DataStar Machine Learning

ADAX

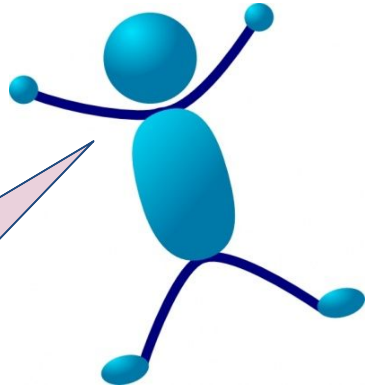
Modules

Session 3: Classification

18 Sept 2017

ADAX

A Case Study



Kuala Lumpur has many 5-star sushi restaurants

What are people saying about the food? The ambiance?

A Case Study

Positive reviews are NOT positive everytime, and maybe about something but NOT everything…

Marufuku Ramen

465 reviews

$$ · Ramen

Kui Shin Bo

550 reviews

$$ · Japanese, Sushi Bars

Ichido

55 reviews

$$$ · Seafood, Japanese, Asian Fusion

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.

A Case Study

Classifying Sentiment of Review

Easily best sushi in Seattle.

Sentence Sentiment Classifier

A Case Study

Classifying Sentiment of Review

All reviews for restaurant

★★★★☆ 7/21/2015

This is probably my favorite place to eat Japanese in Seattle. My boyfriend and I ordered nigiri of scallop, Japanese snapper (seasonal), and the agedashi tofu and 2 special rolls. I would skip the special rolls, because the nigiri and sashimi cuts is where this place excels. The tofu, as recommended by other Yelpers was amazing. It's more chewy and the sauce/gravy is the perfect amount of flavor for the delicate tofu.

★★★★★ 6/11/2015

Dining here at the sushi bar made me feel like sitting front row to an amazing performance. We didn't have resos, banged down to the ID after work, got here breathlessly at 5:10pm, and got the last two seats in the place.

★★★☆☆ 6/9/2015

I came here having high expectations due to the reviews of this place, but i was bit disappointed.
The restaurant is small so do make reservations when you come here. Dishes cost from $4-26 each and dishes are small.

Break all reviews into sentences

The seaweed salad was just OK, vegetable salad was just ordinary.

I like the interior decoration and the blackboard menu on the wall.

All the sushi was delicious.

My wife tried their ramen and it was pretty forgettable.

The sushi was amazing, and the rice is just outstanding.

The service is somewhat hectic.

Easily best sushi in Seattle.

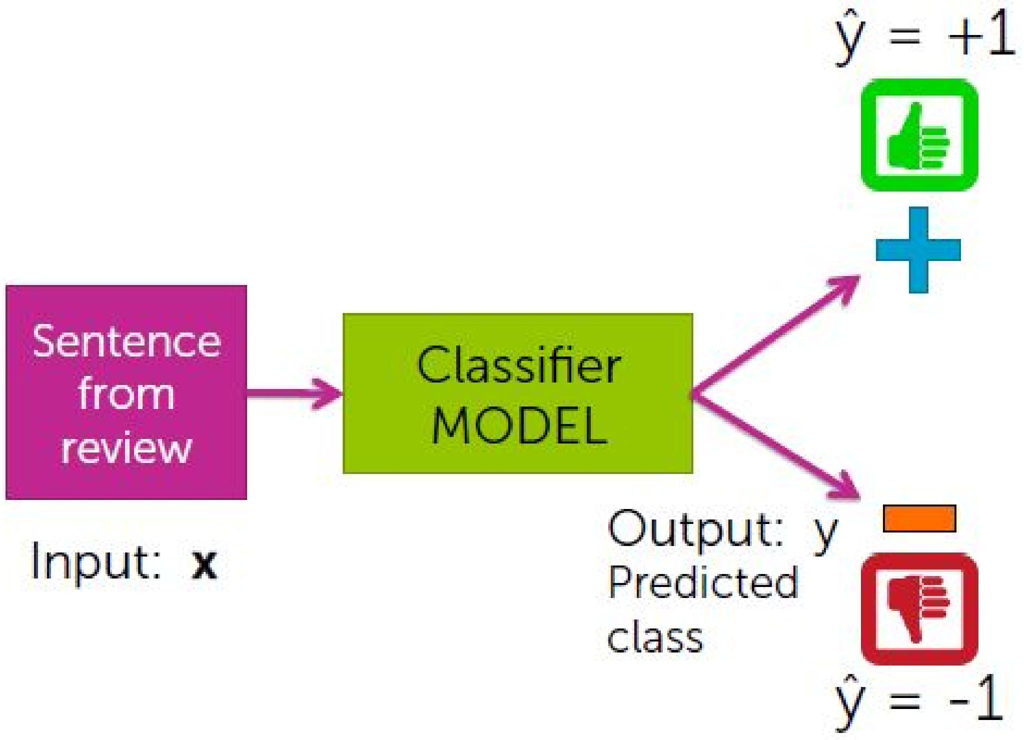
Sentence Sentiment Classifier

Good

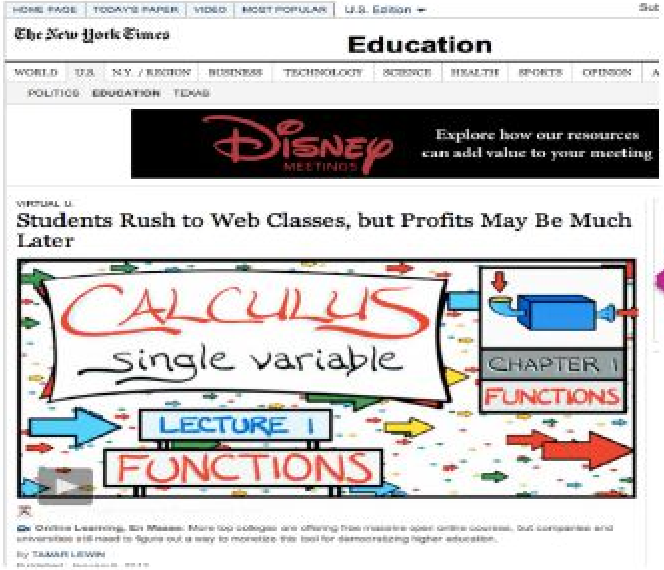
Bad

What is Classification?

What is Classification?



Classification can have more than 2 categories



HOME PAGE | TODAY'S PAPER | VIDEO | MOST POPULAR | U.S. Edition

The New York Times

Education

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

POLITICS EDUCATION TEXAS

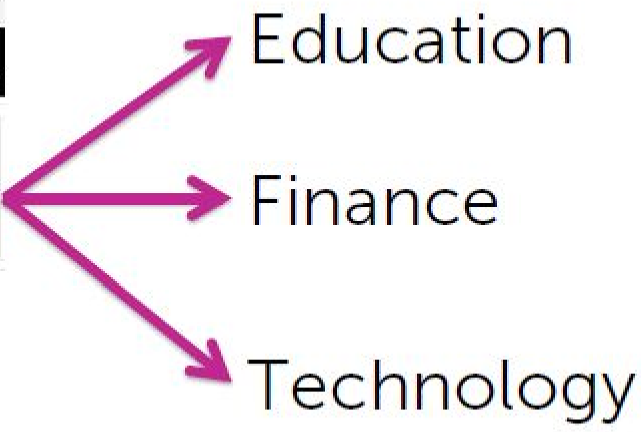
Disney MEETINGS Explore how our resources can add value to your meeting

VIRTUAL U.

Students Rush to Web Classes, but Profits May Be Much Later

Online Learning, En Masse: More top colleges are offering free massive open online courses, but companies and universities still need to figure out a way to monetize this tool for democratizing higher education.

By TAMAR LEWIN



Education

Finance

Technology

Input: x
Webpage

Output: y

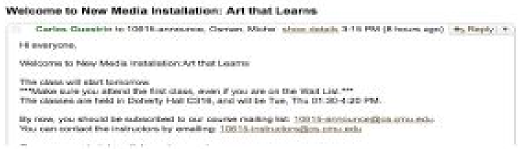
Spam Filtering

Osman Khan to Carlos show details Jan 7 (6 days ago) Reply

sounds good
+ok

Carlos Guestrin wrote:
Let's try to chat on Friday a little to coordinate and more on Sunday in person?

Carlos



Welcome to New Media Installation: Art that Learns

Carlos Guestrin to 10615-announce, Osman, Michel show details 3:15 PM (8 hours ago) Reply

Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only $5.95 for shipping mfw rlk Spam | X

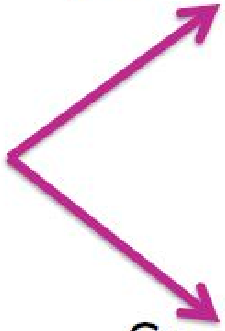
Jaquelyn Halley to nherlein, bcc: thehorney, bcc: ang show details 9:52 PM (1 hour ago) Reply

=== Natural WeightL0SS Solution ===

Vital Acai is a natural WeightL0SS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

- Rapid WeightL0SS
- Increased metabolism - BurnFat & calories easily!
- Better Mood and Attitude
- More Self Confidence
- Cleanse and Detoxify Your Body
- Much More Energy



Not spam

Spam

Input: x

Output: y

Text of email, sender, IP,...

Image Classification

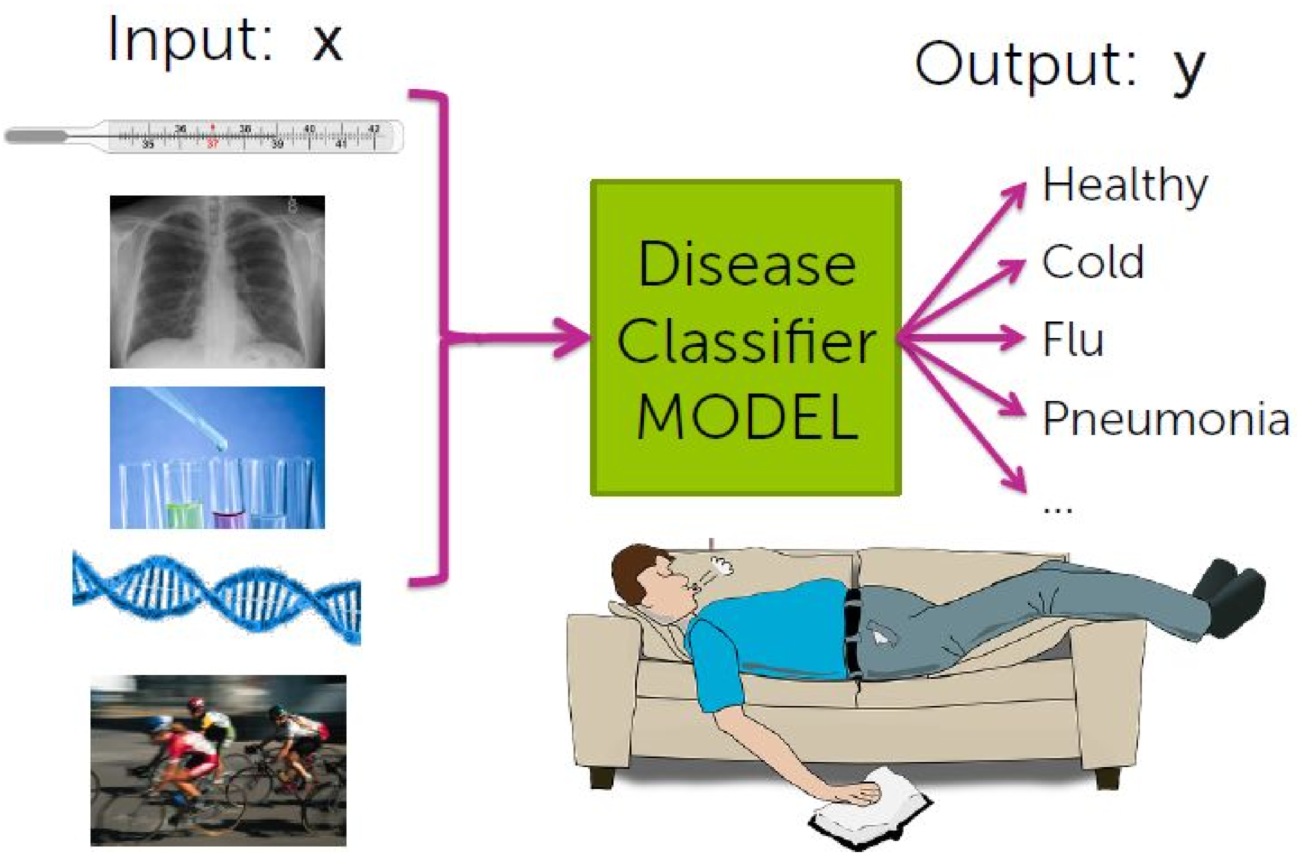
Top Predictions

- Labrador retriever
- golden retriever
- redbone
- bloodhound
- Rhodesian ridgeback

Input: x
Image pixels

Output: y
Predicted object

Medical Diagnosis



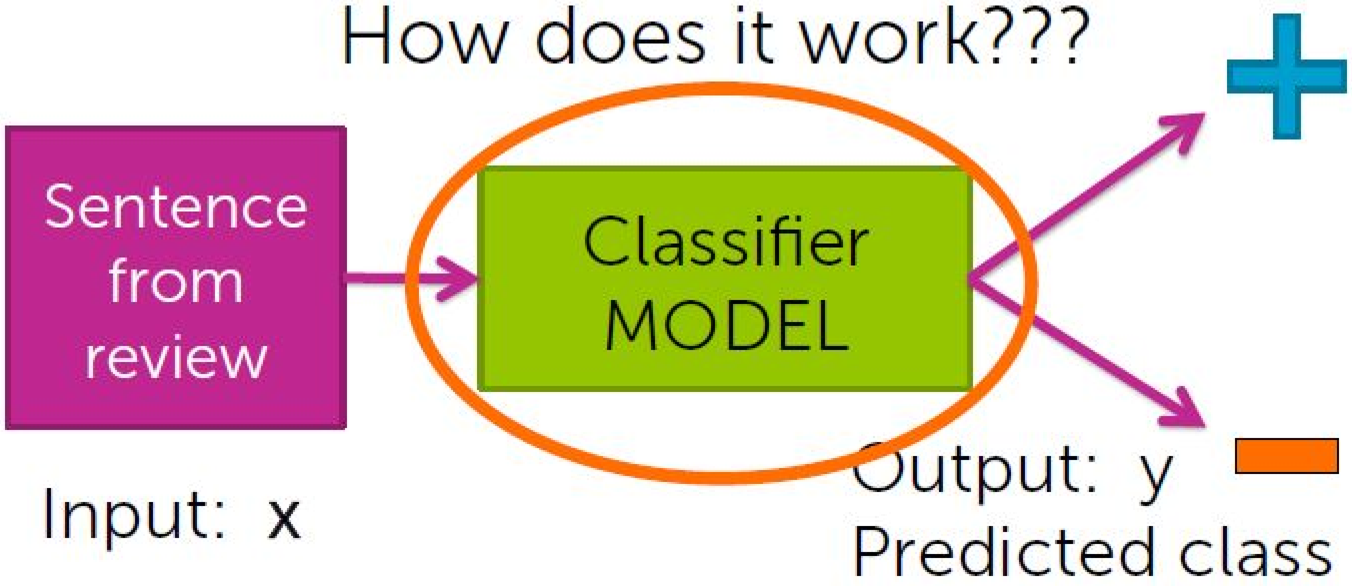
Input: x

Output: y

Disease Classifier MODEL

- Healthy
- Cold
- Flu
- Pneumonia
- …

Intuition of Classifiers



How does it work???

Sentence from review

Classifier MODEL

+

−

Input: x

Output: y

Predicted class

Intuition of Classifiers

List of positive words	List of negative words
great, awesome, good, amazing,…	bad, terrible, disgusting, sucks,…

Sentence from review

Input: x

Simple threshold classifier

Count positive & negative words in sentence

If *number of positive words > number of negative words*:

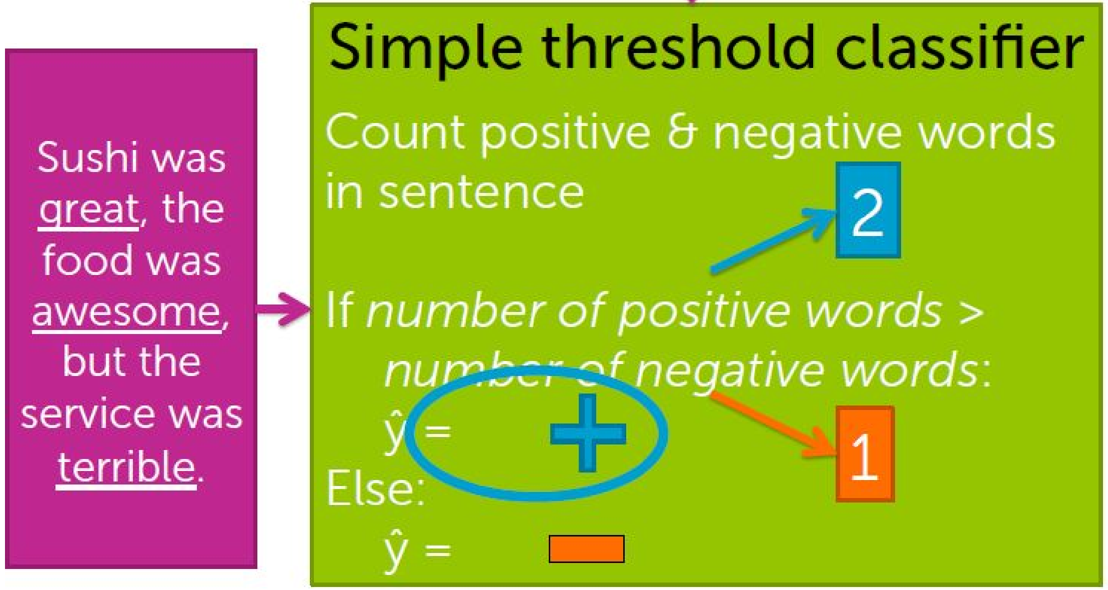
ŷ = +

Else:

ŷ = −

Intuition of Classifiers

List of positive words	List of negative words
great, awesome, good, amazing,…	bad, terrible, disgusting, sucks,…



Sushi was <u>great</u>, the food was <u>awesome</u>, but the service was <u>terrible</u>.

Simple threshold classifier

Count positive & negative words in sentence

2

If *number of positive words > number of negative words*:

ŷ = +

1

Else:

ŷ = −

Problem with threshold classifiers

- How do we get list of positive/negative words?
- Words have different degrees of sentiment:
  - Great > good
  - How do we weigh different words?

Addressed by learning a classifier

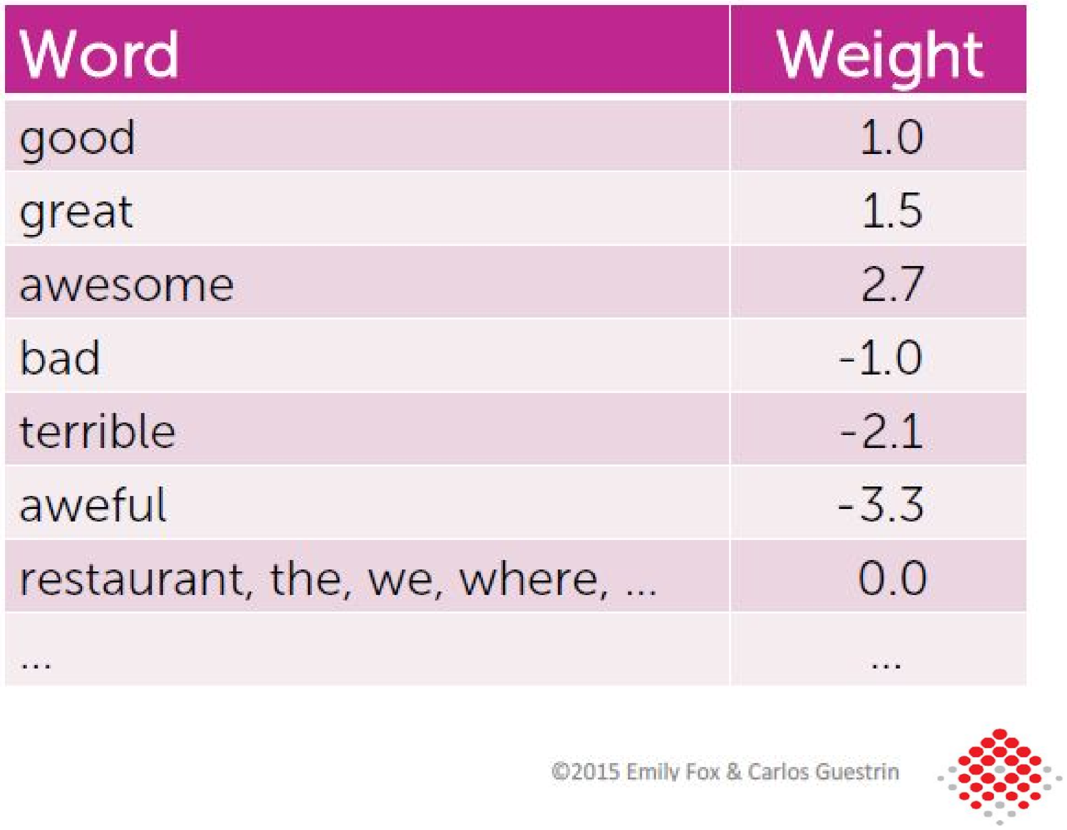
- Single words are not enough:
  - *Good* ➔ Positive
  - *Not good* ➔ Negative

Addressed by more elaborate features

Linear Classifiers

Simple (linear) classifier

Will use training data to learn a weight for each word



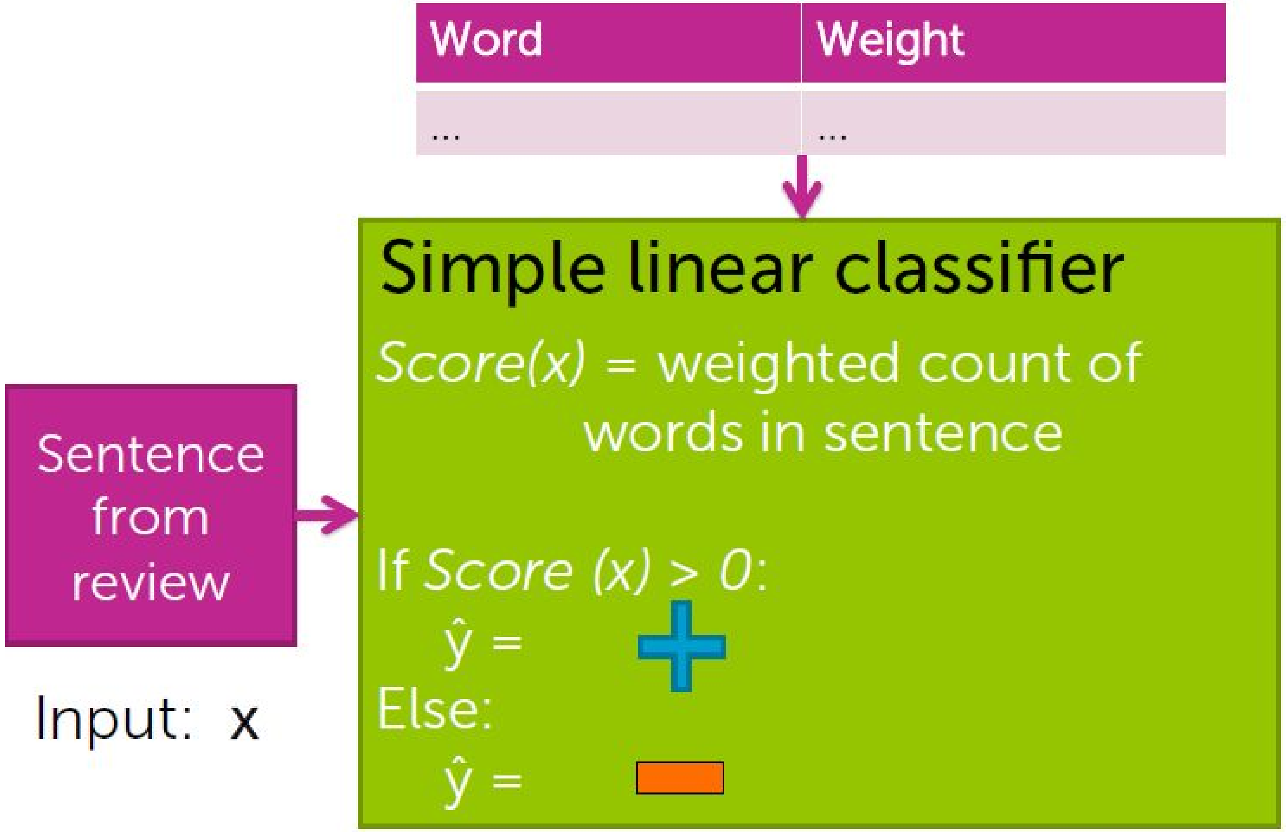
Word	Weight
good	1.0
great	1.5
awesome	2.7
bad	-1.0
terrible	-2.1
aweful	-3.3
restaurant, the, we, where, …	0.0
…	…

Input x:
Sushi was great,
the food was awesome,
but the service was terrible.

Score(x) = ?

Linear classifier:
Output is weighted sum of input

Scoring a sentence



Word	Weight
…	…

Simple linear classifier

Score(x) = weighted count of words in sentence

If *Score (x) > 0*:
  ŷ = +
Else:
  ŷ = −

Sentence from review

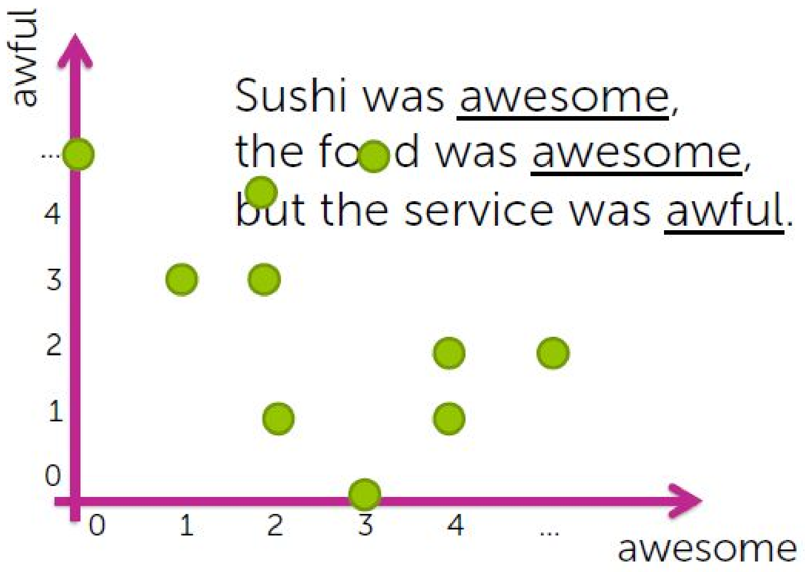
Input: x

Decision boundary

Suppose only 2 words had non-zero weights…

Word	Weight
awesome	1.0
awful	-1.5

Score(x) = 1.0 #awesome – 1.5 #awful

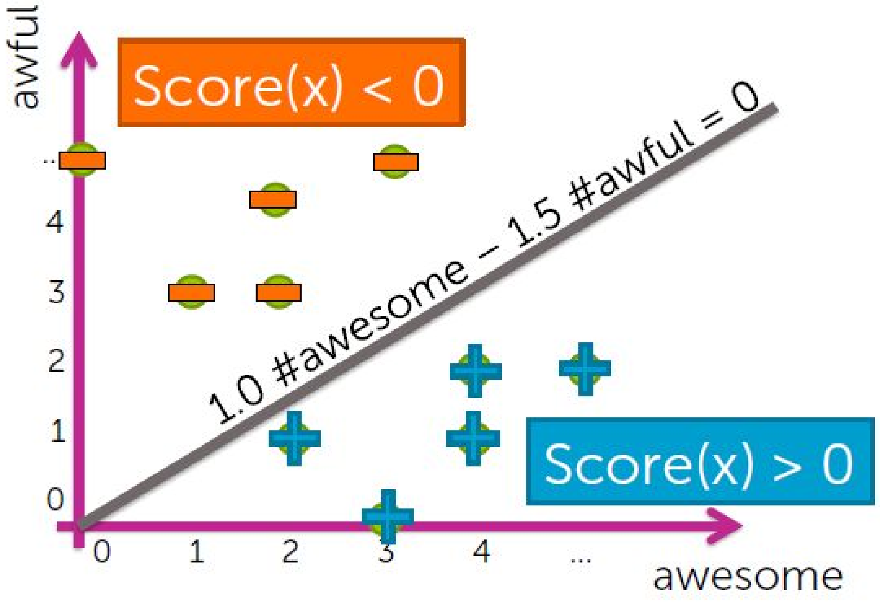


Sushi was awesome,
the food was awesome,
but the service was awful.

Decision boundary

Word	Weight
awesome	1.0
awful	-1.5

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

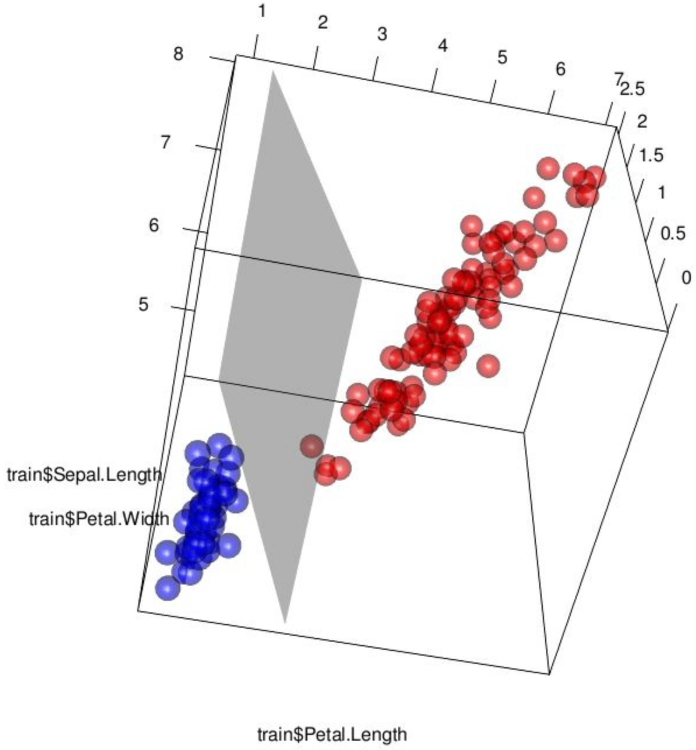


A boundary is to...separate

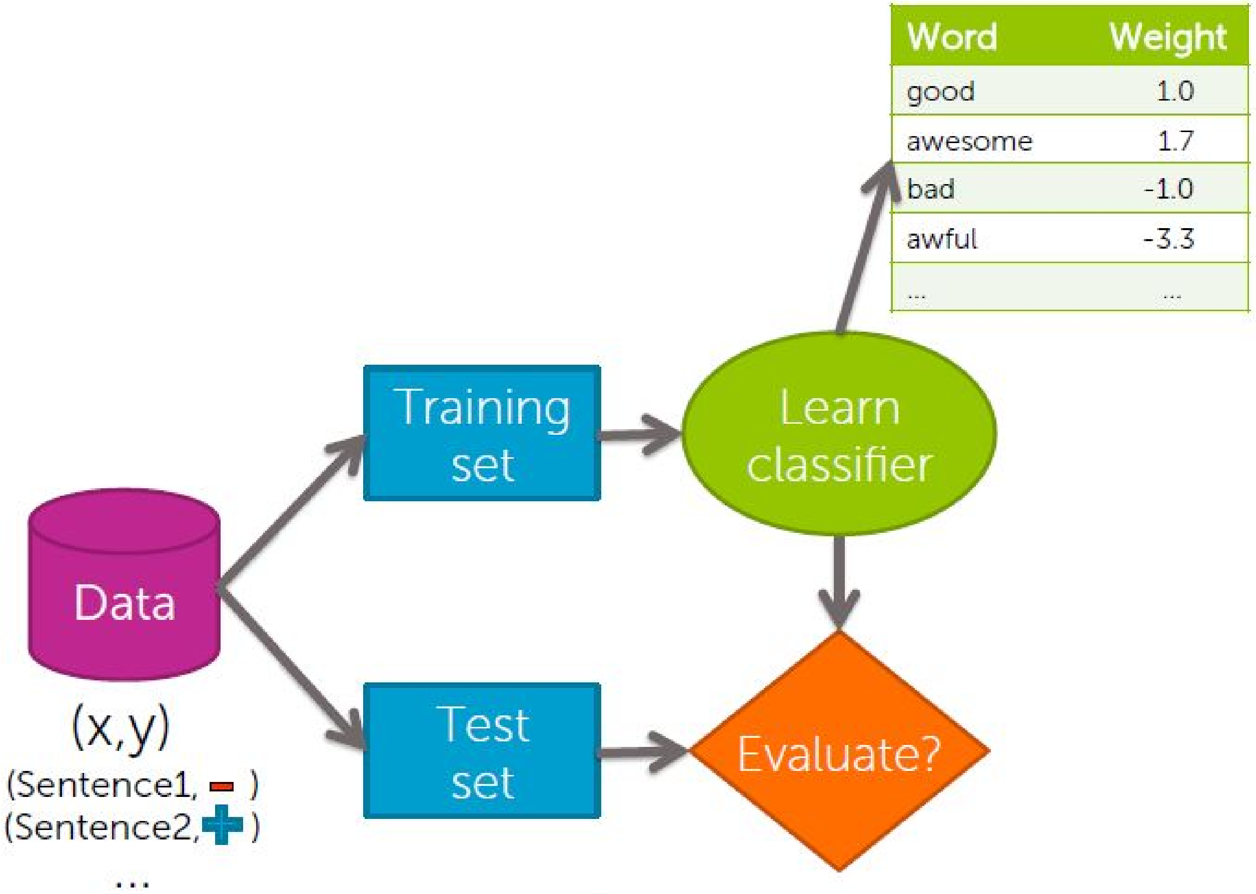
For linear classifiers:

- When 2 weights are non-zero ⇒ **line**
- When 3 weights are non-zero ⇒ **plane**
- When many weights are non-zero ⇒ **hyperplane**

For more general classifiers
⇒ more complicated shapes



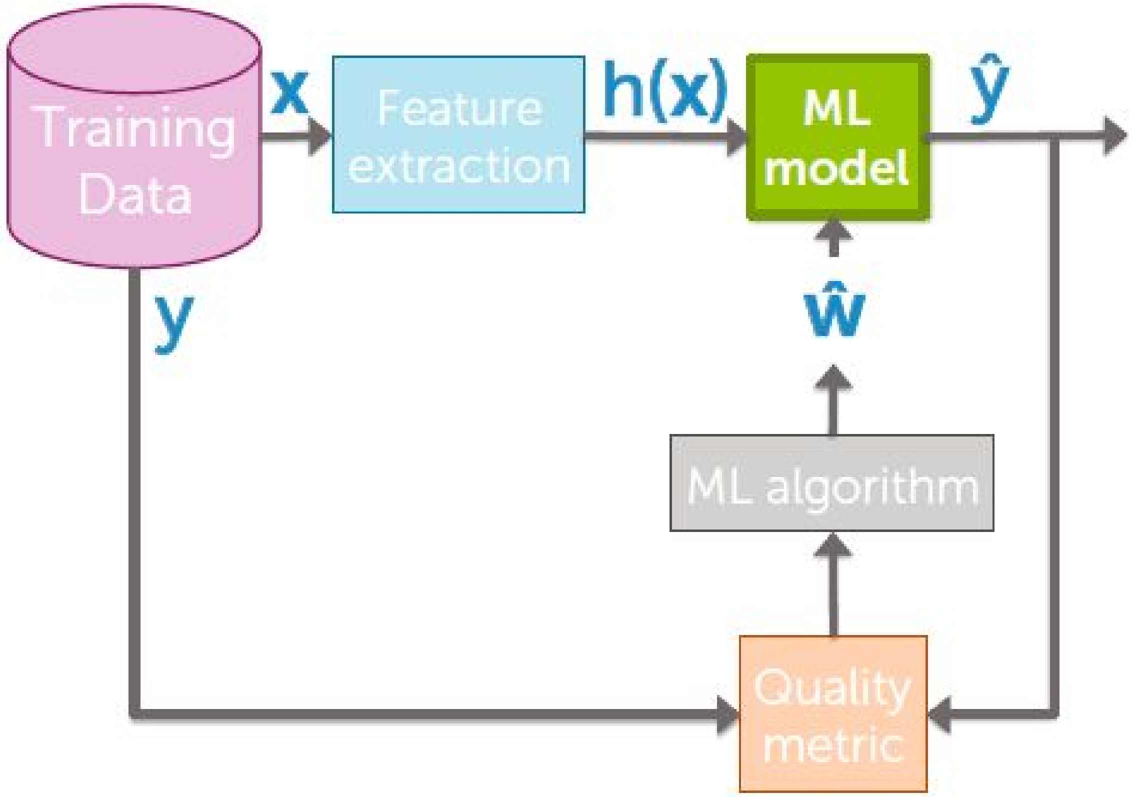
Training a classifier → Learning weights



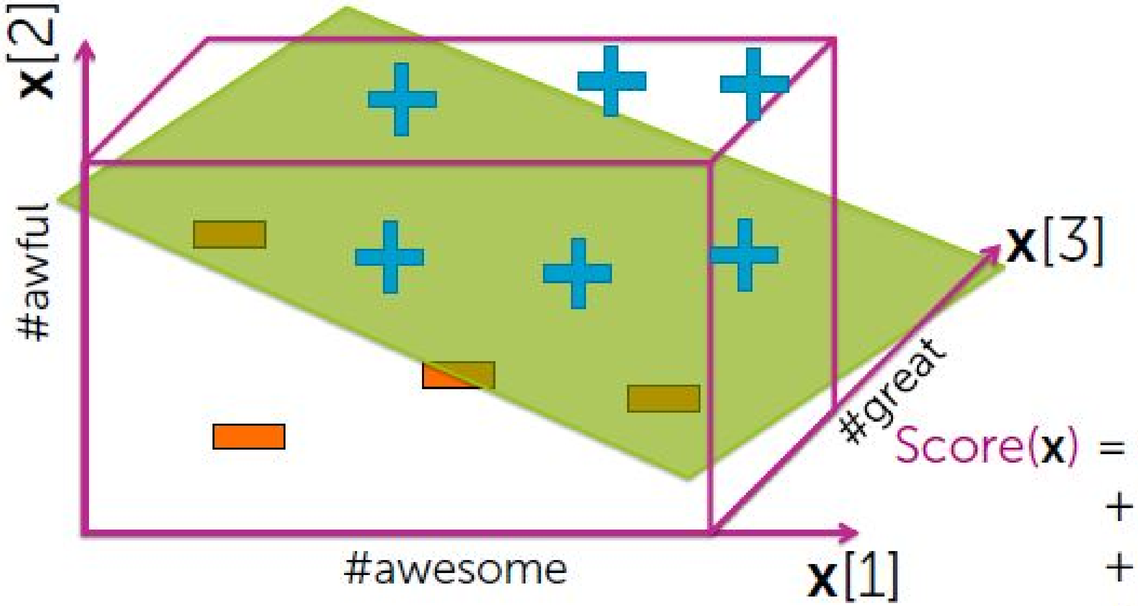
(x,y)
(Sentence1, −)
(Sentence2, +)
…

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

Machine learning system

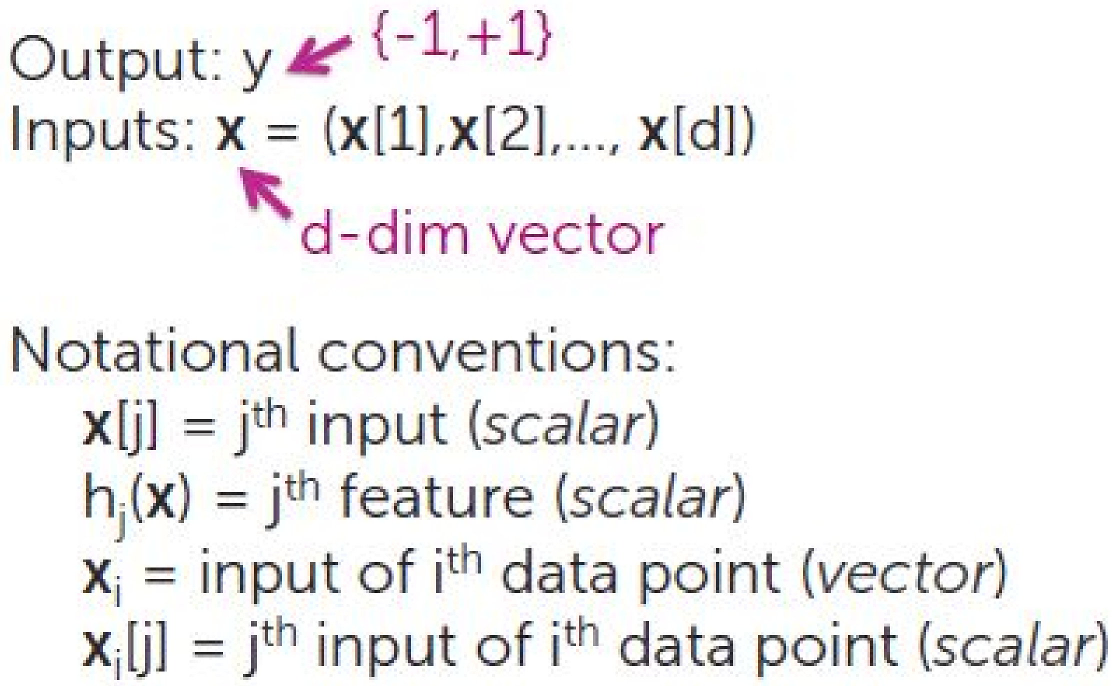


Coefficients of a classifier



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

General notation



Output: y $\{-1,+1\}$

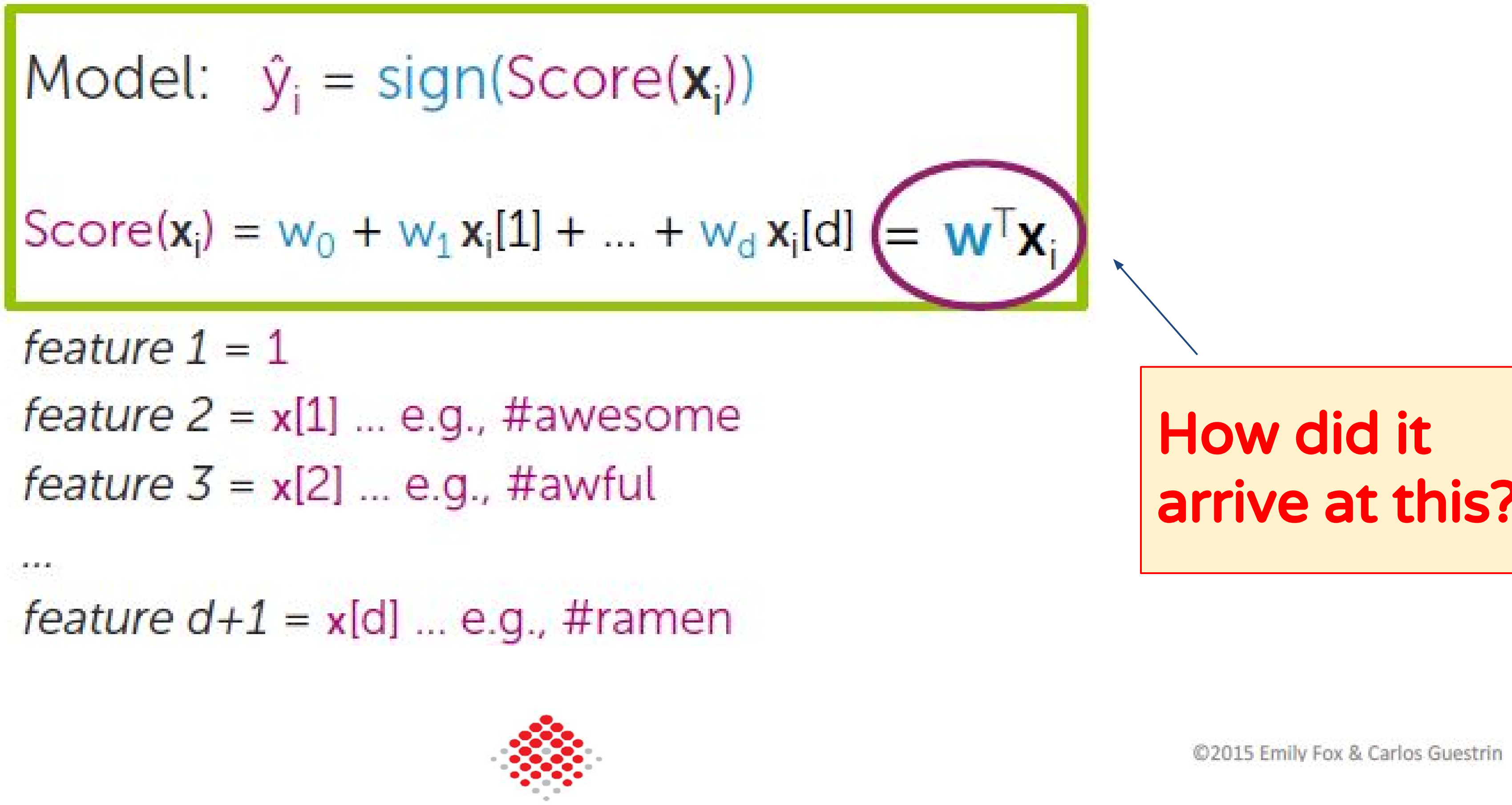
Inputs: $\mathbf{x} = (\mathbf{x}[1],\mathbf{x}[2],\ldots, \mathbf{x}[d])$

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

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

feature 1 = 1

feature 2 = x[1] … e.g., #awesome

feature 3 = x[2] … e.g., #awful

…

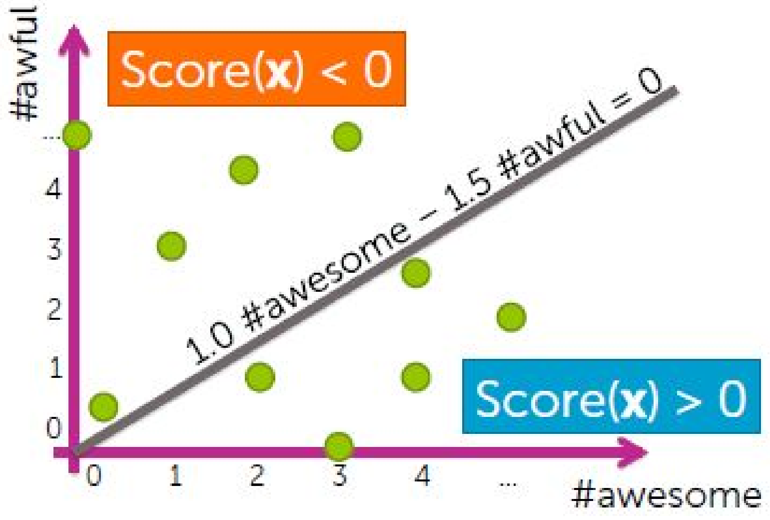
feature d+1 = x[d] … e.g., #ramen

How did it arrive at this?

Decision boundary: Effect of changing coefficients

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

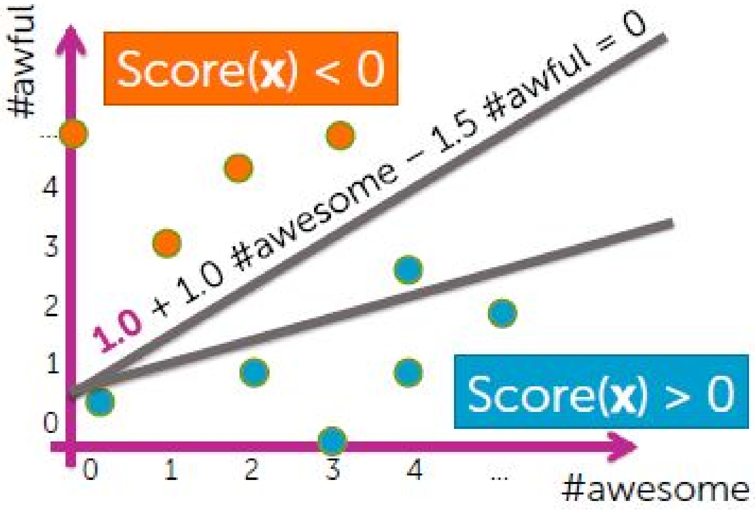
➡ Score(x) = 1.0 #awesome – 1.5 #awful



Decision boundary: Effect of changing coefficients

Input	Coefficient	Value
	w_0	1.0
#awesome	w_1	1.0
#awful	w_2	-3.0

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



More generic features… D-dimensional hyperplane

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

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

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

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

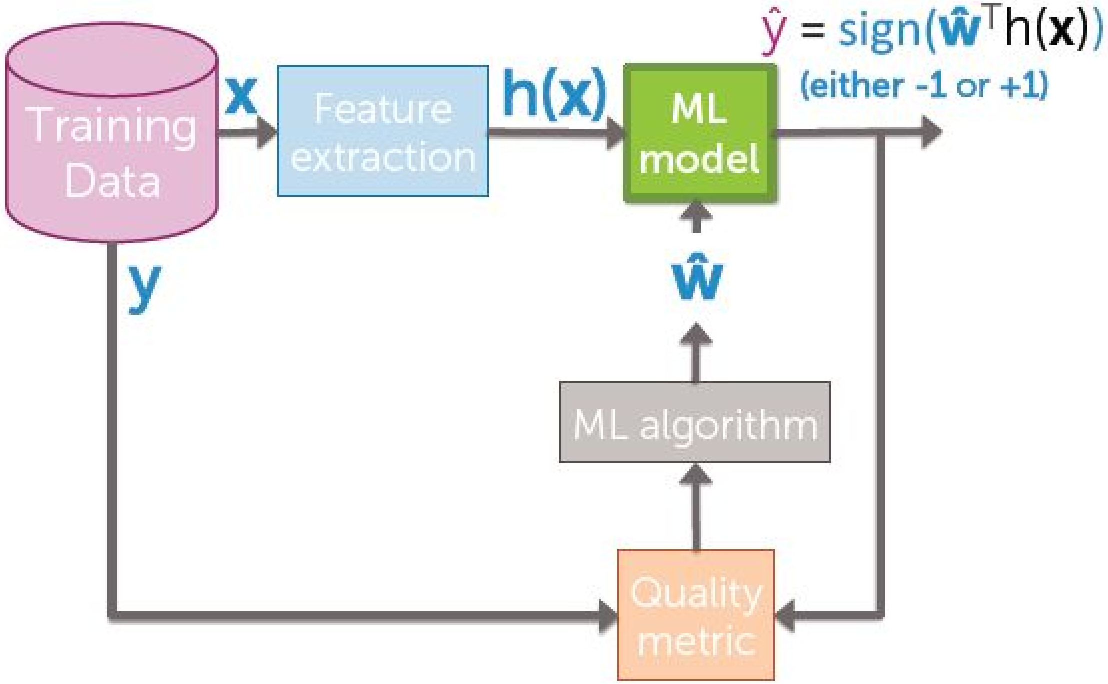
or, $\log(\mathbf{x}[7])\ \mathbf{x}[2]$ = log(#bad) x #awful

or, tf-idf("awful")

…

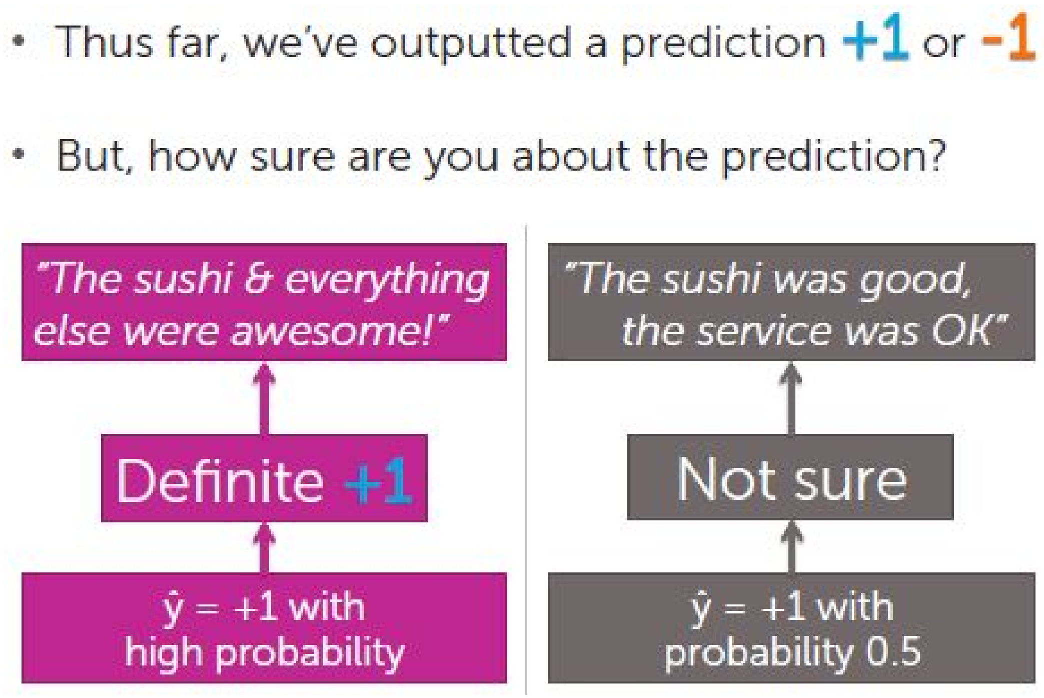
feature D+1 = $h_D(\mathbf{x})$ … some other function of $\mathbf{x}[1], \ldots, \mathbf{x}[d]$

ML model


Training Data
x
Feature extraction
h(x)
ML model
$\hat{y} = \text{sign}(\hat{\mathbf{w}}^T h(\mathbf{x}))$
(either -1 or +1)
y
ŵ
ML algorithm
Quality metric

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?

"The susti & everything else were awesome!"

Definite +1

ŷ = +1 with high probability

"The sushi was good, the service was OK"

Not sure

ŷ = +1 with probability 0.5

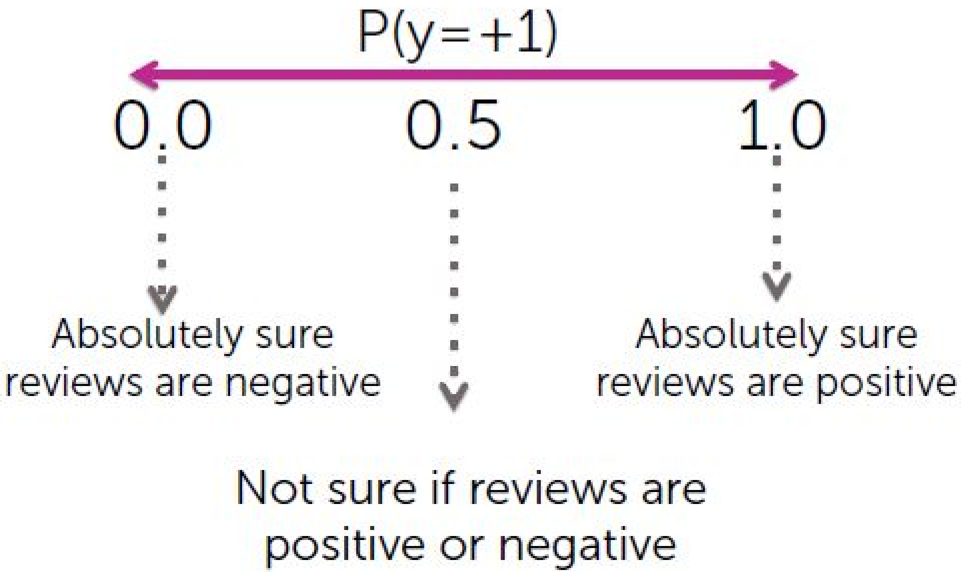
An intuition on probability

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
...	...

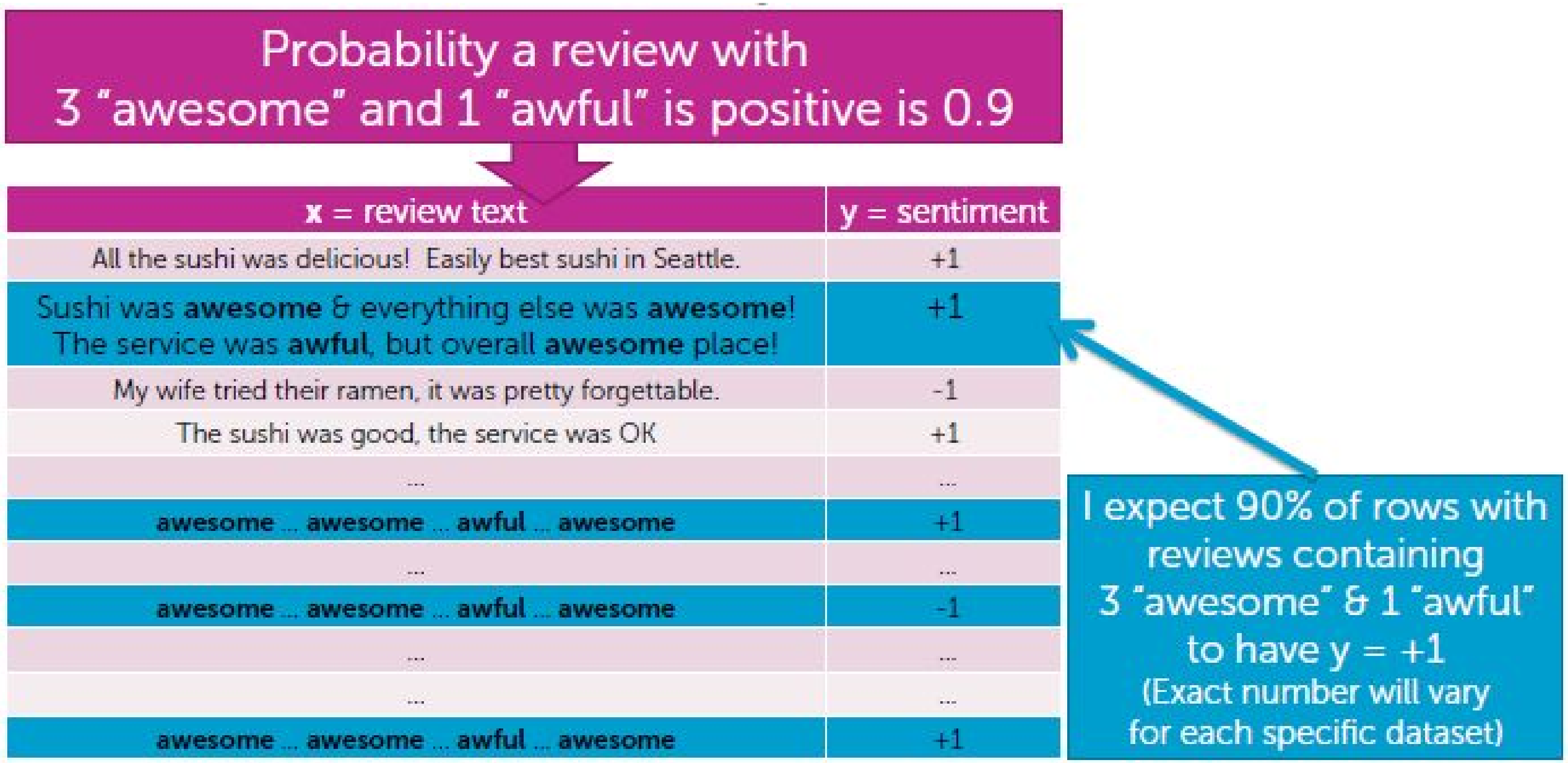
I expect 70% of rows to have y = +1
(Exact number will vary for each specific dataset)

Degrees of belief



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

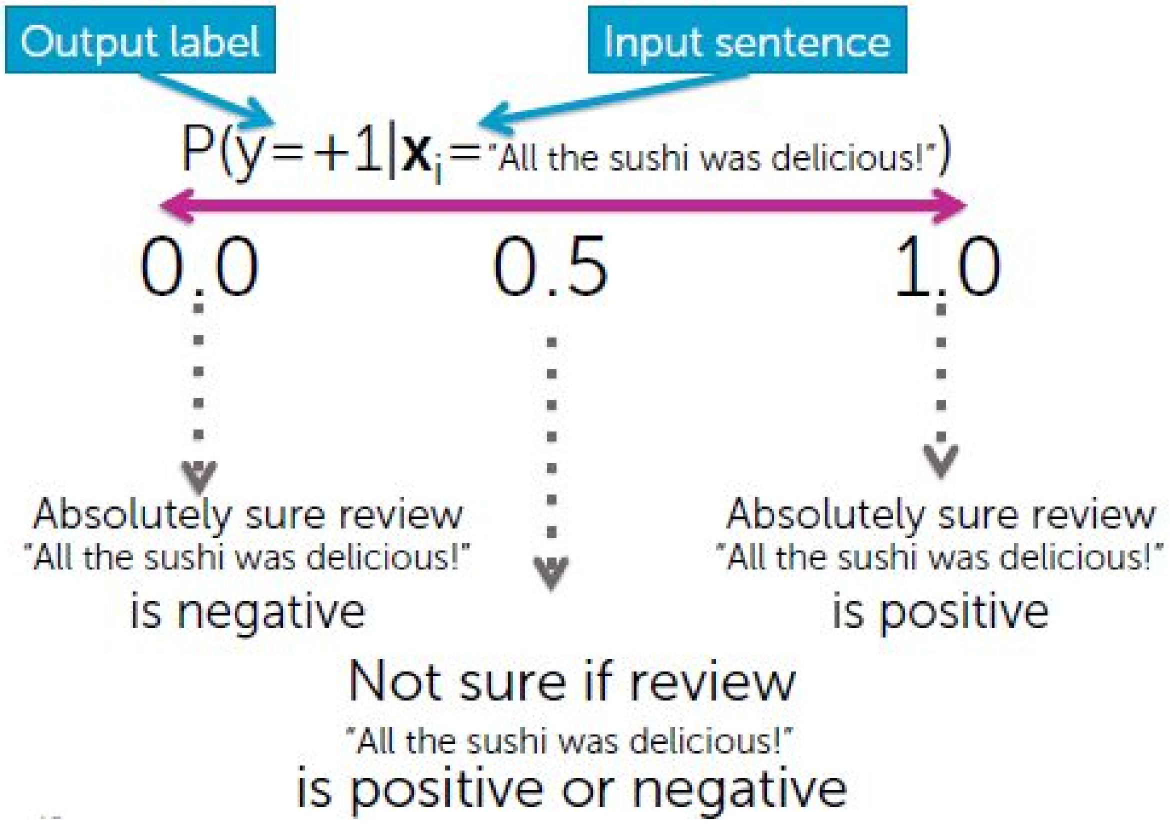


Probability a review with
3 "awesome" and 1 "awful" is positive is 0.9

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



Output label

Input sentence

$P(y=+1|\mathbf{x}_i=$"All the sushi was delicious!"$)$

0.0

0.5

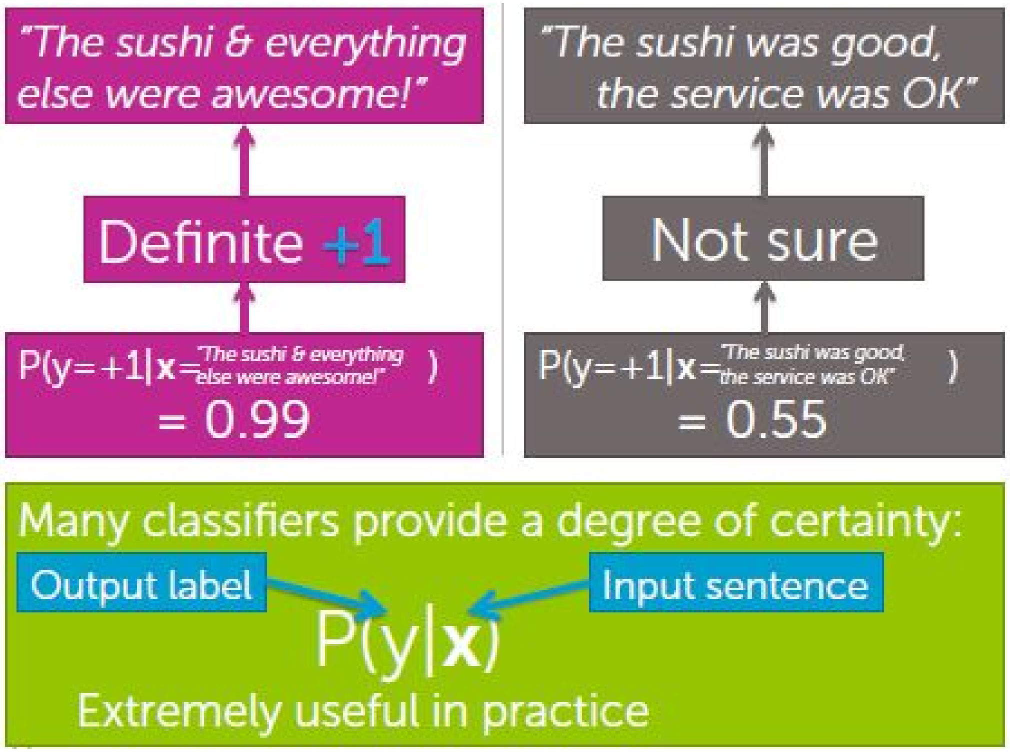
1.0

Absolutely sure review
"All the sushi was delicious!"
is negative

Not sure if review
"All the sushi was delicious!"
is positive or negative

Absolutely sure review
"All the sushi was delicious!"
is positive

How confident is your prediction?



"The sushi & everything else were awesome!"

Definite +1

$P(y=+1|\mathbf{x}=$"The sushi & everything else were awesome!"$)$ $= 0.99$

"The sushi was good, the service was OK"

Not sure

$P(y=+1|\mathbf{x}=$"The sushi was good, the service was OK"$)$ $= 0.55$

Many classifiers provide a degree of certainty:

Output label

Input sentence

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

Extremely useful in practice

Goal: Learn conditional probabilities from data

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

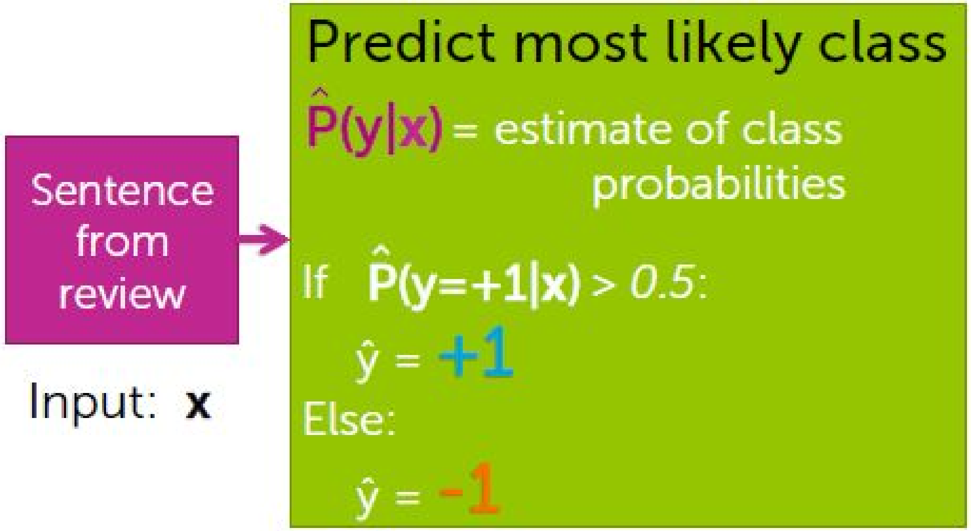
x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
0	2	-1
3	3	-1
4	1	+1
…	…	…

Optimize **quality metric** on training data

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

Useful for predicting $\hat{y}$

Estimate class probabilities



Sentence from review

Input: **x**

Predict most likely class

$\hat{P}(y|x)$ = estimate of class probabilities

If $\hat{P}(y=+1|x) > 0.5$:

$\hat{y} = +1$

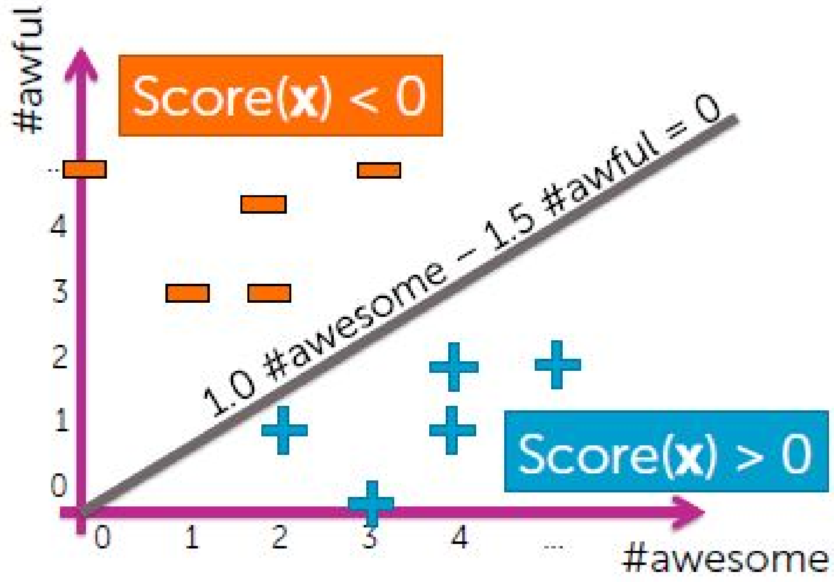
Else:

$\hat{y} = -1$

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

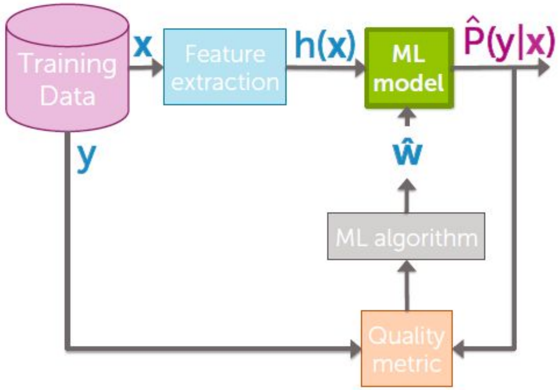
Revisit the “Score”

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

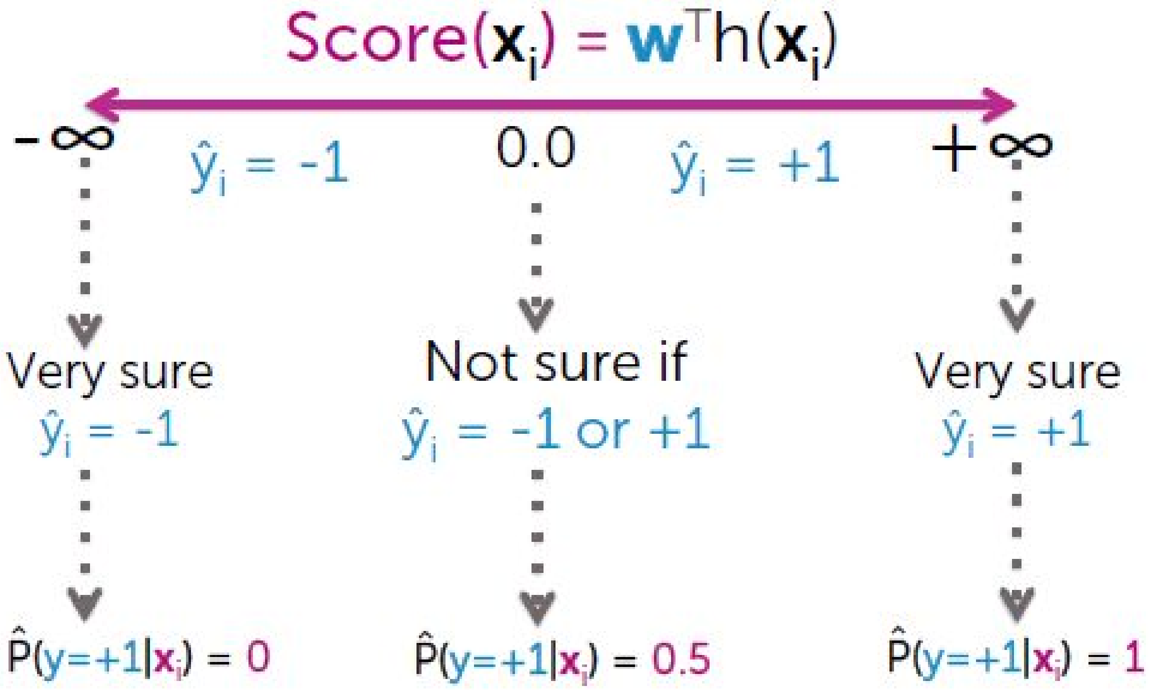


Relate Score($\mathbf{x}_i$) to $\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}})$?

So far, we focused on decision boundaries determined by the “score”



Interpreting "Score(x)"



$\text{Score}(\mathbf{x}_i) = \mathbf{w}^T h(\mathbf{x}_i)$

$-\infty$ $\hat{y}_i = -1$ 0.0 $\hat{y}_i = +1$ $+\infty$

Very sure $\hat{y}_i = -1$

Not sure if $\hat{y}_i = -1$ or $+1$

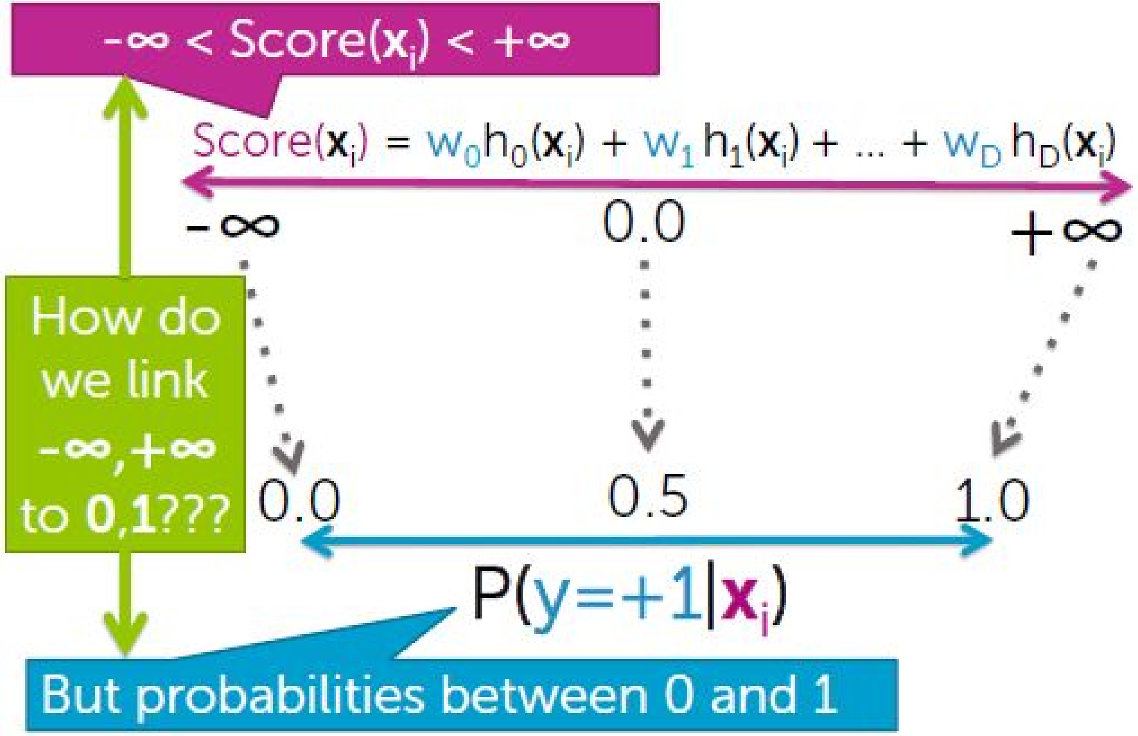
Very sure $\hat{y}_i = +1$

$\hat{P}(y=+1|\mathbf{x}_i) = 0$

$\hat{P}(y=+1|\mathbf{x}_i) = 0.5$

$\hat{P}(y=+1|\mathbf{x}_i) = 1$

Why not we use regression?



$-\infty < \text{Score}(\mathbf{x}_i) < +\infty$

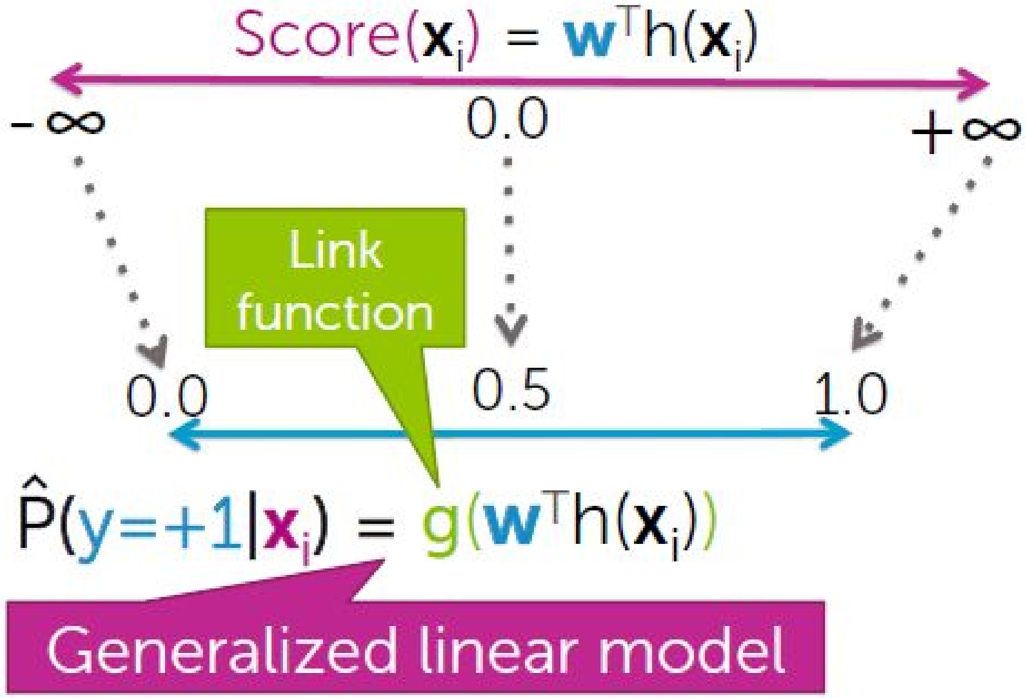
$\text{Score}(\mathbf{x}_i) = w_0 h_0(\mathbf{x}_i) + w_1 h_1(\mathbf{x}_i) + \ldots + w_D h_D(\mathbf{x}_i)$

How do we link $-\infty,+\infty$ to **0,1**???

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

But probabilities between 0 and 1

Link Function: Squeeze to [0, 1]



$\text{Score}(\mathbf{x}_i) = \mathbf{w}^T h(\mathbf{x}_i)$

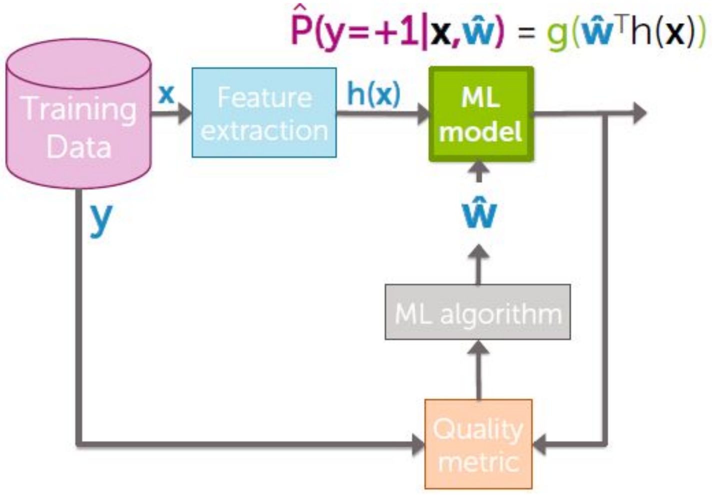
$-\infty$ 0.0 $+\infty$

Link function

0.0 0.5 1.0

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

Generalized linear model



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

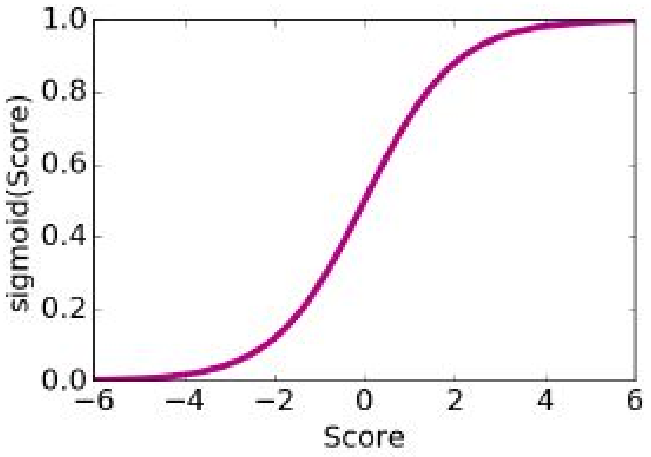
Logistic Regression Classifier

"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)					

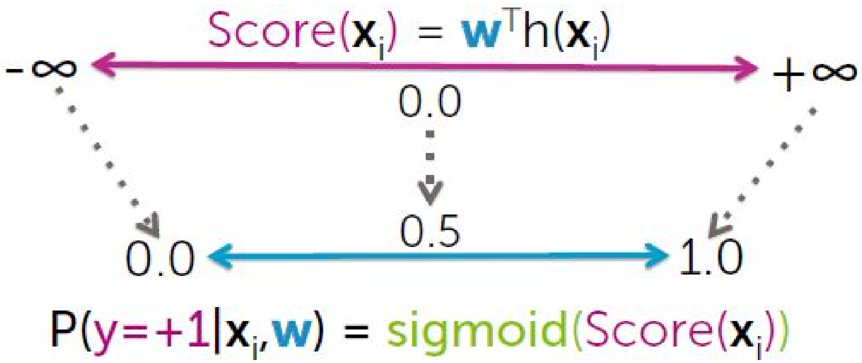


Logistic regression model

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

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

Score



$\text{Score}(\mathbf{x}_i) = \mathbf{w}^T h(\mathbf{x}_i)$

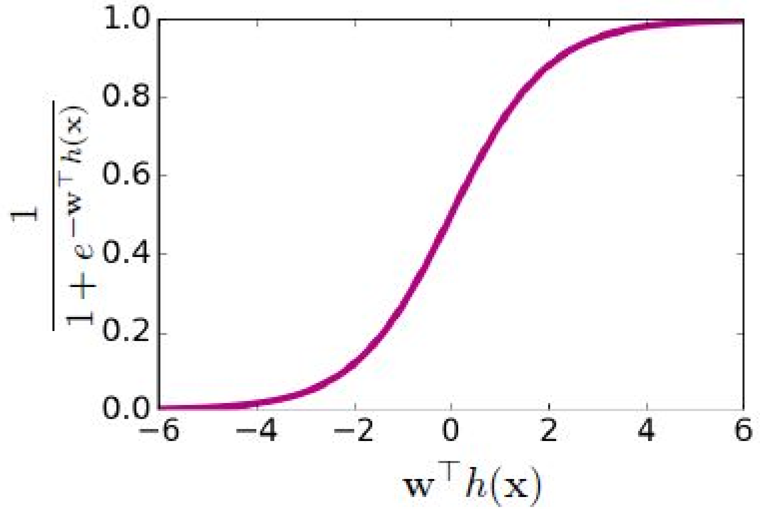
$-\infty$ $+\infty$

0.0

0.5

0.0 1.0

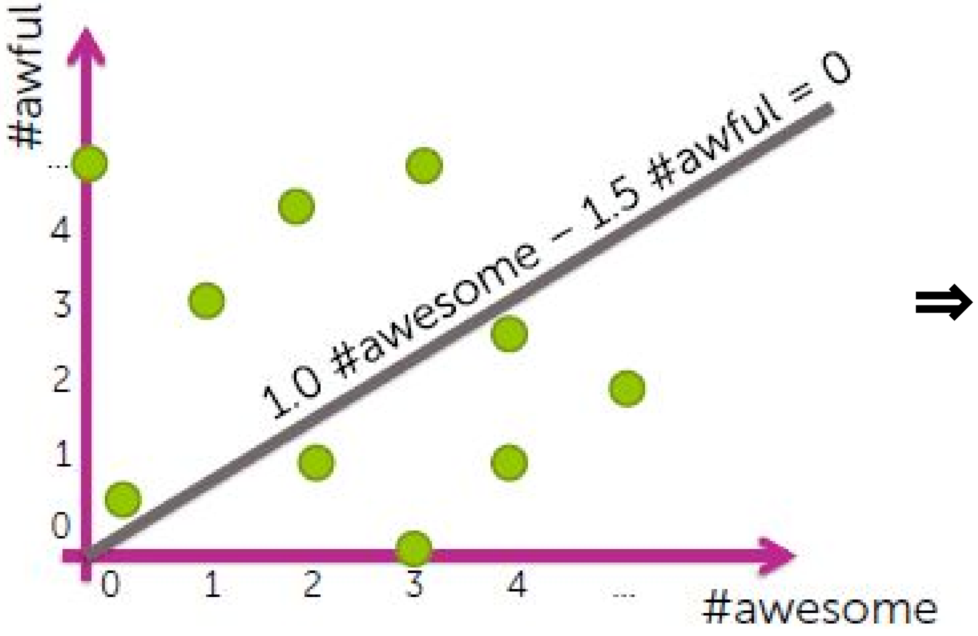
$P(y=+1|\mathbf{x}_i,\mathbf{w}) = \text{sigmoid}(\text{Score}(\mathbf{x}_i))$



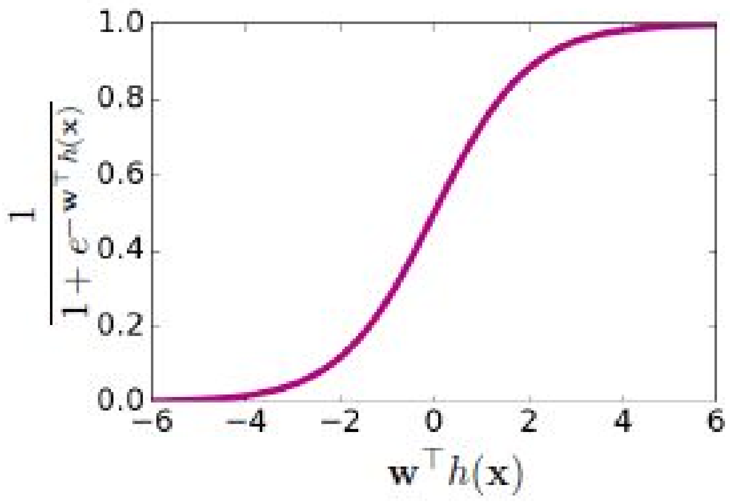
$\text{Score}(\mathbf{x}_i)$	$P(y=+1\|\mathbf{x}_i,\mathbf{w})$

The two steps

Linear decision boundary ⇒ Logistic regression model



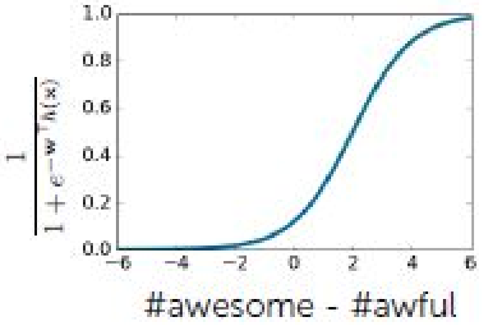
⇒



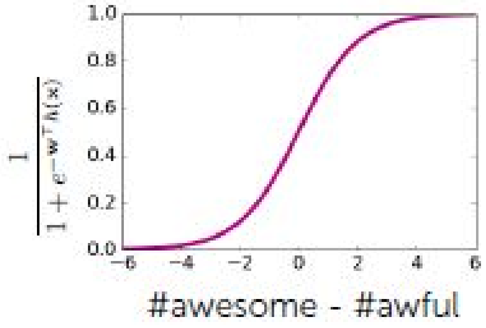
Effects of coefficients on logistic regression model

Linear decision boundary ⇒ Logistic regression model

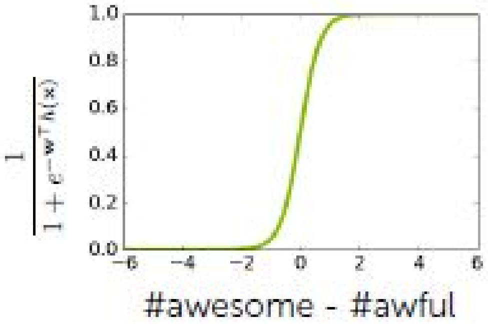
w_0	-2
$w_{\#awesome}$	+1
$w_{\#awful}$	-1



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

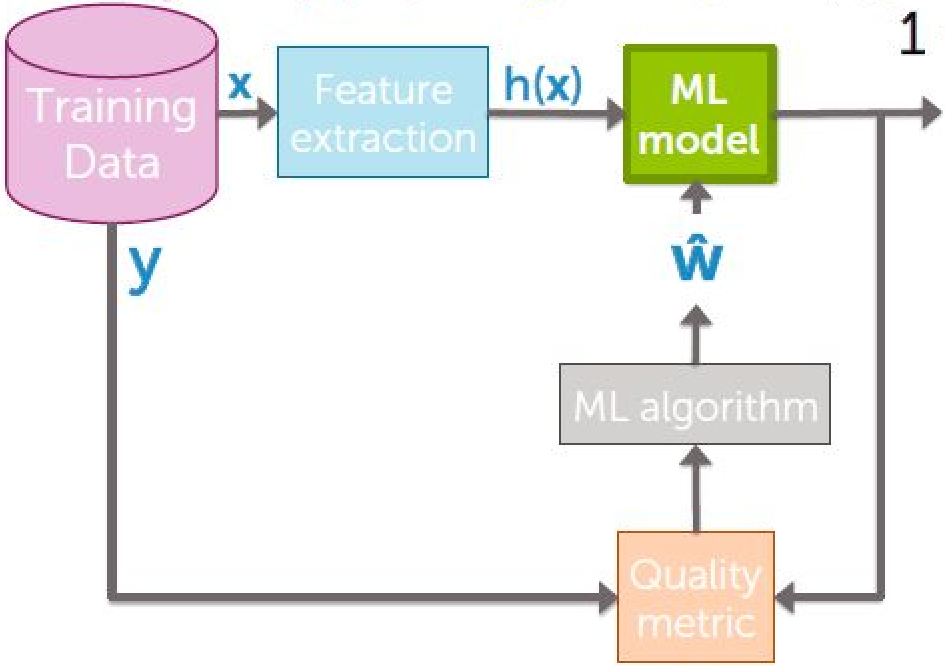


w_0	0
$w_{\#awesome}$	+3
$w_{\#awful}$	-3

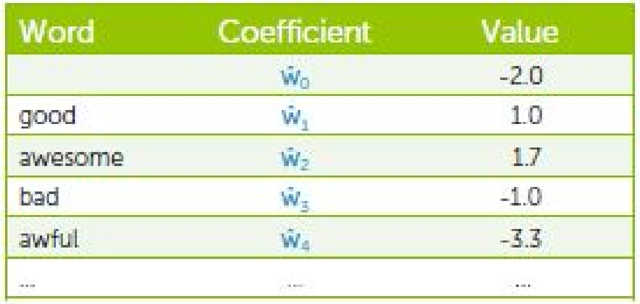


Logistic regression model

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

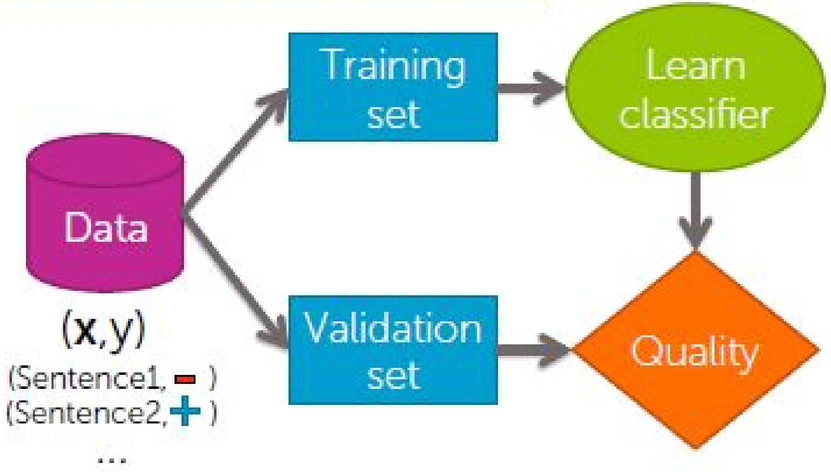


Training a classifier → Learning coefficients



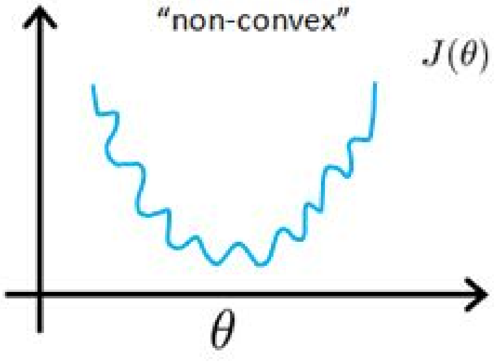
Word	Coefficient	Value
	$\hat{w}_0$	-2.0
good	$\hat{w}_1$	1.0
awesome	$\hat{w}_2$	1.7
bad	$\hat{w}_3$	-1.0
awful	$\hat{w}_4$	-3.3
...	...	...

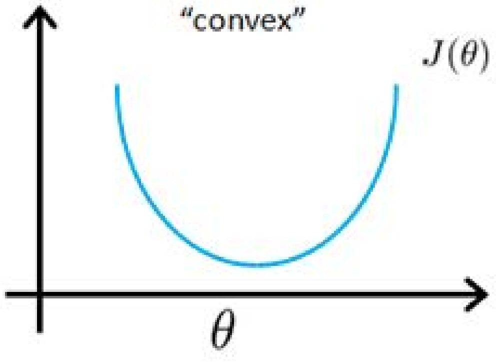
$$\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}}) = \frac{1}{1 + e^{-\hat{\mathbf{w}}\, h(\mathbf{x})}}$$



Find “best” classifier

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





This nonlinear function now causes the cost to be **non-convex**

That means it can easily get stuck in many local minimas

Recall → $\mathrm{RSS}(\mathbf{w}) = \sum_{i=1}^{N} (y_i - h(\mathbf{x}_i)^T\mathbf{w})^2$

which is the sum of cost of all samples for linear regression

Find “best” classifier

Logistic function have to be redefined so that it is convex

Let $\hat{y}$ or $H_w(x)$ be the function of the decision boundary,
the convex “version” of the logistic function is approximated as

$$J = \text{Cost}\,(H_w(x), y) = -y \log(H_w(x)) - (1-y) \log (1-H_w(x))$$

Next, compute the gradient of the cost function, which is part of the update procedure:

$$w^{t+1} := w^{t} - \Sigma\, \underbrace{(H_w(x) - y)\, x}_{\Delta J}$$

Encoding categorical inputs

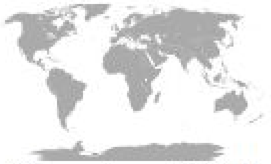
Numerical inputs:

- #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

Categories to numerics

x =

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

196 categories

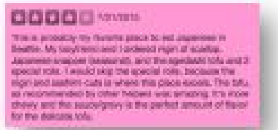
1-hot encoding

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

196 features

---

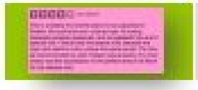
x =



Restaurant review
(Text data)

10,000 words in vocabulary

Bag of words

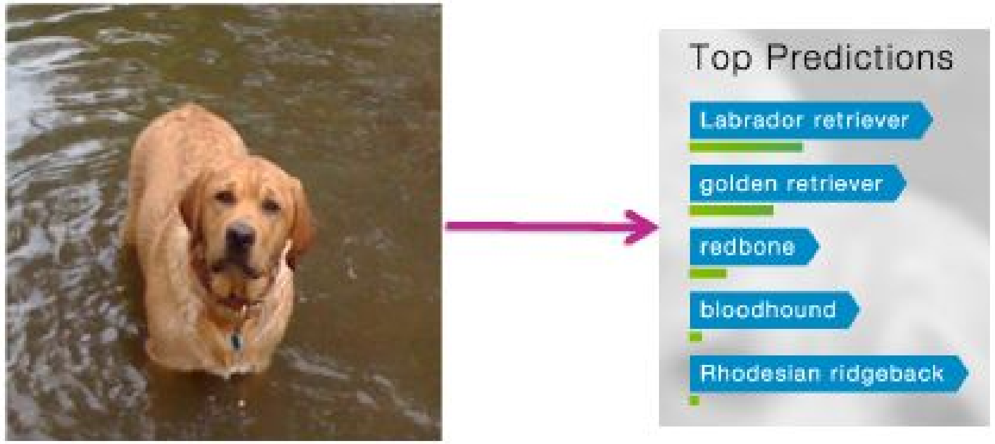
	#awesome	#awful	..	..	#sushi
x	$h_1(x)$	$h_2(x)$	...	$h_{9999}(x)$	$h_{10000}(x)$
	2	0	0	0	3

10,000 features

Multi-class Classification

Using 1-versus-All

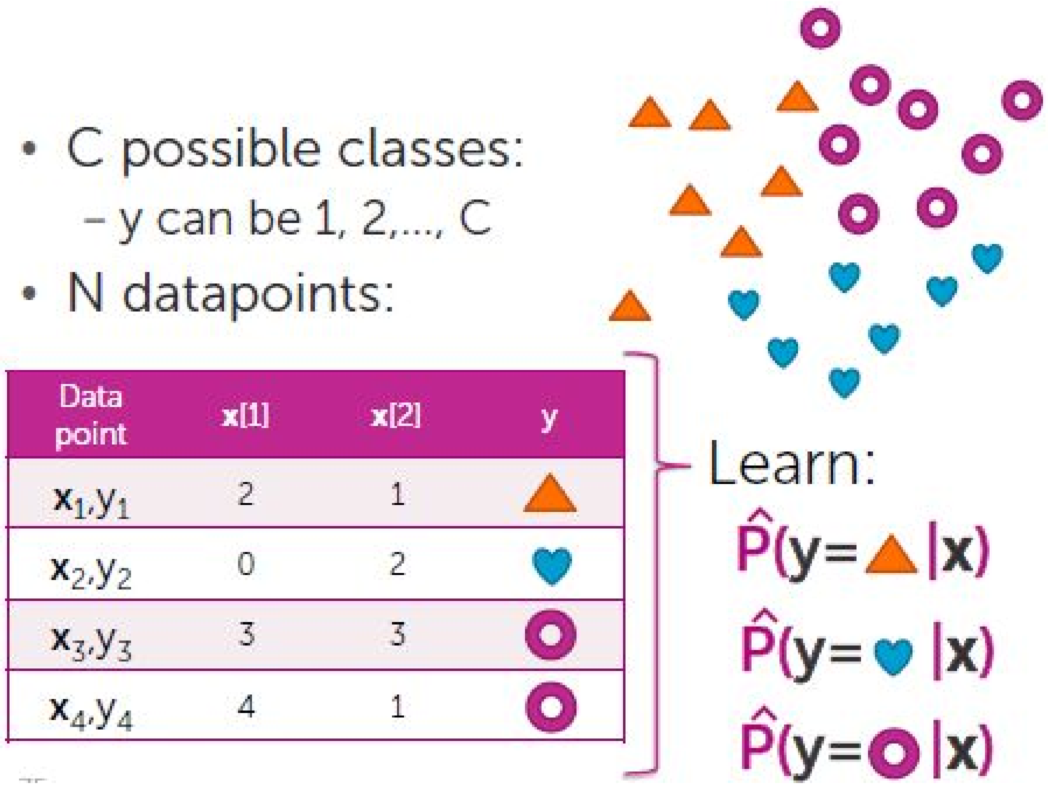
Multiclass classification



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:

$\hat{P}(y=\blacktriangle|\mathbf{x})$

$\hat{P}(y=\heartsuit|\mathbf{x})$

$\hat{P}(y=\bigcirc|\mathbf{x})$

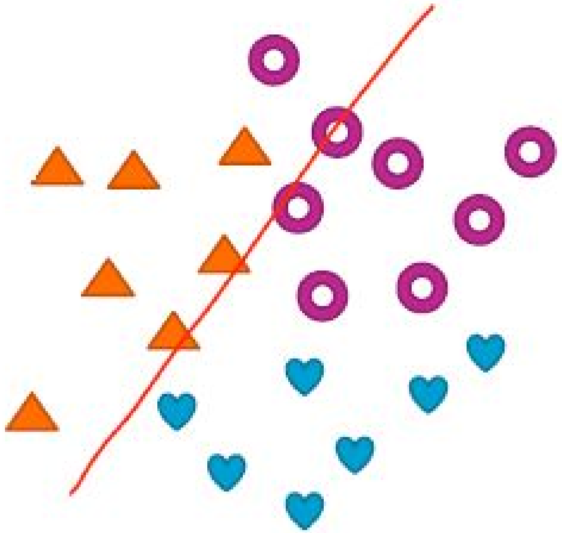
1-versus-all

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

+1 class: points with $y_i=\blacktriangle$
-1 class: points with $y_i=$ ♥ OR ◎

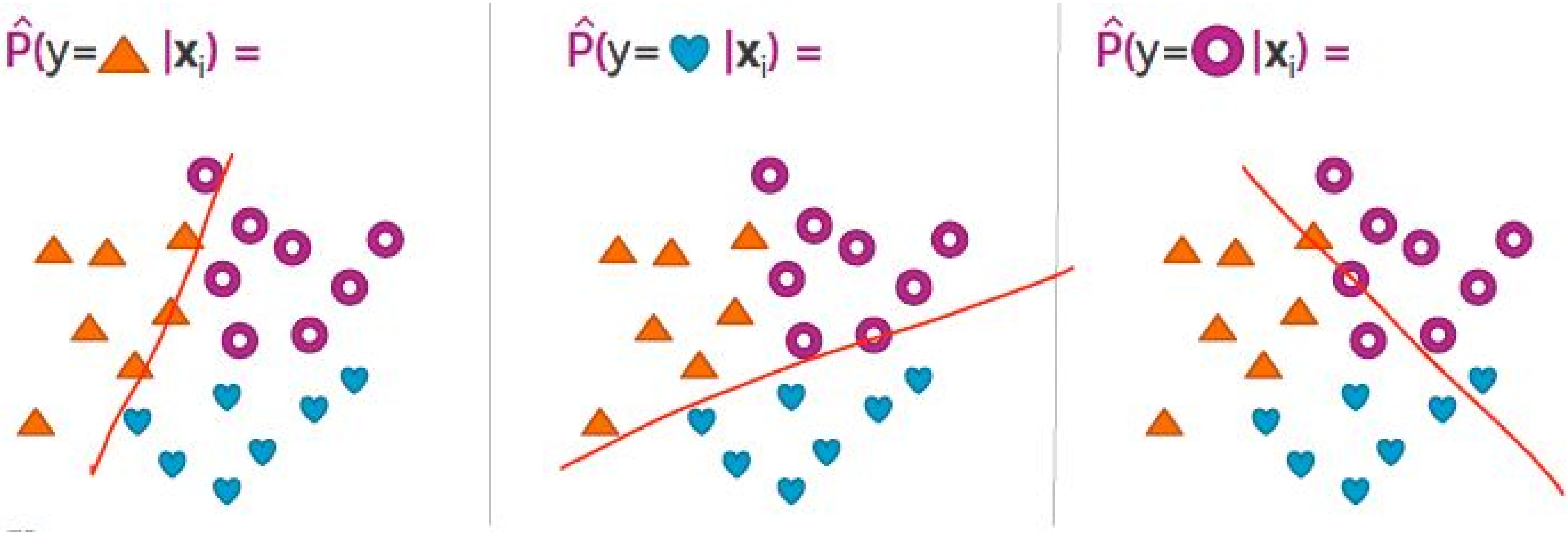
Train classifier: $\hat{P}_{\blacktriangle}(y=+1|\mathbf{x})$

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



1-versus-all

Simple multiclass classification using *C* 2-class models



Multiclass training

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; $\hat{y}$ = 0
For c = 1,…,C:
    If $\hat{P}_c(y=+1|\mathbf{x}_i)$ > max_prob:
        $\hat{y}$ = c
        max_prob = $\hat{P}_c(y=+1|\mathbf{x}_i)$
```

Evaluating Classifiers

Classification error & Accuracy

Error measures fraction of mistakes made in classification:

$$\text{error} = \frac{\text{\# of mistakes}}{\text{Total \# of Sentences}}$$

(Best possible value: 0.0)

Accuracy measures the fraction of correct predictions:

$$\text{accuracy} = \frac{\text{\# of Correct}}{\text{Total \# of Sentences}}$$

(Best possible value: 1.0)

What's a good accuracy?

- For binary classification:
  - Half the time, you'll get it right! (on average)
    ➔ accuracy = 0.5

- For k classes, accuracy = 1/k
  - 0.333 for 3 classes, 0.25 for 4 classes,…

At the very, very, very least,
you should healthily beat random…
Otherwise, it's (usually) pointless…

Is 90% accuracy good? Depends…

2010 data shows:
"90% emails sent are spam!"

Predicting every email is spam gets you 90% accuracy!!!

Majority class prediction

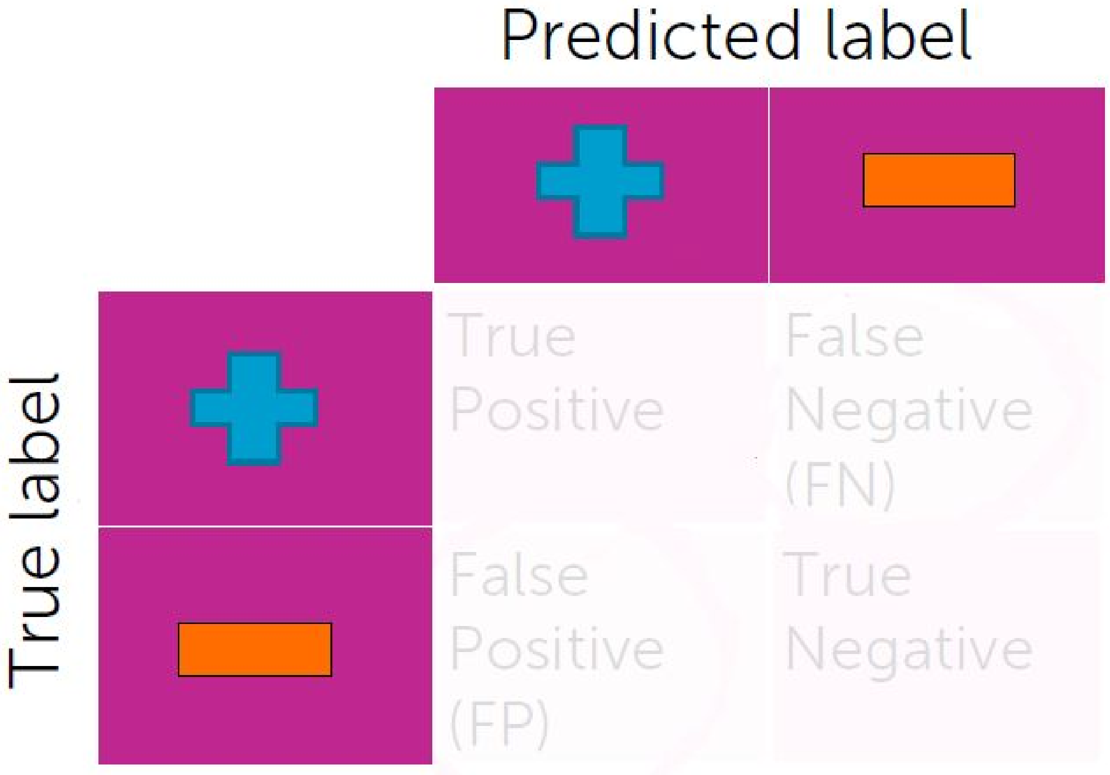
Amazing performance when there is class imbalance
(but silly approach)

- One class is more common than others
- Beats random (if you know the majority class)

Ask the hard questions…

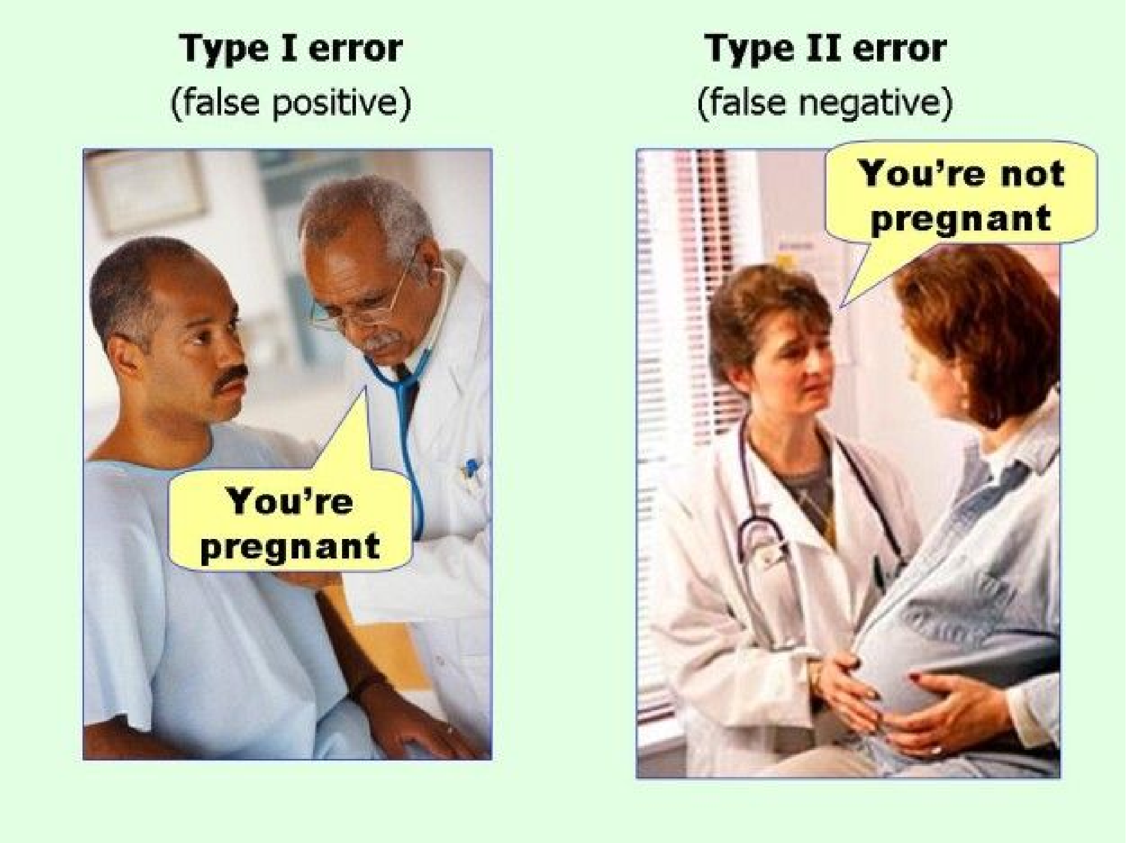
- Is there class imbalance?
- How does it compare to a simple, baseline approach?
  - Random guessing
  - Majority class
  - …
- Most importantly: *what accuracy does my application need?*
  - What is good enough for my user's experience?
  - What is the impact of the mistakes we make?

Types of Errors



		Predicted label	
		+	−
True label	+	True Positive	False Negative (FN)
	−	False Positive (FP)	True Negative

Types of Errors



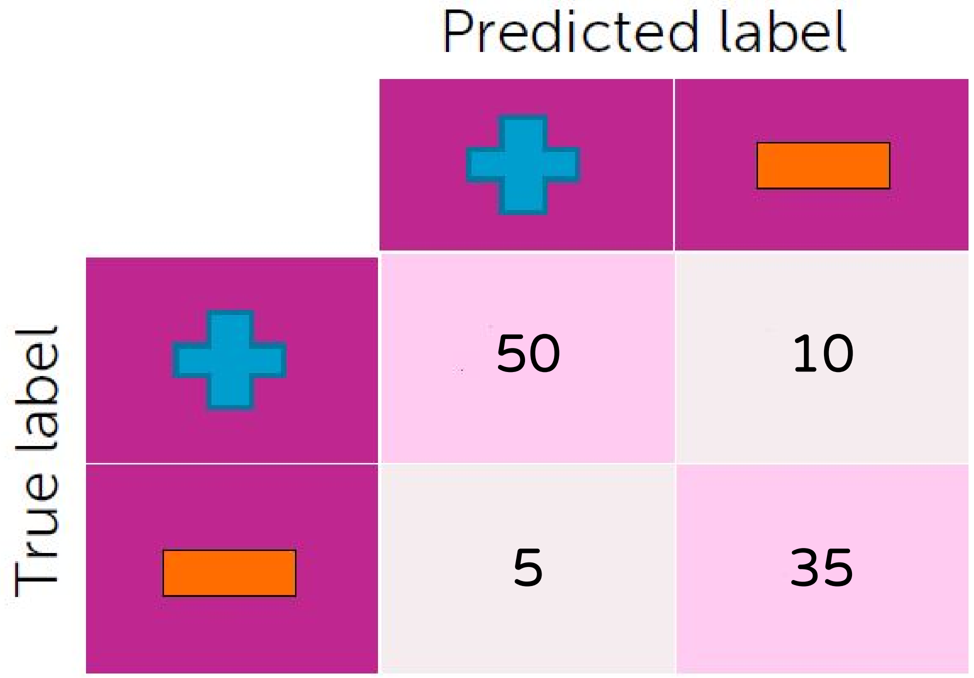
Cost of different types of mistakes

Different (and high) in some applications!

	Spam filtering	Medical diagnosis
False negative	Annoying	Disease not treated
False positive	Email lost	Wasteful treatment

Confusion matrix – binary

Test set: 100 samples



		Predicted label	
		+	−
True label	+	50	10
	−	5	35

Accuracy = ?

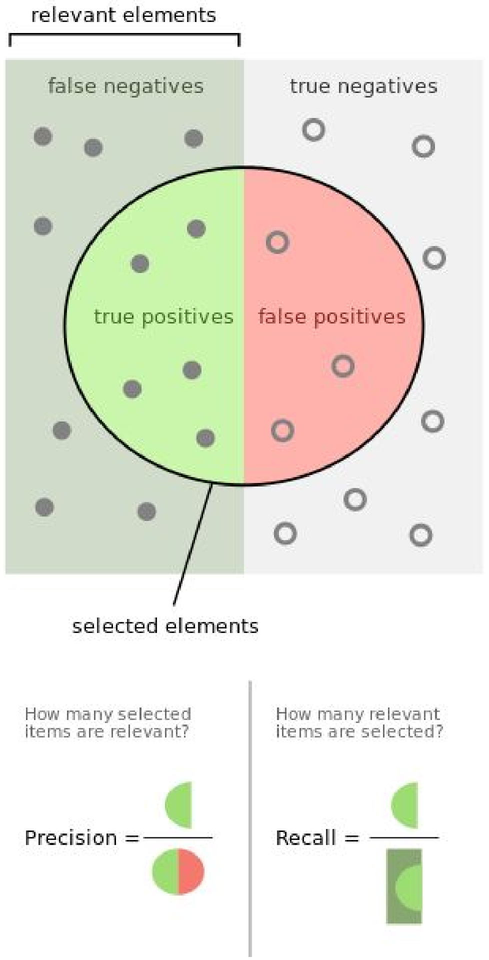
Confusion matrix – multiclass

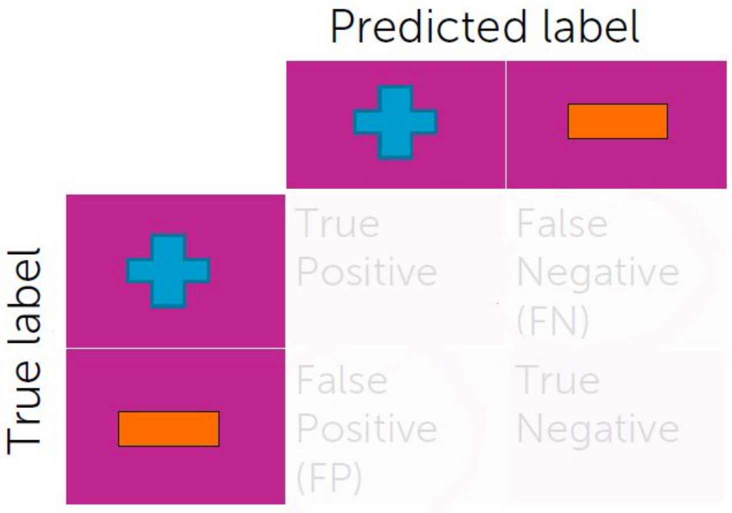
Test set: 100 samples

		Predicted label		
		Healthy	Cold	Flu
True label	Healthy	60	8	2
	Cold	4	12	4
	Flu	0	4 2	8

Accuracy = ?

Precision, Recall

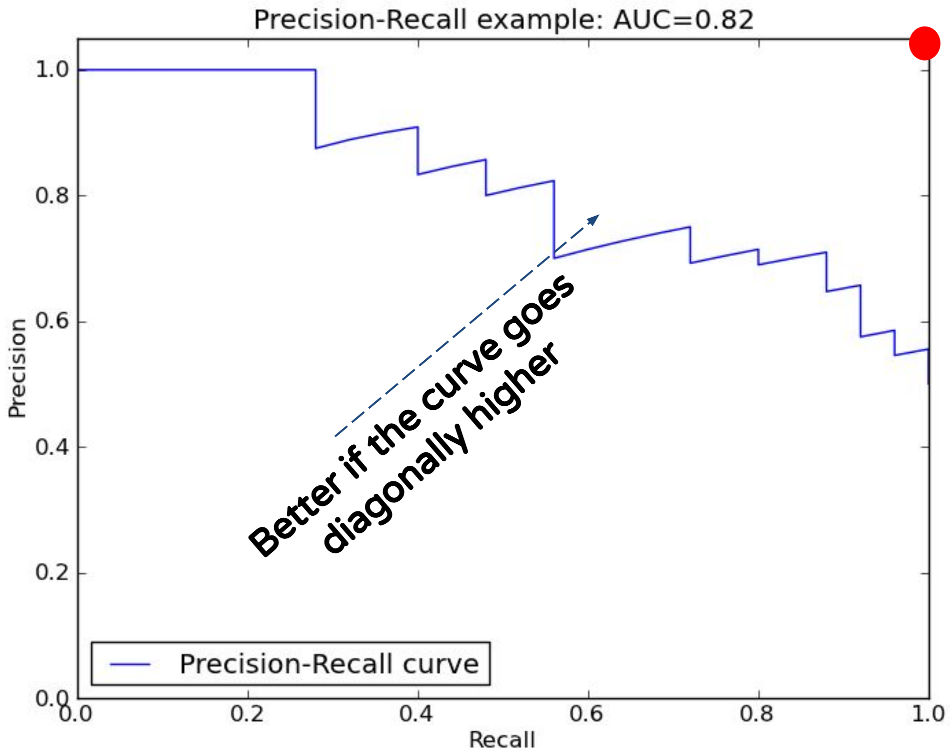




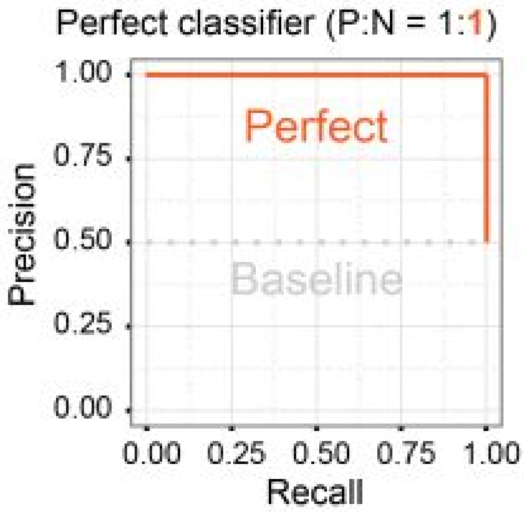
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

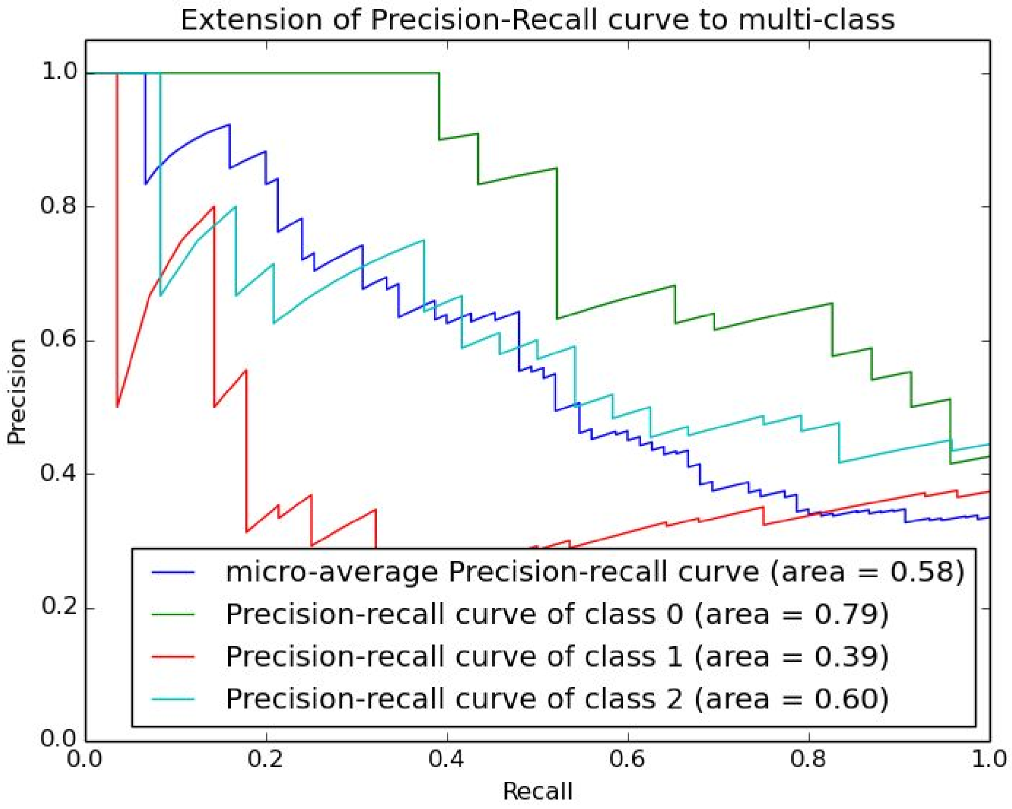
Precision-Recall Curve



This is where you want your performance to be!



Precision-Recall Curve



Can be extended to multiclass case

⇒ You get separate P-R curves for each class!

Go To
Exercises

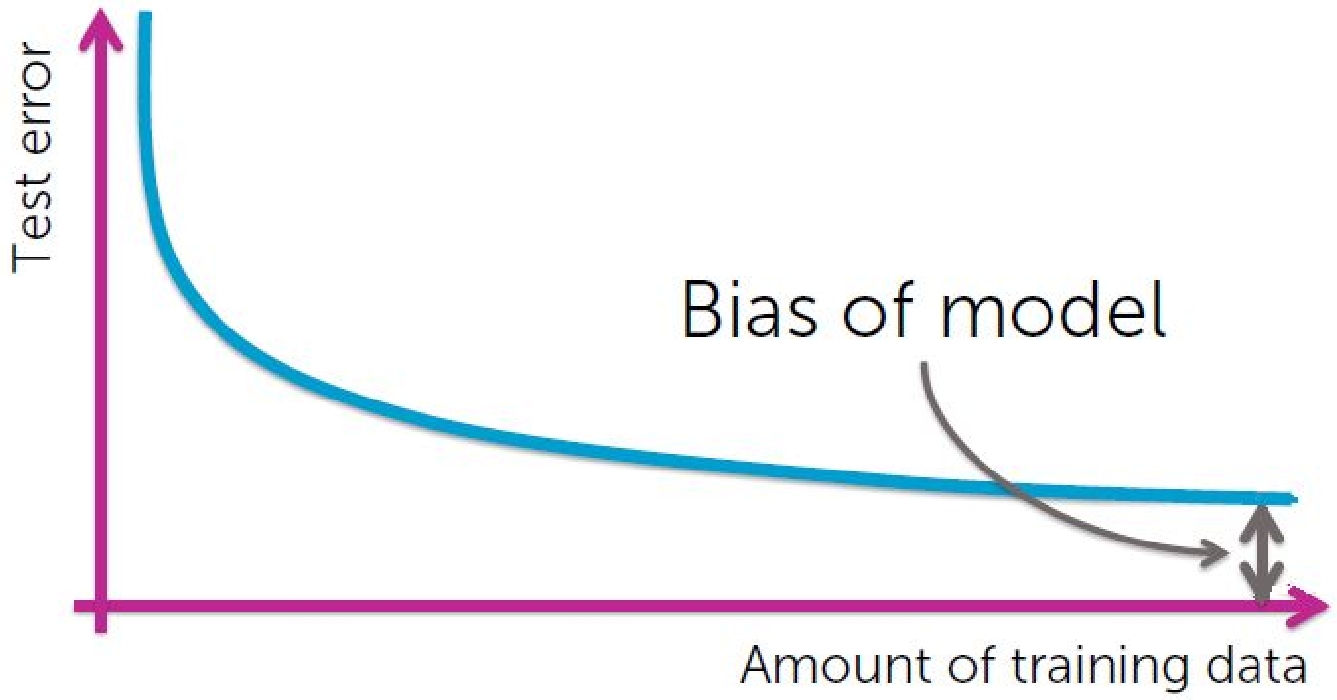
Learning Curves

How much data do I need?

How much data needed for learning?

- The more the merrier ☺
  - But data quality is most important factor
- Theoretical techniques sometimes can bound how much data is needed
  - Typically too loose for practical application
  - But provide guidance
- In practice:
  - More complex models require more data
  - Empirical analysis can provide guidance

Is there a limit? Yes, for most models



More complex models have less bias

Sentiment classifier using single words can do OK, but…

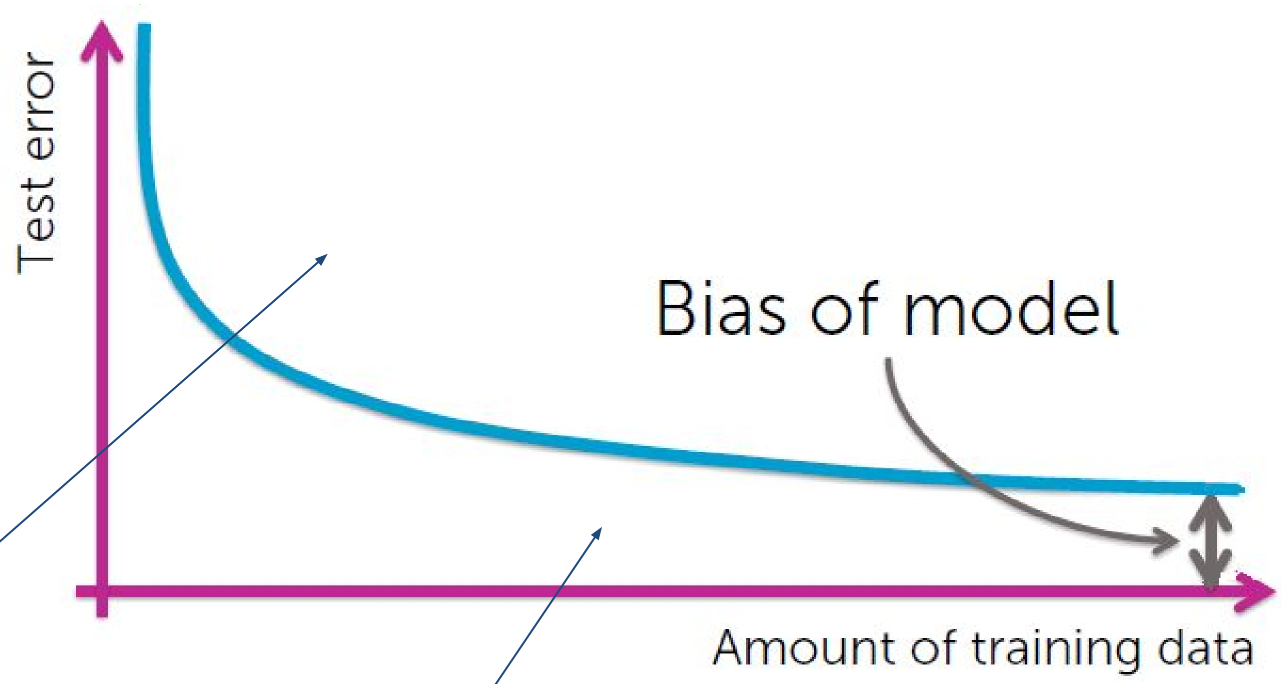
Never classify correctly:
“The sushi was not good.”

More complex model:
consider pairs of words (bigrams)

Word	Weight
good	+1.5
not good	-2.1

Less bias →
potentially more accurate,
needs more data to learn

Models with less bias



Need more data to learn well
But with sufficient data, bias is greatly reduced!

Try Kaggle

Breast Cancer Wisconsin (Diagnostic) Data Set

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data