Linear classifiers: Logistic regression

Emily Fox & Carlos Guestrin

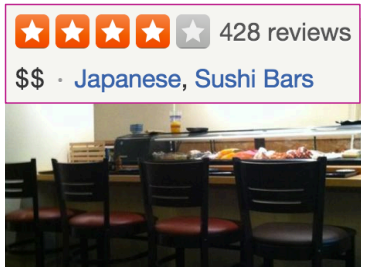
Machine Learning Specialization

University of Washington

Predicting sentiment by topic:
An intelligent restaurant review system

It's a big day & I want to book a table at a nice Japanese restaurant

Positive reviews not positive about everything



Sample review:

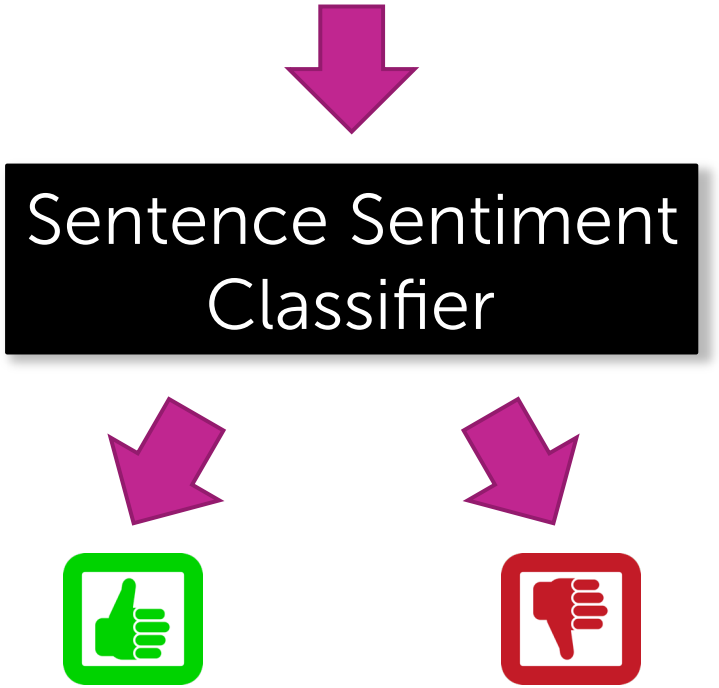
Watching the chefs create incredible edible art made the experience very unique.

My wife tried their ramen and it was pretty forgettable.

All the sushi was delicious! Easily best sushi in Seattle.

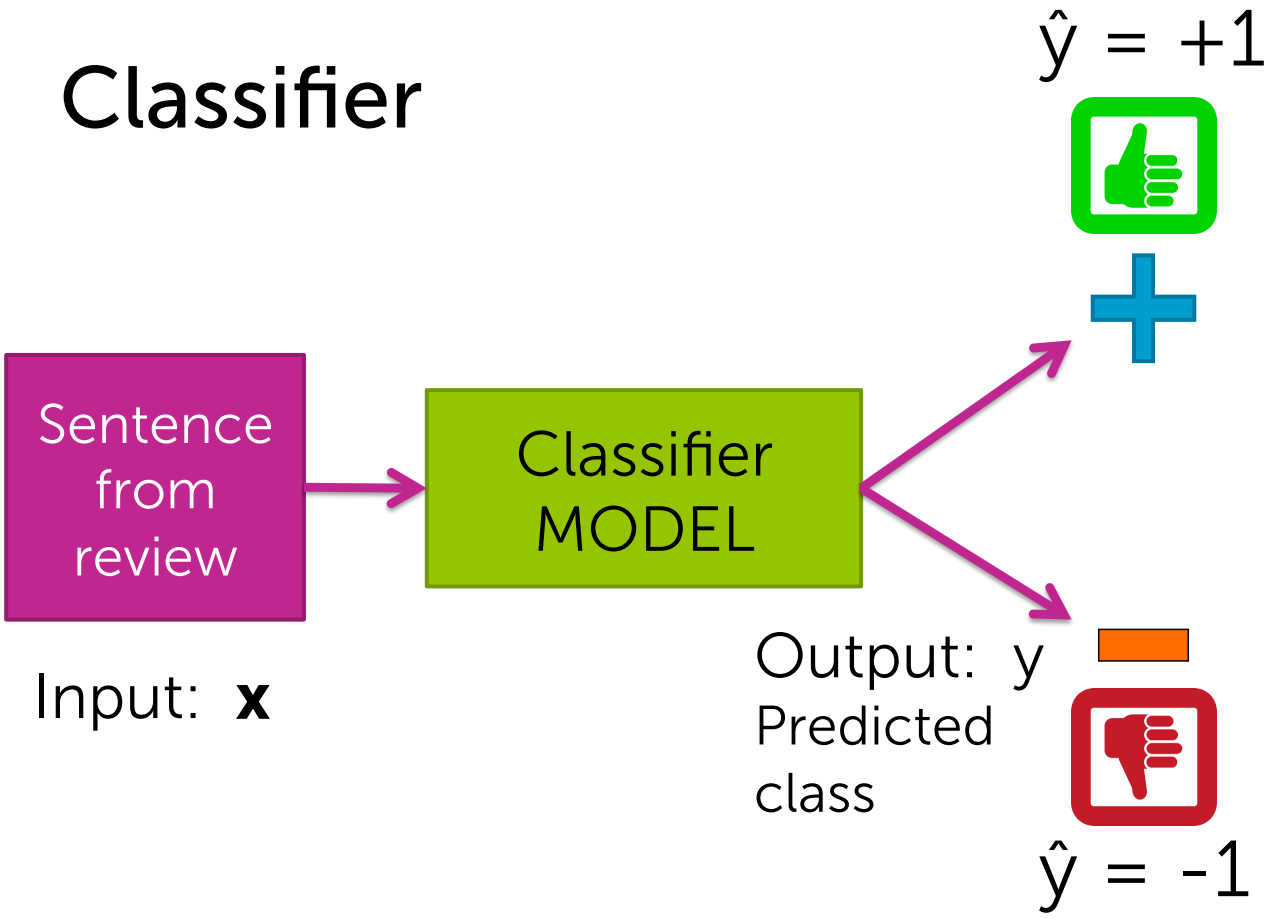
Classifying sentiment of review

Easily best sushi in Seattle.



Linear classifier: Intuition

Classifier



Note: we'll start talking about 2 classes, and address multiclass later

A (linear) classifier

- Will use training data to learn a weight or coefficient for each word

Word	Coefficient
good	1.0
great	1.5
awesome	2.7
bad	-1.0
terrible	-2.1
awful	-3.3
restaurant, the, we, where, …	0.0
…	…

Scoring a sentence

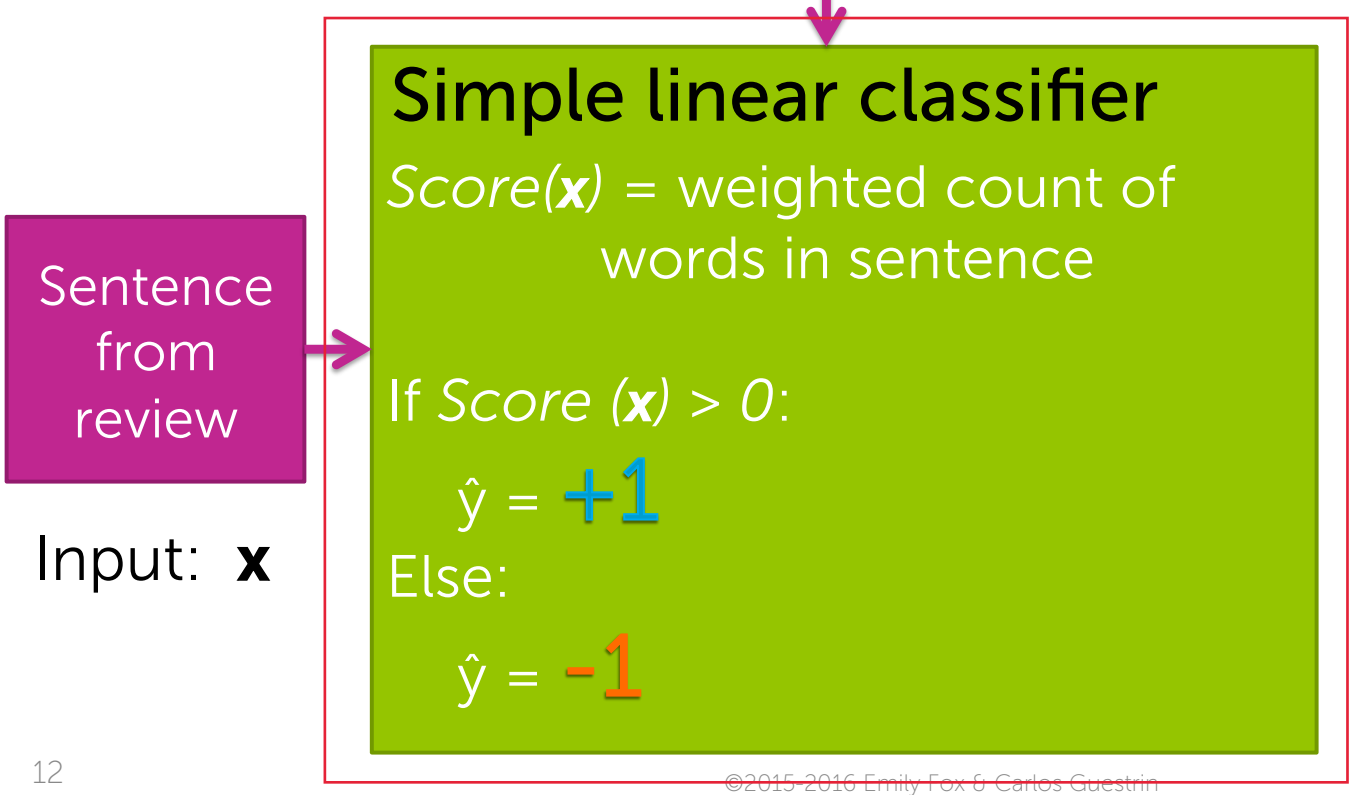
Word	Coefficient
good	1.0
great	1.2
awesome	1.7
bad	-1.0
terrible	-2.1
awful	-3.3
restaurant, the, we, where, …	0.0
…	…

Input $\mathbf{x}_i$:

Sushi was great,
the food was awesome,
but the service was terrible.

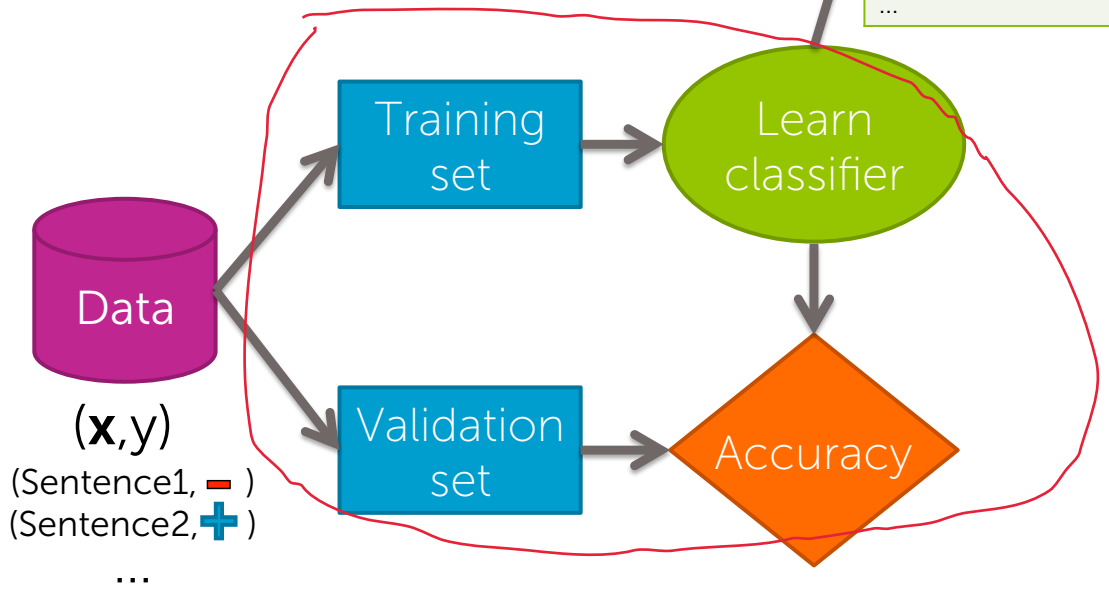
Called a linear classifier, because output is weighted sum of input.

Word	Coefficient
…	…



Training a classifier = Learning the coefficients

Word	Coefficient
good	1.0
awesome	1.7
bad	-1.0
awful	-3.3
…	…

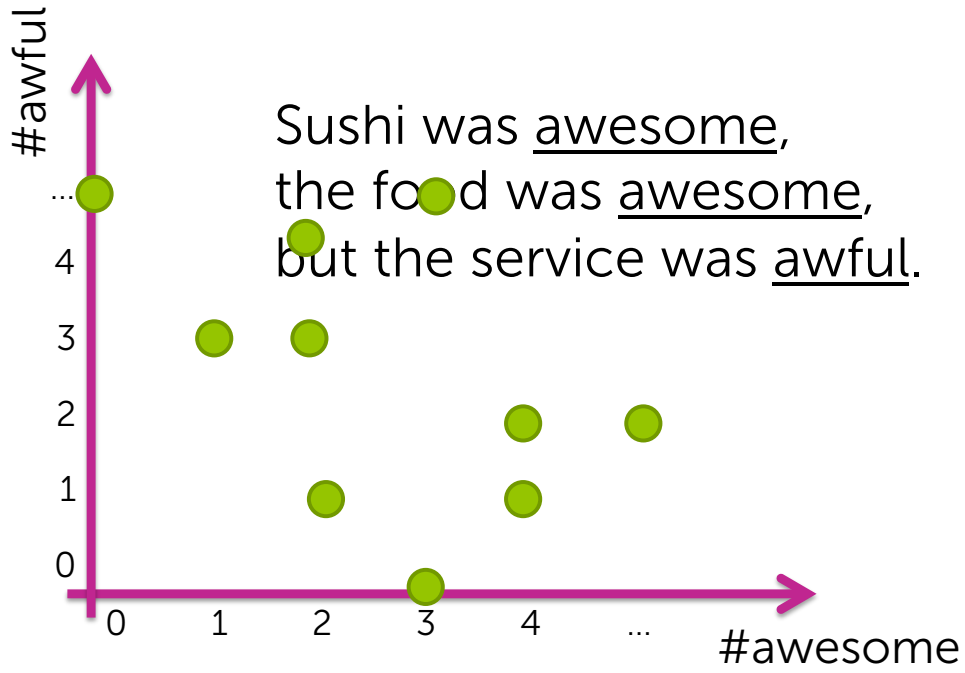


Decision boundaries

Suppose only two words had non-zero coefficient

Word	Coefficient
#awesome	1.0
#awful	-1.5

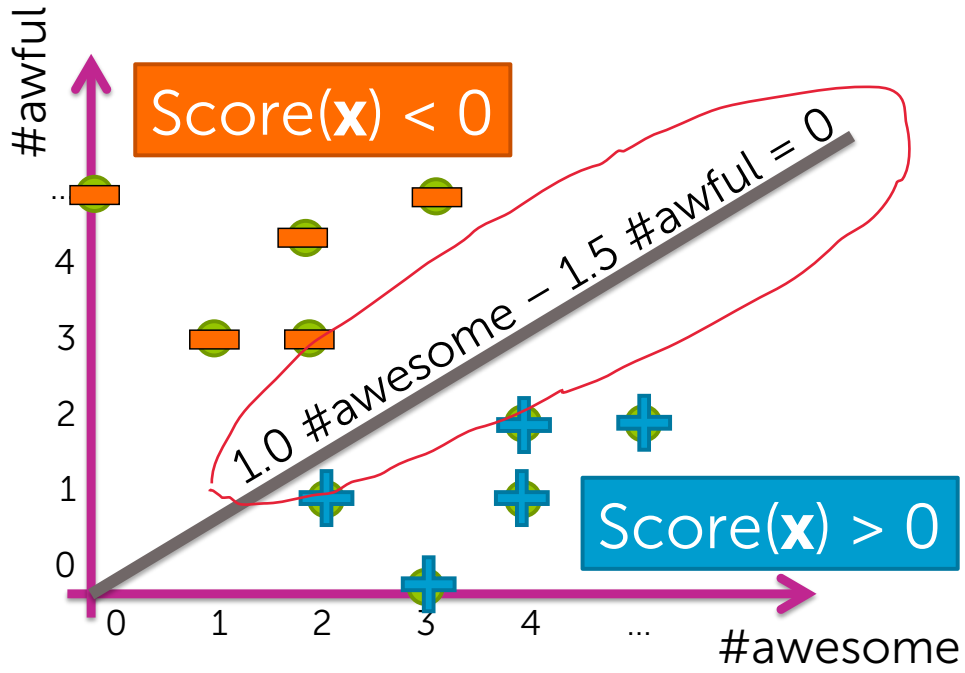
Score(x) = 1.0 #awesome – 1.5 #awful



Decision boundary example

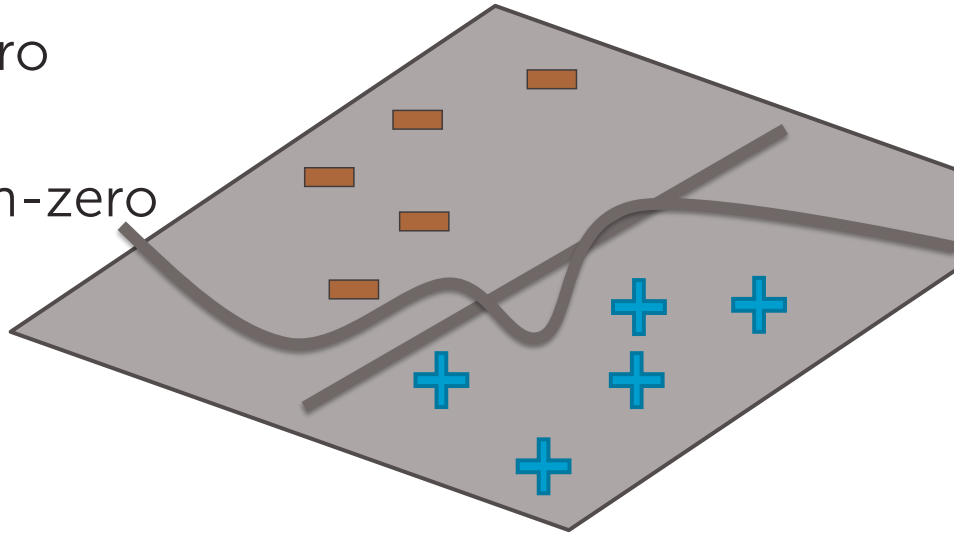
Word	Coefficient
#awesome	1.0
#awful	-1.5

Score(x) = 1.0 #awesome – 1.5 #awful

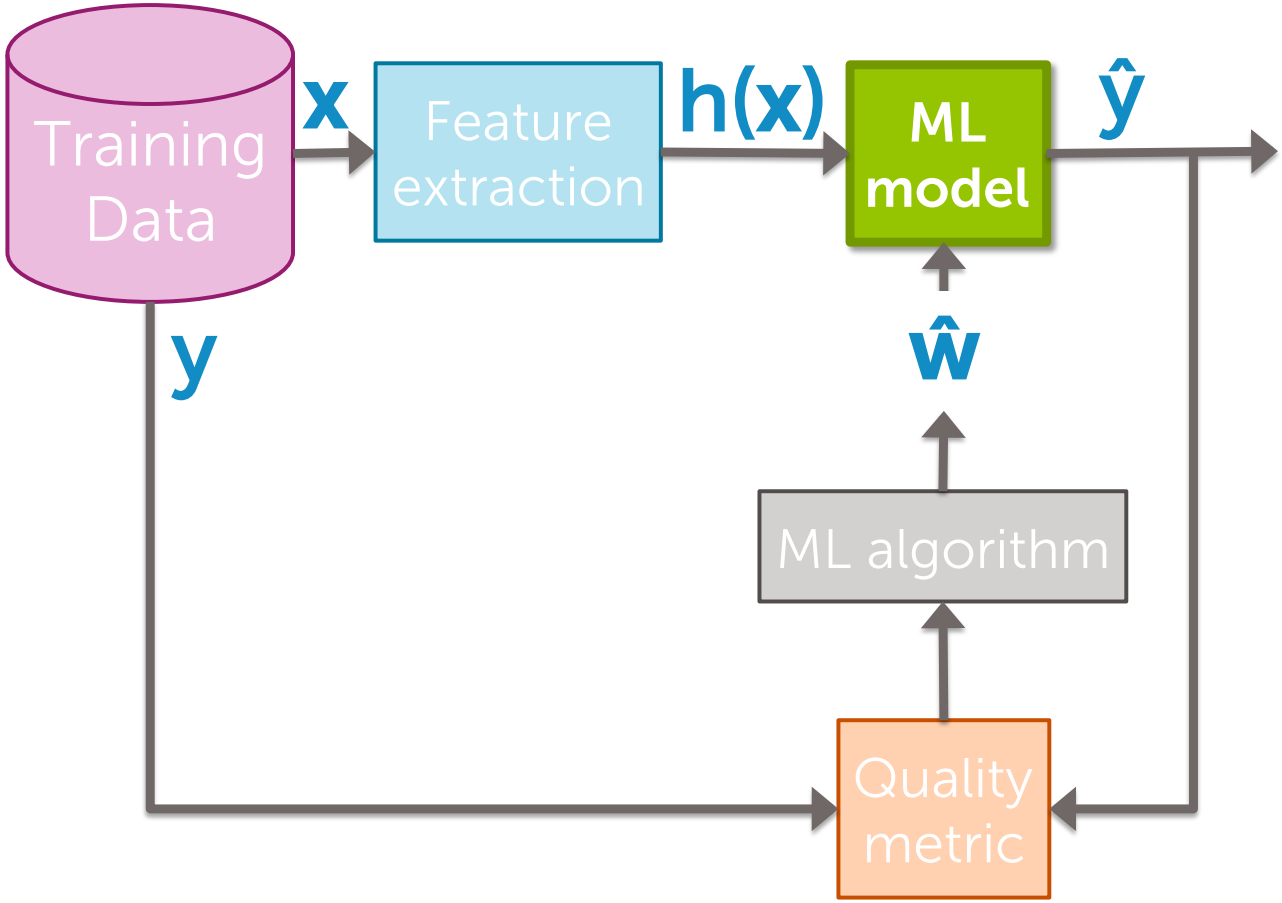


Decision boundary separates positive & negative predictions

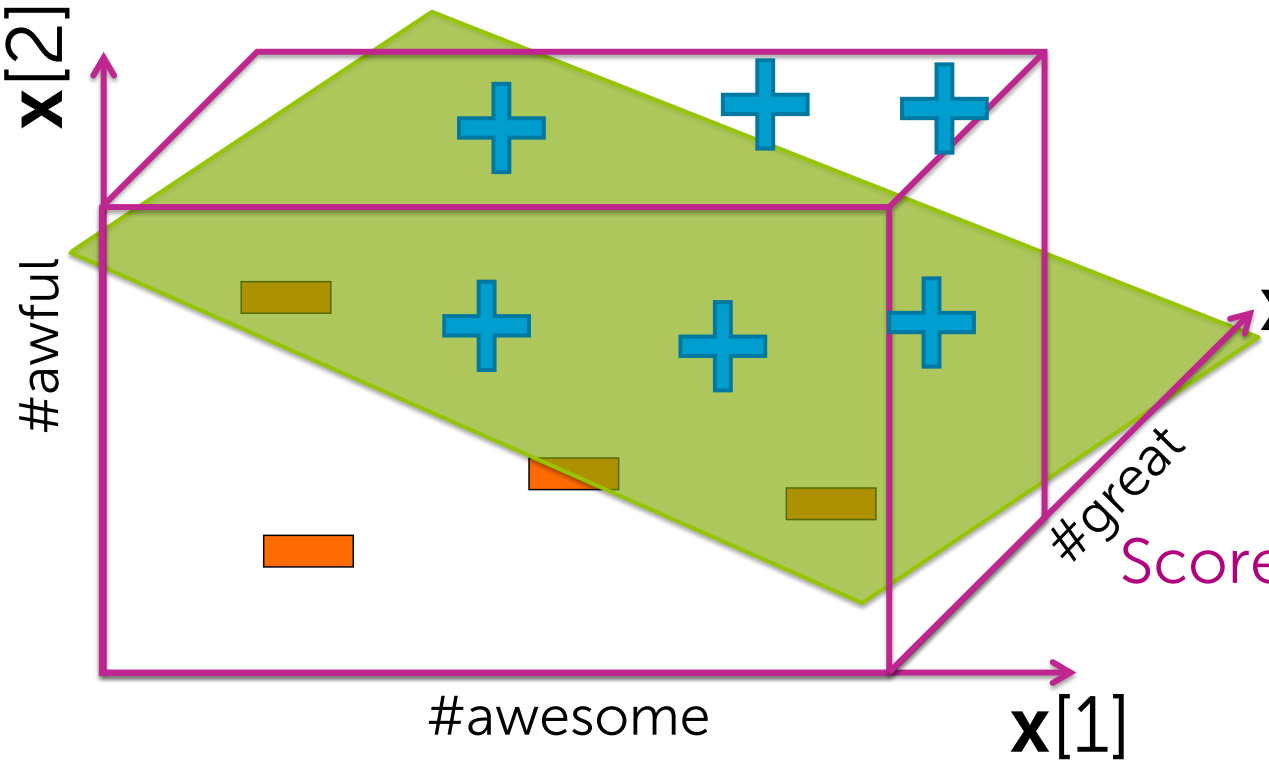
- For linear classifiers:
  - When 2 coefficients are non-zero
    ➔ line
  - When 3 coefficients are non-zero
    ➔ plane
  - When many coefficients are non-zero
    ➔ hyperplane
- For more general classifiers
  ➔ more complicated shapes



Linear classifier: Model



Coefficients of classifier



$$\text{Score}(\mathbf{x}) = w_0 + w_1 \#\text{awesome} + w_2 \#\text{awful} + w_3 \#\text{great}$$

General notation

Output: y ← {-1,+1}

Inputs: $\mathbf{x} = (\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[d])$

← d-dim vector

Notational conventions:

- $\mathbf{x}[j]$ = j^{th} input (*scalar*)
- $h_j(\mathbf{x})$ = j^{th} feature (*scalar*)
- $\mathbf{x}_i$ = input of i^{th} data point (*vector*)
- $\mathbf{x}_i[j]$ = j^{th} input of i^{th} data point (*scalar*)

Simple hyperplane

Model: $\hat{y}_i = \text{sign}(\text{Score}(\mathbf{x}_i))$

$\text{Score}(\mathbf{x}_i) = w_0 + w_1 \mathbf{x}_i[1] + \ldots + w_d \mathbf{x}_i[d] = \mathbf{w}^\top \mathbf{x}_i$

feature 1 = 1

feature 2 = $\mathbf{x}[1]$ … e.g., #awesome

feature 3 = $\mathbf{x}[2]$ … e.g., #awful

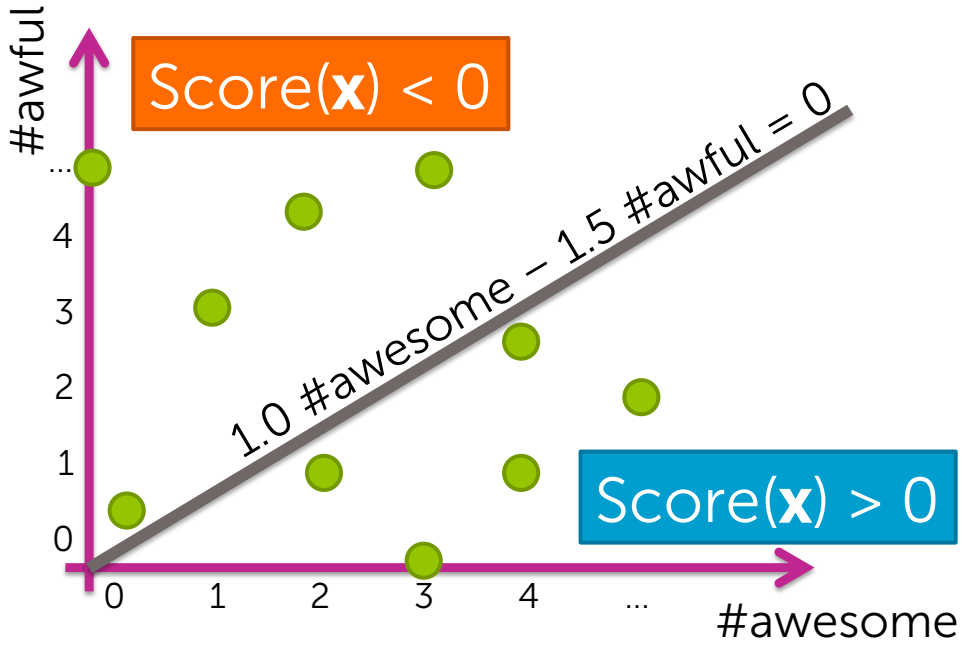
…

feature d+1 = $\mathbf{x}[d]$ … e.g., #ramen

Decision boundary: effect of changing coefficients

Input	Coefficient	Value
	w_0	0.0
#awesome	w_1	1.0
#awful	w_2	-1.5

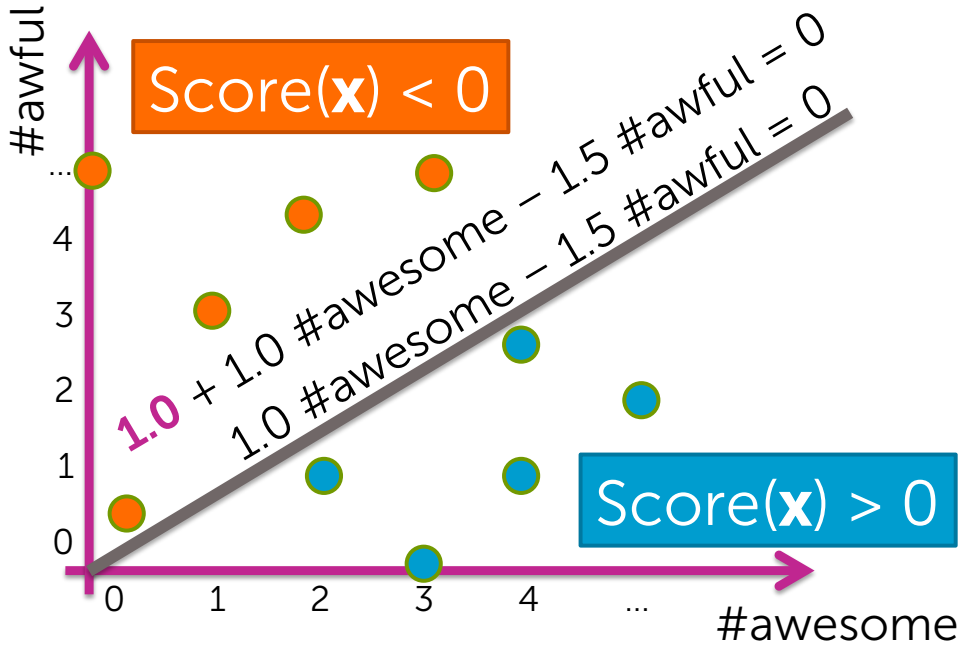
Score(x) = 1.0 #awesome – 1.5 #awful



Decision boundary: effect of changing coefficients

Input	Coefficient	Value
	w_0	**1.0**
#awesome	w_1	1.0
#awful	w_2	-1.5

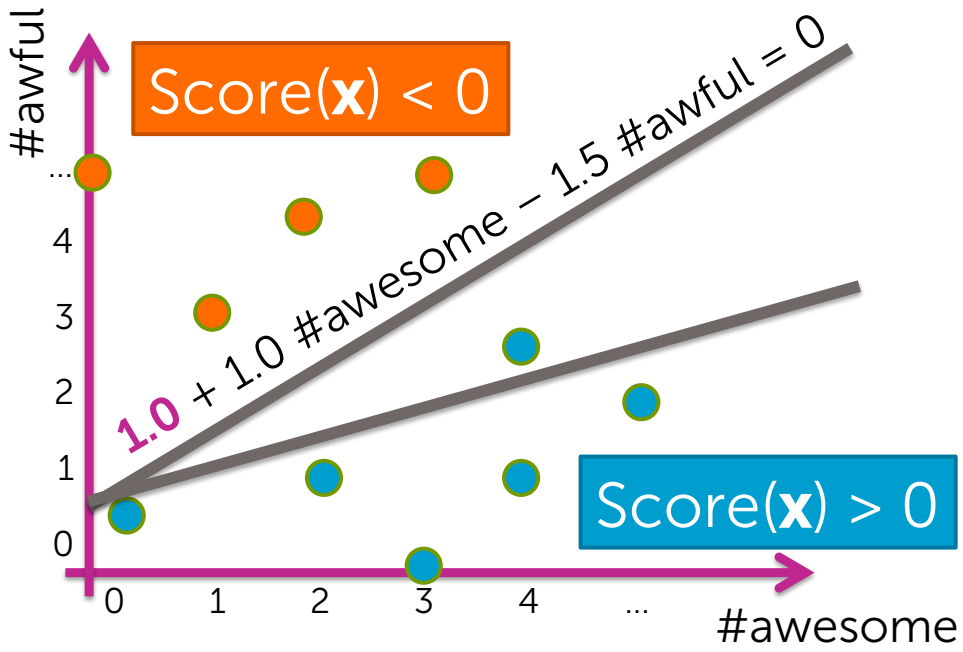
Score(x) = **1.0** #awesome – 1.5 #awful



Decision boundary: effect of changing coefficients

Input	Coefficient	Value
	w_0	**1.0**
#awesome	w_1	1.0
#awful	w_2	**–3.0**

Score(x) = **1.0** + 1.0 #awesome **– 3.0** #awful



More generic features… D-dimensional hyperplane

Model: $\hat{y}_i = \text{sign}(\text{Score}(\mathbf{x}_i))$

$$\text{Score}(\mathbf{x}_i) = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i)$$

$$= \sum_{j=0}^{D} w_j\, h_j(\mathbf{x}_i) = \mathbf{w}^T h(\mathbf{x}_i)$$

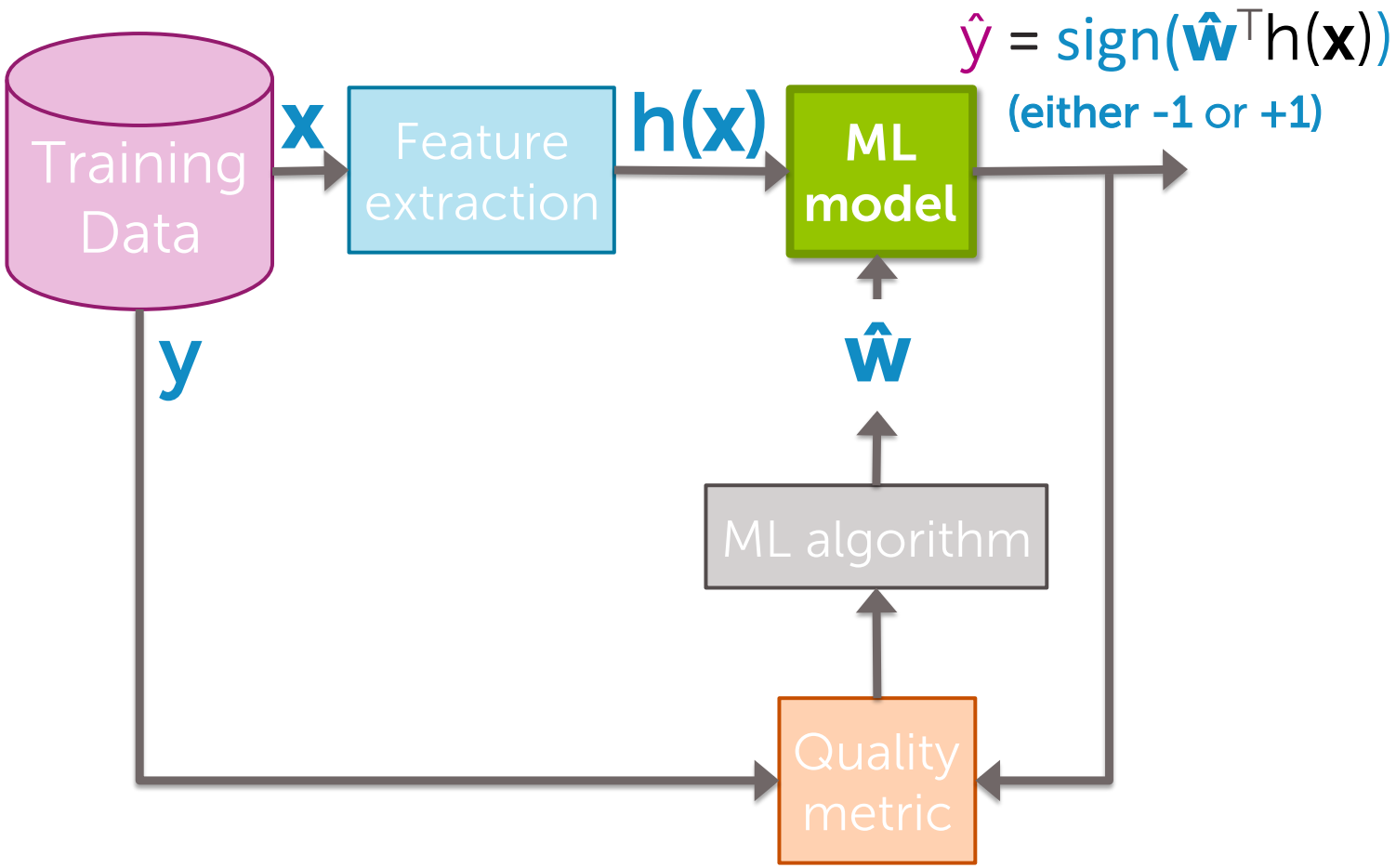
feature 1 = $h_0(\mathbf{x})$ … e.g., 1

feature 2 = $h_1(\mathbf{x})$ … e.g., $\mathbf{x}$[1] = #awesome

feature 3 = $h_2(\mathbf{x})$ … e.g., $\mathbf{x}$[2] = #awful
or, log($\mathbf{x}$[7]) $\mathbf{x}$[2] = log(#bad) x #awful
or, tf-idf("awful")

…

feature D+1 = $h_D(\mathbf{x})$ … some other function of $\mathbf{x}$[1],…, $\mathbf{x}$[d]

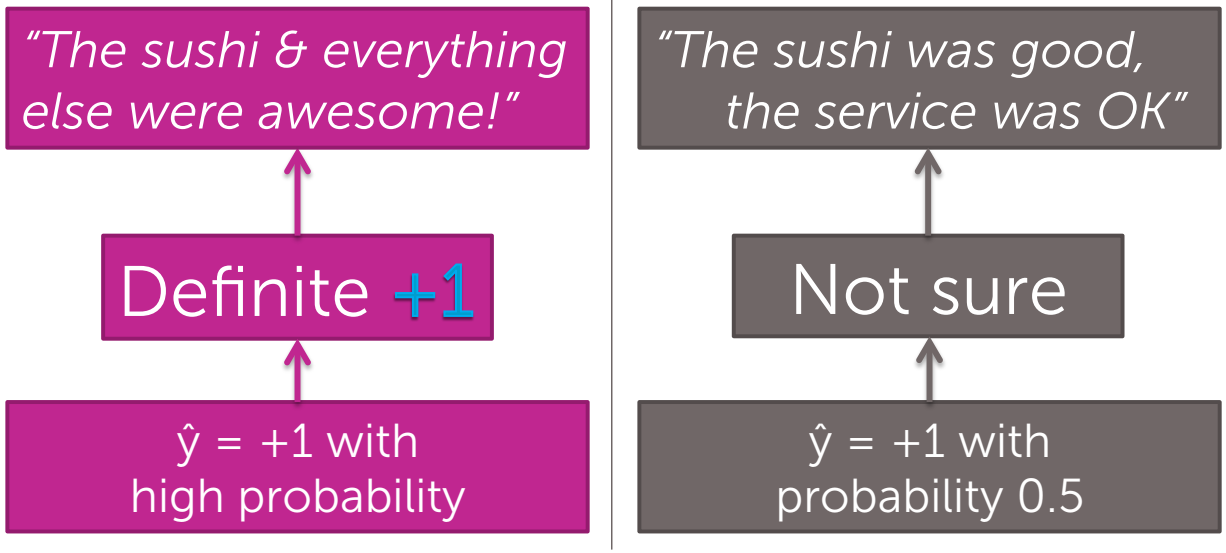


Are you sure about the prediction?

Class probability

How confident is your prediction?

- Thus far, we've outputted a prediction +1 or -1
- But, how sure are you about the prediction?



Basics of probabilities – quick review

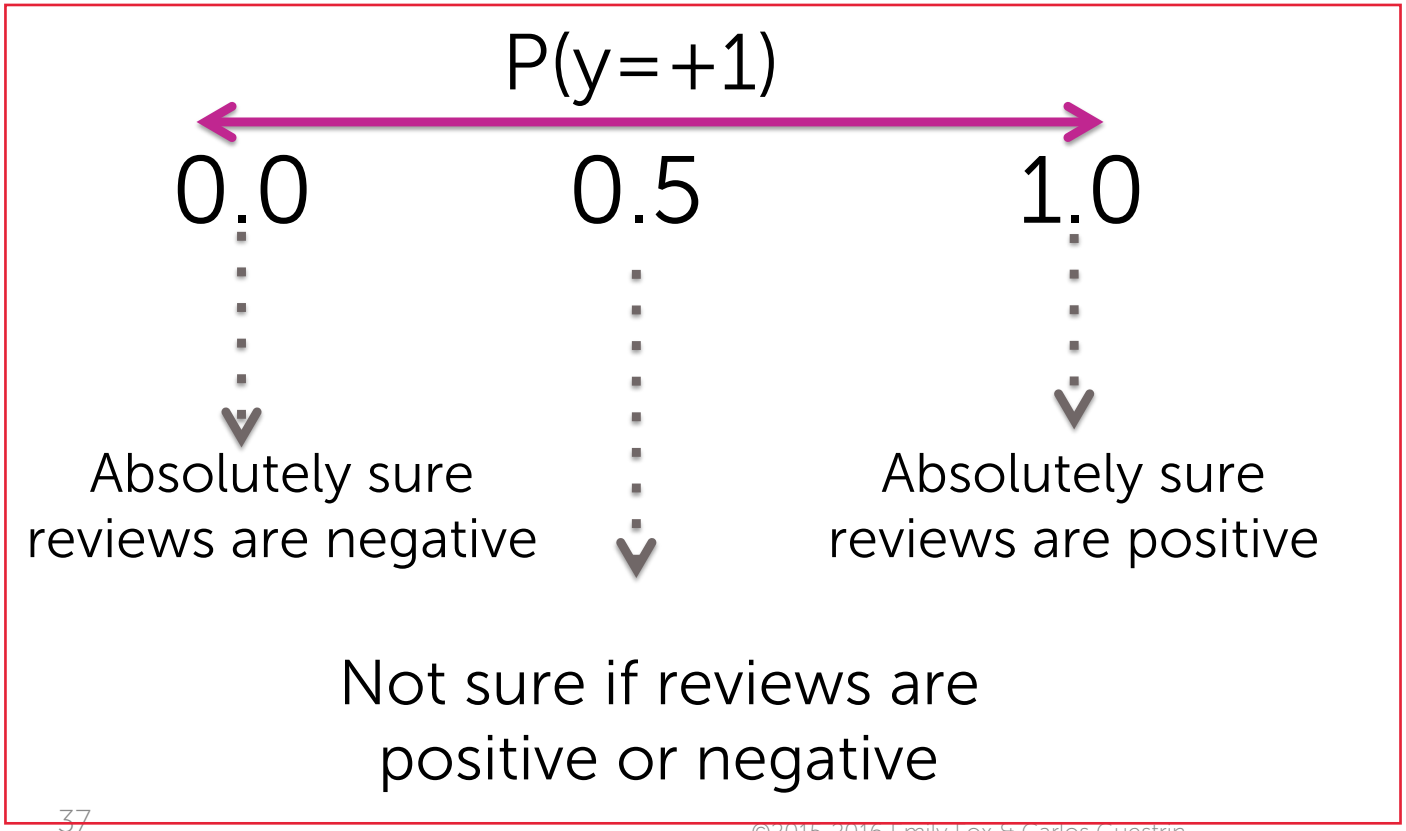
Basic probability

Probability a review is positive is 0.7

x = review text	y = sentiment
All the sushi was delicious! Easily best sushi in Seattle.	+1
The sushi & everything else were awesome!	+1
My wife tried their ramen, it was pretty forgettable.	-1
The sushi was good, the service was OK	+1
…	…

I expect 70% of rows to have y = +1
(Exact number will vary for each specific dataset)

Interpreting probabilities as degrees of belief



Key properties of probabilities

Property	Two class (e.g., y is +1 or -1)	Multiple classes (e.g., y is dog, cat or bird)
Probabilities always between 0 & 1		
Probabilities sum up to 1		

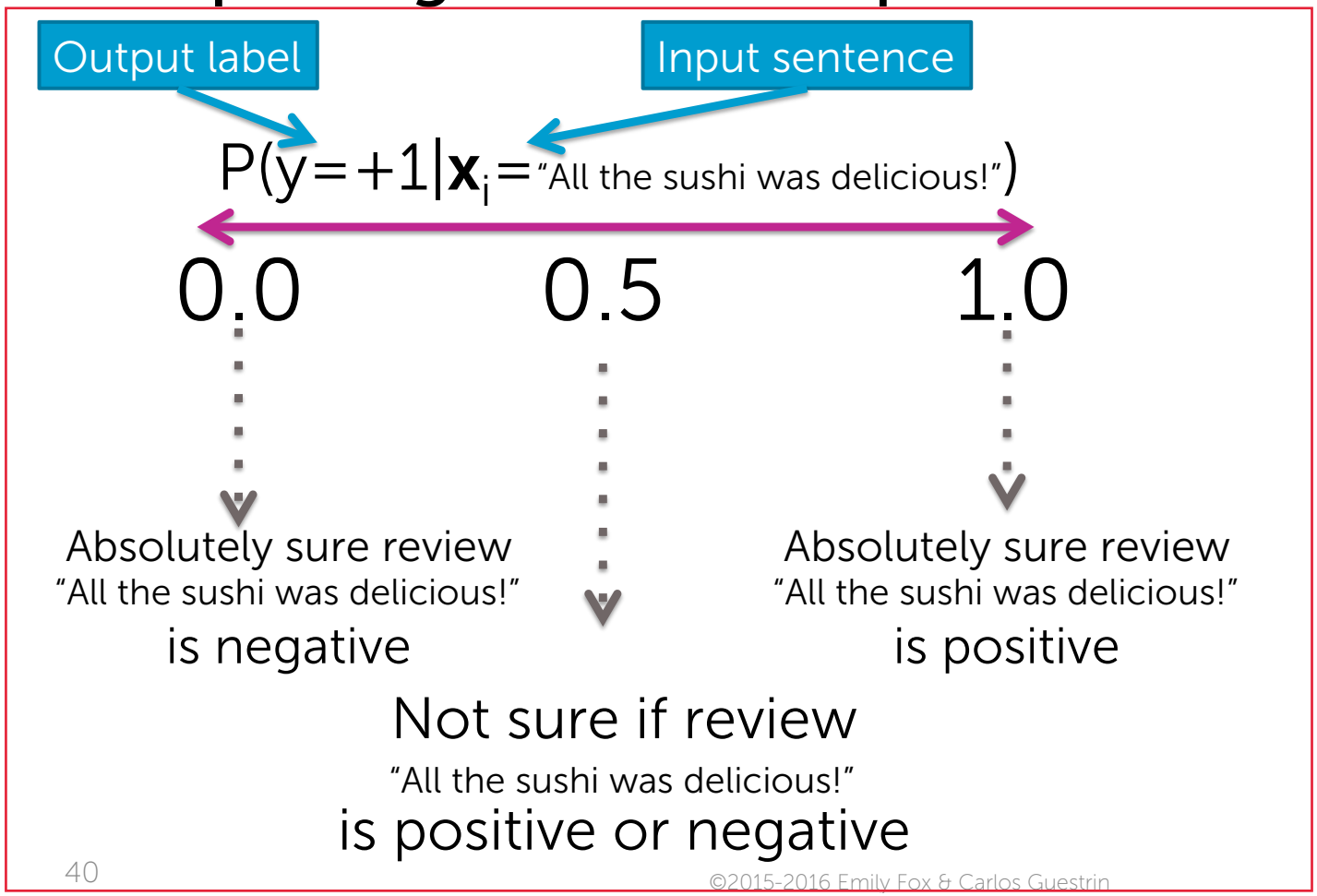
Conditional probability

Probability a review with 3 "awesome" and 1 "awful" is positive is 0.9

x = review text	y = sentiment
All the sushi was delicious! Easily best sushi in Seattle.	+1
Sushi was **awesome** & everything else was **awesome**! The service was **awful**, but overall **awesome** place!	+1
My wife tried their ramen, it was pretty forgettable.	-1
The sushi was good, the service was OK	+1
…	…
awesome … **awesome** … **awful** … **awesome**	+1
…	…
awesome … **awesome** … **awful** … **awesome**	-1
…	…
…	…
awesome … **awesome** … **awful** … **awesome**	+1

I expect 90% of rows with reviews containing 3 "awesome" & 1 "awful" to have y = +1
(Exact number will vary for each specific dataset)

Interpreting conditional probabilities

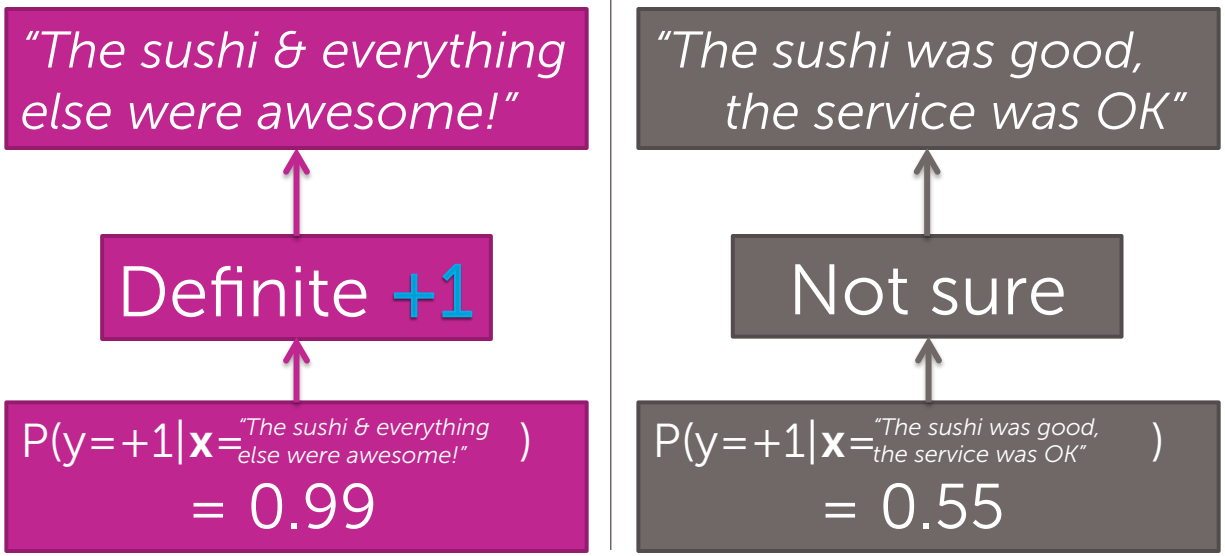


Key properties of conditional probabilities

Property	Two class (e.g., y is +1 or -1, $\mathbf{x}_i$ is review text)	Multiple classes (e.g., y is dog, cat or bird, $\mathbf{x}_i$ is image)
Conditional probabilities always between 0 & 1		
Conditional probabilities sum up to 1 over y, but not over **x**		

Using probabilities in classification

How confident is your prediction?



Many classifiers provide a degree of certainty:

Output label → $P(y|\mathbf{x})$ ← Input sentence

Extremely useful in practice

Goal: Learn conditional probabilities from data

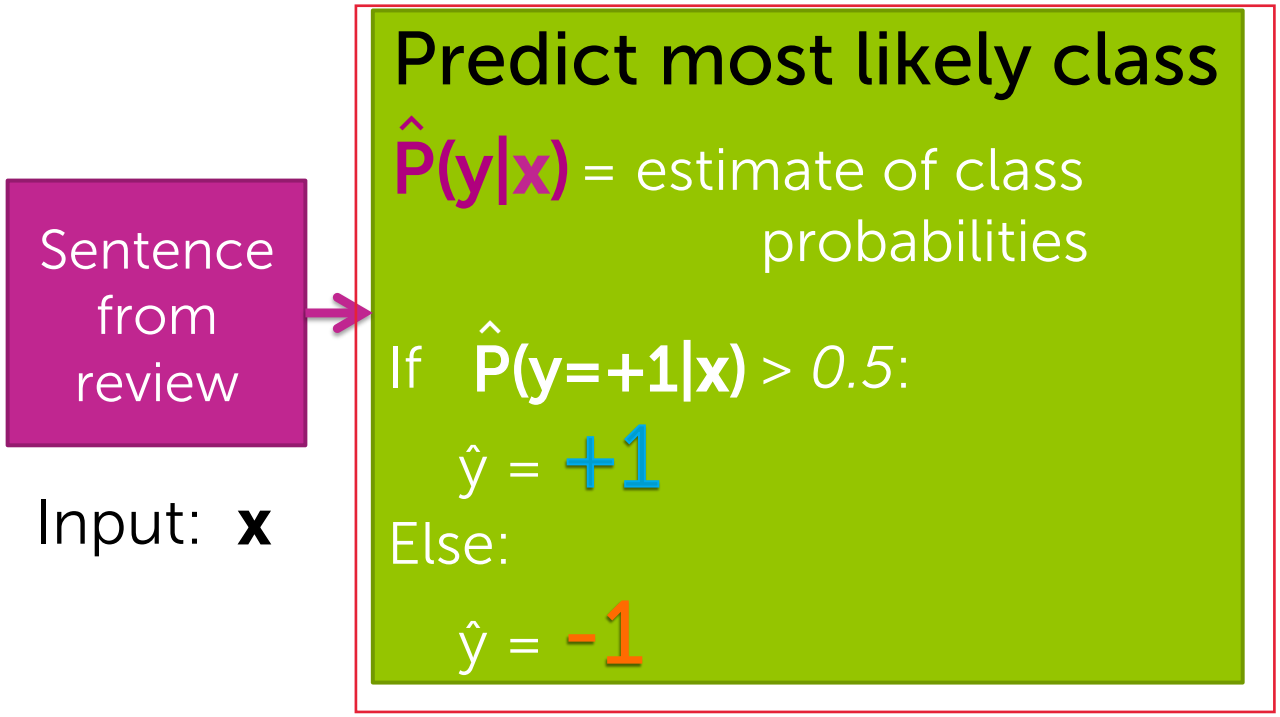
Training data: N observations $(\mathbf{x}_i, y_i)$

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
0	2	-1
3	3	-1
4	1	+1
…	…	…

Optimize **quality metric** on training data

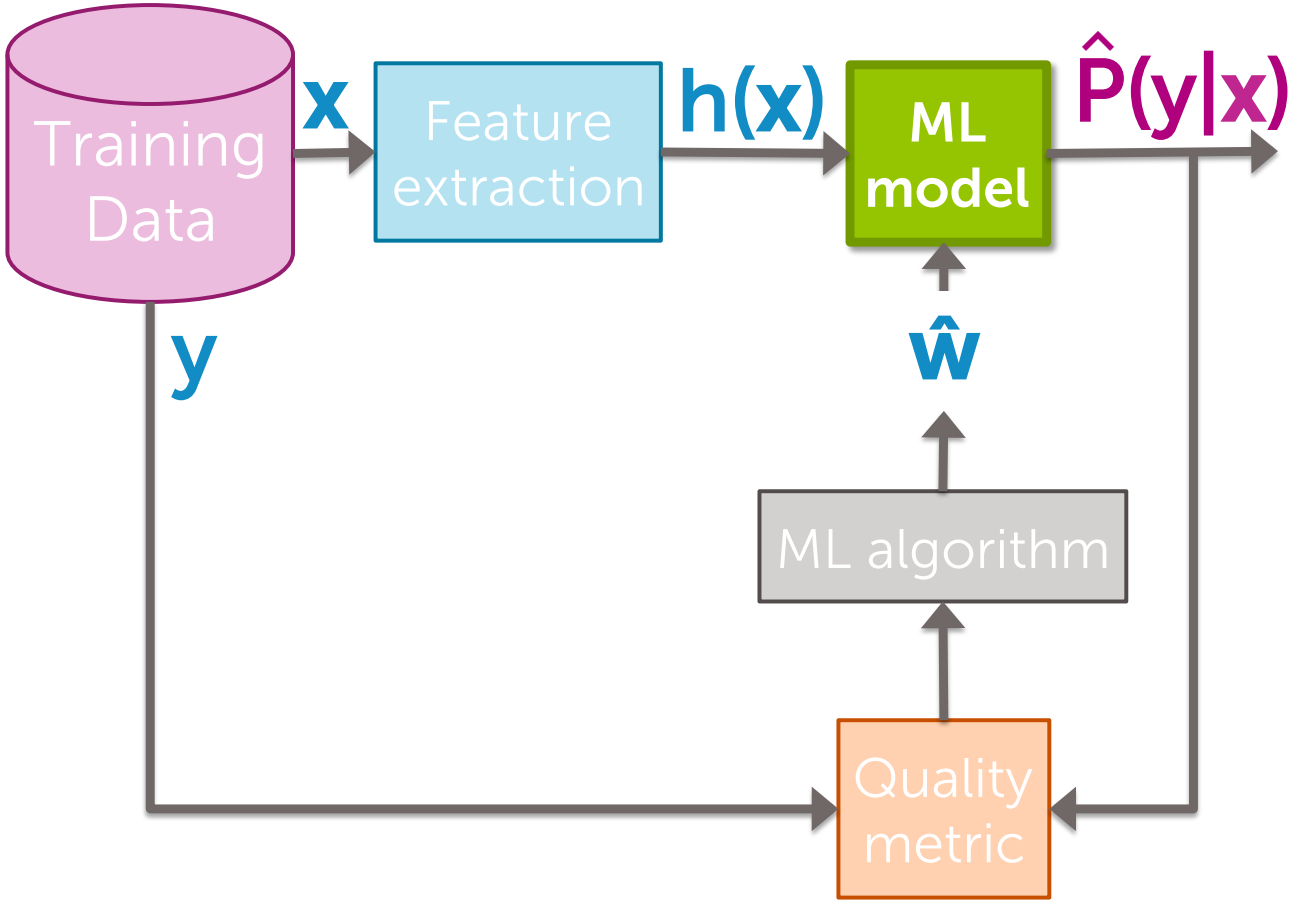
Find best model $\hat{P}$ by finding best $\hat{\mathbf{w}}$

Useful for predicting $\hat{y}$



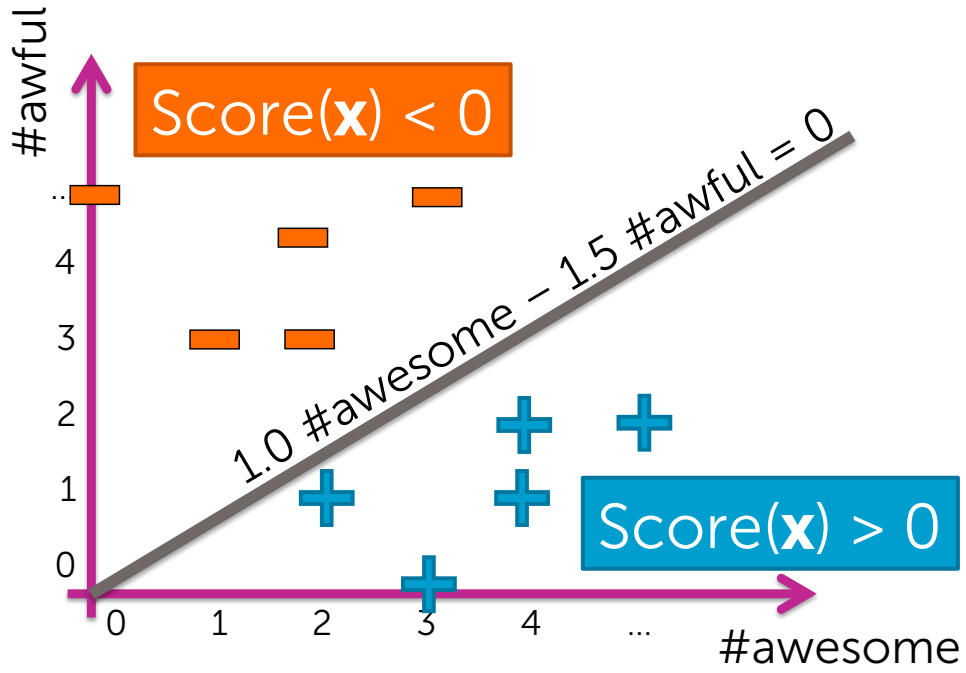
- Estimating $\hat{P}(y|x)$ improves **interpretability**:
  - Predict ŷ = +1 **and** tell me how sure you are

Predicting class probabilities with generalized linear models



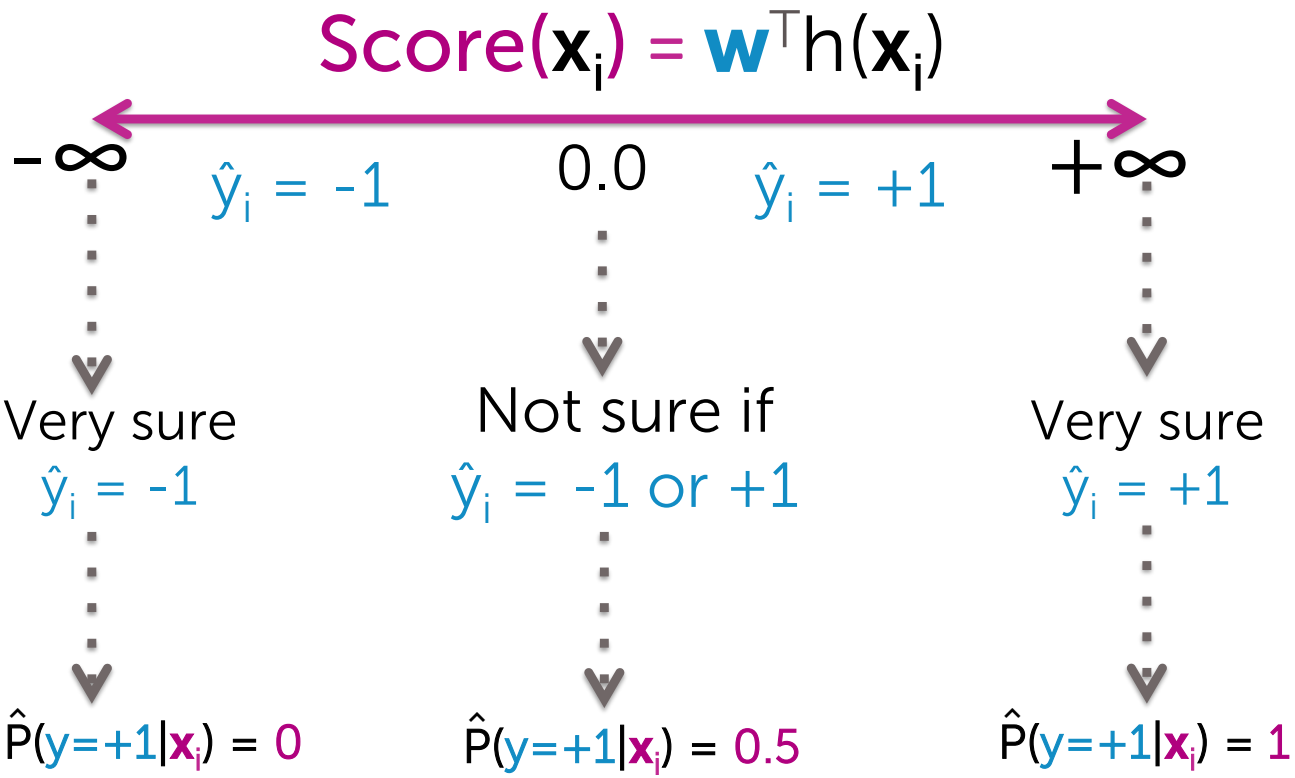
Thus far, we focused on decision boundaries

$$\text{Score}(\mathbf{x}_i) = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i)$$
$$= \mathbf{w}^T h(\mathbf{x}_i)$$

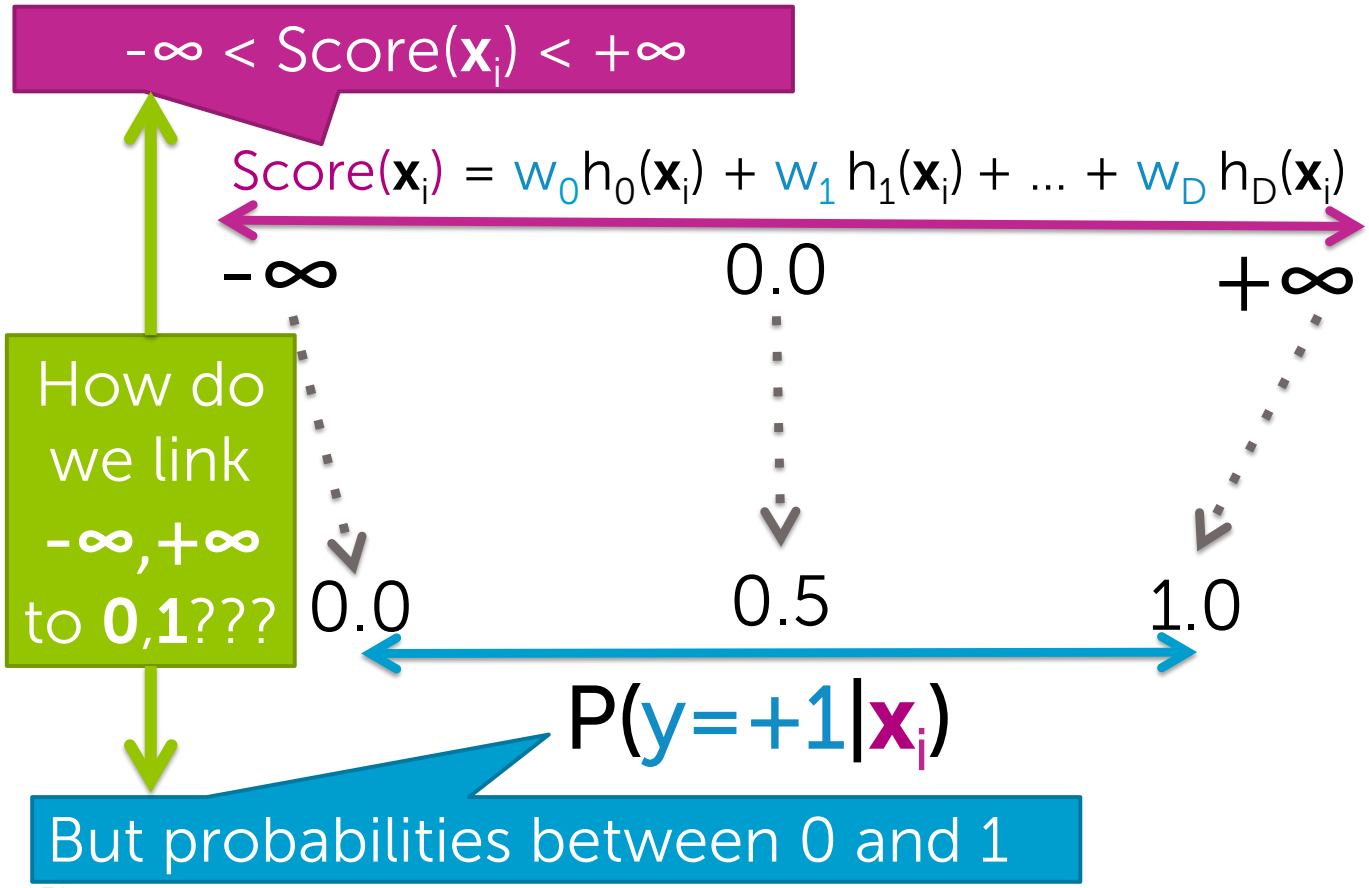


Relate $\text{Score}(\mathbf{x}_i)$ to $\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}})$?

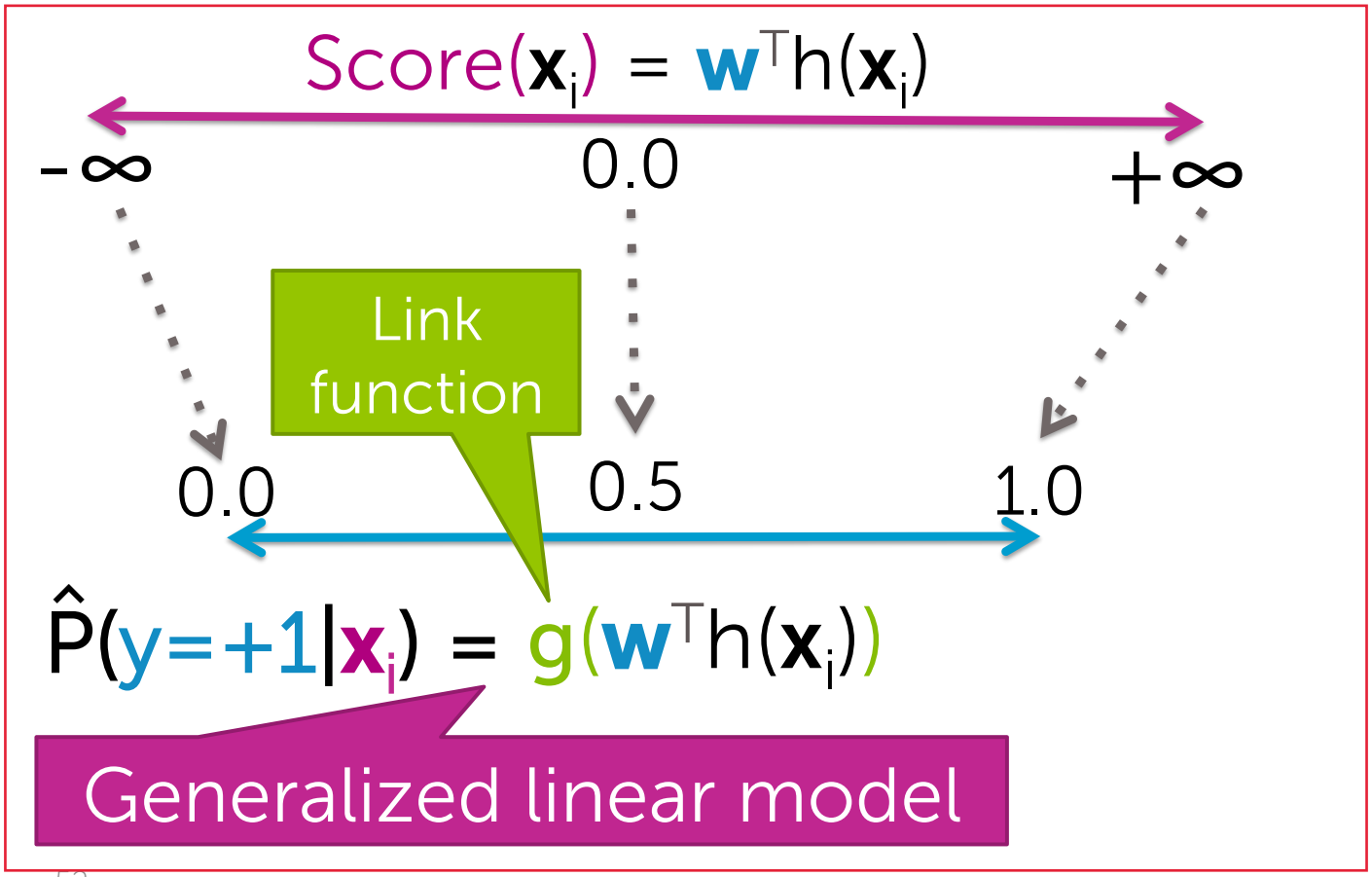
Interpreting Score($\mathbf{x}_i$)



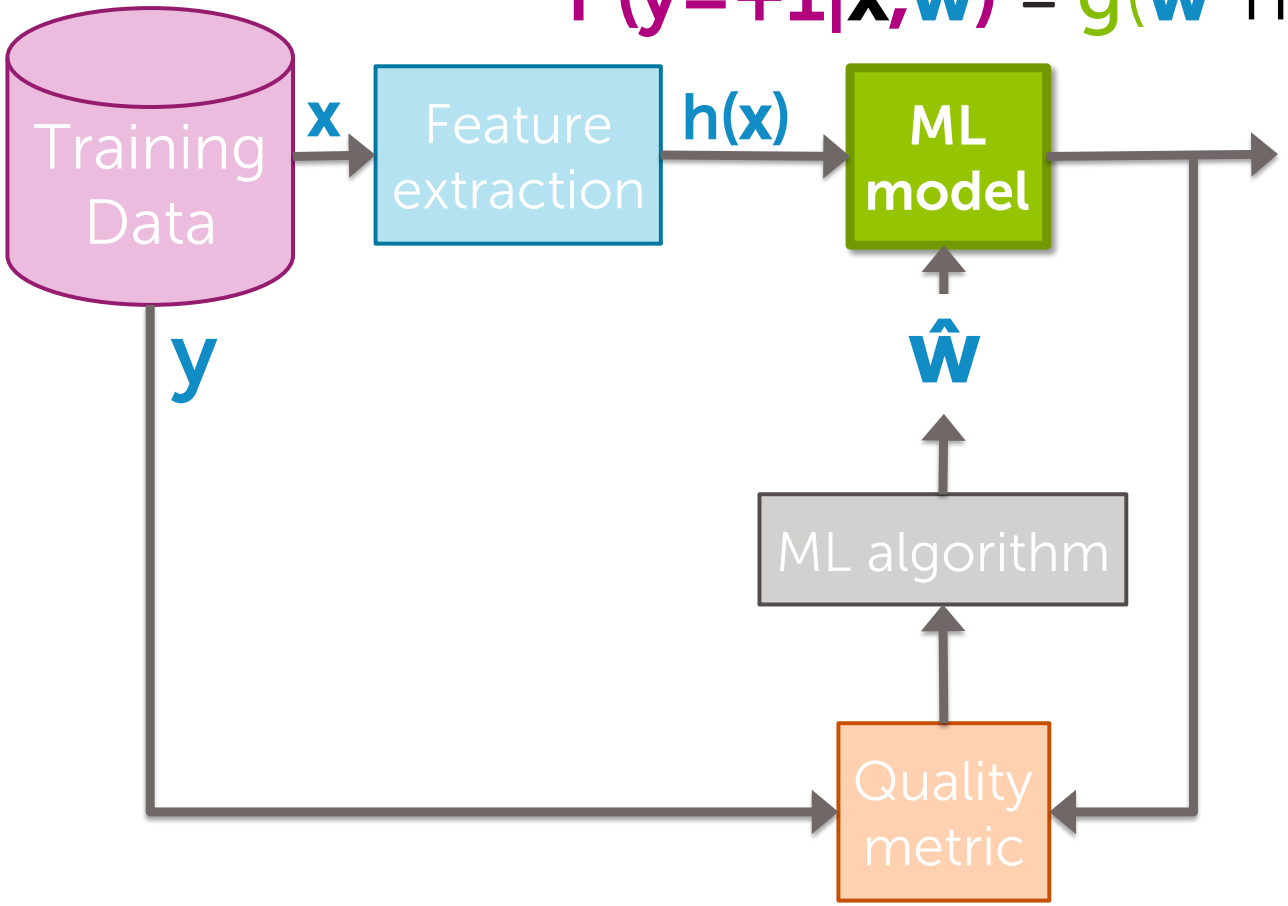
Why not just use regression to build classifier?



Link function: squeeze real line into [0,1]



$$\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}}) = g(\hat{\mathbf{w}}^T h(\mathbf{x}))$$



Logistic regression classifier:

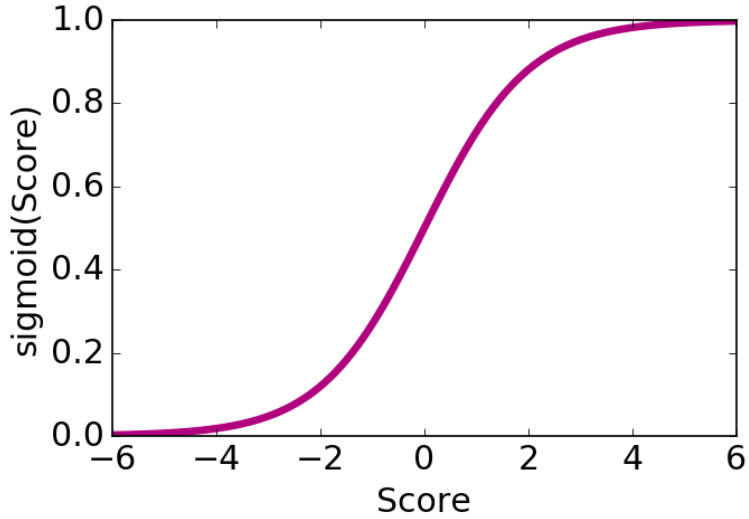
linear score with

logistic link function

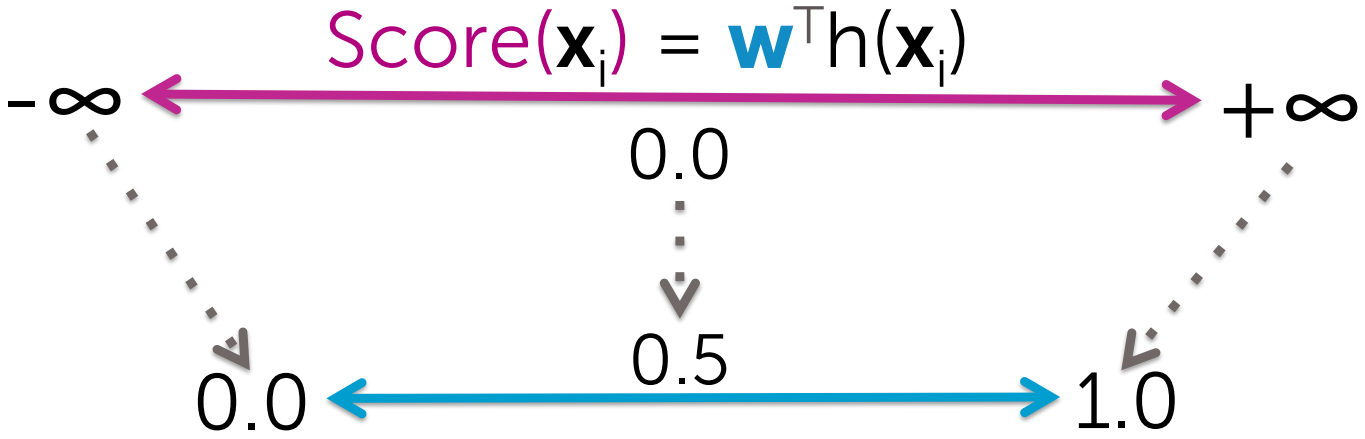
Logistic function (sigmoid, logit)

$$sigmoid(\text{Score}) = \frac{1}{1 + e^{-\text{Score}}}$$

Score	$-\infty$	-2	0.0	+2	$+\infty$
sigmoid(Score)					



Logistic regression model

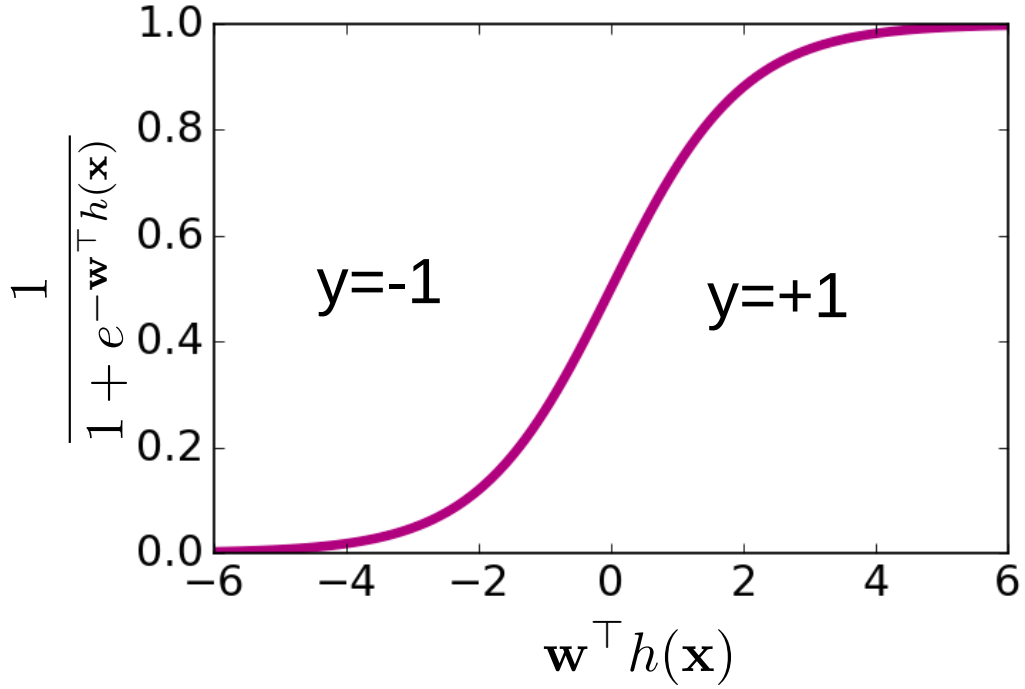


$P(y=+1|\mathbf{x}_i,\mathbf{w}) = \text{sigmoid}(\text{Score}(\mathbf{x}_i))$

$= 1/\ (1+\exp(-w^t h(x)))$

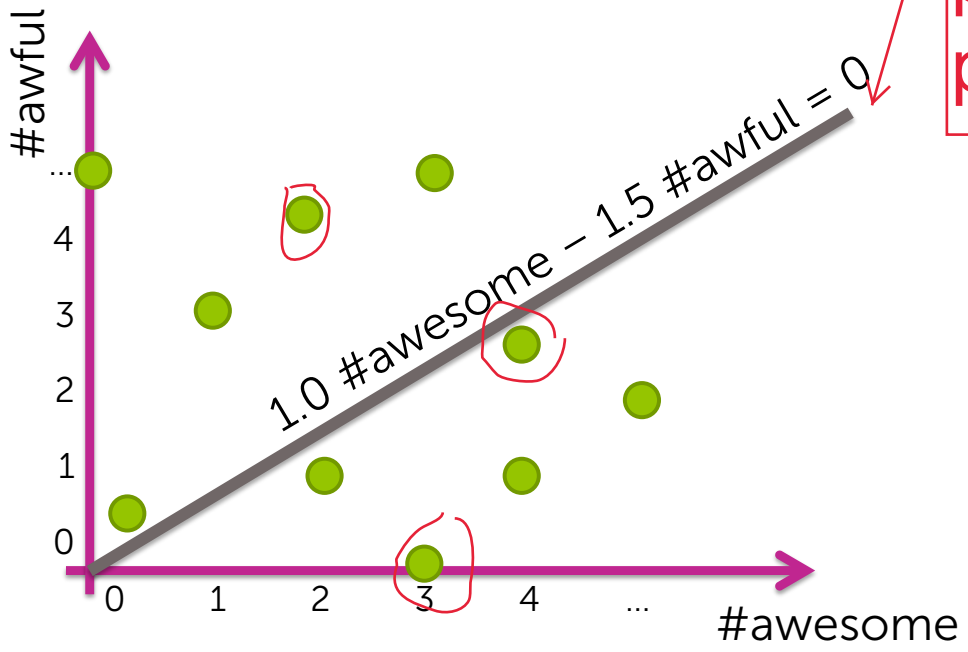
Understanding the logistic regression model

$$P(y = +1 \mid \mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^\top h(\mathbf{x})}}$$

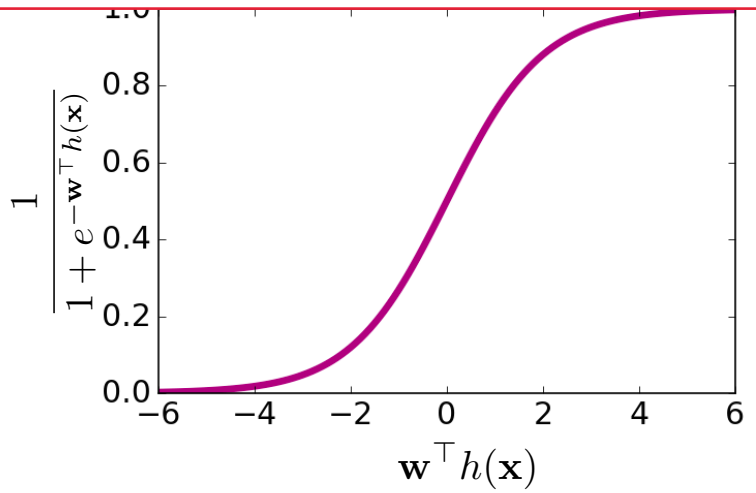


Score($\mathbf{x}_i$)	P(y=+1|$\mathbf{x}_i$,$\mathbf{w}$)

Logistic regression ➔ Linear decision boundary

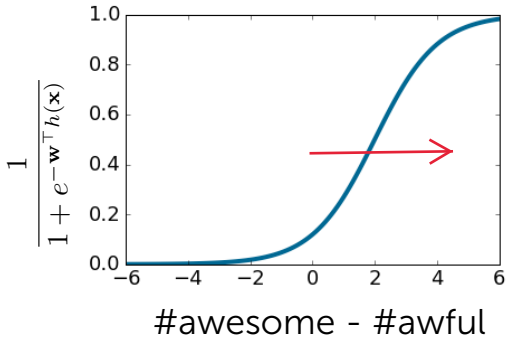


points closed to boundary, p closed to 0.5

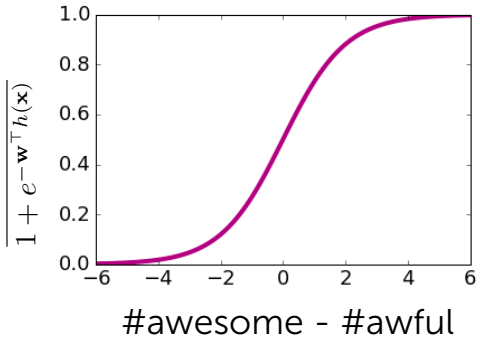


Effect of coefficients on logistic regression model

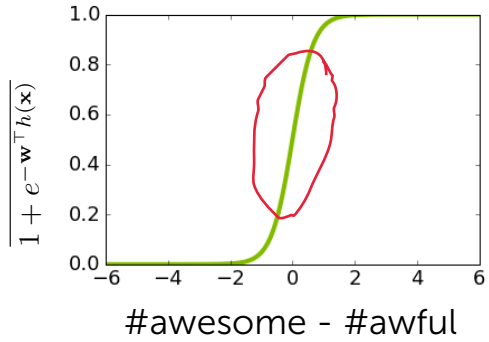
w_0	-2
$w_{\#awesome}$	+1
$w_{\#awful}$	-1



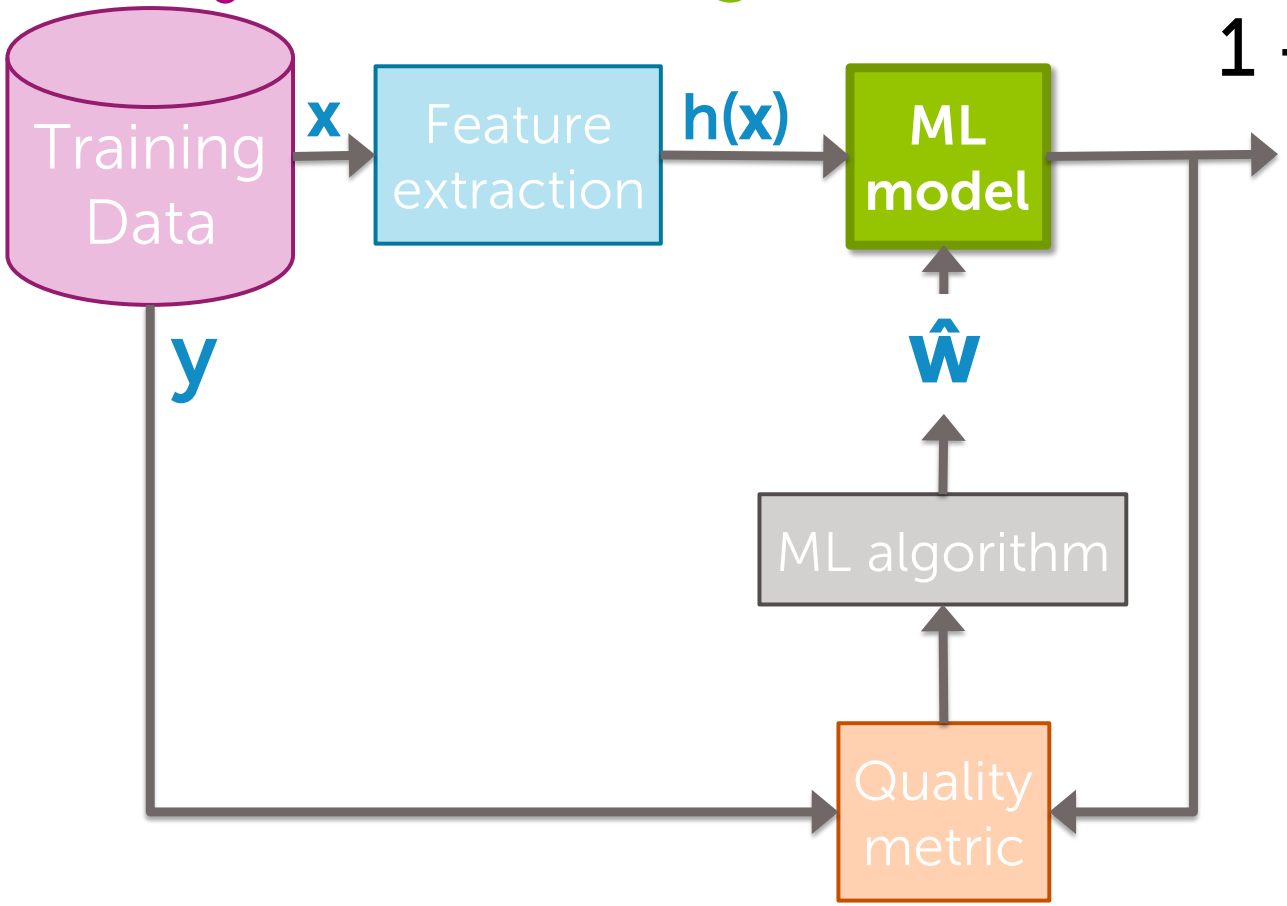
w_0	0
$w_{\#awesome}$	+1
$w_{\#awful}$	-1



w_0	0
$w_{\#awesome}$	+3
$w_{\#awful}$	-3



$$\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}}) = \text{sigmoid}(\hat{\mathbf{w}}^T h(\mathbf{x})) = \frac{1}{1 + e^{-\hat{\mathbf{w}}^T h(\mathbf{x})}}$$

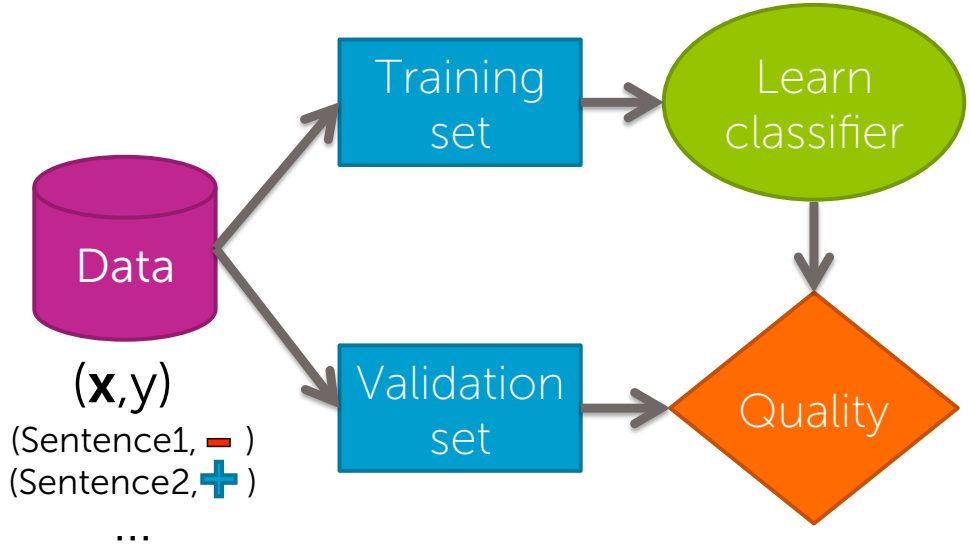


Overview of learning logistic regression model

Training a classifier = Learning the coefficients

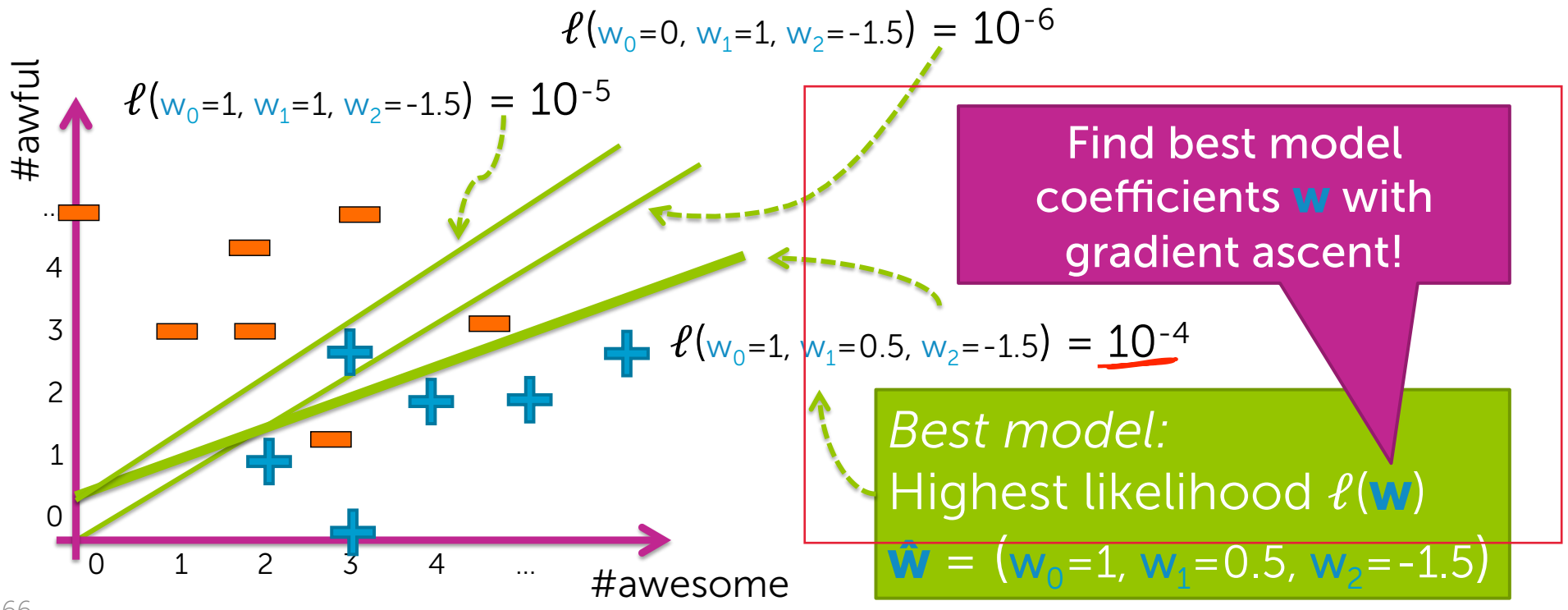
Word	Coefficient	Value
	$\hat{w}_0$	-2.0
good	$\hat{w}_1$	1.0
awesome	$\hat{w}_2$	1.7
bad	$\hat{w}_3$	-1.0
awful	$\hat{w}_4$	-3.3
…	…	…

$$\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}}) = \frac{1}{1 + e^{-\hat{\mathbf{w}}^T h(\mathbf{x})}}$$



Find "best" classifier =
Maximize quality metric over all possible w_0, w_1, w_2

Likelihood $\ell(\mathbf{w})$



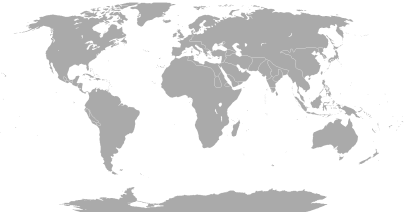
Encoding categorical inputs

Categorical inputs

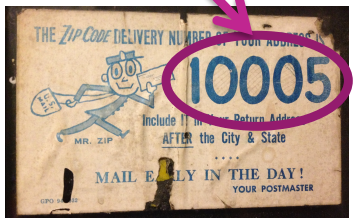
- Numeric inputs:
  - #awesome, age, salary,…
  - Intuitive when multiplied by coefficient
    - e.g., 1.5 #awesome
- Categorical inputs:

Numeric value, but should be interpreted as category
(98195 not about 9x larger than 10005)

Gender
(Male, Female,…)



Country of birth
(Argentina, Brazil, USA,…)

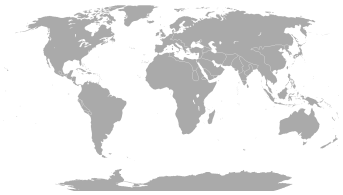


Zipcode
(10005, 98195,…)

How do we multiply category by coefficient???
Must convert categorical inputs into numeric features

Encoding categories as numeric features

$\mathbf{x}$ =



Country of birth
(Argentina, Brazil, USA,...)

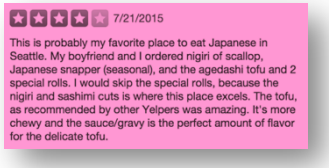
196 categories

1-hot encoding

x	$h_1(x)$	$h_2(x)$	...	$h_{195}(x)$	$h_{196}(x)$
	Argentina	Brazil	...	...	Zimbabwe
Brazil	0	1	0	0	0
Zimbabwe	0	0	0	0	1

196 features

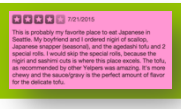
$\mathbf{x}$ =



Restaurant review
(Text data)

10,000 words in vocabulary

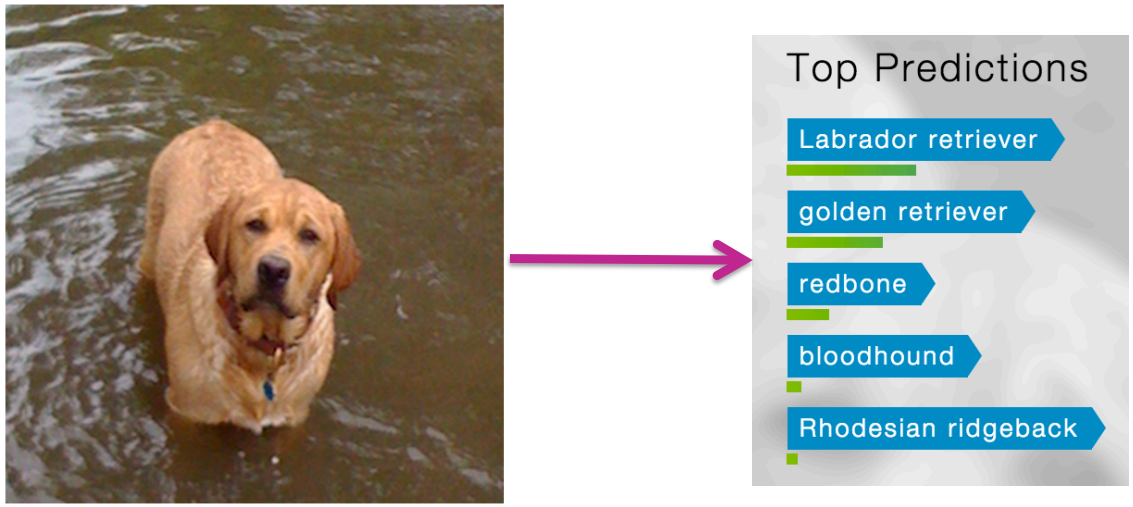
Bag of words

x	$h_1(x)$	$h_2(x)$	...	$h_{9999}(x)$	$h_{10000}(x)$
	#awesome	#awful	..	..	#sushi
	2	0	0	0	3

10,000 features

Multiclass classification using 1 versus all

Multiclass classification



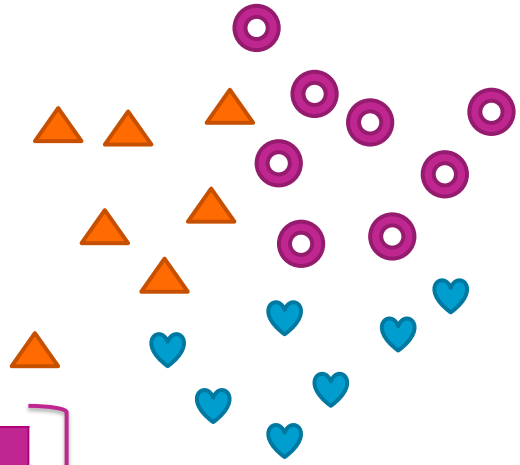
Input: **x**
Image pixels

Output: **y**
Object in image

Multiclass classification formulation

- C possible classes:
  - y can be 1, 2,…, C
- N datapoints:

Data point	x[1]	x[2]	y
$\mathbf{x}_1, y_1$	2	1	▲
$\mathbf{x}_2, y_2$	0	2	♥
$\mathbf{x}_3, y_3$	3	3	◎
$\mathbf{x}_4, y_4$	4	1	◎



Learn:

$\hat{P}(y=▲|x)$

$\hat{P}(y=♥|x)$

$\hat{P}(y=◎|x)$

1 versus all:

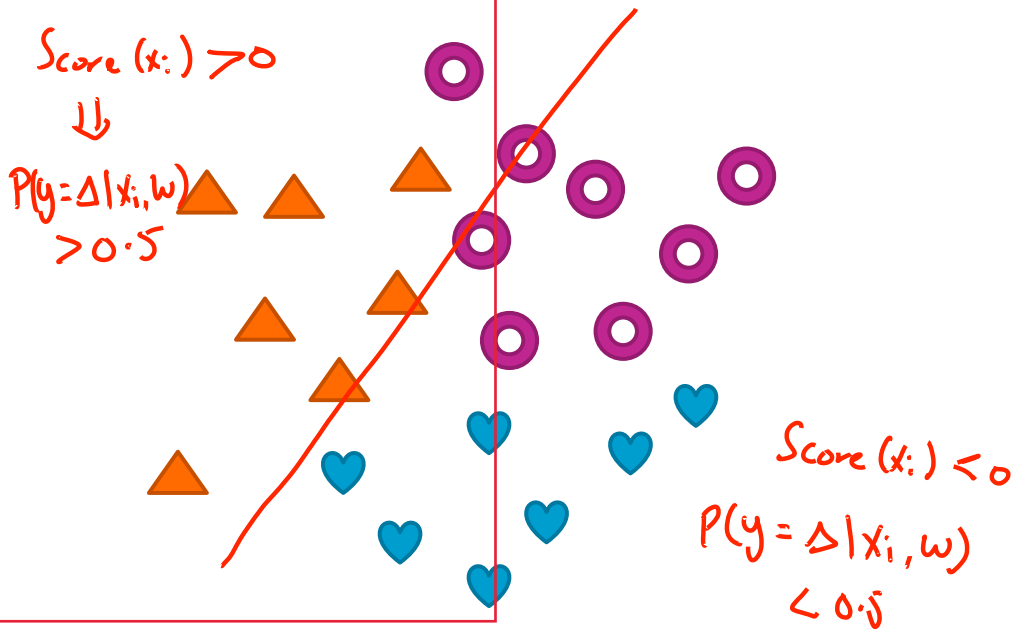
Estimate $\hat{P}(y=\blacktriangle|x)$ using 2-class model

+1 class: points with $y_i=$ ▲

-1 class: points with $y_i=$ 💙 OR ◎

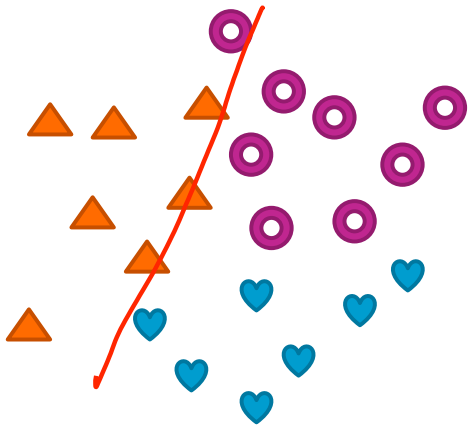
Train classifier: $\hat{P}_{\blacktriangle}(y=+1|x)$

Predict: $\hat{P}(y=\blacktriangle|x_i)= \hat{P}_{\blacktriangle}(y=+1|x_i)$

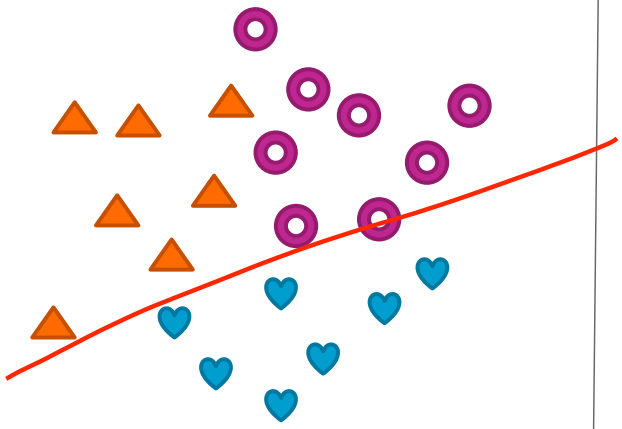


1 versus all: simple multiclass classification using C 2-class models

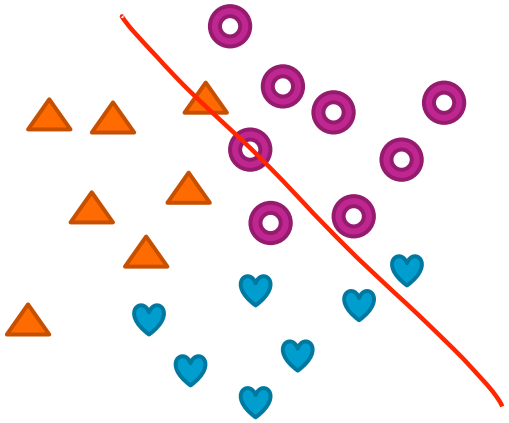
$\hat{P}(y=\blacktriangle \mid x_i) = \hat{P}_{\blacktriangle}(y=+1 \mid x_i, w_{\blacktriangle})$

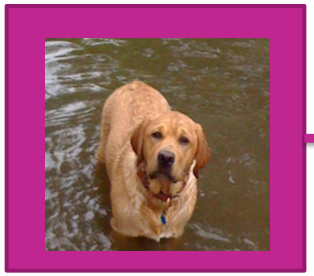


$\hat{P}(y=\heartsuit \mid x_i) = \hat{P}_{\heartsuit}(y=+1 \mid x_i, w_{\heartsuit})$



$\hat{P}(y=\bigcirc \mid x_i) = \hat{P}_{\bigcirc}(y=+1 \mid x_i, w_{\bigcirc})$





Input: $\mathbf{x}_i$

Multiclass training

$\hat{P}_c(y=+1|\mathbf{x})$ = estimate of 1 vs all model for each class

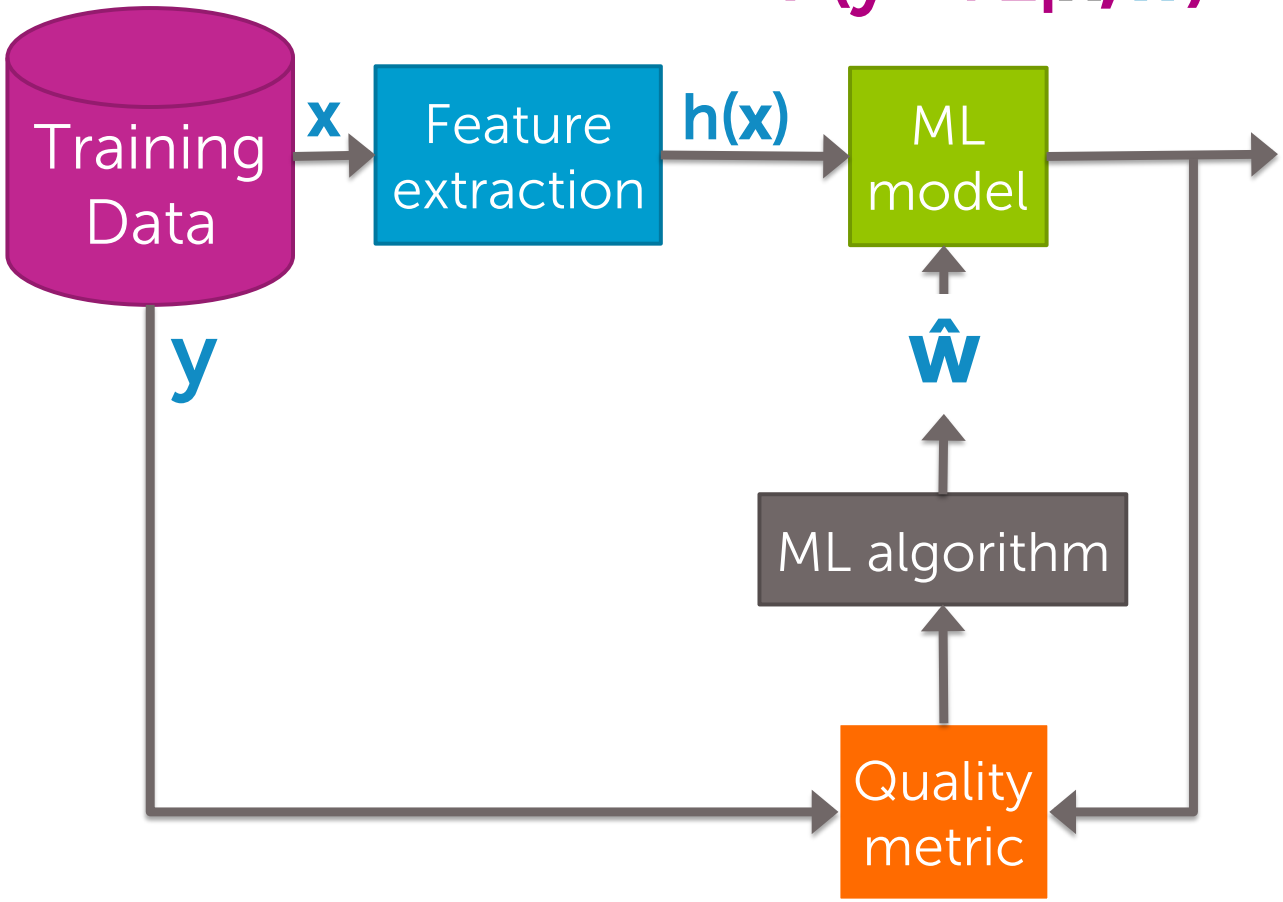
Predict most likely class

```
max_prob = 0; ŷ = 0
For c = 1,…,C:
    If P̂c(y=+1|xi) > max_prob:
        ŷ = c
        max_prob = P̂c(y=+1|xi)
```

predict using the max probability from multiclass training

Summary of logistic regression classifier

$$\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}}) = \frac{1}{1 + e^{-\hat{\mathbf{w}}^T h(\mathbf{x})}}$$



What you can do now...

- Describe decision boundaries and linear classifiers
- Use class probability to express degree of confidence in prediction
- Define a logistic regression model
- Interpret logistic regression outputs as class probabilities
- Describe impact of coefficient values on logistic regression output
- Use 1-hot encoding to represent categorical inputs
- Perform multiclass classification using the 1-versus-all approach