



# Naïve Bayes

Chris Piech  
CS109, Stanford University

# Review

# Event Shorthand

## Without shorthand

$$P(Y = y | X_1 = x_1)$$

---

## Our shorthand notation

$y$

is shorthand for the event:

$Y = y$

$x_1$

is shorthand for the event:

$X_1 = x_1$

---

## Now with shorthand

$$P(y|x_1)$$

# Event Shorthand

## MAP, without shorthand

$$\hat{\theta}_{MAP} = \operatorname{argmax}_{\theta} f(\Theta = \theta | X^{(1)} = x^{(1)}, \dots, X^{(n)} = x^{(n)})$$

---

## Our shorthand notation

$\theta$  is shorthand for the event:  $\Theta = \theta$

$x^{(i)}$  is shorthand for the event:  $X^{(i)} = x^{(i)}$

---

## MAP, now with shorthand

$$\hat{\theta}_{MAP} = \operatorname{argmax}_{\theta} f(\theta | x^{(1)}, \dots, x^{(n)})$$

# MLE vs MAP

**Data:**  $x^{(1)}, \dots, x^{(n)}$

## Maximum Likelihood Estimation

$$\hat{\theta}_{MLE} = \operatorname{argmax}_{\theta} f(x^{(1)}, \dots, x^{(n)} | \theta)$$

## Maximum A Posteriori

$$\hat{\theta}_{MAP} = \operatorname{argmax}_{\theta} f(\theta | x^{(1)}, \dots, x^{(n)})$$

# MLE vs MAP

**Data:**  $x^{(1)}, \dots, x^{(n)}$

## Maximum Likelihood Estimation

$$\begin{aligned}\hat{\theta}_{MLE} &= \operatorname{argmax}_{\theta} f(x^{(1)}, \dots, x^{(n)} | \theta) \\ &= \operatorname{argmax}_{\theta} \left( \sum_i \log f(x^{(i)} | \theta) \right)\end{aligned}$$

## Maximum A Posteriori

$$\begin{aligned}\hat{\theta}_{MAP} &= \operatorname{argmax}_{\theta} f(\theta | x^{(1)}, \dots, x^{(n)}) \\ &= \operatorname{argmax}_{\theta} \left( \log(g(\theta)) + \sum_{i=1}^n \log(f(x^{(i)} | \theta)) \right)\end{aligned}$$

# Multinomial

Each experiment has  $M$  possible outcomes. What is the likelihood of a particular count of each outcome?

*multinomial is parameterized by  $p_i$ :  
the likelihood of outcome  $i$  on any one experiment.*



# Multinomial

Each experiment has  $M$  possible outcomes. What is the likelihood of a particular count of each outcome?

*multinomial is parameterized by  $p_i$ :  
the likelihood of outcome  $i$  on any one experiment.*

Dice:

$$M = 6$$

$$p_i = 1/6$$



# MLE for Multinomial

MLE estimate of  
the probability of  
outcome  $i$

$$p_i = \frac{n_i}{n}$$

number of  
observed outcomes  
of type  $i$

number of  
observations

$\theta_i$  is  $p$   
For a multinomial

# MAP for Multinomial, Leplace Prior

MAP estimate of  
the probability of  
outcome  $i$

$$p_i = \frac{n_i + 1}{n + m}$$

number of observed outcomes of type  $i$

number of observations

number of outcome types

$\theta$  is  $p$   
For a multinomial

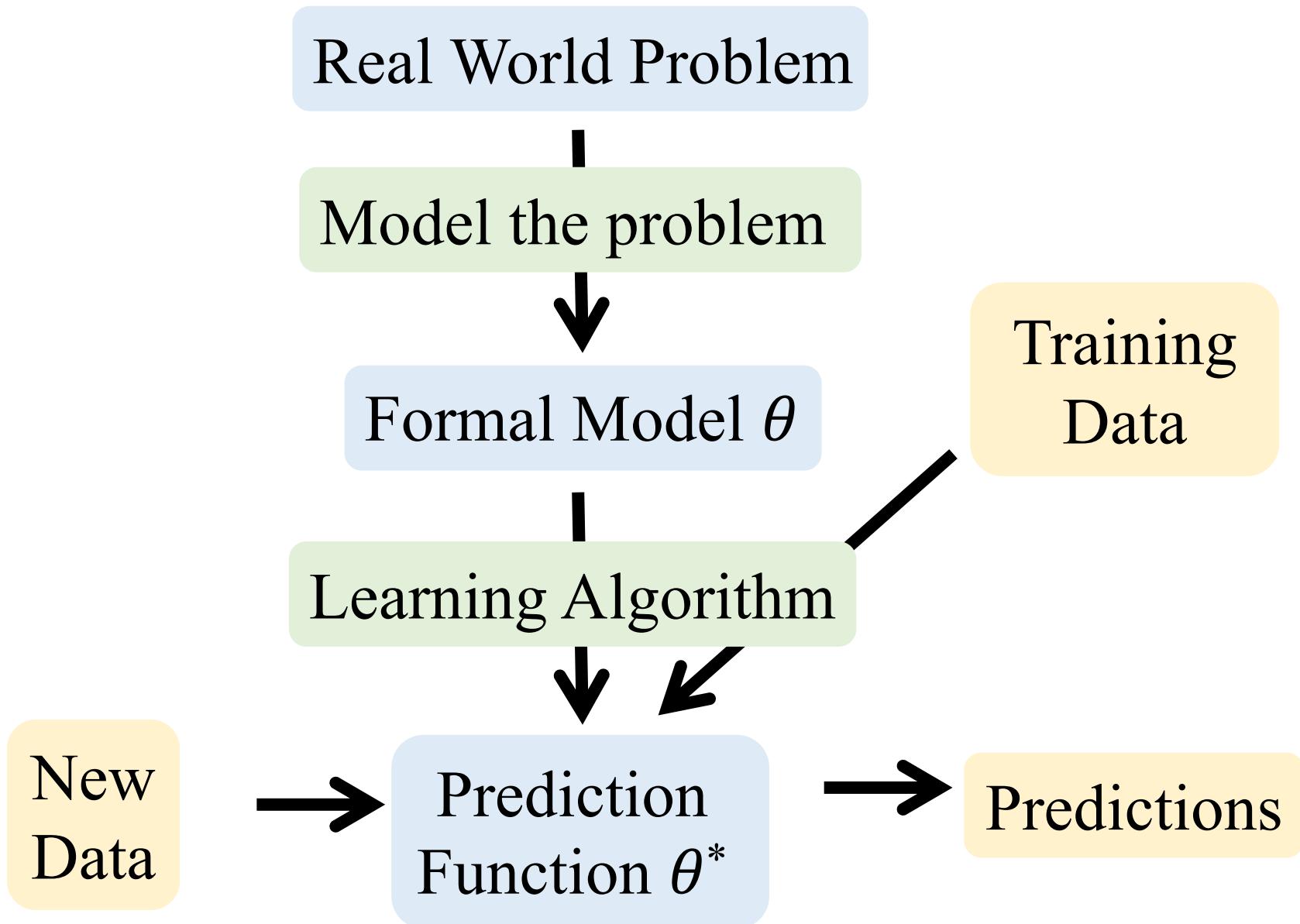


End Review

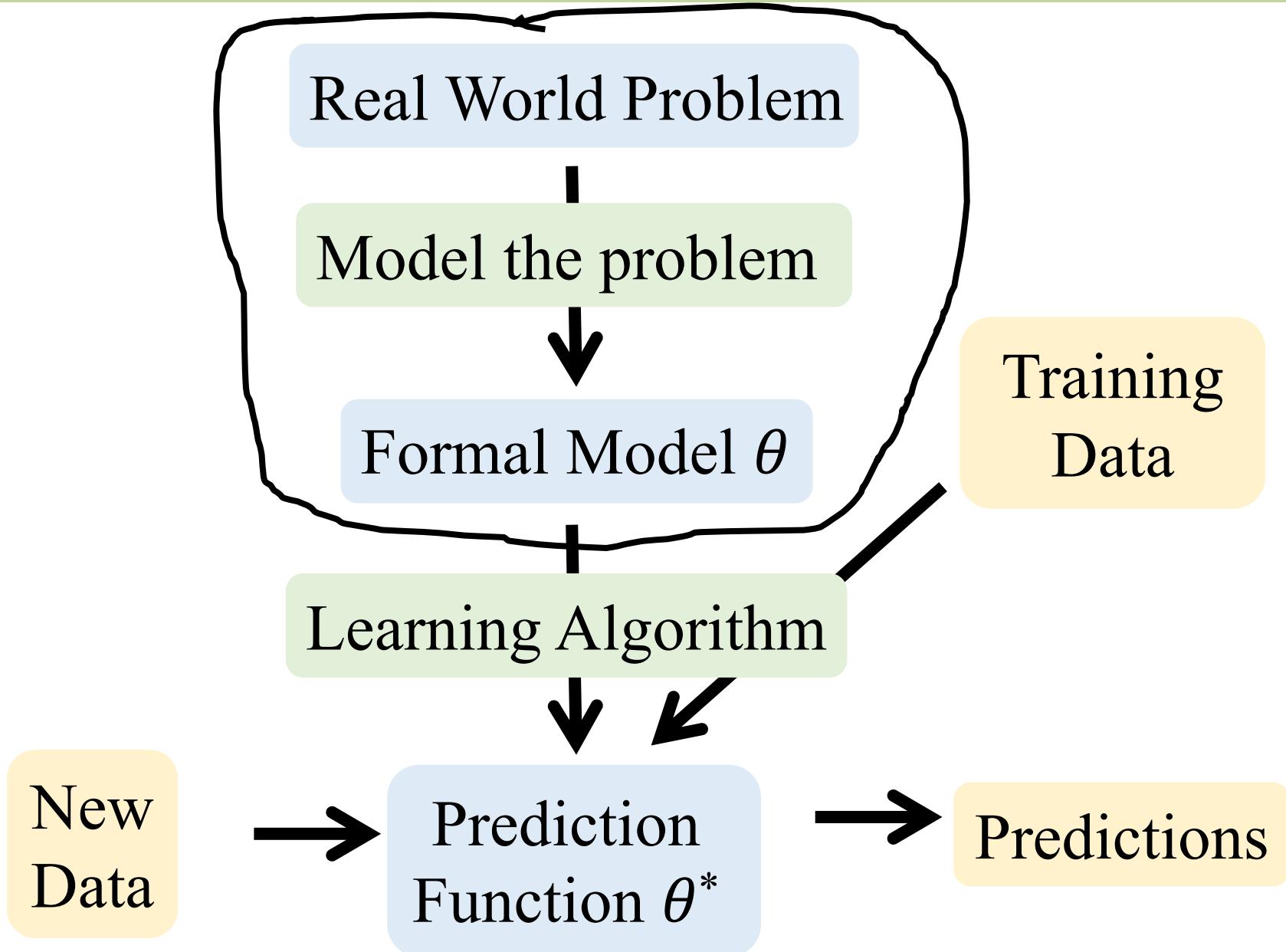
The last estimator has risen...

# Machine Learning

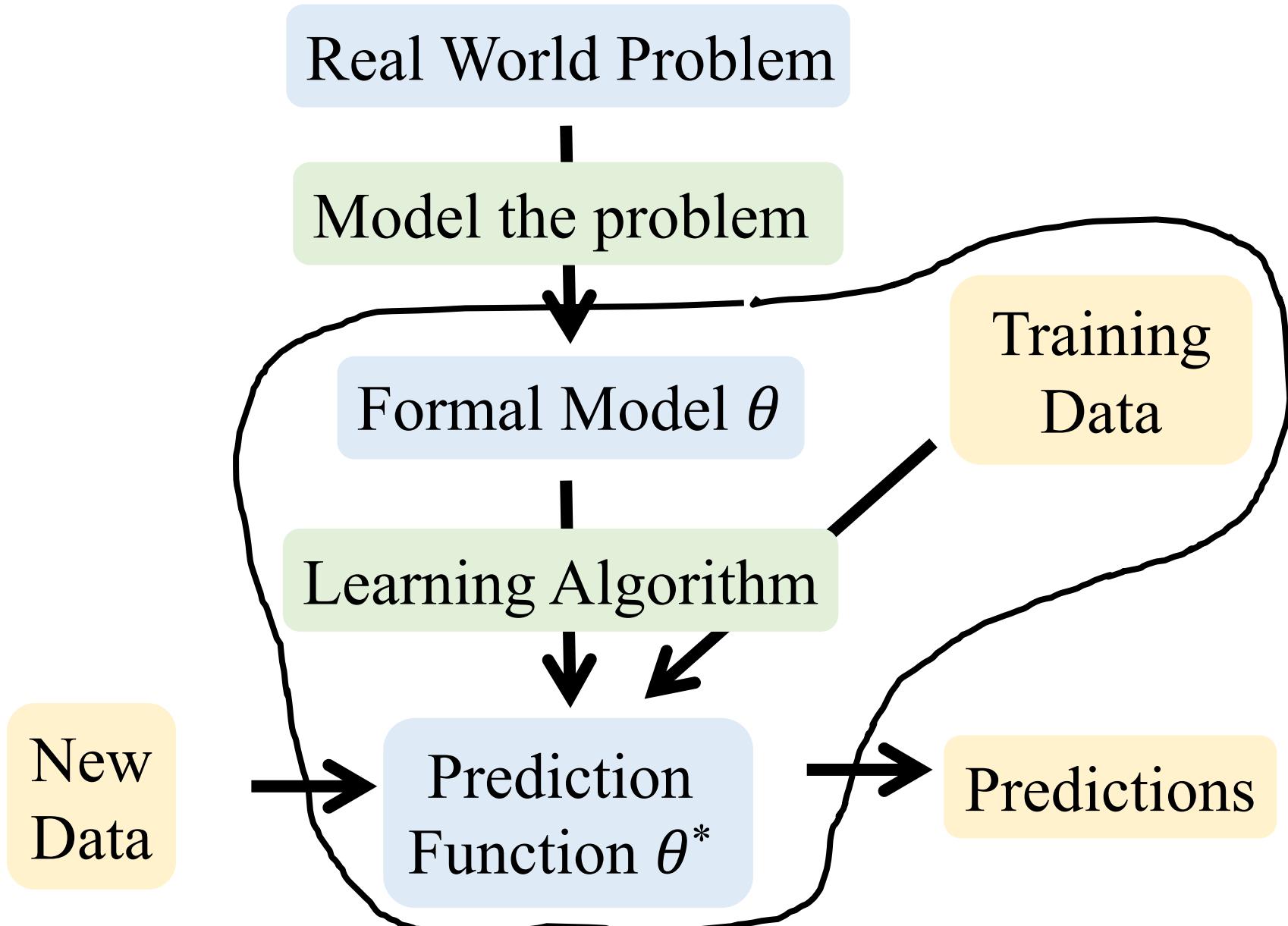
# Supervised Learning



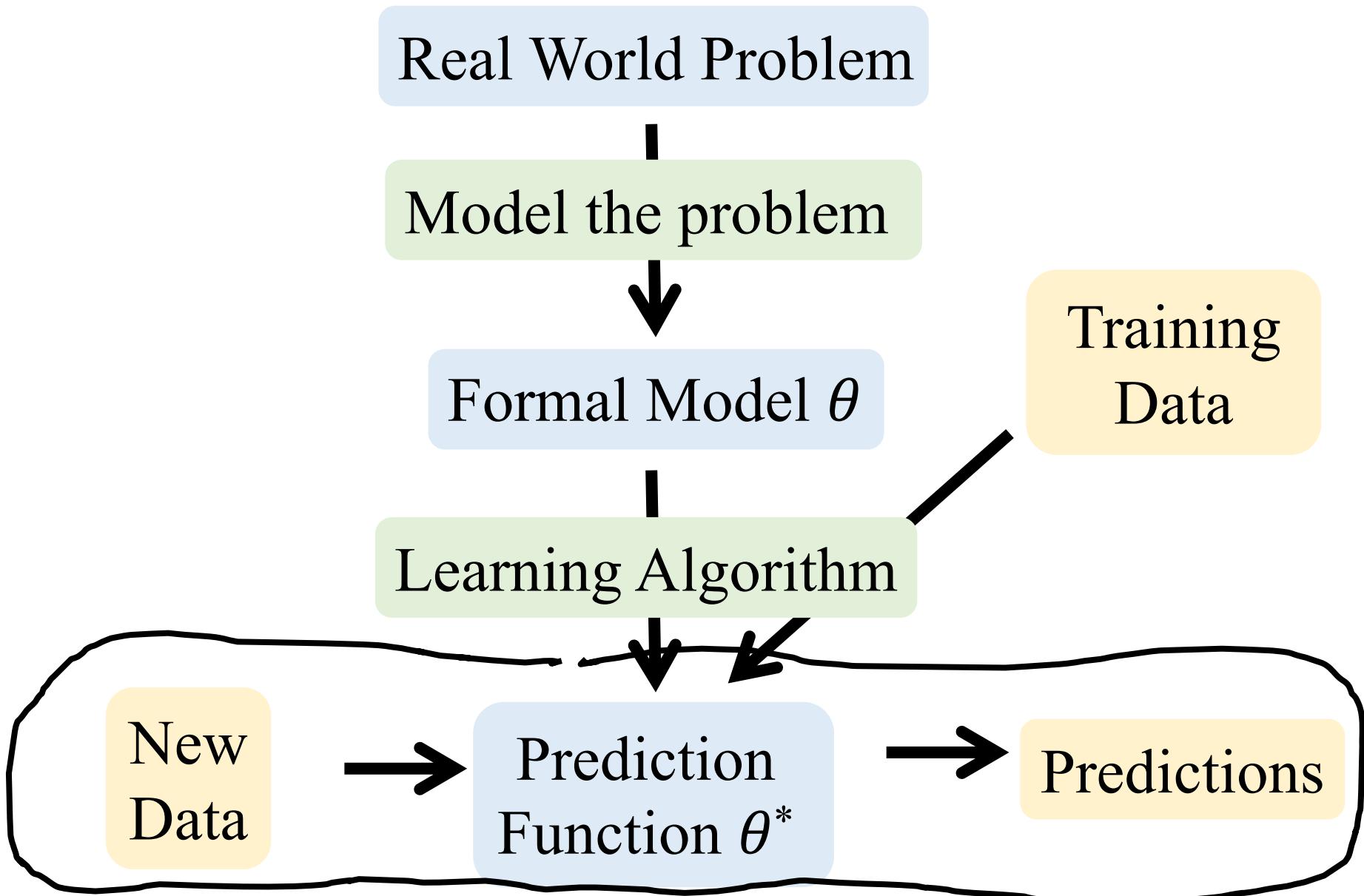
# Modelling



# Training\*



# Make Predictions\*



# Machine Learning: Formally

- Many different forms of “Machine Learning”
  - We focus on the problem of *prediction*
- Want to make a prediction based on observations
  - Vector  $\mathbf{X}$  of  $m$  observed variables:  $\mathbf{X} = [X_1 \dots X_m]$
  - Based on observed  $\mathbf{X}$ , want to predict unseen variable  $\mathbf{Y}$ 
    - $\mathbf{Y}$  called “output feature/variable” (or the “dependent variable”)
  - Seek to “learn” a function  $g(\mathbf{X})$  to predict  $\mathbf{Y}$ :
    - $\hat{\mathbf{Y}} = g(\mathbf{X})$
    - When  $\mathbf{Y}$  is discrete, prediction of  $\mathbf{Y}$  is called “classification”
    - When  $\mathbf{Y}$  is continuous, prediction of  $\mathbf{Y}$  is called “regression”

# Training Data

Training Data: assignments all random variables  $\mathbf{X}$  and  $\mathbf{Y}$

Assume IID data:

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots (\mathbf{x}^{(n)}, y^{(n)})$$

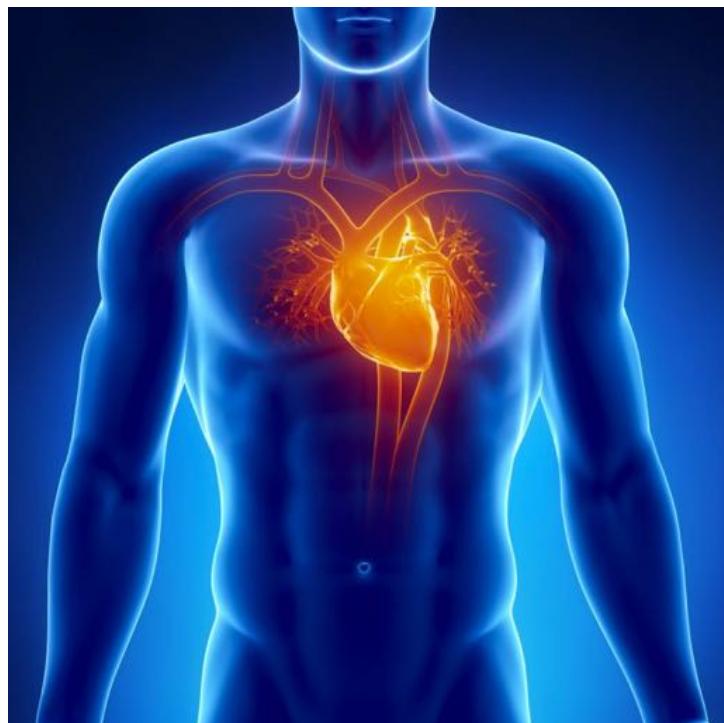
*n training datapoints*

$$m = |\mathbf{x}^{(i)}|$$

Each datapoint has  $m$  features and a single output

# Example Datasets

Heart



Ancestry



Netflix

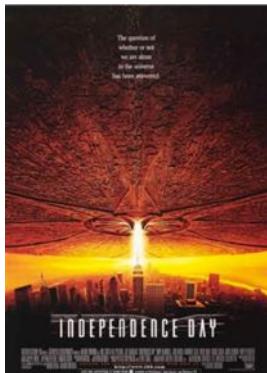


# Target Movie “Like” Classification

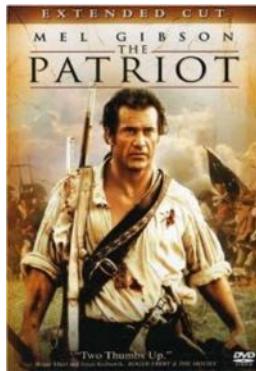
	Movie 1	Movie 2	Movie $m$	Output
User 1	1	0	1	1
User 2	1	1	0	0
		⋮		⋮
User $n$	0	0	1	1

# Single Instance

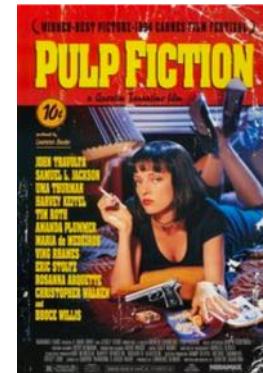
Movie 1



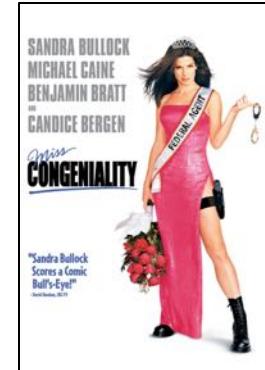
Movie 2



Movie  $m$



Output



User 1

1

0

1

1

User 2

1

1

0

0

⋮

⋮

User  $n$

0

0

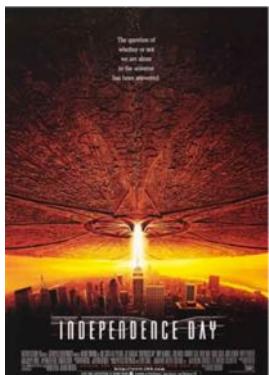
1

1

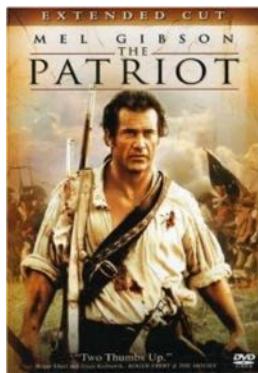
$(\mathbf{x}^{(i)}, y^{(i)})$  such that  $1 \leq i \leq n$

# Feature Vector

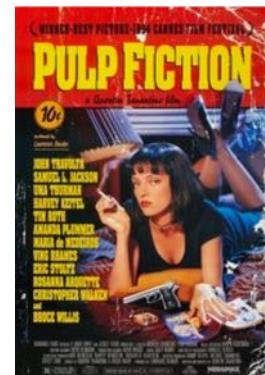
Movie 1



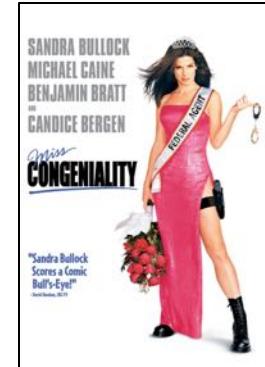
Movie 2



Movie  $m$



Output



User 1

1

0

1

1

User 2

1

1

0

0

⋮

⋮

User  $n$

0

0

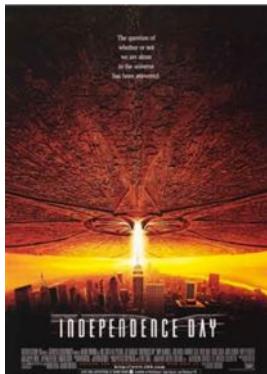
1

1

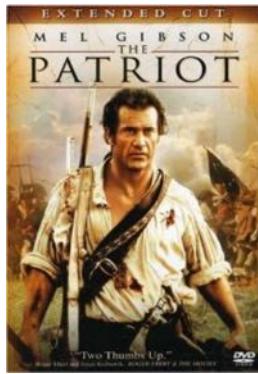
$(\mathbf{x}^{(i)}, y^{(i)})$  such that  $1 \leq i \leq n$

# Output Value

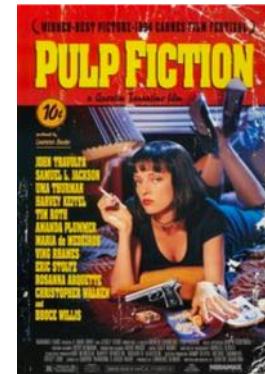
Movie 1



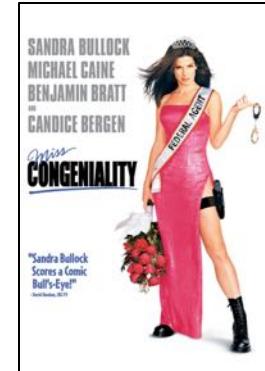
Movie 2



Movie  $m$



Output



User 1

1

0

1

1

User 2

1

1

0

0

⋮

⋮

User  $n$

0

0

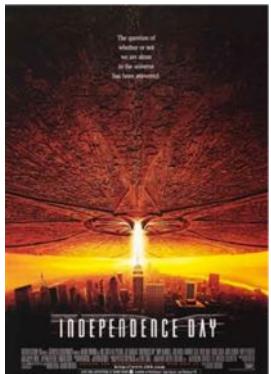
1

1

$(\mathbf{x}^{(i)}, y^{(i)})$  such that  $1 \leq i \leq n$

# Single Feature Value

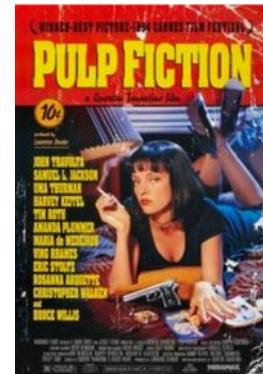
Movie 1



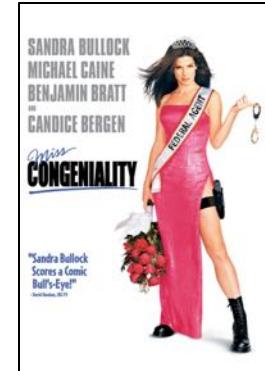
Movie 2



Movie  $m$



Output



User 1

1

0

1

1

User 2

1

1

0

0

⋮

⋮

User  $n$

0

0

1

1

In general:  $\mathbf{x}_j^{(i)}$

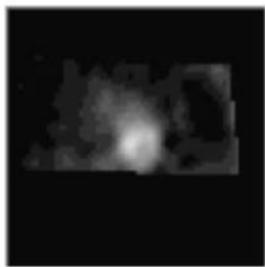
In this case:  $\mathbf{x}_m^{(2)}$

# Healthy Heart Classifier

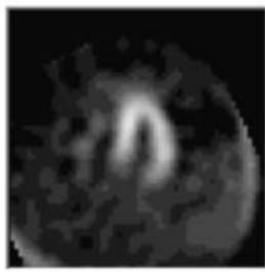
	ROI 1	ROI 2	...	ROI $m$	Output
Heart 1	0	1		1	0
Heart 2	1	1		1	0
			:		:
Heart $n$	0	0		0	1

# Healthy Heart Classifier

ROI 1



ROI 2



...

ROI  $m$

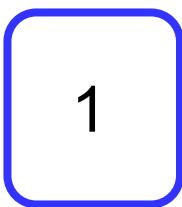


Output



Heart 1

0



1

0

Heart 2

1

1

1

0

⋮

Heart  $n$

0

0

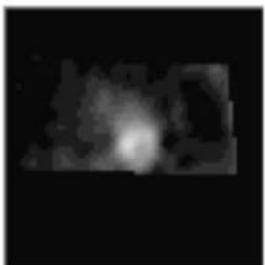
0

1

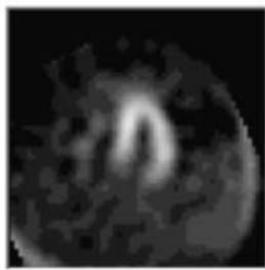
$$x_2^{(1)}$$

# Healthy Heart Classifier

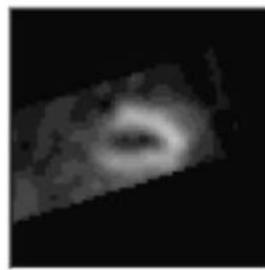
ROI 1



ROI 2



ROI  $m$



...

Output



Heart 1

0

1

1

0

Heart 2

1

1

1

0

:

:

Heart  $n$

0

0

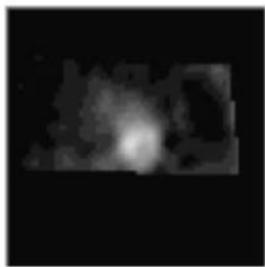
0

1

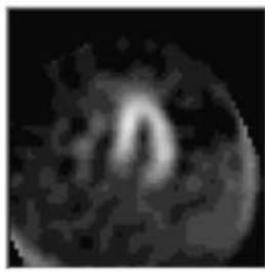
$$(\mathbf{x}^{(2)}, y^{(2)})$$

# Healthy Heart Classifier

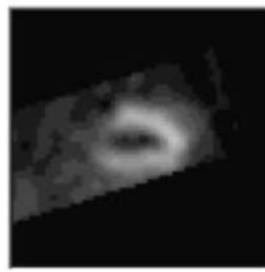
ROI 1



ROI 2



ROI  $m$



...

Output



Heart 1

0

1

1

0

Heart 2

1

1

1

0

:

Heart  $n$

0

0

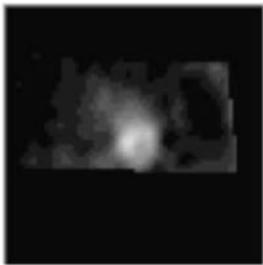
0

1

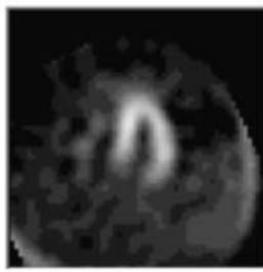
$\mathbf{x}^{(2)}$

# Healthy Heart Classifier

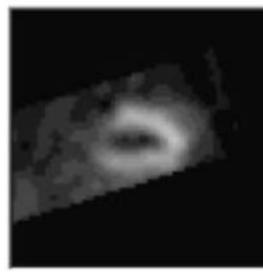
ROI 1



ROI 2



ROI  $m$



Output



Heart 1

0

1

1

0

Heart 2

1

1

1

0

⋮

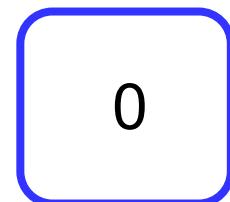
Heart  $n$

0

0

0

1



⋮

$$y^{(2)}$$

# Ancestry Classifier

	SNP 1	SNP 2	SNP $m$	Output
User 1	1	0	1	0
User 2	0	0	1	1
		⋮		⋮
User $n$	1	1	0	1

# Regression: Predicting Real Numbers

Opposing team ELO	Points in last game	At Home?	Output
		...	 # Points
Game 1 84	105	1	120
Game 2 90	102	0	95
	:		:
Game $n$ 74	120	0	115

# Training Data

Training Data: assignments all random variables  $\mathbf{X}$  and  $\mathbf{Y}$

Assume IID data:

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots (\mathbf{x}^{(n)}, y^{(n)})$$

*n training datapoints*

$$m = |\mathbf{x}^{(i)}|$$

Each datapoint has  $m$  features and a single output

ML is ubiquitous

# Linear Regression

# Linear Regression

Opposing team ELO	Points in last game	At Home?	Output
		...	
Game 1 84	105	1	120
Game 2 90	102	0	95
	:		:
Game $n$ 74	120	0	115

# Linear Regression

$X_1$  = Opposing team ELO

$X_2$  = Points in last game

$X_3$  = Curry playing?

$X_4$  = Playing at home?

---

$Y$  = Warriors points

# Linear Regression

$Y = \text{Warriors points}$

$$\begin{aligned}\hat{Y} &= \theta_1 X_1 + \theta_2 X_2 + \dots \theta_{n-1} X_{n-1} + \theta_n 1 \\ &= \theta^T \mathbf{X}\end{aligned}$$

---

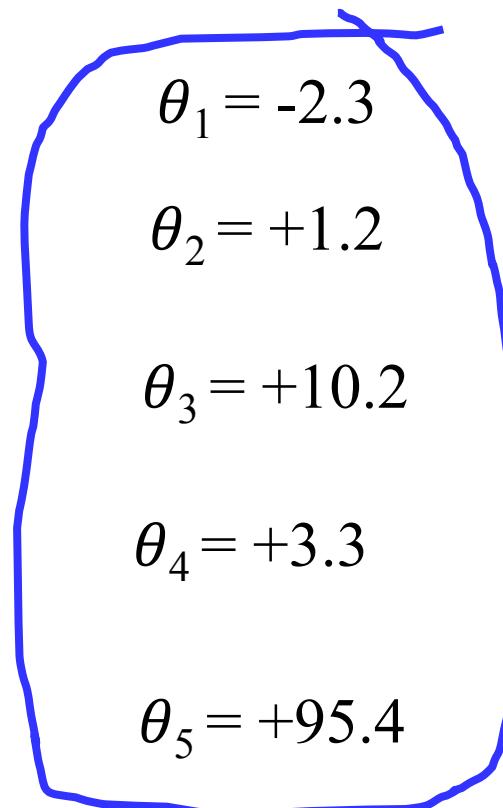
$X_1$  = Opposing team ELO

$X_2$  = Points in last game

$X_3$  = Curry playing?

$X_4$  = Playing at home?

$X_5 = 1$



# Classification

# Classification is Building a Harry Potter Hat



$$\mathbf{x} = [0, 1, \dots, 1]$$

# Healthy Heart Classifier

	ROI 1	ROI 2	...	ROI $m$	Output
Heart 1	0	1		1	0
Heart 2	1	1		1	0
			:		:
Heart $n$	0	0		0	1

# Ancestry Classifier

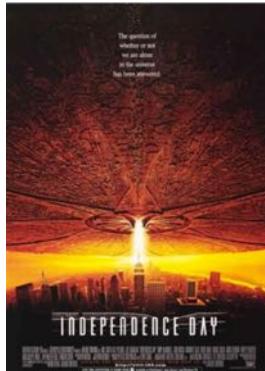
	SNP 1	SNP 2	SNP $m$	Output
User 1	1	0	1	0
User 2	0	0	1	1
		⋮		⋮
User $n$	1	1	0	1

# NETFLIX

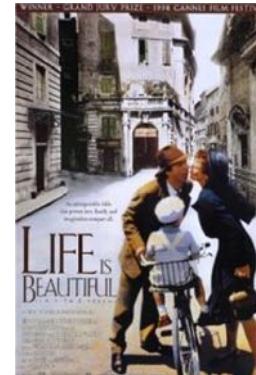
**And Learn**

# Target Movie “Like” Classification

Feature 1



Output



User 1      1

1

User 2      1

0

⋮

User  $n$       0

1

$$x_j^{(i)} \in \{0, 1\}$$

$$y^{(i)} \in \{0, 1\}$$

How could we predict the class label:  
will the user like life is beautiful?

# Fake Algorithm: Brute Bayes Classifier

# Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x})$$

Prediction: will they like L.I.B?

If  $y = 1$ , they like L.I.B?

Whether or not they liked Independence day

Simply chose the class label that is the most likely given the data

This is for one user

# Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x})$$

Simply chose the class label that is the most likely given the data

This is for one user

# Brute Force Bayes

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$

Simply chose the class label that is the most likely given the data

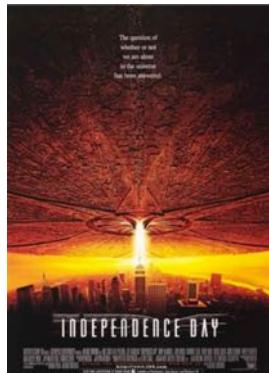
This is for one user

\* Note how similar this is to Hamilton example ☺

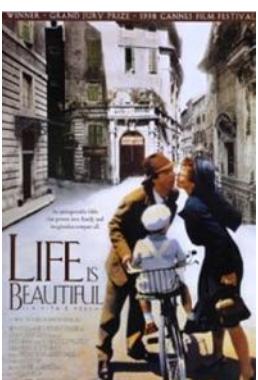
# What are the Parameters?

# Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$



Conditional  
probability  
table



$\mathbf{Y} = 0$

$x_1 = 0$	$\theta_0$
$x_1 = 1$	$\theta_1$

$\mathbf{Y} = 1$

$x_1 = 0$	$\theta_2$
$x_1 = 1$	$\theta_3$

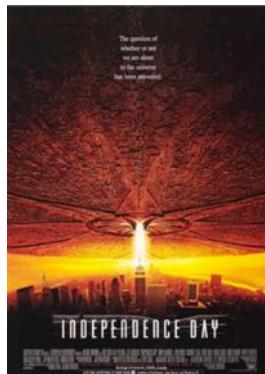


$\mathbf{Y} = 0$	$\theta_4$
$\mathbf{Y} = 1$	$\theta_5$

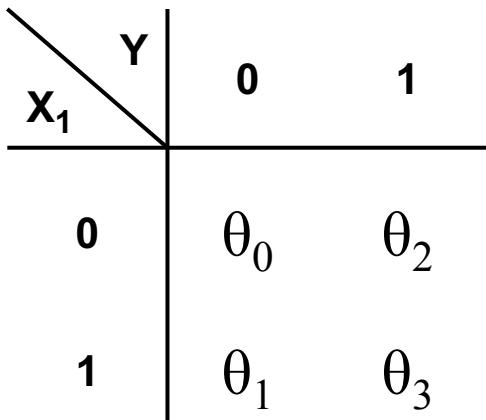
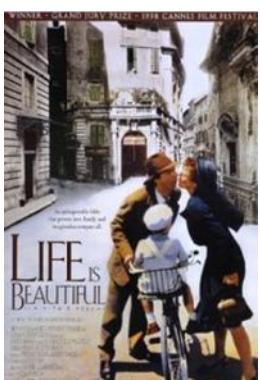
Learn these during training

# Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$



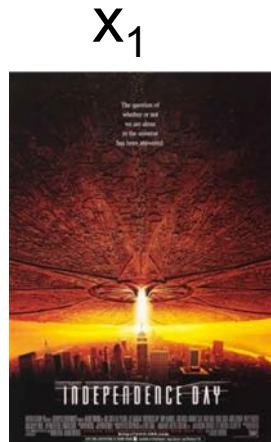
Conditional  
probability  
table



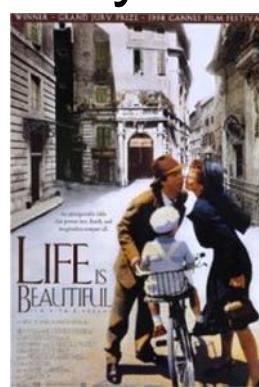
$Y = 0$	$\theta_4$
$Y = 1$	$\theta_5$

Learn these during training

# Training



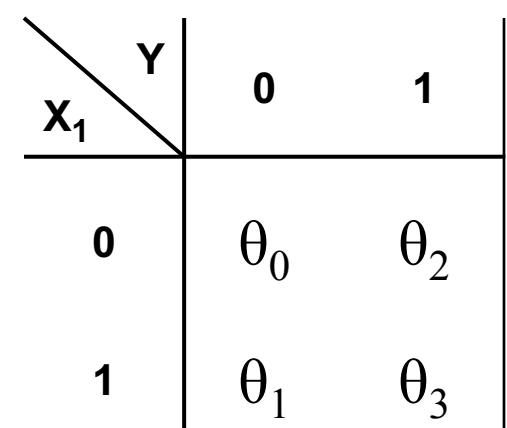
$x_1$



$y$

User 1	1
User 2	0
User $n$	0

$$P(\mathbf{x}|y)$$



What is  $P(X_1 | Y = 0)$ ?  
What is  $P(X_1 | Y = 1)$ ?  
Both multinomials  
with two outcomes

# MLE Estimate



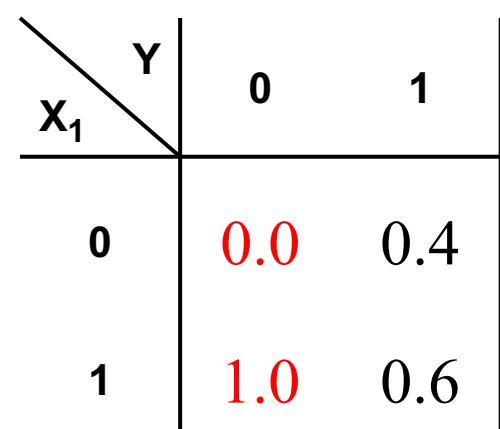
$x_1$



$y$

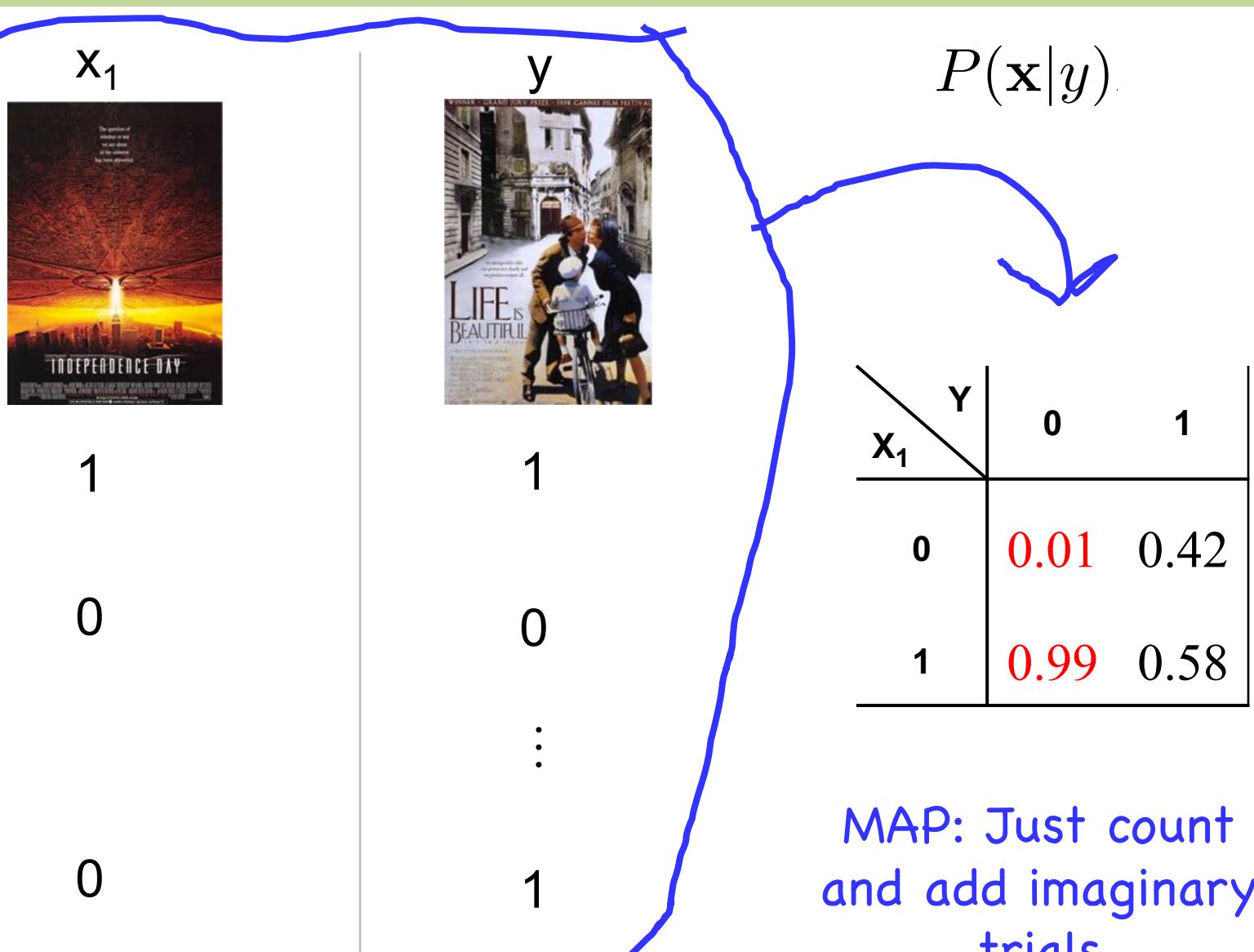
User 1	1
User 2	0
User $n$	0

$P(\mathbf{x}|y)$



MLE: Just count

# MAP Estimate



# Testing

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$



Y = 0	0.21
Y = 1	0.79

---

Test user: Likes independence day

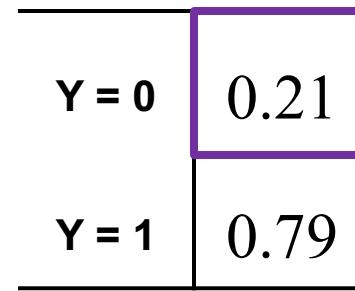
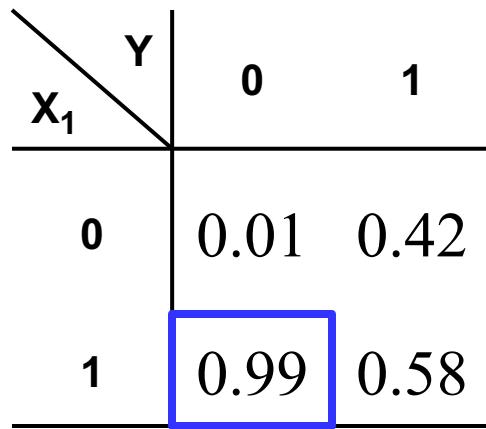
$$P(x_1 = 1|y = 0)P(y = 0)$$

vs

$$P(x_1 = 1|y = 1)P(y = 1)$$

# Testing

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$



---

Test user: Likes independence day

$$P(x_1 = 1|y = 0)P(y = 0)$$

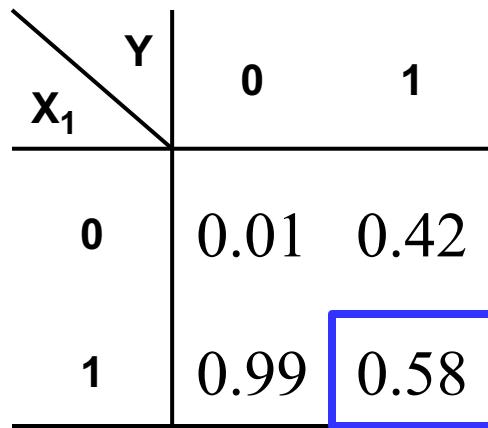
0.208

vs

$$P(x_1 = 1|y = 1)P(y = 1)$$

# Testing

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$



Y = 0	0.21
Y = 1	0.79

---

Test user: Likes independence day

$$P(x_1 = 1|y = 0)P(y = 0) \quad 0.208$$

vs

$$P(x_1 = 1|y = 1)P(y = 1) \quad 0.458$$

That was pretty good!

# Brute Force Bayes $m = 2$

	$x_1$	$x_2$	$y$
User 1	1	0	1
User 2	1	0	0
			:
User $n$	0	1	1

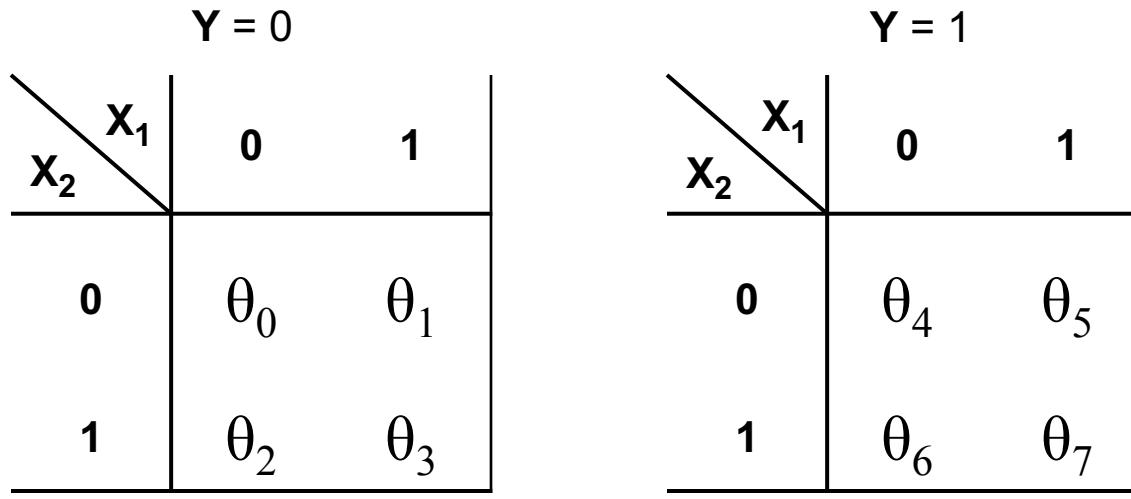
# Brute Force Bayes m = 2

Simply chose the class label that is the most likely given the data

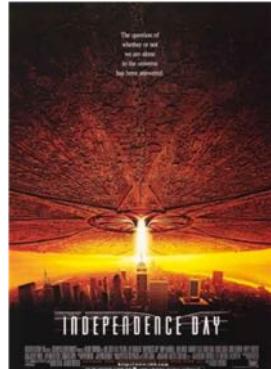
$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$


# Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$



$x_1$



$x_2$



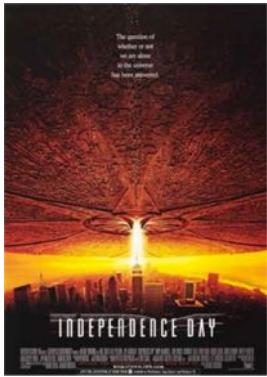
$y$



Fine

# Brute Force Bayes $m = 3$

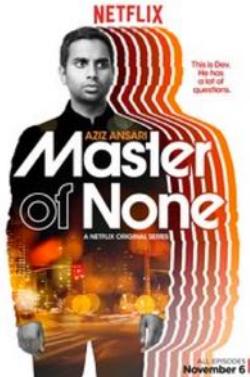
$x_1$



$x_2$



$x_3$



$y$



User 1

1

0

1

1

User 2

1

0

1

0

:

User  $n$

0

1

1

1

# Brute Force Bayes m = 3

Simply chose the class label that is the most likely given the data

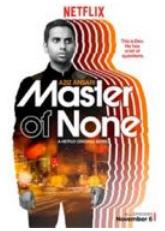
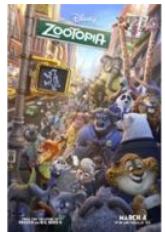
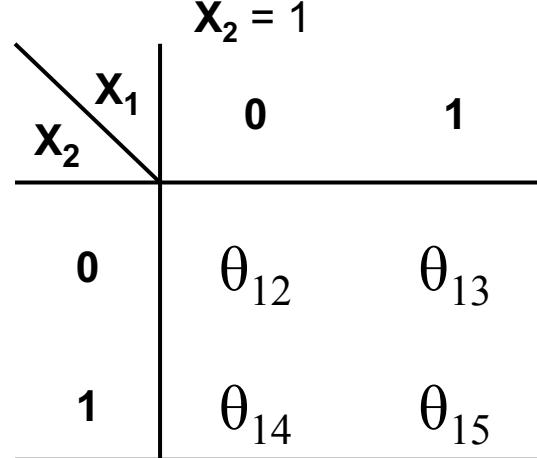
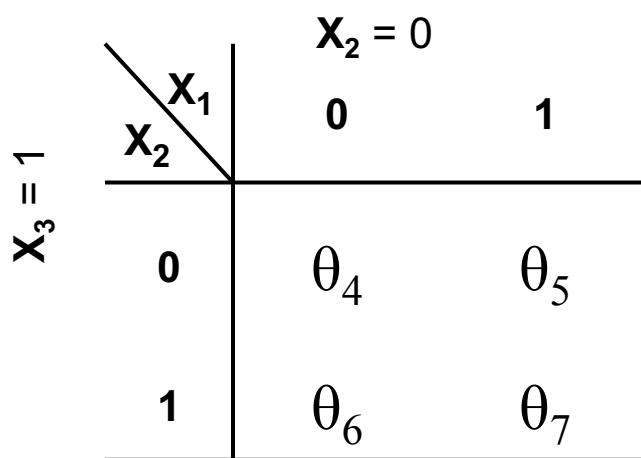
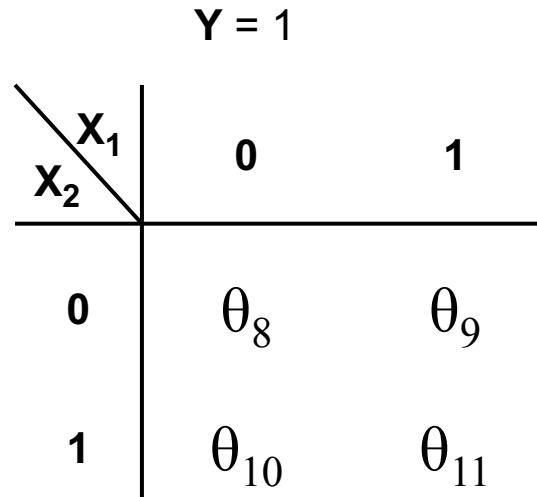
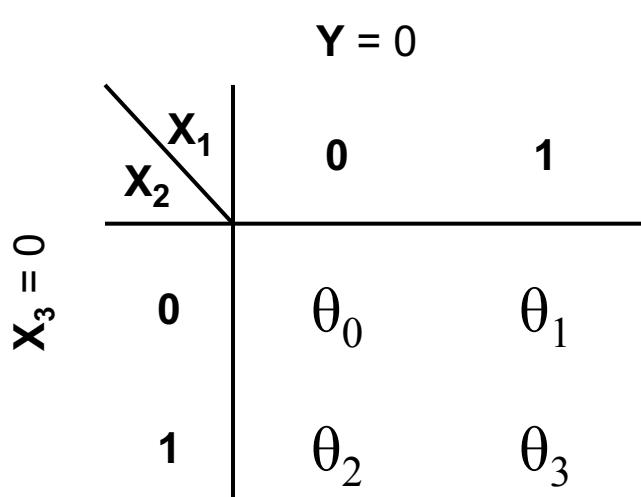
$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$



$$P(x_1, x_2, x_3|y)$$

# Brute Force Bayes

$$\hat{y} = \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$



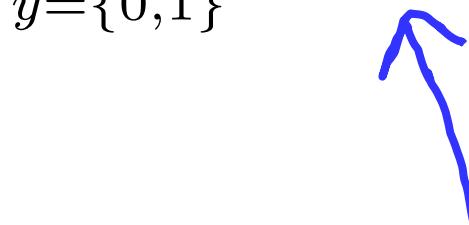
And if  $m=100$ ?

# Brute Force Bayes m = 100

Simply chose the class label that is the most likely given the data

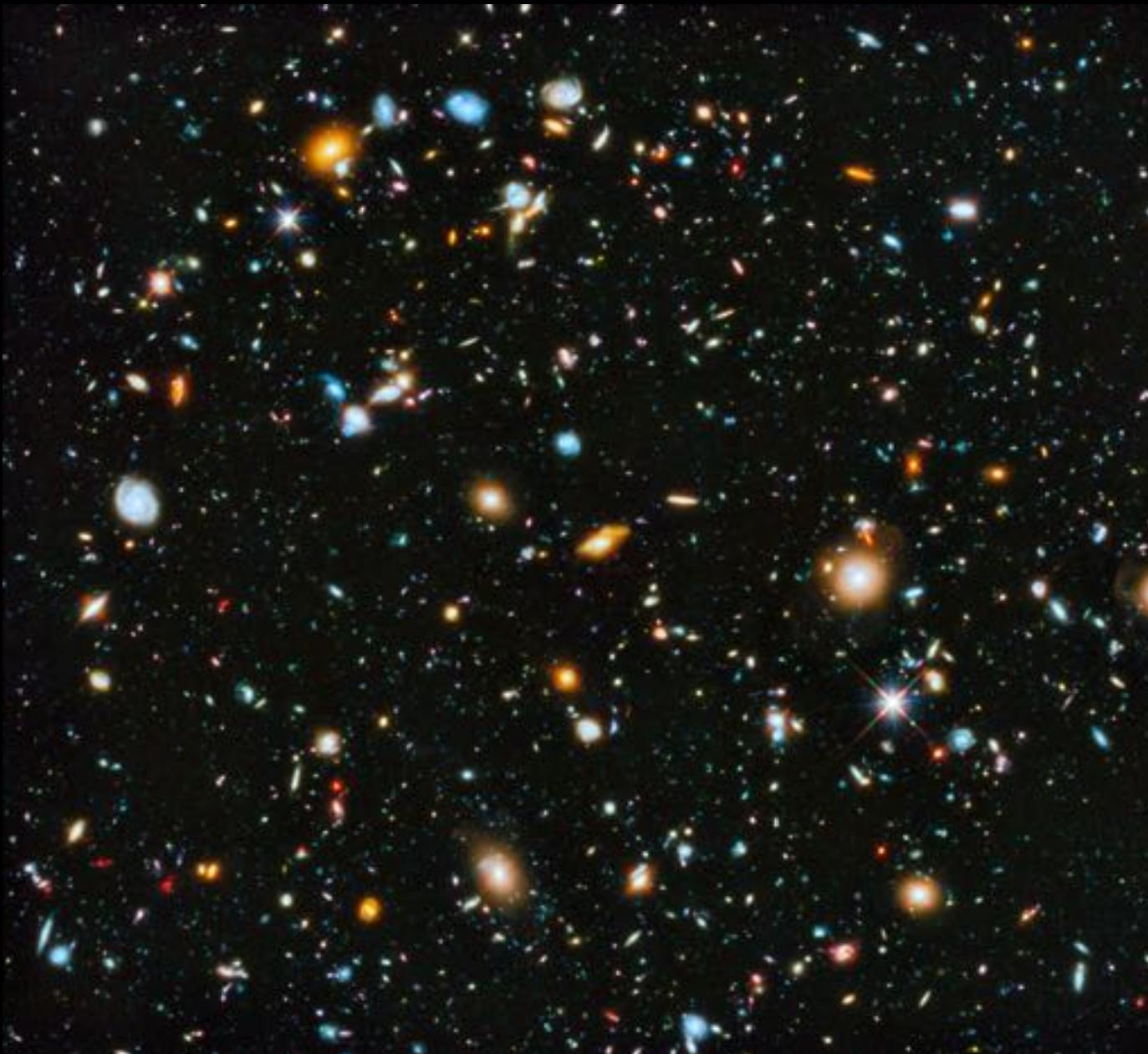
$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}\end{aligned}$$

$$= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)$$



$$P(x_1, x_2, x_3, \dots, x_{100}|y)$$

# Oops... Number of atoms in the universe



What is the big O for # parameters?  
 $m = \# \text{ features.}$

# Big O of Brute Force Joint

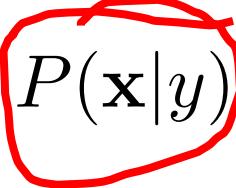
What is the big O for # parameters?  
 $m = \# \text{ features.}$

$$O(2^m)$$

Assuming each feature  
is binary...

Not going to cut it!

# What is the problem here?

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\ &= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$


---

$$P(\mathbf{x}|y) = P(x_1, x_2, \dots, x_m | y)$$

# Naïve Bayes Assumption

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y=\{0,1\}} P(y|\mathbf{x}) \\&= \operatorname{argmax}_{y=\{0,1\}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\&= \operatorname{argmax}_{y=\{0,1\}} P(\mathbf{x}|y)P(y)\end{aligned}$$

---

$$P(\mathbf{x}|y) = P(x_1, x_2, \dots, x_m|y)$$

$$= \prod_i P(x_i|y)$$

The Naïve Bayes  
assumption



## Naïve Bayes Assumption:

$$P(\mathbf{x}|y) = \prod_i P(x_i|y)$$



# Naïve Bayes Classifier

# Naïve Bayes

Our  
prediction  
for  $y$

Is a function  
of  $x$

That chooses the  
best value of  $y$   
given  $x$

$$\hat{y} = g(\mathbf{x}) = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(y|\mathbf{x})$$

$$= \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(\mathbf{x}|y) \hat{P}(y) \quad \text{Bayes rule!}$$

$$= \operatorname{argmax}_y \left( \prod_{i=1}^n \hat{P}(x_i|y) \right) \hat{P}(y)$$

Naïve Bayes  
Assumption

$$= \operatorname{argmax}_y \log \hat{P}(y) + \sum_{i=1}^m \log \hat{P}(x_i|y)$$

This log version is useful for  
numerical stability



# Naïve Bayes Example

- Predict  $Y$  based on observing variables  $X_1$  and  $X_2$ 
  - $X_1$  and  $X_2$  are both indicator variables
    - $X_1$  denotes “likes Star Wars”,  $X_2$  denotes “likes Harry Potter”
  - $Y$  is indicator variable: “likes Lord of the Rings”
    - Use training data to estimate params:  $\hat{P}(x_i|y)$   $\hat{P}(y)$

$X_1 \backslash Y$	0	1	MLE estimates		$X_2 \backslash Y$	0	1	MLE estimates		$Y$	#	MLE est.
0	3	10	0.23	0.77	0	5	8	0.38	0.62	0	13	0.43
1	4	13	0.24	0.76	1	7	10	0.41	0.59	1	17	0.57

- Say someone likes **Star Wars ( $X_1 = 1$ )**, but not **Harry Potter ( $X_2 = 0$ )**
- Will they like “Lord of the Rings”? Need to predict  $Y$ :

$$\hat{y} = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(\mathbf{x}|y) \hat{P}(y) = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(x_1|y) \hat{P}(x_2|y) \hat{P}(y)$$

# Naïve Bayes Example

- Predict  $Y$  based on observing variables  $X_1$  and  $X_2$ 
  - $X_1$  and  $X_2$  are both indicator variables
    - $X_1$  denotes “likes Star Wars”,  $X_2$  denotes “likes Harry Potter”
  - $Y$  is indicator variable: “likes Lord of the Rings”
    - Use training data to estimate params:  $\hat{P}(x_i|y)$   $\hat{P}(y)$

$Y \backslash X_1$	0	1	MLE estimates	
0	3	10	0.23	0.77
1	4	13	0.24	0.76

$Y \backslash X_2$	0	1	MLE estimates	
0	5	8	0.38	0.62
1	7	10	0.41	0.59

$Y$	#	MLE est.
0	13	0.43
1	17	0.57

- Say someone likes **Star Wars ( $X_1 = 1$ )**, but not **Harry Potter ( $X_2 = 0$ )**
- Will they like “Lord of the Rings”? Need to predict  $Y$ .

$$\hat{y} = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(X_1 = x_1 | Y = y) \hat{P}(X_2 = x_2 | Y = y) \hat{P}(Y = y)$$

# Naïve Bayes Example

- Predict  $Y$  based on observing variables  $X_1$  and  $X_2$ 
  - $X_1$  and  $X_2$  are both indicator variables
    - $X_1$  denotes “likes Star Wars”,  $X_2$  denotes “likes Harry Potter”
  - $Y$  is indicator variable: “likes Lord of the Rings”
    - Use training data to estimate params:  $\hat{P}(x_i|y)$   $\hat{P}(y)$

$X_1 \backslash Y$	0	1	MLE estimates		$X_2 \backslash Y$	0	1	MLE estimates		$Y$	#	MLE est.
0	3	10	0.23	0.77	0	5	8	0.38	0.62	0	13	0.43
1	4	13	0.24	0.76	1	7	10	0.41	0.59	1	17	0.57

- Say someone likes **Star Wars ( $X_1 = 1$ )**, but not **Harry Potter ( $X_2 = 0$ )**
- Will they like “Lord of the Rings”? Need to predict  $Y$ :

$$\hat{y} = \operatorname{argmax}_{y \in \{0,1\}} \hat{P}(X_1 = 1|Y = y) \hat{P}(X_2 = 0|Y = y) \hat{P}(Y = y)$$

# One SciFi/Fantasy to Rule them All

$X_1$	0	1	MLE estimates	
$Y$				
0	3	10	0.23	0.77
1	4	13	0.24	0.76

$X_2$	0	1	MLE estimates	
$Y$				
0	5	8	0.38	0.62
1	7	10	0.41	0.59

$Y$	#	MLE est.
0	13	0.43
1	17	0.57

$$\hat{y} = \underset{y \in \{0,1\}}{\operatorname{argmax}} \hat{P}(X_1 = 1|Y = y) \hat{P}(X_2 = 0|Y = y) \hat{P}(Y = y)$$

- Let  $Y = 0$   $\hat{P}(X_1 = 1|Y = 0) \hat{P}(X_2 = 0|Y = 0) \hat{P}(Y = 0)$   
 $= (0.77)(0.38)(0.43) = 0.126$
- Let  $Y = 1$   $\hat{P}(X_1 = 1|Y = 1) \hat{P}(X_2 = 0|Y = 1) \hat{P}(Y = 1)$   
 $= (0.76)(0.41)(0.57) = 0.178$

Since term is greatest when  $Y = 1$ , we predict  $\hat{Y} = 1$

$$P(Y = 1) = K \cdot 0.178 \quad P(Y = 0) = K \cdot 0.126 \quad K = \frac{1}{0.126 + 0.178}$$

# MAP Naïve Bayes

- Predict  $Y$  based on observing variables  $X_1$  and  $X_2$ 
  - $X_1$  and  $X_2$  are both indicator variables
    - $X_1$  denotes “likes Star Wars”,  $X_2$  denotes “likes Harry Potter”
  - $Y$  is indicator variable: “likes Lord of the Rings”
    - Use training data to estimate PMFs:  $\hat{P}(x_i|y)$   $\hat{P}(y)$

$Y$	$X_1$	0	1	MAP estimates
0	3	10		
1	4	13		

$Y$	$X_2$	0	1	MAP estimates
0	5	8		
1	7	10		

$Y$	#	MAP est.
0	13	
1	17	

What prior?

# MAP Naïve Bayes

- Predict  $Y$  based on observing variables  $X_1$  and  $X_2$ 
  - $X_1$  and  $X_2$  are both indicator variables
    - $X_1$  denotes “likes Star Wars”,  $X_2$  denotes “likes Harry Potter”
  - $Y$  is indicator variable: “likes Lord of the Rings”
    - Use training data to estimate PMFs:  $\hat{P}(x_i|y)$   $\hat{P}(y)$

		$X_1$		MAP estimates	
		0	1	0.27	0.73
Y	0	3	10	0.27	0.73
	1	4	13		

		$X_2$		MAP estimates	
		0	1	0	1
Y	0	5	8	0	1
	1	7	10		

	Y	#	MAP est.
0	0	13	
	1	17	

Laplace!

$$p_i = \frac{n_i + 1}{n + m}$$

$$p_i = \frac{n_i + 1}{n + 2}$$

# MAP Naïve Bayes

- Predict  $Y$  based on observing variables  $X_1$  and  $X_2$ 
  - $X_1$  and  $X_2$  are both indicator variables
    - $X_1$  denotes “likes Star Wars”,  $X_2$  denotes “likes Harry Potter”
  - $Y$  is indicator variable: “likes Lord of the Rings”
    - Use training data to estimate PMFs:  $\hat{P}(x_i|y)$   $\hat{P}(y)$

		$X_1$		MAP estimates	
		0	1	0.27	0.73
Y	0	3	10	0.27	0.73
	1	4	13	0.26	0.74

		$X_2$		MAP estimates	
		0	1	0.4	0.6
Y	0	5	8	0.4	0.6
	1	7	10	0.42	0.58

Y	#	MAP est.
0	13	0.45
1	17	0.55

Laplace!

$$p_i = \frac{n_i + 1}{n + m}$$

$$p_i = \frac{n_i + 1}{n + 2}$$



Training Naïve Bayes, is estimating parameters for a multinomial.

Thus training is just counting.

# What is Bayes Doing in my Mail Server

- This is spam:



Let's get Bayesian on your spam:

## A Bayesian Approach to Filtering Junk E-Mail

Mehran Sahami\* Susan Dumais† David Heckerman† Eric Horvitz†

\*Gates Building 1A  
Computer Science Department  
Stanford University  
Stanford, CA 94305-9010  
sahami@cs.stanford.edu

†Microsoft Research  
Redmond, WA 98052-6309  
(adumais, heckerman, horvitz@microsoft.com)

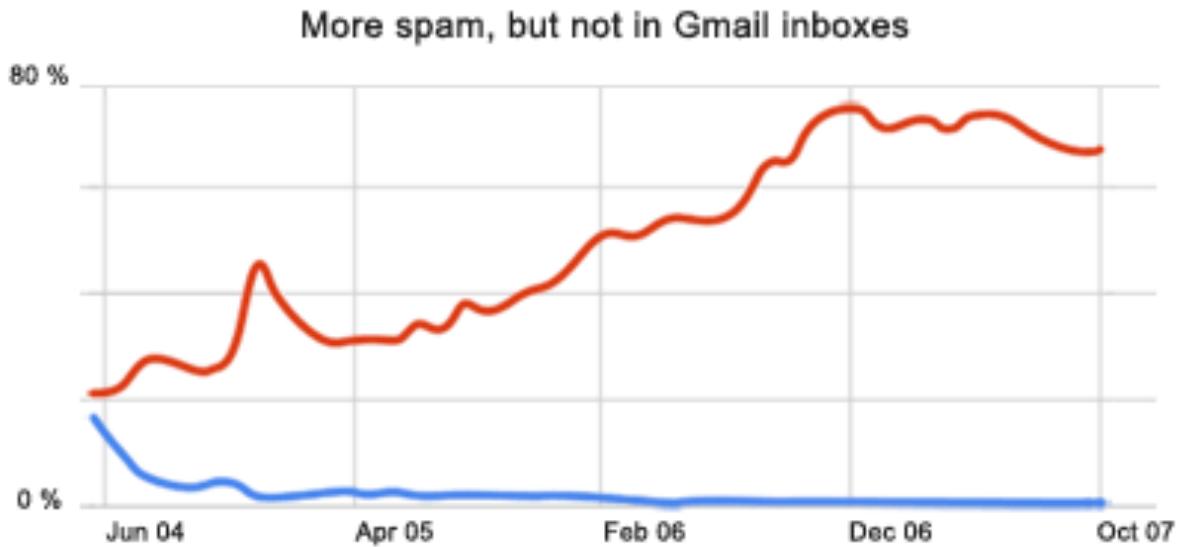
### Abstract

In addressing the growing problem of junk E-mail on the Internet, we examine methods for the automated

contain offensive material (such as graphic pornography), there is often a higher cost to users of actually viewing this mail than simply the time to sort out the junk. Lastly, junk mail not only wastes user time, but

# Spam, Spam... Go Away!

- The constant battle with spam



- Spam prevalence: % of all incoming Gmail traffic (before filtering) that is spam
- Missed spam: % of total spam reported by Gmail users

As the amount of spam has increased, Gmail users have received less of it in their inboxes, reporting a rate less than 1%.

*“And machine-learning algorithms developed to merge and rank large sets of Google search results allow us to combine hundreds of factors to classify spam.”*

# Email Classification

- Want to predict if an email is spam or not
  - Start with the input data
    - Consider a lexicon of  $m$  words (Note: in English  $m \approx 100,000$ )
    - Define  $m$  indicator variables  $\mathbf{X} = \langle X_1, X_2, \dots, X_m \rangle$
    - Each variable  $X_i$  denotes if word  $i$  appeared in a document or not
    - Note:  $m$  is huge, so make “Naive Bayes” assumption
  - Define output classes  $Y$  to be: {spam, non-spam}
  - Given training set of  $N$  previous emails
    - For each email message, we have a training instance:  $\mathbf{X} = \langle X_1, X_2, \dots, X_m \rangle$  noting for each word, if it appeared in email
    - Each email message is also marked as spam or not (value of  $Y$ )

# Training the Classifier

- Given  $N$  training pairs:

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$$

- Learning

- Estimate probabilities  $P(Y)$  and each  $P(X_i | Y)$  for all  $i$ 
  - Many words are likely to not appear at all in given set of email
- Laplace estimate:  $\hat{p}(X_i = 1 | Y = \text{spam})_{\text{Laplace}} = \frac{(\# \text{spam emails with word } i) + 1}{\text{total } \# \text{spam emails} + 2}$

- Classification

- For a new email, generate  $\mathbf{X} = \langle X_1, X_2, \dots, X_m \rangle$
- Classify as spam or not using:  $\hat{Y} = \arg \max_y \hat{P}(\mathbf{X} | Y) \hat{P}(Y)$
- Employ Naive Bayes assumption:  $\hat{P}(\mathbf{X} | Y) = \prod_{i=1}^m \hat{P}(X_i | Y)$



Training Naïve Bayes, is estimating parameters for a multinomial.

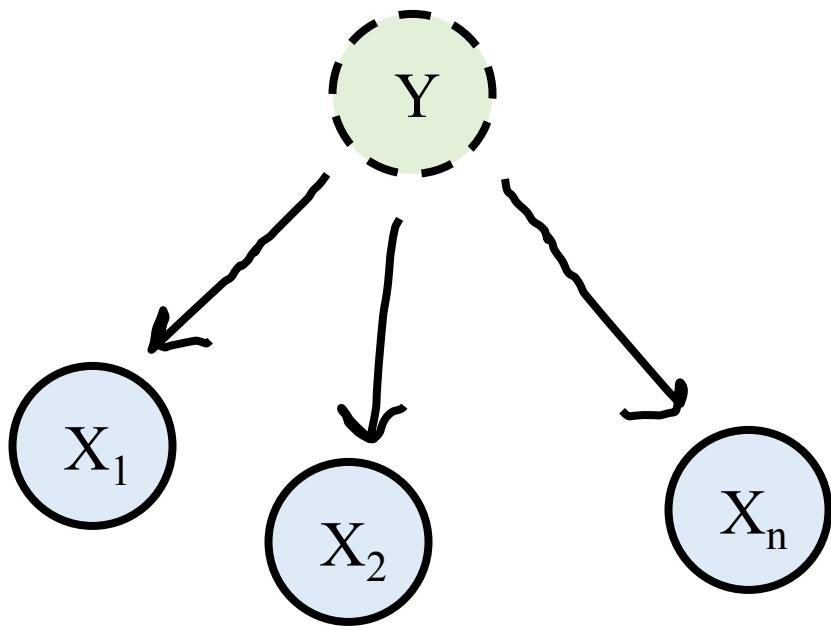
Thus it is just counting.

# How Does This Do?

- After training, can test with another set of data
  - “Testing” set also has known values for Y, so we can see how often we were right/wrong in predictions for Y
  - Spam data
    - Email data set: 1789 emails (1578 spam, 211 non-spam)
    - First, 1538 email messages (by time) used for training
    - Next 251 messages used to test learned classifier
  - Criteria:
    - Precision = # *correctly* predicted class Y / # predicted class Y
    - Recall = # *correctly* predicted class Y / # real class Y messages

	Spam		Non-spam	
	Precision	Recall	Precision	Recall
<b>Words only</b>	97.1%	94.3%	87.7%	93.4%
<b>Words + add'l features</b>	<b>100%</b>	98.3%	96.2%	<b>100%</b>

# Naïve Bayes as Bayes Net



Assumption:

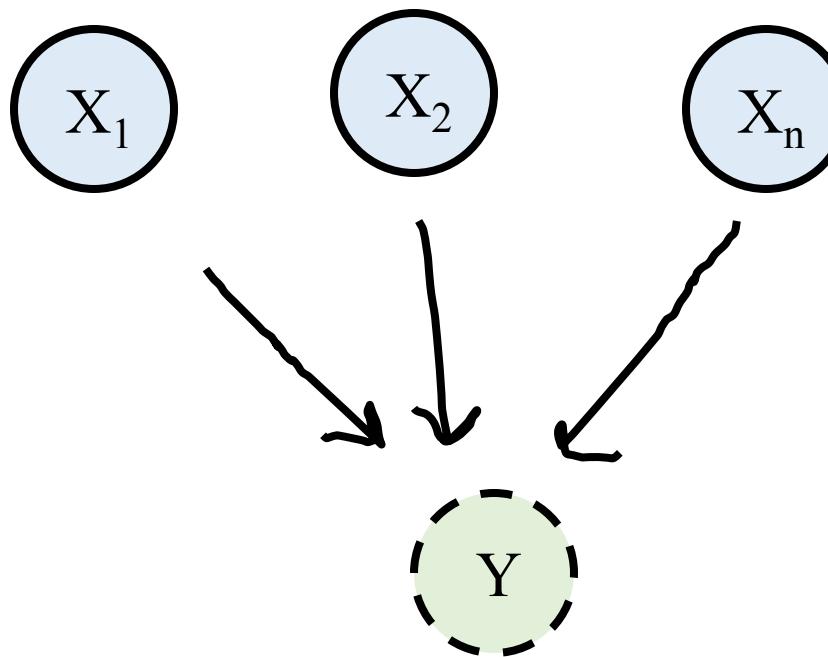
$$P(\mathbf{x}, y) = P(y) \prod_i P(x_i|y)$$

Parameters:

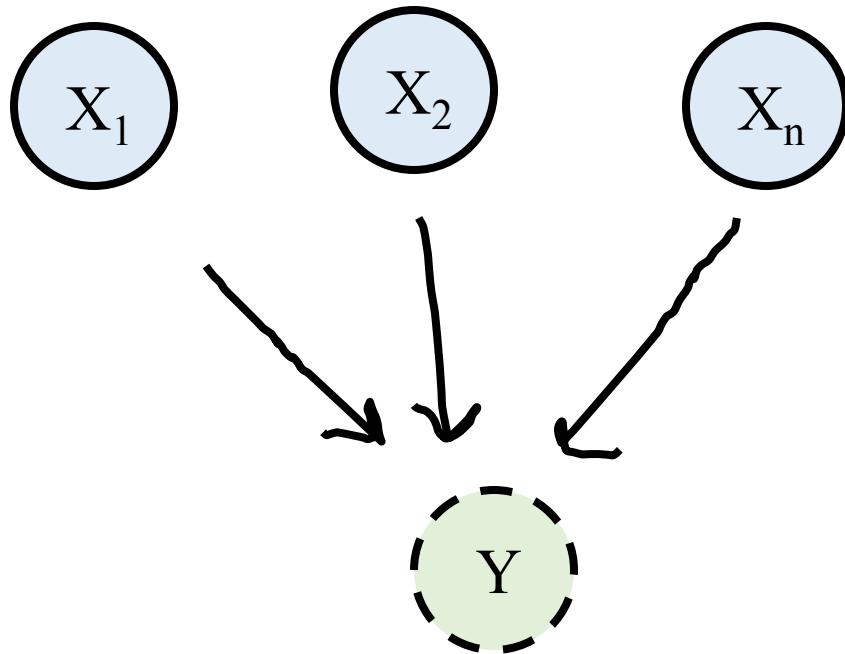
$P(X_i = x_i | \text{Parents of } X_i \text{ take on specified values})$

$$P(Y = y)$$

# Why not this?



# Why not this?



Assumption:

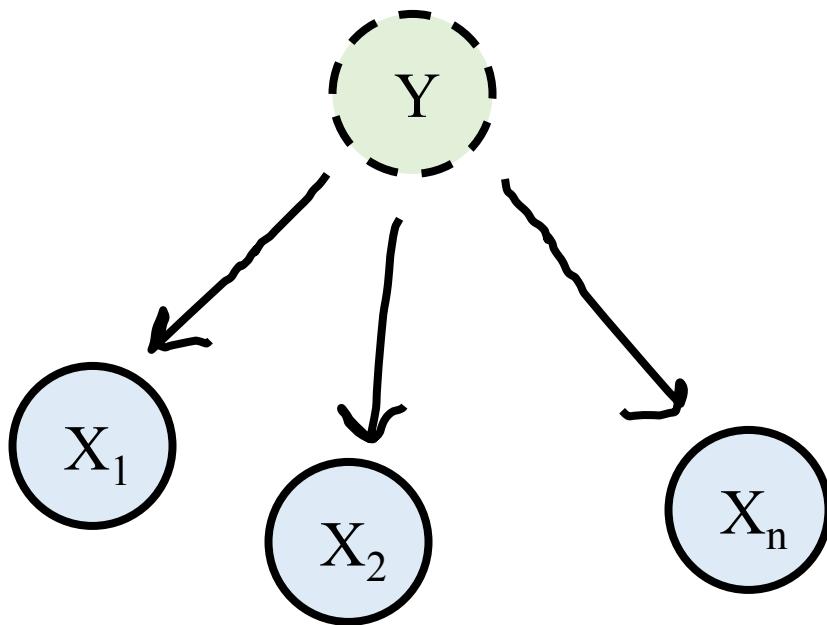
$$P(\mathbf{x}, y) = P(y|\mathbf{x}) \prod_i P(x_i)$$

Parameters:

$P(Y = y | \text{Parents of } Y \text{ take on specified values})$

$P(X_i = x_i)$

# General Bayes Net Learning



Assumption:

$$P(\mathbf{x}, y) = P(y) \prod_i P(x_i|y)$$

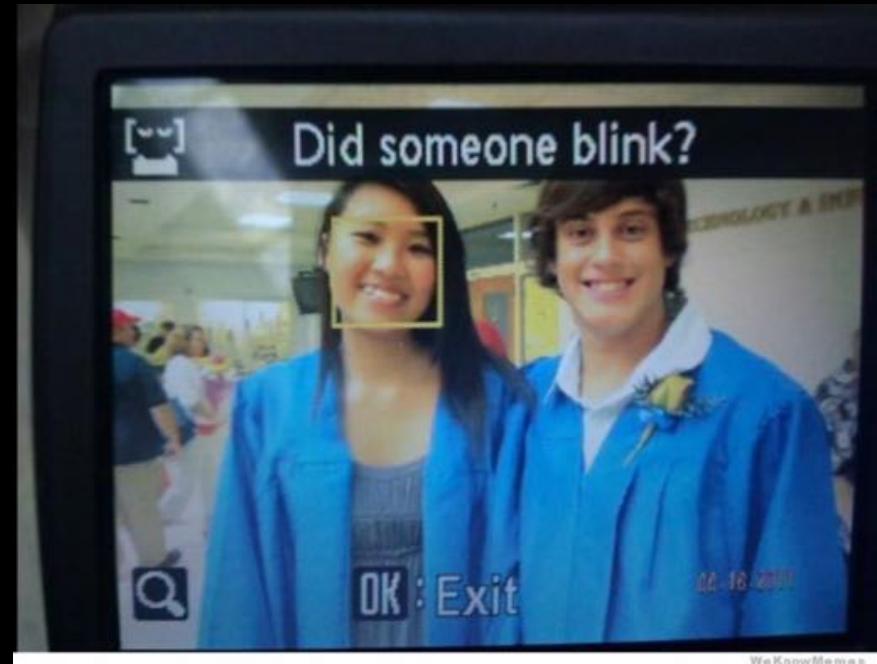
Parameters:

$P(X_i = x_i | \text{Parents of } X_i \text{ take on specified values})$

$$P(Y = y)$$

# On biased datasets

# Ethics and Datasets?



Sometimes machine learning feels universally unbiased.

We can even prove our estimators are “unbiased” 😊

Google/Nikon/HP had biased datasets

Ancestry dataset prediction

East Asian

or

Ad Mixed American (Native, European and  
African Americans)

It is much easier  
to write a binary classifier  
when learning ML  
for the first time

# Learn Two Things From This

1. What classification with DNA Single Nucleotide Polymorphisms looks like.
2. The importance of choosing the right data to learn from. Your results will be as biased as your dataset.

Know it so you can beat it!

Ethics in Machine Learning  
is a whole new field