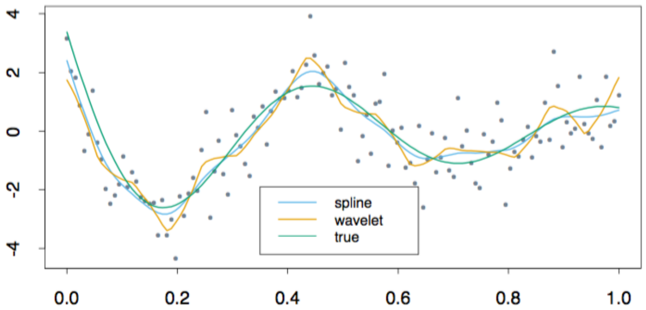
Predict continuous value:
ex: stock market, credit score, temperature, Netflix rating



Classification

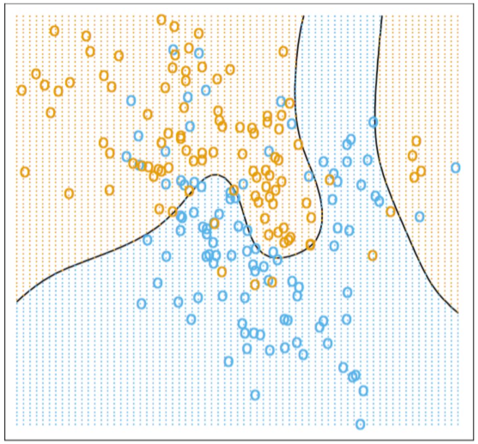
Predict categorical value:
loan or not? spam or not? what disease is this?

Flavors of ML



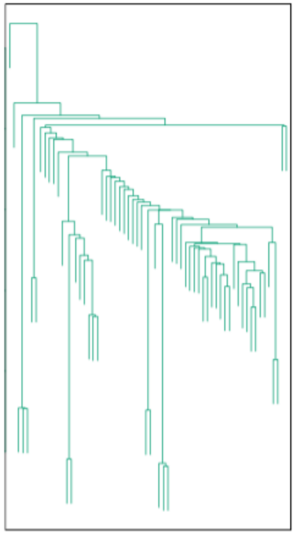
Regression

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



Classification

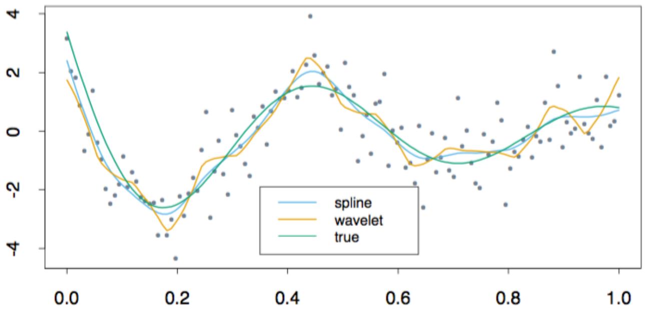
Predict categorical value:
loan or not? spam or not? what disease is this?



Unsupervised Learning

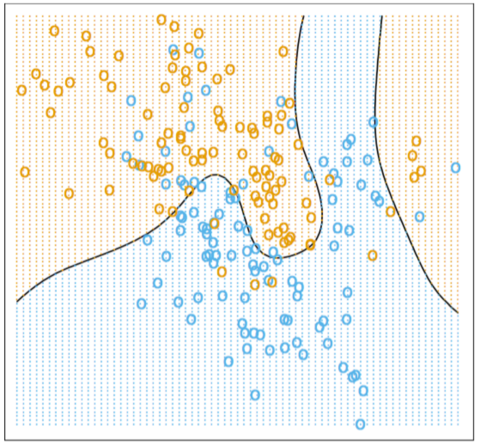
Predict structure:
tree of life from DNA, find similar images, community detection

Flavors of ML



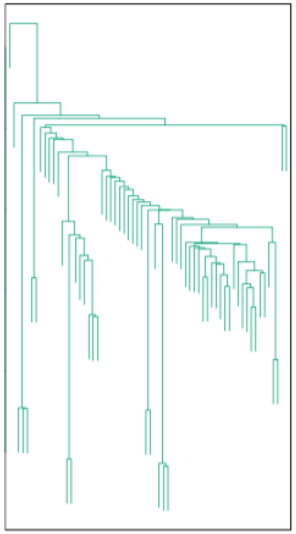
Regression

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



Classification

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



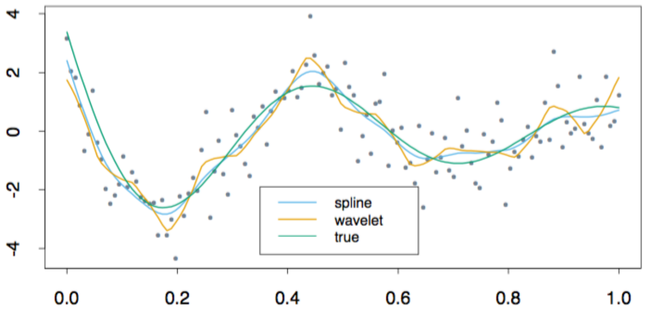
Unsupervised Learning

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

Mix of statistics (theory) and algorithms (programming)

Flavors of ML

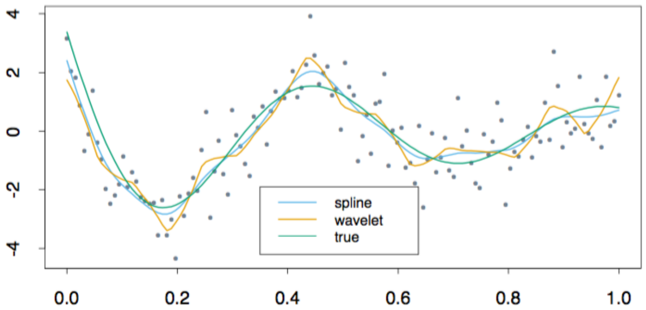
Flavors of ML



Regression

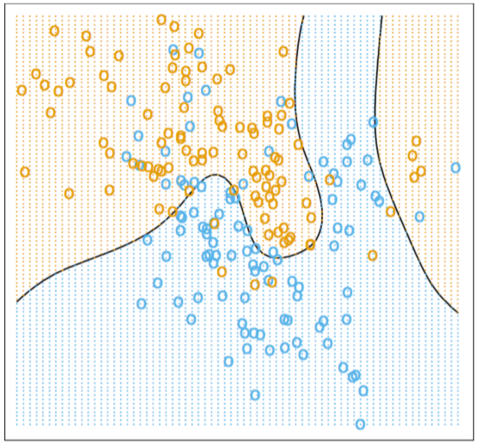
Predict continuous value:
ex: stock market, credit score, temperature, Netflix rating

Flavors of ML



Regression

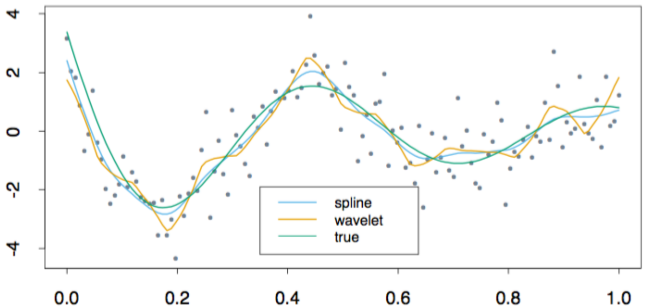
Predict continuous value:
ex: stock market, credit score, temperature, Netflix rating



Classification

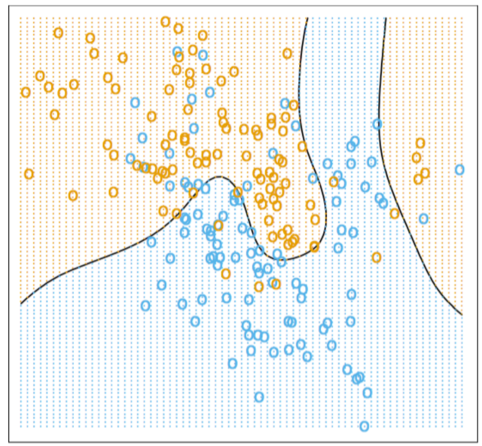
Predict categorical value:
loan or not? spam or not? what disease is this?

Flavors of ML



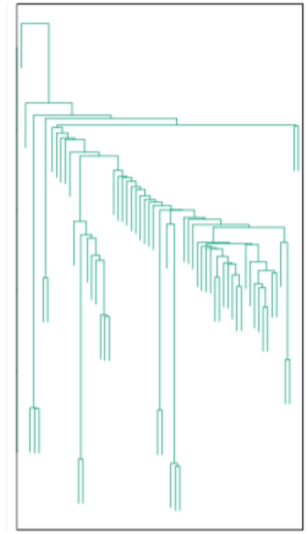
Regression

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



Classification

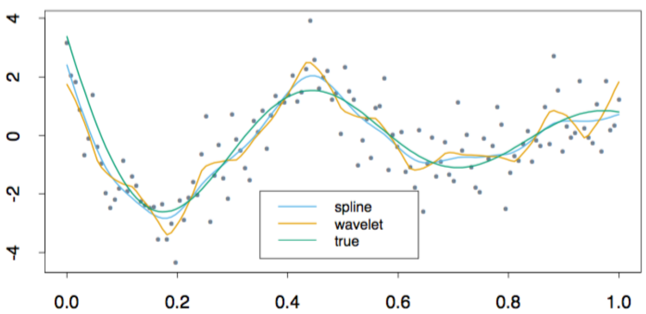
Predict categorical value:
loan or not? spam or not? what disease is this?



Unsupervised Learning

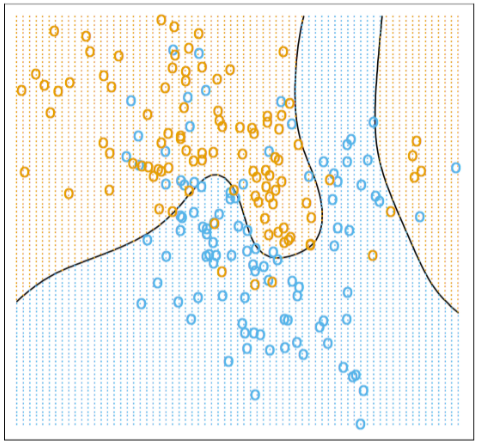
Predict structure:
tree of life from DNA, find similar images, community detection

Flavors of ML



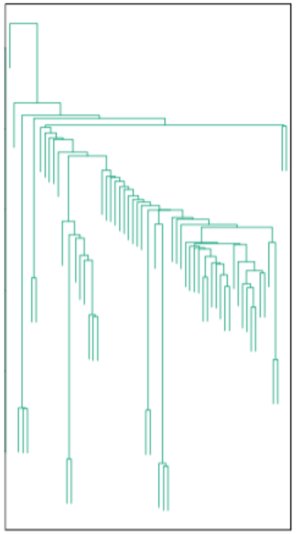
Regression

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



Classification

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



Unsupervised Learning

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

Mix of statistics (theory) and algorithms (programming)

CSE446/546: Machine Learning

Instructor: Jamie Morgenstern

Contact: cse446-staff@cs.washington.edu

Course Website: https://courses.cs.washington.edu/courses/cse446/23wi

CSE446/546: Machine Learning

Instructor: Jamie Morgenstern

Contact: cse446-staff@cs.washington.edu

Course Website: https://courses.cs.washington.edu/courses/cse446/23wi

What this class is:

- **Fundamentals of ML:** bias/variance tradeoff, overfitting, optimization and computational tradeoffs, supervised learning (e.g., linear, boosting, deep learning), unsupervised models (e.g. k-means, EM, PCA)
- **Preparation for further learning:** the field is fast-moving, you will be able to apply the basics and teach yourself the latest

CSE446/546: Machine Learning

Instructor: Jamie Morgenstern

Contact: cse446-staff@cs.washington.edu

Course Website: https://courses.cs.washington.edu/courses/cse446/23wi

What this class is:

- **Fundamentals of ML:** bias/variance tradeoff, overfitting, optimization and computational tradeoffs, supervised learning (e.g., linear, boosting, deep learning), unsupervised models (e.g. k-means, EM, PCA)
- **Preparation for further learning:** the field is fast-moving, you will be able to apply the basics and teach yourself the latest

What this class is not:

- **Survey course:** laundry list of algorithms, how to win Kaggle
- **An easy course:** familiarity with intro linear algebra and probability are assumed, homework will be time-consuming

Prerequisites

- Formally:
  - MATH 308, CSE 312, STAT 390 or equivalent
- Familiarity with:
  - Linear algebra
    - linear dependence, rank, linear equations, SVD
  - Multivariate calculus
  - Probability and statistics
    - Distributions, marginalization, moments, conditional expectation
  - Algorithms
    - Basic data structures, complexity
- "Can I learn these topics concurrently?"
- Use HW0 to judge skills
- **See website for review materials!**

Grading

- *5 homework*
  - *Each contains both theoretical questions and will have programming*
  - *Collaboration okay but must write who you collaborated with.* ***You must write, submit, and understand your answers and code (which run on autograder)***
  - ***WHITEBOARD POLICY***
  - *Do not Google for answers.*
- *2 exams, a midterm and a final*

Homework

- HW 0 is out (**Due next Wednesday 10/6 Midnight**)
  - Short *review*
  - Work individually, treat as barometer for readiness
- HW 1,2,3,4
  - They are not easy or short. Start early.
- Submit to Gradescope
- Regrade requests on Gradescope
- **There is no credit for late work, 5 late days**

Homework

- HW 0 is out (**Due next Wednesday 10/6 Midnight**)
  - Short *review*
  - Work individually, treat as barometer for readiness
- HW 1,2,3,4
  - They are not easy or short. Start early.
- Submit to Gradescope
- Regrade requests on Gradescope
- **There is no credit for late work, 5 late days**

Homework

- HW 0 is out (**Due next Wednesday 10/6 Midnight**)
  - Short *review*
  - Work individually, treat as barometer for readiness
- HW 1,2,3,4
  - They are not easy or short. Start early.
- Submit to Gradescope
- Regrade requests on Gradescope
- **There is no credit for late work, 5 late days**

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

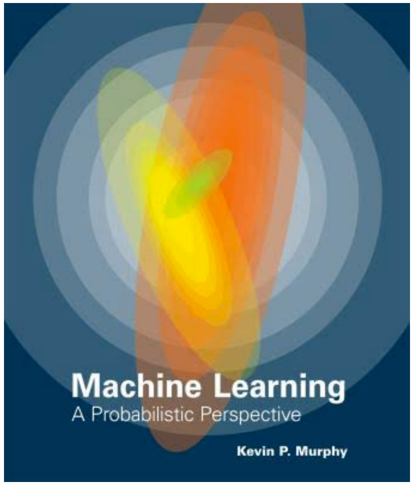
See course website for tutorials and references.

Communication Chanels

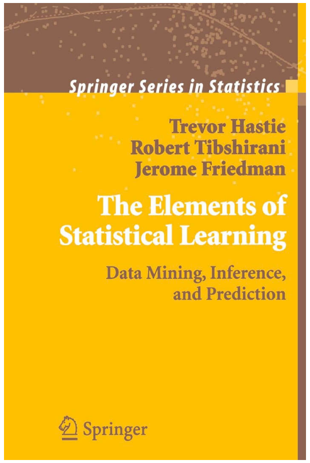
- **Announcements, questions about class, homework help**
  - EdStem (invitation sent, contact TAs if you need access)
  - Weekly Section
  - Office hours
- **Regrade requests**
  - Directly to Gradescope
- **Personal concerns**
  - Email: cse446-staff@cs.washington.edu
- **Anonymous feedback**
  - See website for link

Textbooks

- Required Textbook:
  - ***Machine Learning: a Probabilistic Perspective***; **Kevin Murphy**



- Optional Books (free PDF):
  - *The Elements of Statistical Learning: Data Mining, Inference, and Prediction;* Trevor Hastie, Robert Tibshirani, Jerome Friedman



Addcodes

- Email: Elle Brown (elleanb@cs.washington.edu) for addcodes

Enjoy!

- ML is becoming ubiquitous in science, engineering and beyond
- It's one of the hottest topics in industry today
- This class should give you the basic foundation for applying ML and developing new methods
- The fun begins…

Maximum Likelihood Estimation

Jamie Morgenstern

Your first consulting job

- *Billionaire*: I have a special coin, if I flip it, what's the probability it will be heads?
- *You*: Please flip it a few times:

$D = (HTHHT\ldots)$

$\underline{\qquad\qquad}$ n flips

k heads

- *You*: The probability is:

$\frac{k}{n}$

- *Billionaire:* Why?

Coin – Binomial Distribution

- **Data**: sequence *D= (HHTHT…)*, **k heads** out of **n flips**
- **Hypothesis:** P(Heads) = θ, P(Tails) = 1-θ
  - Flips are i.i.d.:
    - Independent events
    - Identically distributed according to Binomial distribution

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

Maximum Likelihood Estimation

- **Data**: sequence *D= (HHTHT…),* **k heads** out of **n flips**
- **Hypothesis:** P(Heads) = θ, P(Tails) = 1-θ

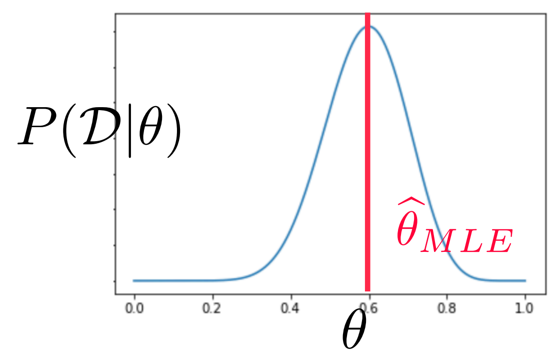
$$P(\mathcal{D}|\theta) = \theta^k (1-\theta)^{n-k}$$

When
x > y
log x > log y

- Maximum likelihood estimation (MLE): Choose θ that maximizes the probability of observed data:

$$\widehat{\theta}_{MLE} = \arg\max_{\theta} \; P(\mathcal{D}|\theta)$$

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



Your first learning algorithm

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

$$= \arg\max_{\theta} \; \log \theta^k (1-\theta)^{n-k}$$

- Set derivative to zero: $\boxed{\dfrac{d}{d\theta} \log P(\mathcal{D}|\theta) = 0}$

$$\frac{d}{d\theta}\, k\log\theta + (n-k)\log(1-\theta)$$

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

$$\Leftrightarrow \frac{k}{\theta} = \frac{n-k}{(1-\theta)}$$

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

$$\frac{1}{\theta} - 1 = \frac{n}{k} - 1$$

$$\frac{1}{\theta} = \frac{n}{k} \Rightarrow \boxed{\frac{k}{n} = \theta}$$

How many flips do I need?

$$\widehat{\theta}_{MLE} = \frac{k}{n}$$

- *You*: flip the coin 5 times. *Billionaire*: I got 3 heads.

$$\widehat{\theta}_{MLE} = \frac{3}{5} = .6$$

- *You*: flip the coin 50 times. *Billionaire*: I got 20 heads.

$$\widehat{\theta}_{MLE} = \frac{2}{5} = .4$$

- *Billionaire:* Which one is right? Why?

Simple bound (based on Hoeffding's inequality)

- For **n flips** and **k heads** the MLE is **unbiased** for true θ^*:

$$\widehat{\theta}_{MLE} = \frac{k}{n} \qquad \mathbb{E}[\widehat{\theta}_{MLE}] = \theta^*$$

- Hoeffding's inequality says that for any $\varepsilon>0$:

$$P(|\widehat{\theta}_{MLE} - \theta^*| \geq \epsilon) \leq 2e^{-2n\epsilon^2}$$

PAC Learning

- PAC: Probably Approximate Correct
- *Billionaire*: I want to know the parameter θ^*, within $\varepsilon = 0.1$, with probability at least $1-\delta = 0.95$. How many flips?

$$P(|\widehat{\theta}_{MLE} - \theta^*| \geq \epsilon) \leq 2e^{-2n\epsilon^2}$$

What about continuous variables?

- *Billionaire*: 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 properties of Gaussians

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

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

MLE for Gaussian

$\in \mathbb{R}_{\geq 0}$

- Prob. of i.i.d. samples $D=\{x_1,\ldots,x_N\}$ (e.g., exam scores):

$$P(\mathcal{D}|\mu,\sigma) = P(x_1,\ldots,x_n|\mu,\sigma)$$
$$= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \prod_{i=1}^n e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$\frac{d}{d\mu} = 0$

- Log-likelihood of data:

$\frac{d}{d\mu}$ $-2\frac{(x_i-\mu)}{2\sigma^2}$

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

$$\frac{d}{d\mu}\log P(D|\mu,\sigma) = +\sum_i \frac{(x_i-\mu)}{\sigma^2}$$

Your second learning algorithm: MLE for mean of a Gaussian

- What's MLE for mean?

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

$$\sum \frac{x_i - \mu)}{\cancel{\sigma^2}} = 0$$

$$\sum_i (x_i - \hat{\mu}) = 0$$

$$\sum x_i = n\hat{\mu}$$

$$\hat{\mu} = \frac{\sum x_i}{n}$$

MLE for variance

- Again, set derivative to zero:

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

Learning Gaussian parameters

$\mathbb{E}_{\mathcal{D}}[\hat{\mu}_{MLE}] = \mu$

- MLE:

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

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

- MLE for the variance of a Gaussian is **biased**

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

$\sigma^2_{MLE} = \frac{1}{n}\sum (x_i - \hat{\mu}_{MLE})^2$

- Unbiased variance estimator:

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

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) $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

Maximum Likelihood Estimation

Observe $X_1, X_2, \dots, X_n$ 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) $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

Properties (under benign regularity conditions—smoothness, identifiability, etc.):

- Asymptotically consistent and normal: $\frac{\widehat{\theta}_{MLE} - \theta_*}{\widehat{se}} \sim \mathcal{N}(0,1)$
- Asymptotic Optimality, minimum variance (see Cramer-Rao lower bound)

Recap

- Learning is…
  - Collect some data
    - E.g., coin flips
  - Choose a hypothesis class or model
    - E.g., binomial
  - Choose a loss function
    - E.g., data likelihood
  - Choose an optimization procedure
    - E.g., set derivative to zero to obtain MLE
  - Justifying the accuracy of the estimate
    - E.g., Hoeffding's inequality

Maximum Likelihood Estimation, cont.

Machine Learning – CSE446

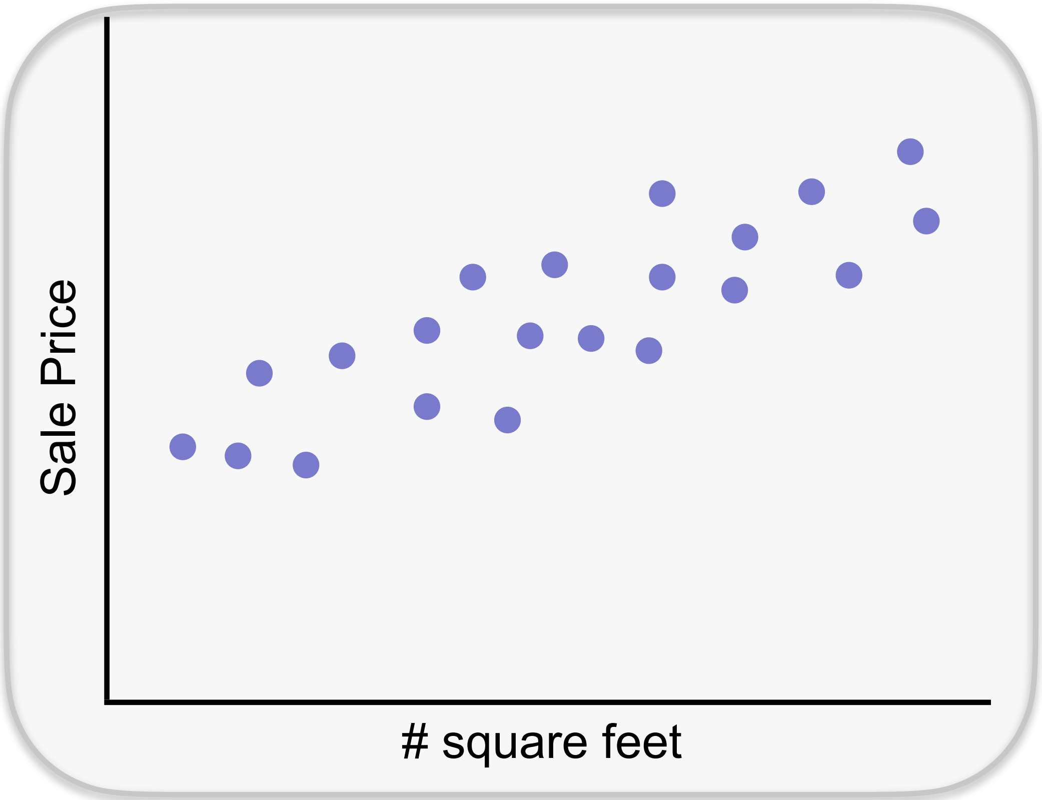
Jamie Morgenstern
University of Washington

The 1d regression problem

Given past sales data on zillow.com, predict:

y = **House sale price** *from*

x = **# sq. ft.**



Training Data:

$$\{(x_i, y_i)\}_{i=1}^n$$

Model:

$$y_i = x_i w + b + \epsilon_i$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Linear model

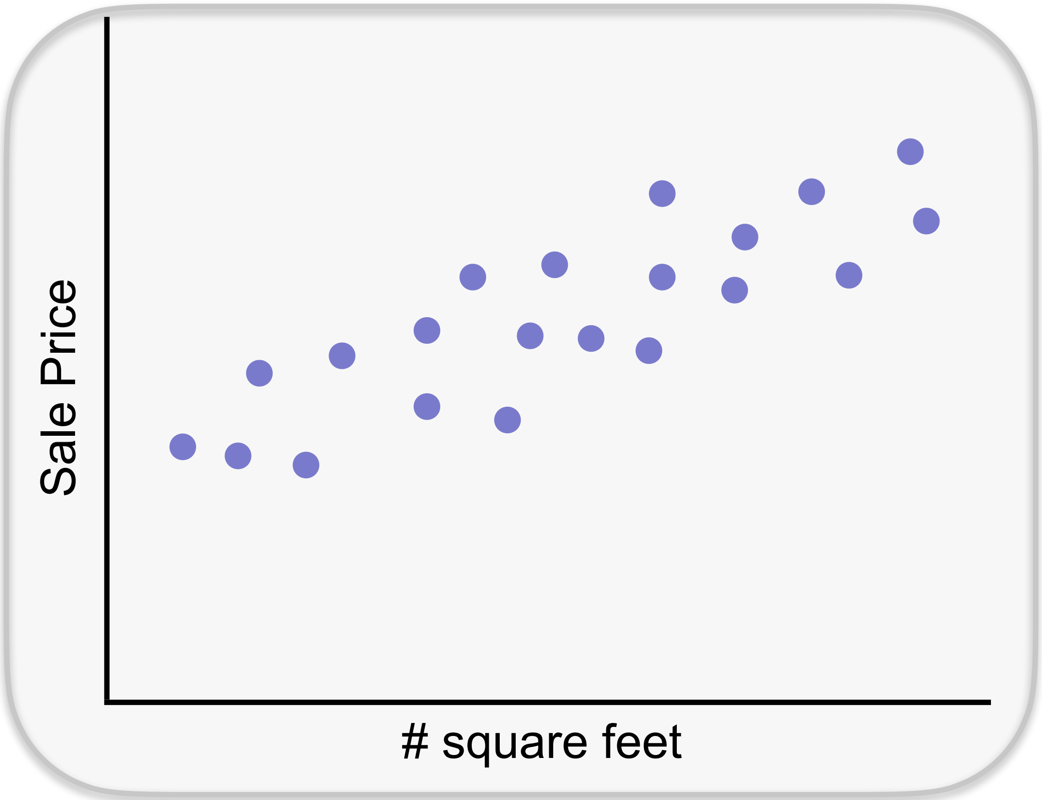
Noise model

The 1d regression problem

Given past sales data on zillow.com, predict:

y = **House sale price** *from*

x = **# sq. ft.**



Training Data:

$$\{(x_i, y_i)\}_{i=1}^n$$

Model:

$$y_i = x_i w + b + \epsilon_i$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Loss function:

$$\sum_{i=1}^n -\log(p(y_i | x_i, w, b))$$

The 1d regression problem

Loss function:

$$\sum_{i=1}^{n} -\log(p(y_i|x_i, w, b))$$

$$= \sum_{i=1}^{n} -\log(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{\frac{-(y_i - (wx_i + b))^2}{2\sigma^2}\right\})$$

Model:

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

$$y_i = x_i w + b + \epsilon_i$$

The 1d regression problem

Loss function:

$$\sum_{i=1}^{n} -\log(p(y_i|x_i, w, b))$$

$$= \sum_{i=1}^{n} -\log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{\frac{-(y_i - (wx_i + b))^2}{2\sigma^2}\right\}\right)$$

Model:

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

$$y_i = x_i w + b + \epsilon_i$$

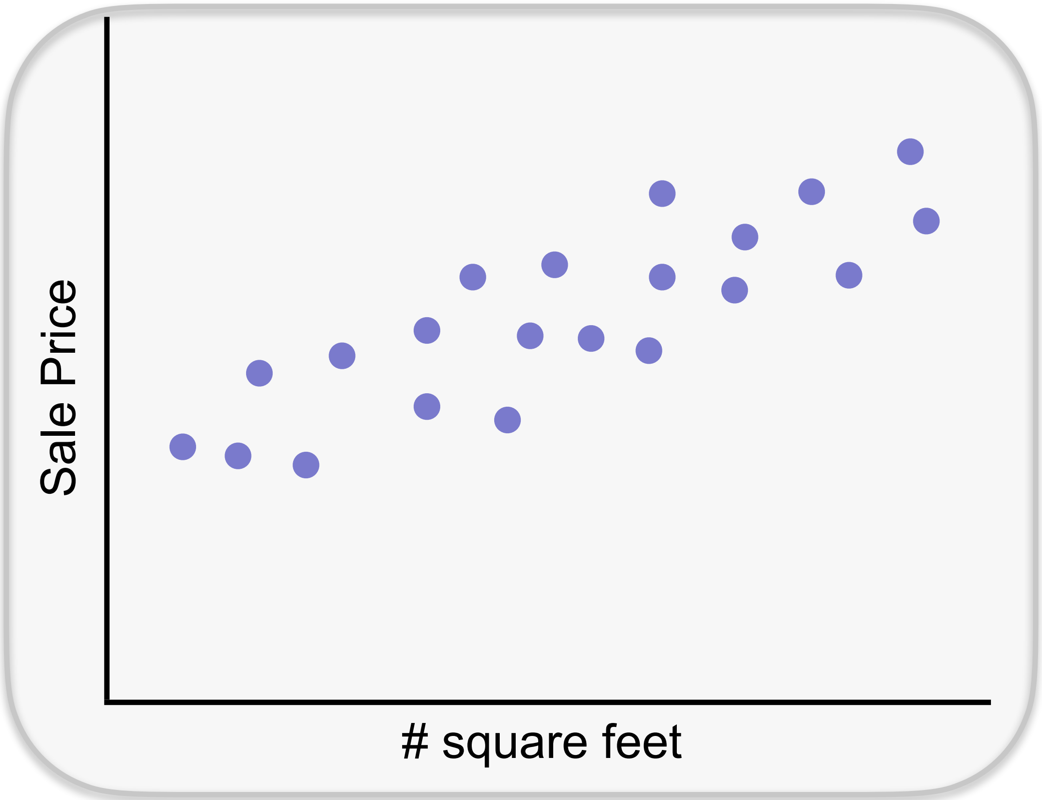
$$\arg\min_{w,b} \sum_{i=1}^{n} -\log(p(y_i|x_i, w, b)) \equiv \arg\min_{w,b} \sum_{i=1}^{n} (y_i - (wx_i + b))^2$$

The 1d regression problem

Given past sales data on zillow.com, predict:

y = **House sale price** *from*

x = **# sq. ft.**



Training Data:

$$\{(x_i, y_i)\}_{i=1}^n$$

Model:

$$y_i = x_i w + b + \epsilon_i$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Loss function:

$$\sum_{i=1}^n (y_i - (wx_i + b))^2$$