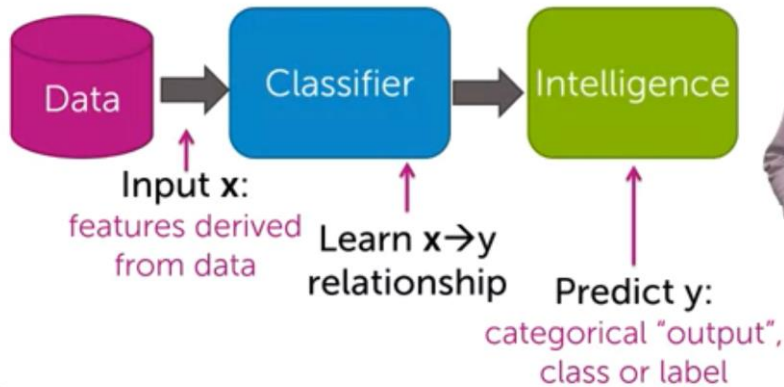


What is classification?

From features to predictions



5

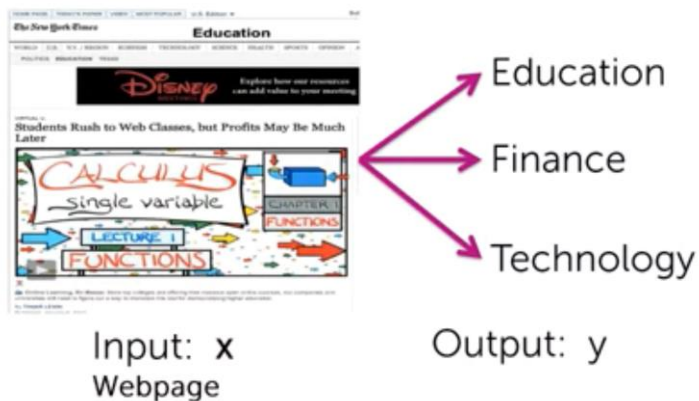
©2015-2016 Emily Fox & Carlos Guestrin

Learning Specialization

Multi-class classification

Given a web page, we have to find out whether that page belongs to 'Education', 'finance', 'technology' based on the content of that webpage.

Example multiclass classifier *Output y has more than 2 categories*



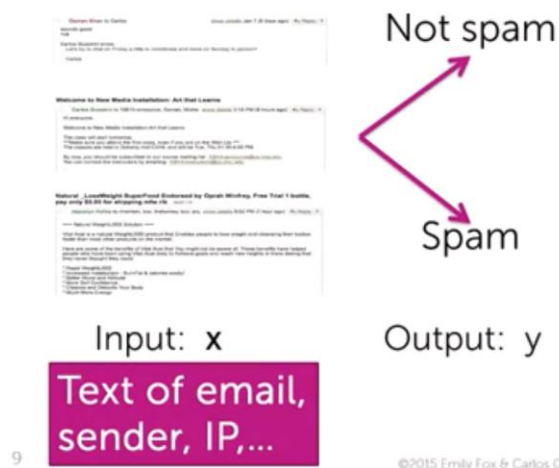
8

©2015 Emily Fox & Carlos Guestrin

Learning Specialization

Famous Example : Spam filtering

Spam filtering



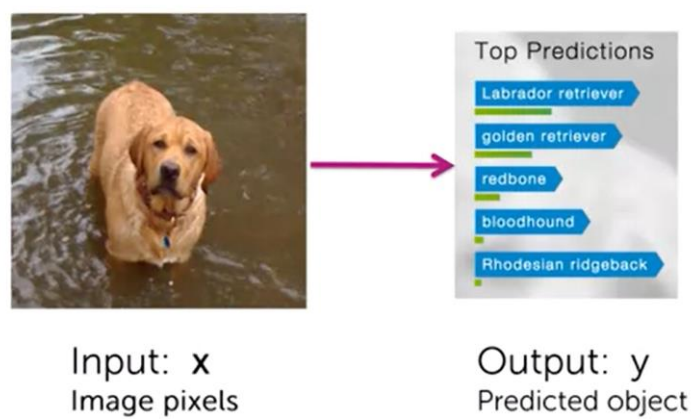
9

©2015 Emily Fox & Carlos Guestrin



Learning Specialization

Image classification



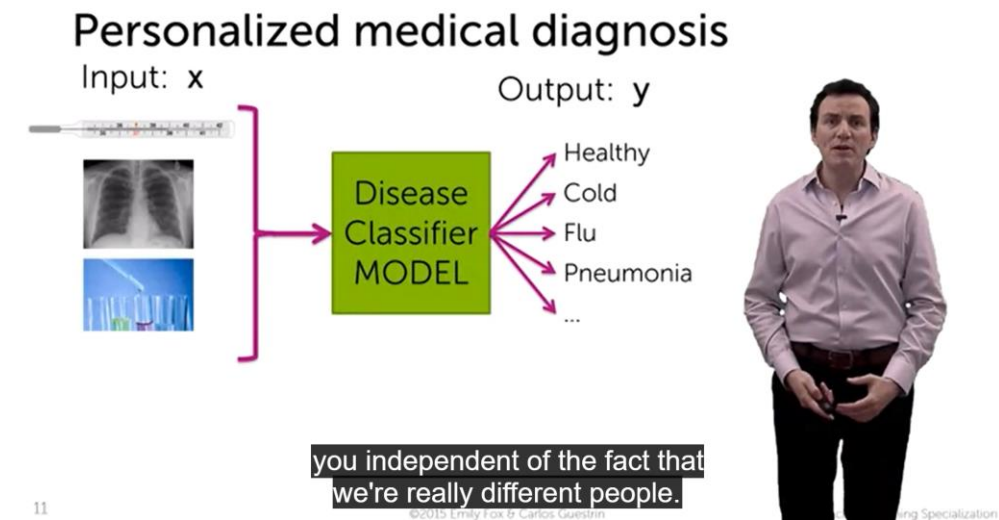
10

©2015 Emily Fox & Carlos Guestrin



Learning Specialization

This classifier is for all the patients ir-respective of their personal habits

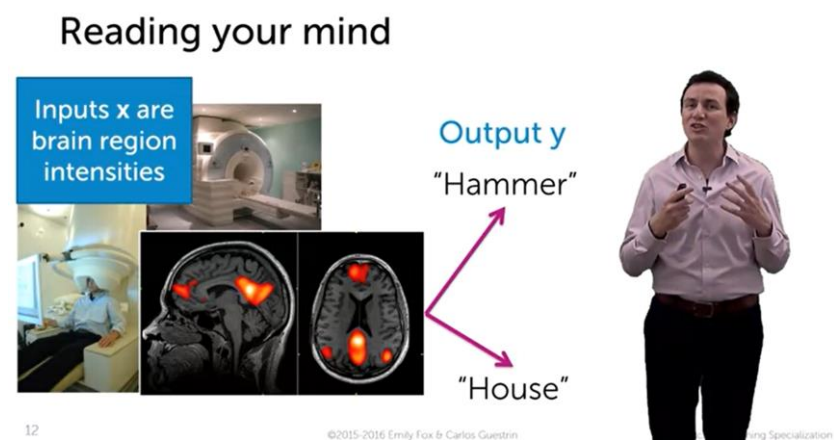


But personalized medical diagnosis tests our DNA, food-habits, genetical problems, total body metabolism etc to decide what treatment is going to be most effective for that individual person instead a giving a routine flat general treatment.

This is the real world example of classification problem.

Mind-Reading :

A person sees an hammer. A FMRI is taken for that particular person, From that Scanned images we can conclude that what a person is seeing whether an 'hammer' or a 'house'



Linear Classifier

1) Linear Classifier Model

Simple hyperplane

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

$$\text{Score}(\mathbf{x}_i) = w_0 + \overset{\text{awesome}}{w_1 x_i[1]} + \dots + \overset{\text{awful}}{w_d x_i[d]} = \mathbf{w}^T \mathbf{x}_i$$

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

26

©2015-2016 Emily Fox & Carlos Guestrin

$$\begin{array}{c|c} \mathbf{w}^T & \mathbf{x}_i \\ \hline w_0 & x_i[0] \\ w_1 & x_i[1] \\ w_2 & x_i[2] \\ \vdots & \vdots \\ w_d & x_i[d] \end{array}$$

For a single row representing $\rightarrow \mathbf{x}_i$

Score for that single row $\rightarrow \text{score}(\mathbf{x}_i)$

Sign of the score of that single row $\rightarrow \hat{y}^{(i)} = \text{Sign}(\text{score}(\mathbf{x}_i))$

i from 1 to N (row wise) $[\mathbf{x}_i]$

The Role of sign

If the score > 0 then predict $\rightarrow +1$

If the score < 0 then predict $\rightarrow -1$

At 0, we have the choice to predict either -1/+1. You make an arbitrary choice

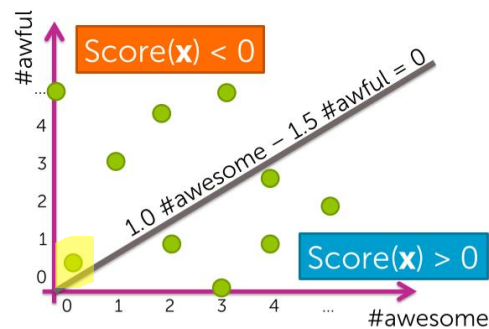
2) Effects of coefficient values on decision boundary

Initially the intercept $\rightarrow 0$

Decision boundary: effect of changing coefficients

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

$$\text{Score}(\mathbf{x}) = 1.0 \text{ #awesome} - 1.5 \text{ #awful}$$



27

©2015-2016 Emily Fox & Carlos Guestrin

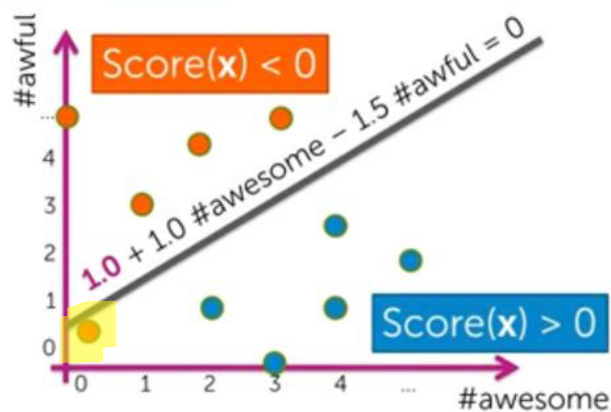
Machine Learning Specialization

Now the intercept $\rightarrow 1$ and the line slightly shifts up, so the orange point which was -ve before, now becomes +ve

Decision boundary: effect of changing coefficients

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

$$\text{Score}(\mathbf{x}) = 1.0 + 1.0 \text{ #awesome} - 1.5 \text{ #awful}$$



28

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

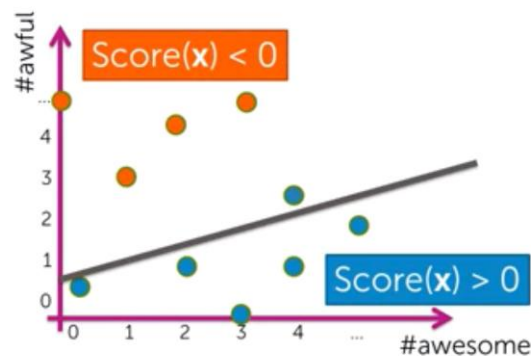


After changing the $w_2 = -3.0$, the line gets modified, and the blue point which was +ve Now becomes -ve.

Decision boundary: effect of changing coefficients

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

$$\text{Score}(\mathbf{x}) = 1.0 + 1.0 \text{ #awesome} - 3.0 \text{ #awful}$$



29

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

From this we can conclude that the coefficients are playing a very important role in the classification.

3) Using features of inputs

More generic features... D-dimensional hyperplane

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

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

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

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

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

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

...

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

30

©2015-2016 Emily Fox & Carlos Guestrin

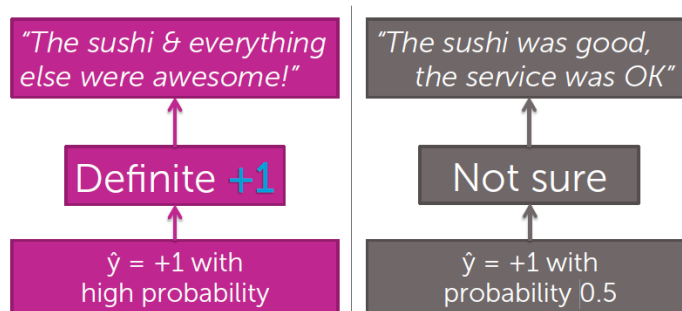
Machine Learning Specialization

Probabilities and its basics

Not all our output will be exactly +1 or -1, especially the output of logistic regression will be like 0.432 (or) 0.211. So in-order to conclude them as +1 or -1 (Here probability comes into picture)

How confident is your prediction?

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



33

©2015-2016 Emily Fox & Carlos Guestrin

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)

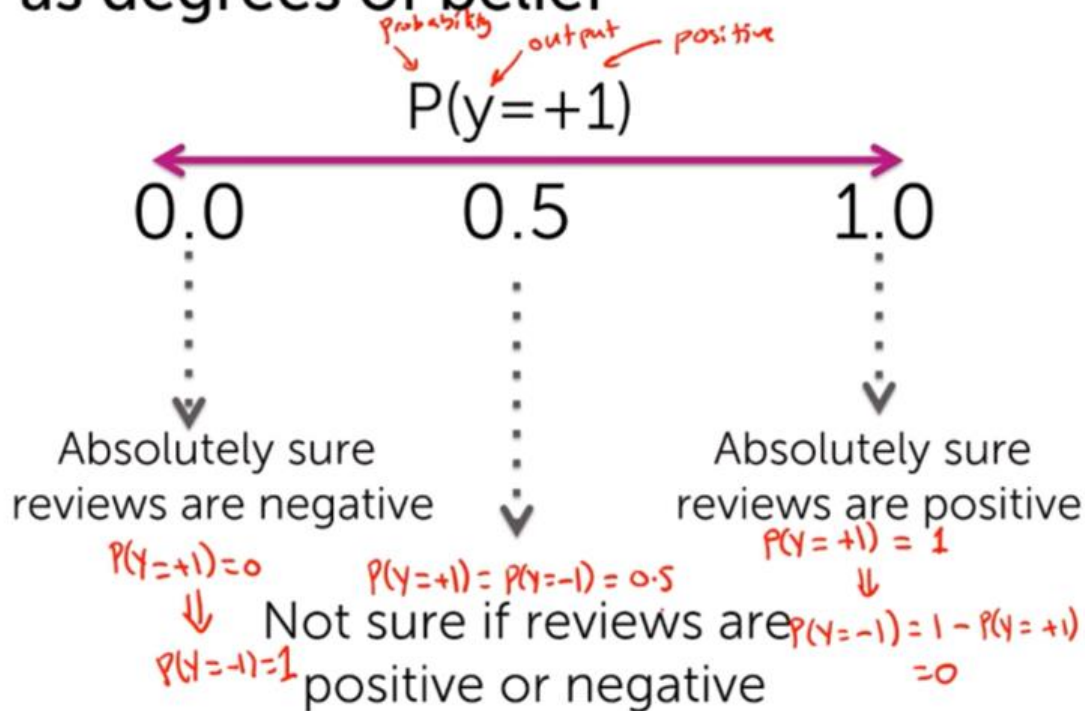
36

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

From the above it is assumed on an average,
70% → +ve reviews and the
remaining 30% → -ve reviews

Interpreting probabilities as degrees of belief



37

©2015-2016 Emily Fox & Carlos Guestrin

$P(y=+1) \rightarrow 1$ [Complete +ve reviews], $P(y=-1) \rightarrow 0$ [No negative reviews]

$P(y=-1) \rightarrow 1$ [Complete -ve reviews], $P(y=+1) \rightarrow 0$ [No positive reviews]

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	$0 \leq P(y=+1) \leq 1$ $0 \leq P(y=-1) \leq 1$	$0 \leq P(y=\text{dog}) \leq 1$ $0 \leq P(y=\text{cat}) \leq 1$ $0 \leq P(y=\text{bird}) \leq 1$
Probabilities sum up to 1	$P(y=+1) + P(y=-1) = 1$	$P(y=\text{dog}) + P(y=\text{cat}) + P(y=\text{bird}) = 1$

Conditional Probability

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)

39

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

The given condition is that ("3 awesome and 1 awful"). In this given condition, it is observed that 90% of the reviews are +ve and the remaining 10% are -ve.

week-1 logistic-regression-model-annotated.pdf - Drawboard PDF

Interpreting conditional probabilities

Output label: positive
Input sentence: "All the sushi was delicious!"

Probability: $P(y=+1|x_i = \text{"All the sushi was delicious!"})$

0.0 0.5 1.0

Absolutely sure review "All the sushi was delicious!" is negative
 $P(y=-1|x_i) = 1$
 $P(y=+1|x_i) = 1 - P(y=-1|x_i) = 1 - 1 = 0$
 $P(y=-1|x_i) = 0$

Not sure if review "All the sushi was delicious!" is positive or negative
 $P(y=+1|x_i) = P(y=-1|x_i) = 0.5$

Absolutely sure review "All the sushi was delicious!" is positive
 $P(y=+1|x_i) = 1$
 $P(y=-1|x_i) = 1 - P(y=+1|x_i) = 1 - 1 = 0$
 $P(y=-1|x_i) = 0$

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

20:28 02-07-2020

Key properties of conditional probabilities



Property	Two class (e.g., y is +1 or -1, x_i is review text)	Multiple classes (e.g., y is dog, cat or bird, x_i is image)
Conditional probabilities always between 0 & 1	$0 \leq P(y=+1 x_i) \leq 1$ $0 \leq P(y=-1 x_i) \leq 1$	$0 \leq P(y=\text{dog} x_i) \leq 1$ $0 \leq P(y=\text{cat} x_i) \leq 1$ $0 \leq P(y=\text{bird} x_i) \leq 1$
Conditional probabilities sum up to 1 over y , but not over x	$P(y=+1 x_i) + P(y=-1 x_i) = 1$ But $\sum_x P(y=+1 x) \neq 1$ $\sum_{i=1}^N P(y=+1 x_i) \neq 1$	$P(y=\text{dog} x_i) + P(y=\text{cat} x_i) + P(y=\text{bird} x_i) = 1$

41

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

Probabilities used in Classification

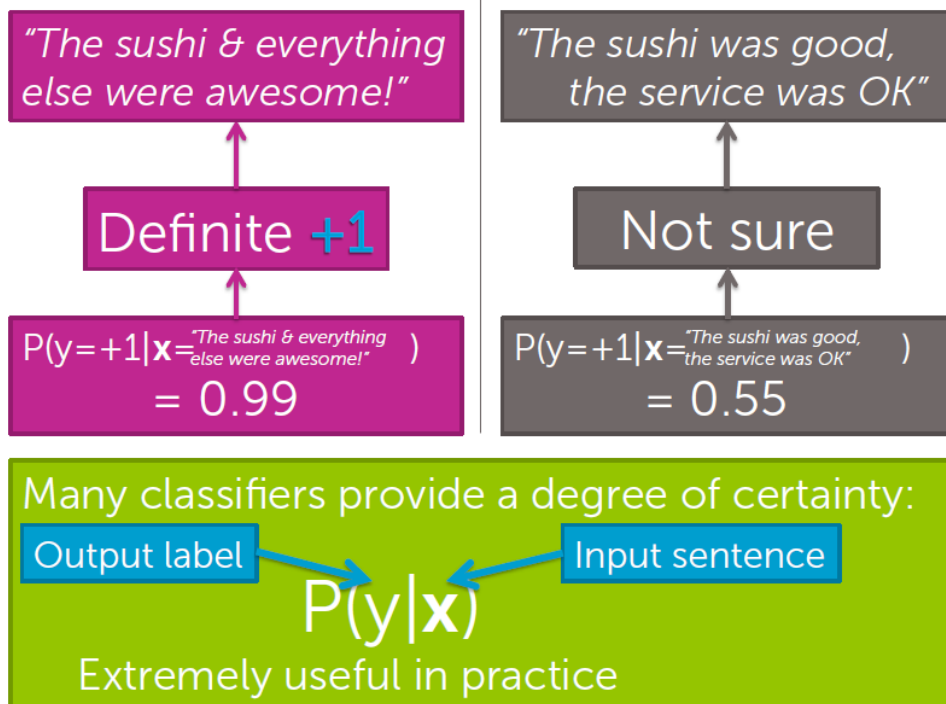
Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

\nearrow
 given

$P(A|B) \rightarrow$ The probability of How much A is in B

How confident is your prediction?



44

©2015-2016 Emily Fox & Carlos Guestrin

Goal: Learn conditional probabilities from data

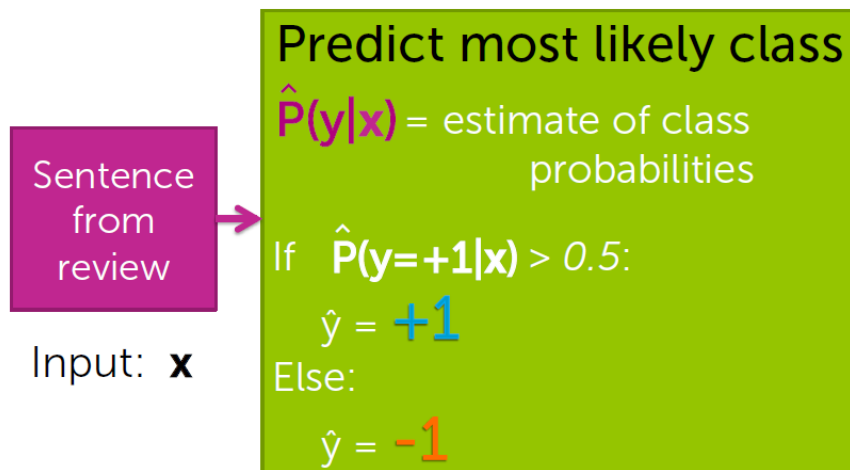
Training data: N observations (\mathbf{x}_i, y_i)

$x[1] = \text{\#awesome}$	$x[2] = \text{\#awful}$	$y = \text{sentiment}$
2	1	+1
0	2	-1
3	3	-1
4	1	+1
...

Optimize **quality metric**
on training data

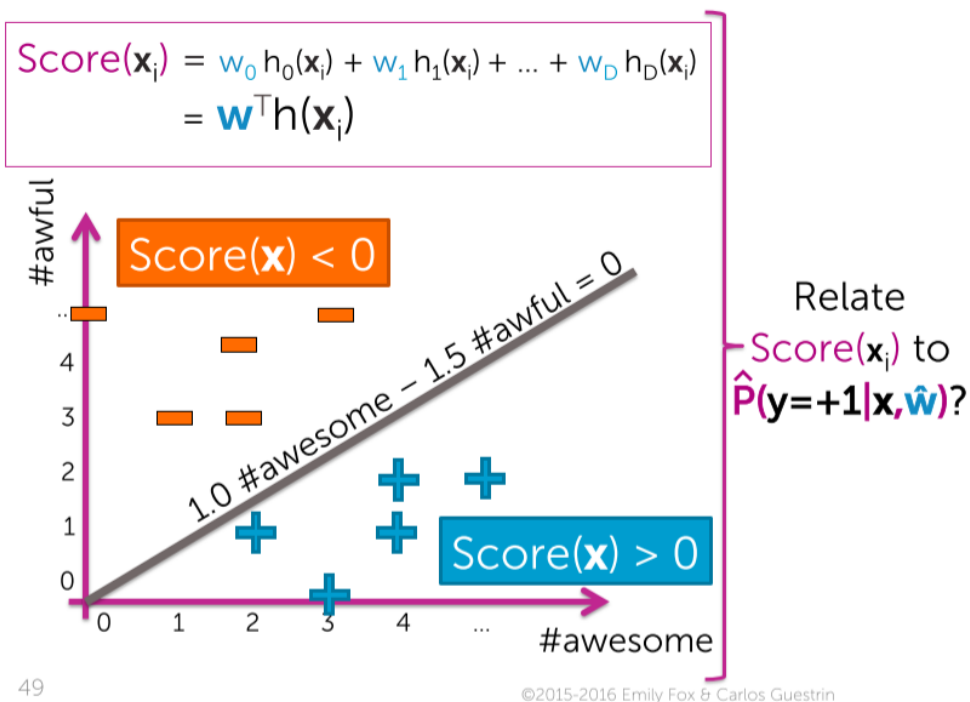
Find best model $\hat{\mathbf{p}}$
by finding best $\hat{\mathbf{w}}$

Useful for
predicting \hat{y}



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

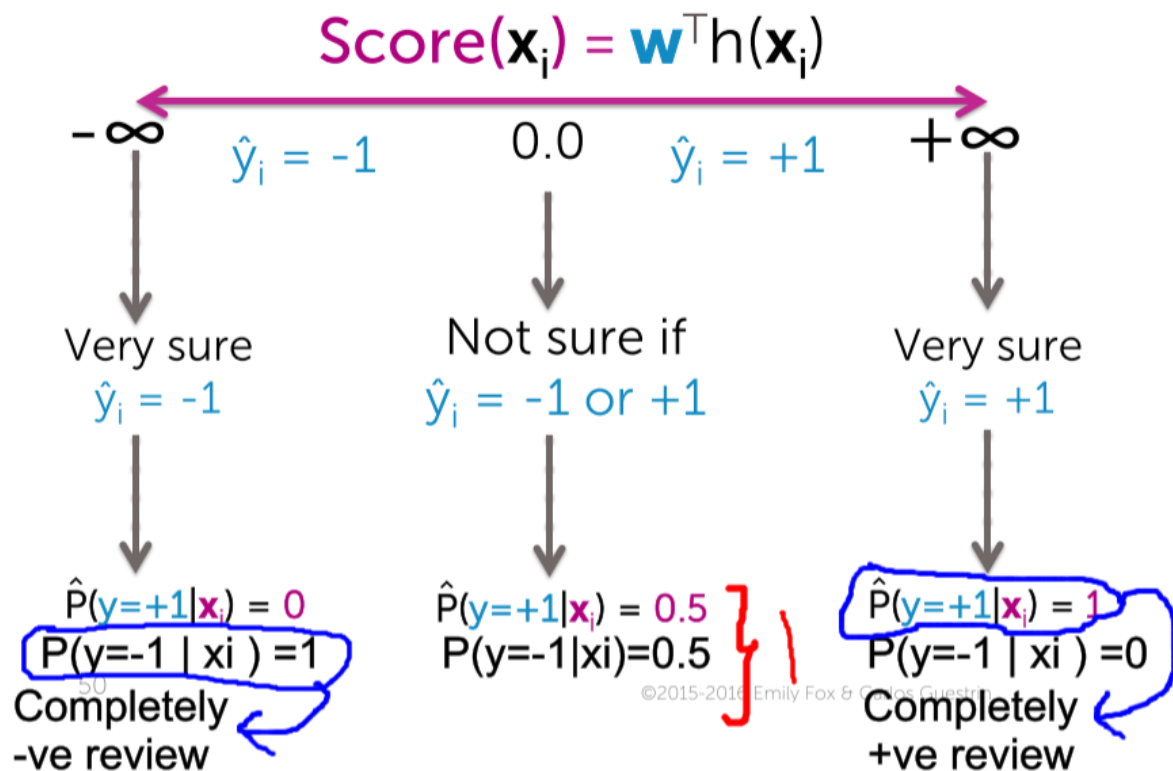
Logistic Regression



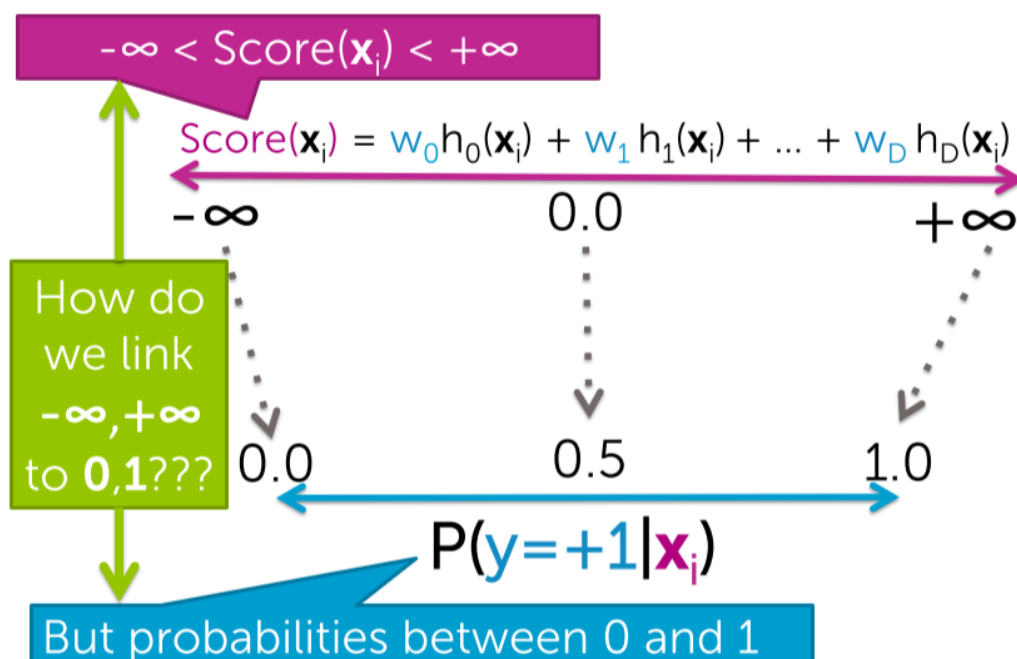
We know below the line the score(\mathbf{x})>0 and above the line the score(\mathbf{x})<0. But we don't know how much far is it less/greater than 0

1) Predicting the class probabilities with (generalized) linear model

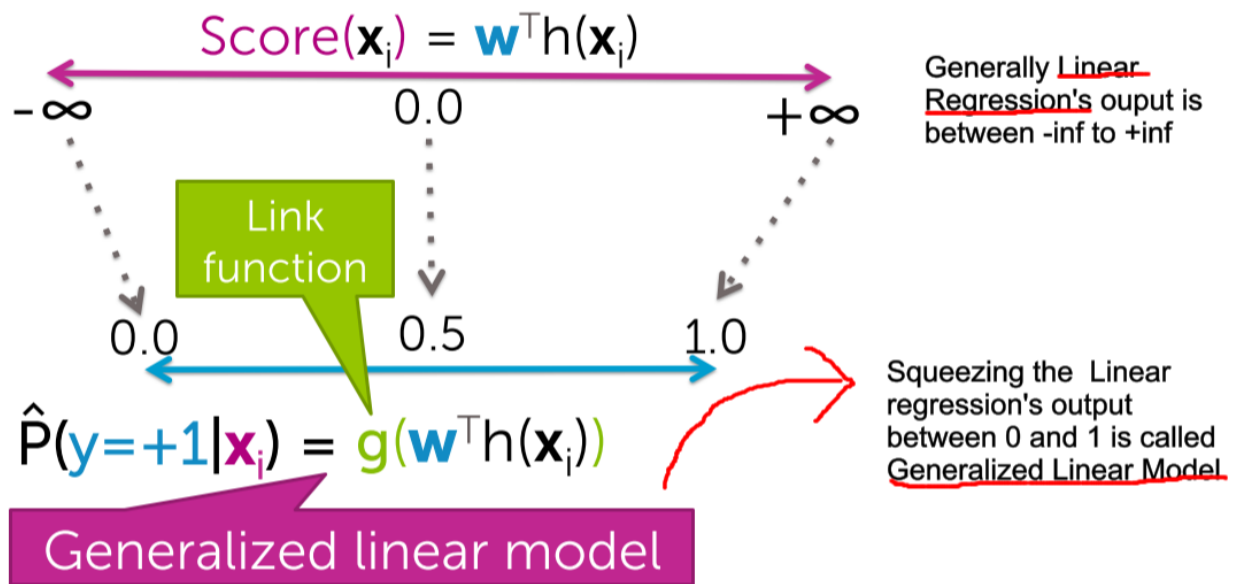
Interpreting $\text{Score}(\mathbf{x}_i)$



Why not just use regression to build classifier?



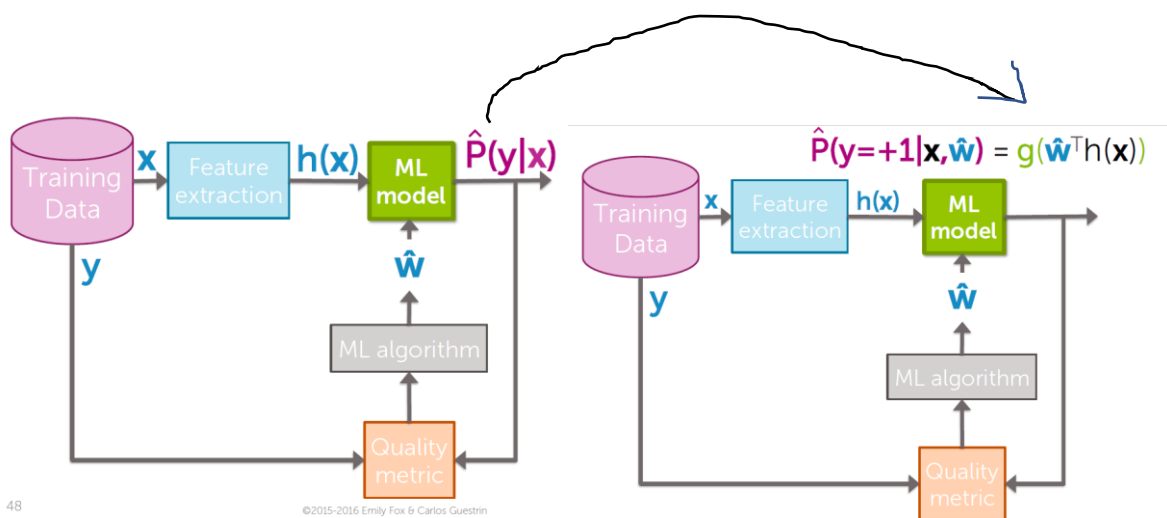
Link function: squeeze real line into [0,1]



52

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization



2) The sigmoid(logistic) Link function

Logistic function (sigmoid, logit)

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

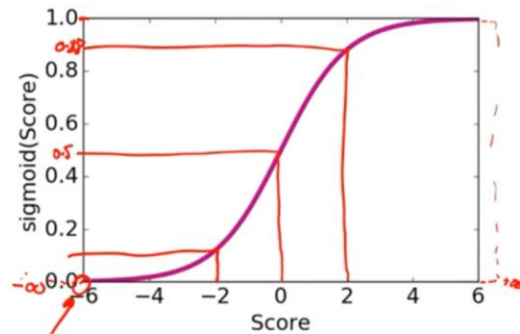
input

Score	$-\infty$	-2	0.0	+2	$+\infty$
sigmoid(Score)	$\frac{1}{1+e^{\infty}}$ $= \frac{1}{1+\infty}$ $= 0$	0.12	$\text{Sigmoid}(0)$ $= \frac{1}{1+e^0}$ $= \frac{1}{1+1}$ $= 0.5$	0.88	$\frac{1}{1+e^{-\infty}}$ $= 1$

$$e^{\infty} = \infty$$

$$e^0 = 1$$

$$e^{-\infty} = 0$$



56

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

We are giving the score whose output is $(-\infty \text{ to } +\infty)$

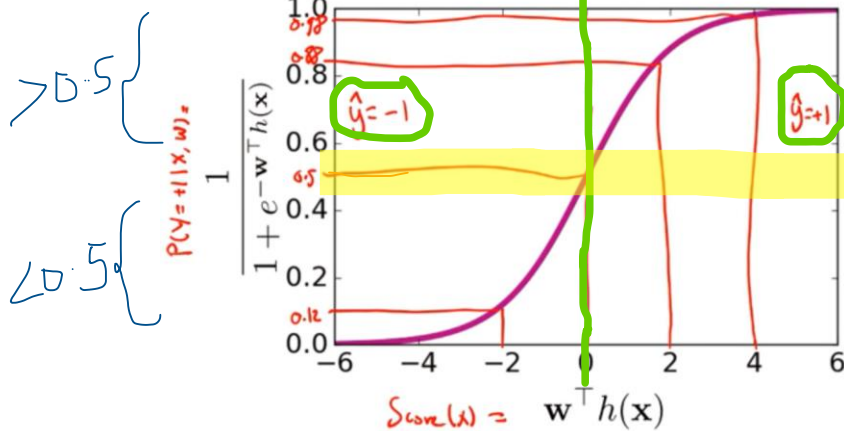
Wants to change the output $(0 \text{ to } +\infty)$

This is done using Link function which is Sigmoid function

2) Logistic Regression Model

Understanding the logistic regression model

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



-ve reviews

+ve reviews

Score(\mathbf{x}_i)	$P(y=+1 \mathbf{x}_i, \mathbf{w})$
0	0.5
-2	$0.12 < 0.5 \Rightarrow \hat{y} = -1$
2	$0.88 \Rightarrow \hat{y} = +1$
4	$0.98 \Rightarrow \hat{y} = +1$

58

©2015-2016 Emily Fox & Carlos Guestrin

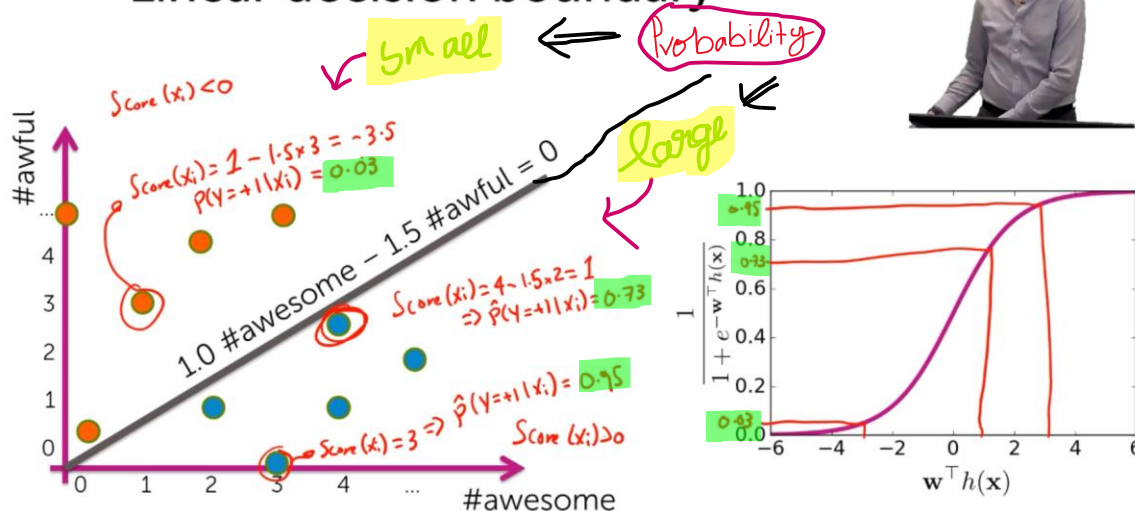
Machine Learning Specialization

Taking Score(x-axis) and Probability(y-axis)

Effect of coefficient values on predicted probabilities

Logistic regression →

Linear decision boundary



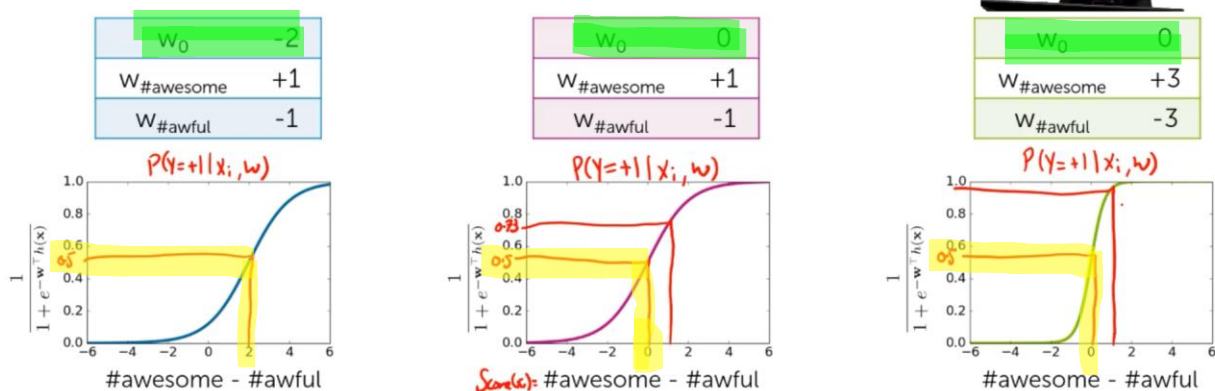
59

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

4) Effect of coefficient on Logistic Regression Model

Effect of coefficients on logistic regression model



Case-2

Case-1

Case-3

60

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

Case-1

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

$\text{Score}(x_i) \Rightarrow w_0 + w_{\# \text{ awesome}} + w_{\# \text{ awful}} + \text{constant}$
 $\Rightarrow +1 - 1 + 0$

$\text{Score}(x_i) \Rightarrow 0$

$f(x) \Rightarrow \frac{1}{1 + e^{-(\text{score}(x_i))}}$
 $\Rightarrow \frac{1}{1 + e^0} \Rightarrow \frac{1}{2} \Rightarrow 0.5$

$f(x) \Rightarrow 0.5$

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

Increasing # of awesome

$\text{Score}(x_i) \Rightarrow 2 - 1 + 0 \Rightarrow 1$

$f(x) \Rightarrow \frac{1}{1 + e^{-1}} \Rightarrow \frac{1}{1.3678} \Rightarrow 0.73$

Case-2

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

$\text{Score}(x_i) \Rightarrow w_0 + w_{\# \text{ awesome}} + w_{\# \text{ awful}} + w_0$
 $\text{Score}(x_i) \Rightarrow 2$

Now if my make the $\text{Score}(x_i) \Rightarrow 2$, then only
 I will get probability $\Rightarrow 0.5$

Case-3

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

$\text{Score}(x_i) \Rightarrow w_0 + w_{\# \text{ awesome}} + w_{\# \text{ awful}} + w_0$
 $\Rightarrow 3 - 3 \Rightarrow 0$

$f(x) \Rightarrow 0.5$

Increasing # of awesome

Then the probability $\Rightarrow 0.9$ (nearly 1)

Conclusion

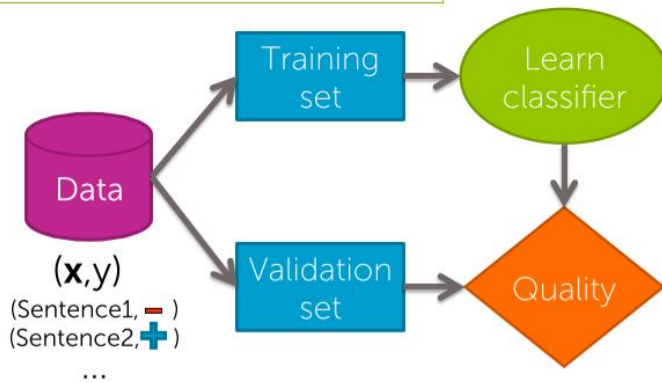
- See the effect of coefficients affecting the probability. We can conclude if the model has some bigger coefficients then the probabilities can be found more quickly.
- Changing the constant will shift the line to the left and to the right.

Now we want to find the coefficients that best fits

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
...

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



The data is splitted up into training and validation set.

In training set, running a learning algorithm and output the parameter w^\wedge .

This w^\wedge is fitted into the model, to estimate the probability that the input sentence either +ve (or) -ve.

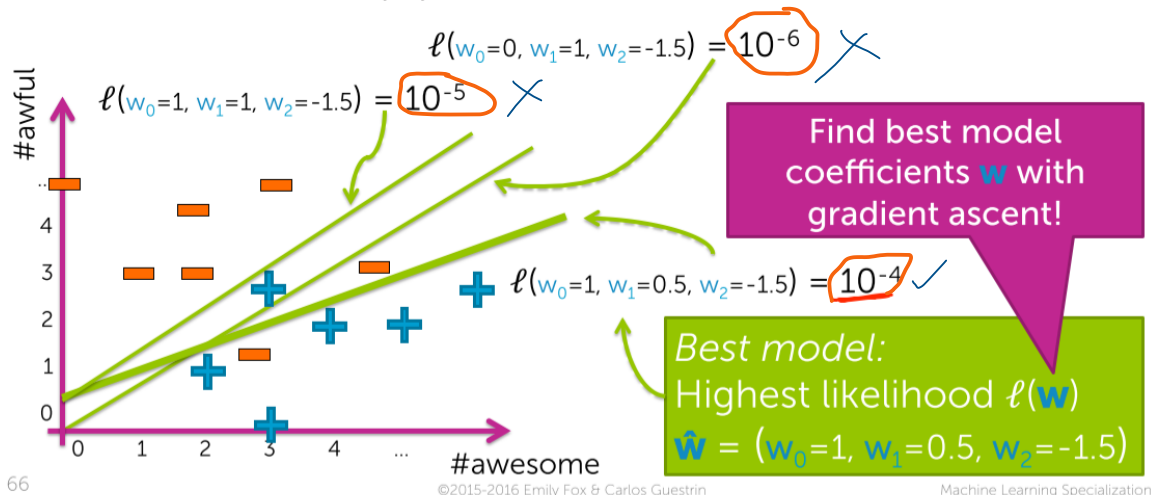
Now we can take the validation set and fit into the model . And can predict the quality metric, error etc..

How to choose w^* ????

Find "best" classifier =

Maximize quality metric over all possible w_0, w_1, w_2

Likelihood $\ell(w)$



Quality metric \rightarrow likelihood function $\ell(w)$

We are saying among the three $10^{-4} \rightarrow$ best likelihood

(Since for the best classifier \rightarrow Maximise the likelihood $\ell(w)$)

We want the required w^* , in which $\ell(w) \rightarrow$ maximum. So in-order to choose that w^* gradient ascent comes to the picture.

Categorical inputs

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

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

- Categorical inputs:



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

Basically in numerical data,

in score function → we will multiply the numeric with the coefficient.

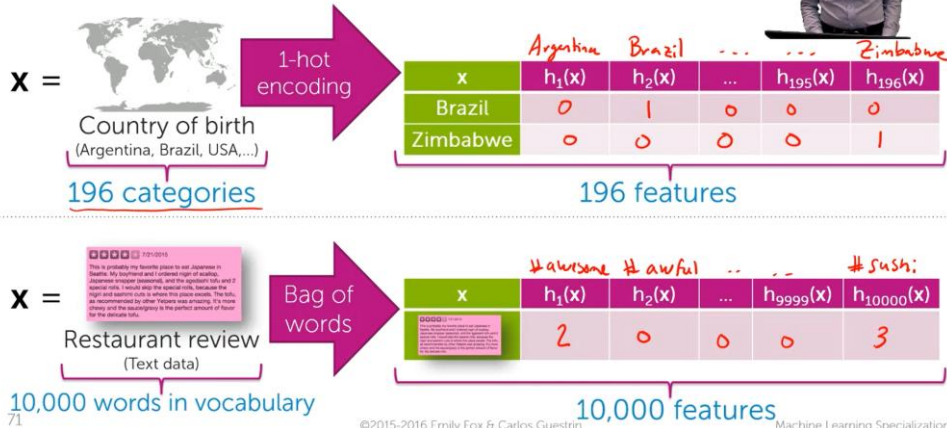
The zip-code is 10005, 98195 etc. It is not meant that 98195 is 9 times of 10005. These are different postal codes representing different parts of the country. Hence they are not numerical features, they are categorical features.

But how to multiply the coefficient with the categorical values???

This is achieved by **encoding technique**

Encoding Categorical inputs

Encoding categories as numeric features



If somebody is born in Brazil → then for Brazil put 1 and for everything put 0

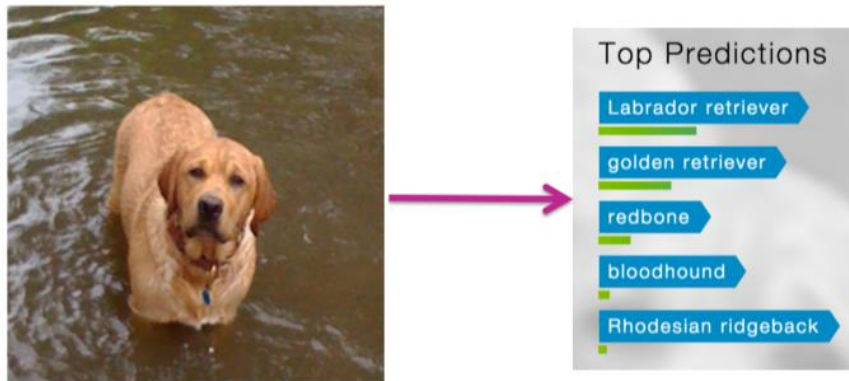
If somebody is born in Zimbabwe → then for Zimbabwe put 1 and for everything put 0

How to encode a restaurant's review???

Take 1 review, and put down the word count in the respective places
In the above case, 1st review contains (2 → awesome), (0 → awful), ..., (3 → sushi)

Multiclass classification (1 versus all)

Multiclass classification



Input: x
Image pixels

Output: y
Object in image

In this image \rightarrow there is a dog

Our aim is to predict 1) Whether is it a dog ?? 2) What kind of dog is it???

Here we are not having only 2 categories. There are nearly 1000's of categories.

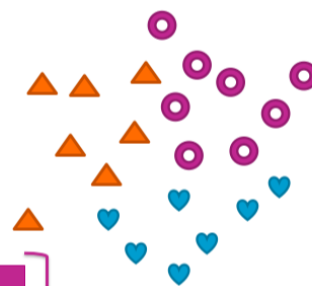
How to solve this??? \rightarrow **One versus ALL**

Eg: Triangle, donut and hearts are different classes

Multiclass classification formulation

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

Data point	$x[1]$	$x[2]$	y
x_1, y_1	2	1	▲
x_2, y_2	0	2	♥
x_3, y_3	3	3	⊙
x_4, y_4	4	1	⊙



Learn:

$$\hat{P}(y = \text{▲} | x)$$

$$\hat{P}(y = \text{♥} | x)$$

$$\hat{P}(y = \text{⊙} | x)$$

There are 3 classes,

What I need to know is for a particular input, whether is it a triangle, hearts or donut.

Now I want to classify the triangle from the rest

1 versus all:

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

+1 class: points with $y_i = \triangle$
-1 class: points with $y_i = \heartsuit$ OR \odot

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

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

More likely to be \triangle

Score(x_i) > 0
 \Downarrow
 $P(y=\triangle | x_i, w) > 0.5$



Not more likely to be a \triangle

Score(x_i) ≤ 0
 $P(y=\triangle | x_i, w) < 0.5$

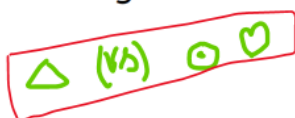
77

©2015-2016 Emily Fox & Carlos Guestrin

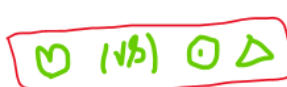
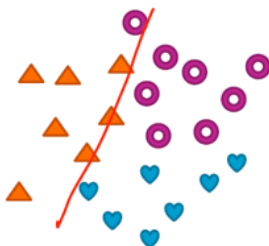
Machine Learning Specialization

Now we are going to make a classifier to learn \rightarrow that separates the triangle from the donuts and hearts. This train classifier outputs +1, if the input x is more likely to be a triangle.

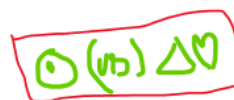
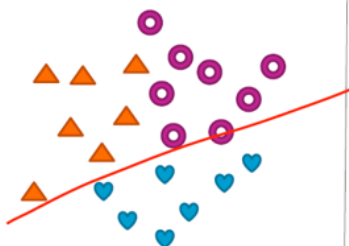
1 versus all: simple multiclass classification using C 2-class models



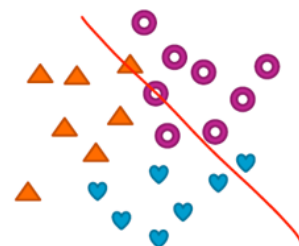
$$\hat{P}(y=\triangle | x_i) = \hat{P}_\triangle(y=+1|x_i, w)$$



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



$$\hat{P}(y=\odot | x_i) = \hat{P}_\odot(y=+1|x_i, w)$$



78

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

In the same data-set

1st classifier → triangle . Note down the probability


2nd classifier → hearts. Note down the probability

3rd classifier → donot . Note down the probability

Among these three classifier , capture a classifier which has the highest probability. For eg: If the 2nd classifier (i.e heart classifier) has the highest probability means that classifier (i.e w (heart)) make the classification more exact and accurate.

Take the dog classification example

Iterate over all the classifier and finally note which classifier has the highest probability and that classifier classifies correctly when compared to the rest of the classifier.



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 $c = 1, \dots, C$:
If $\hat{P}_c(y=+1|\mathbf{x}_i) > \text{max_prob}$:
 $\hat{y} = c$
 max_prob = $\hat{P}_c(y=+1|\mathbf{x}_i)$

