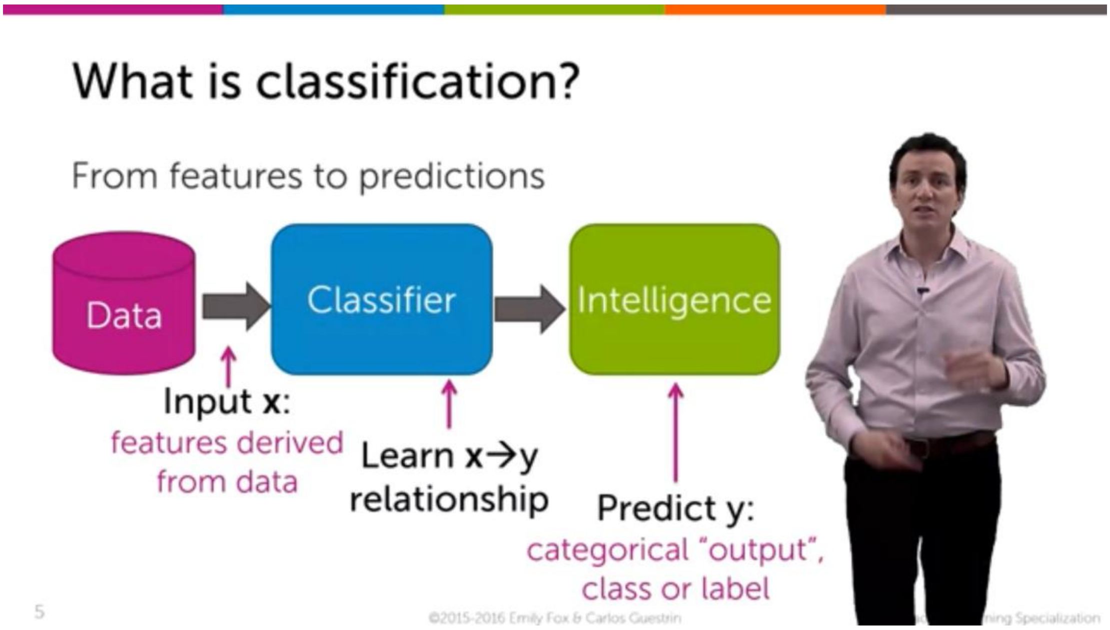
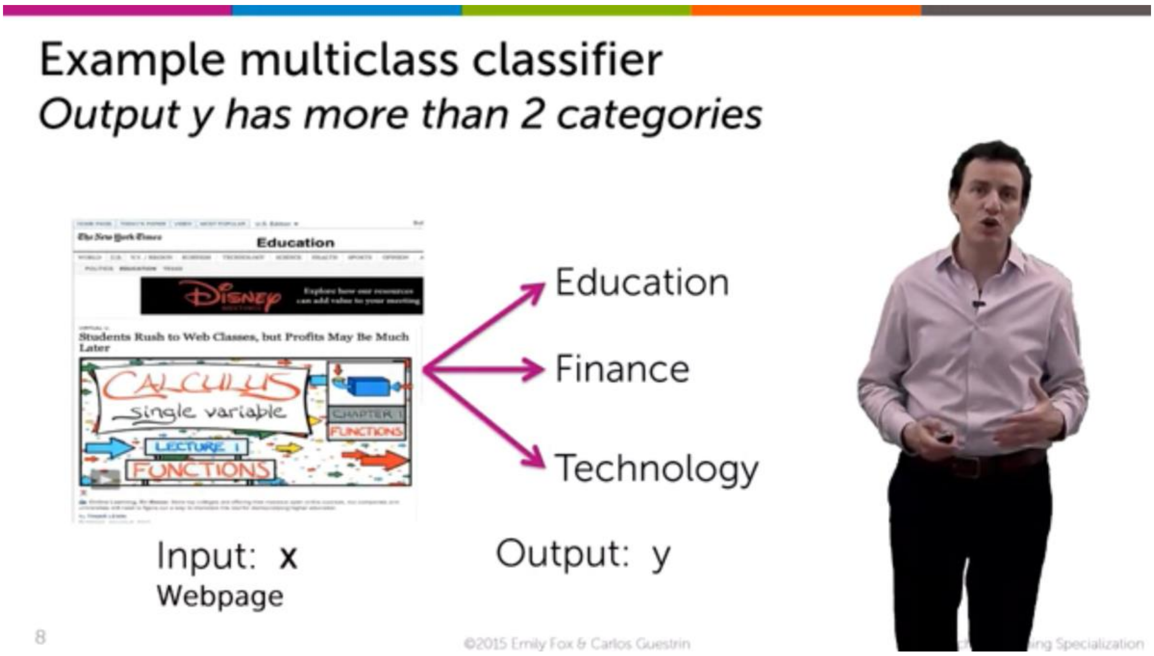
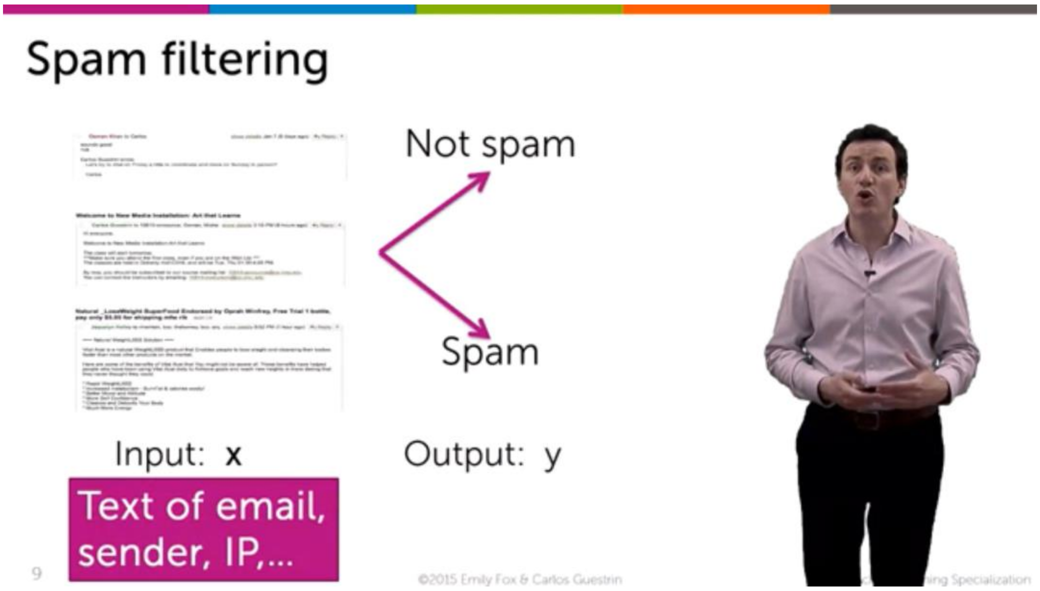


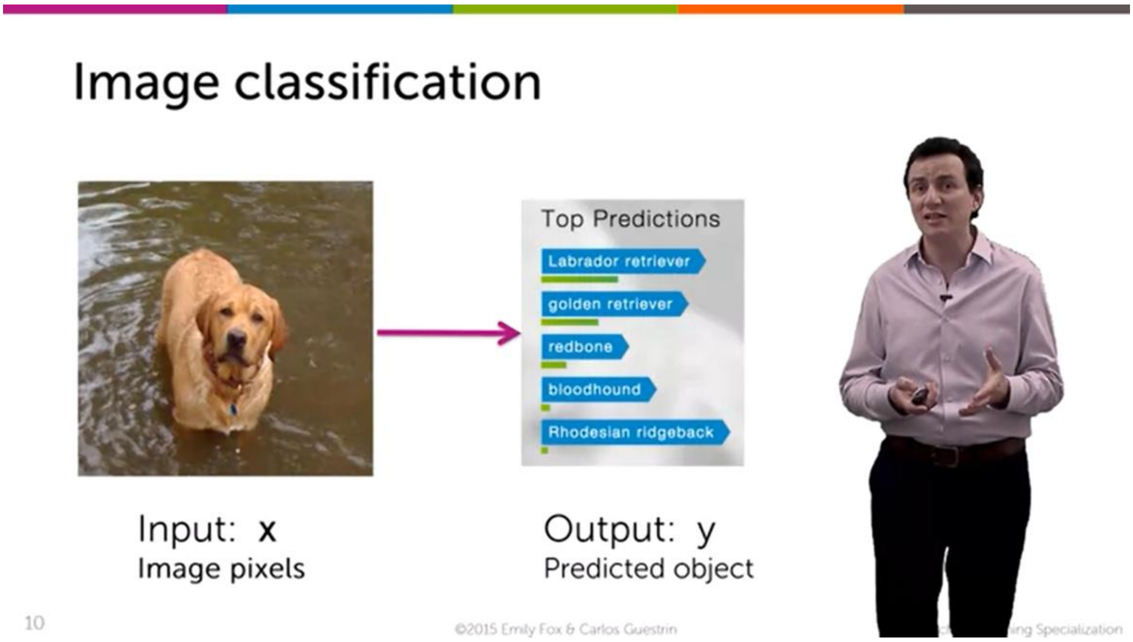
Multi-class classification

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

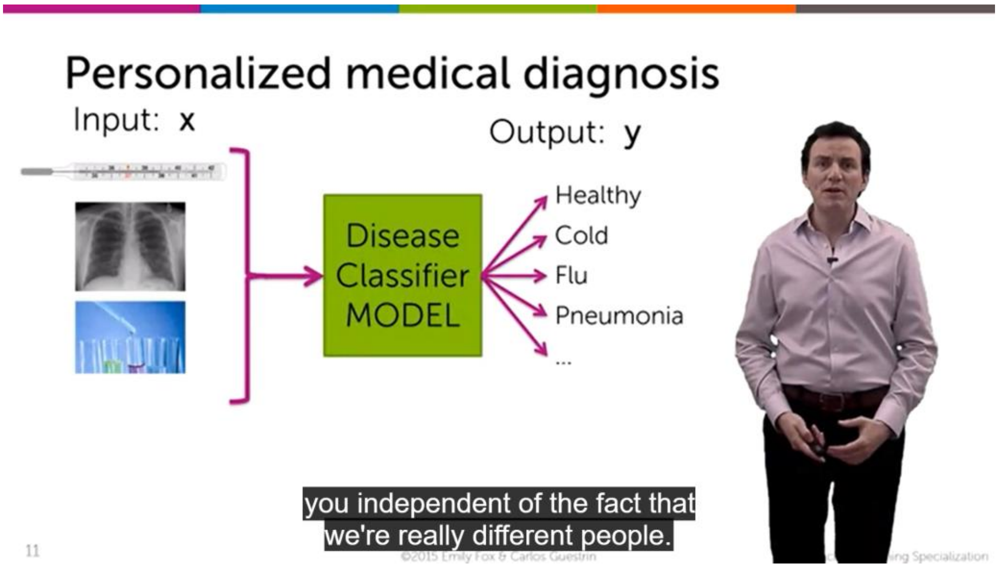


Famous Example : Spam filtering





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

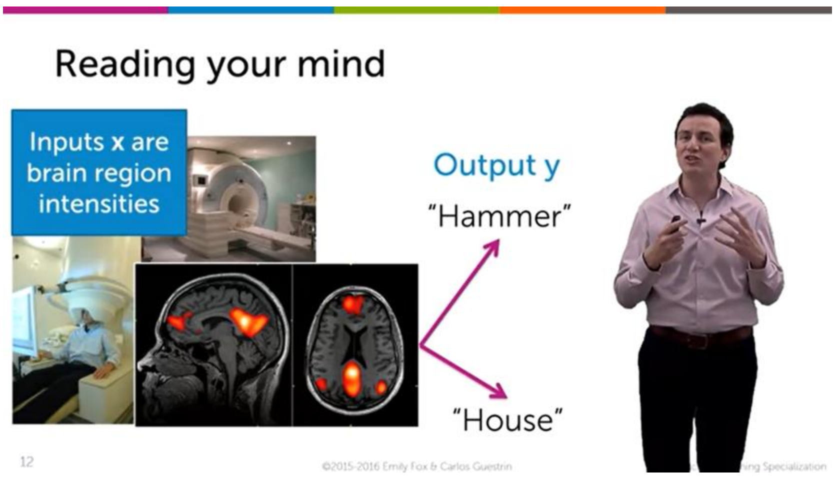


But personalized medical diagnosis tests our DNA, food-habits, genetical problems, total body metabolism etc to decide what treatment is going to be most effective for that individual person instead a giving a routine flat general treatment.

This is the real world example of classification problem.

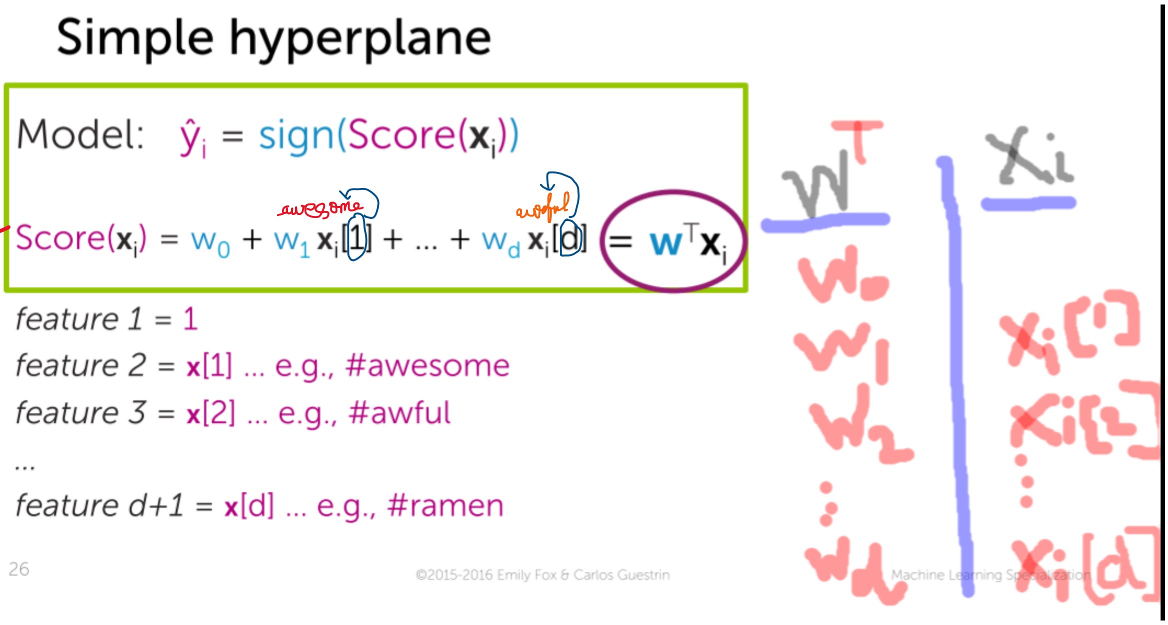
Mind-Reading :

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



For a single row representing → xi
Score for that single row → score(xi)
Sign of the score of that single row → y^(i) = Sign(score(xi))

i from 1 to N (row wise)   [xi]

The Role of *sign*

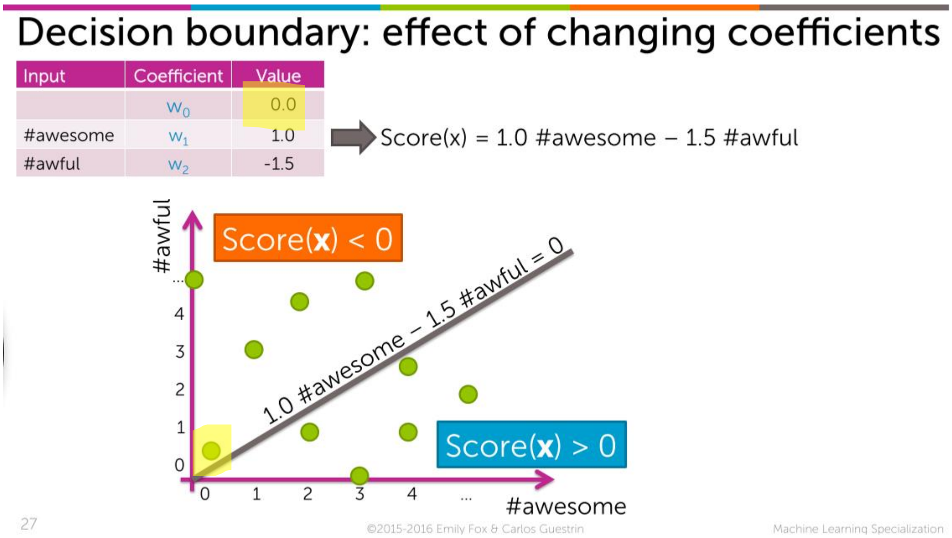
If the score>0 then predict → +1

If the score<0 then predict → -1

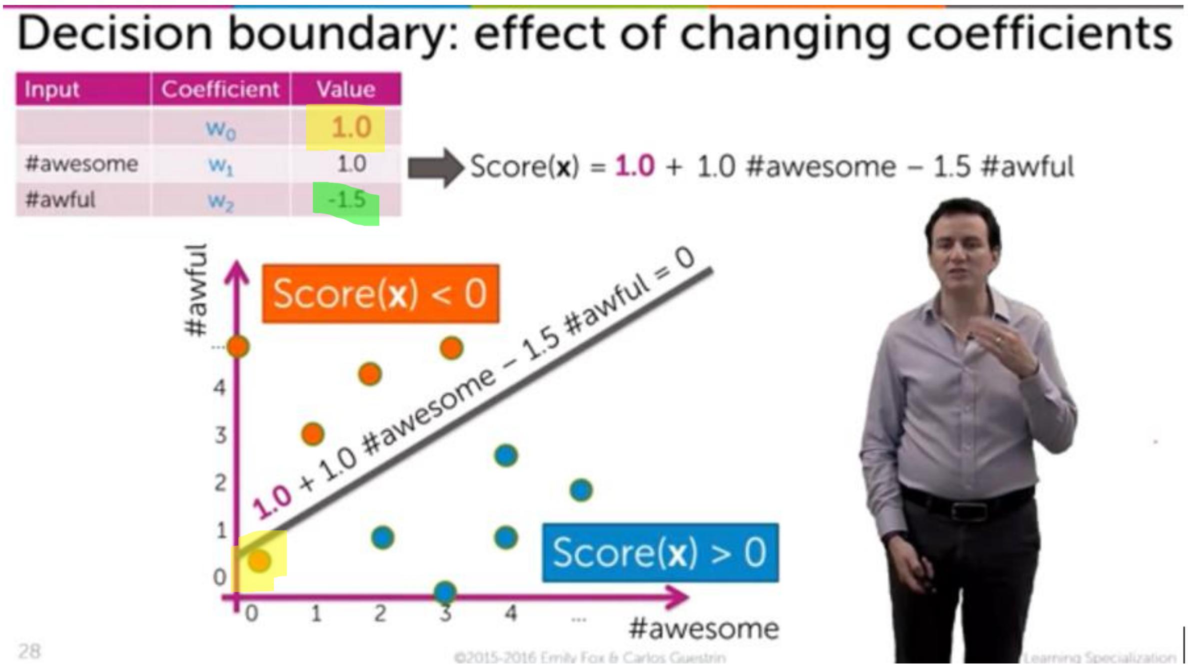
At 0, we have the choice to predict either -1/+1. You make an arbitrary choice

2) Effects of coefficient values on decision boundary

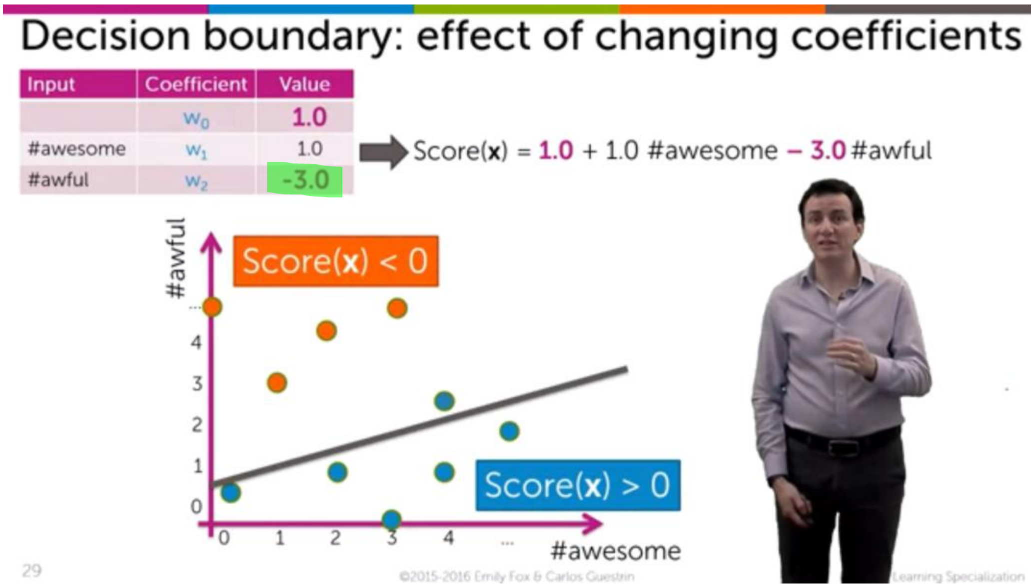
Initially the intercept → 0



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

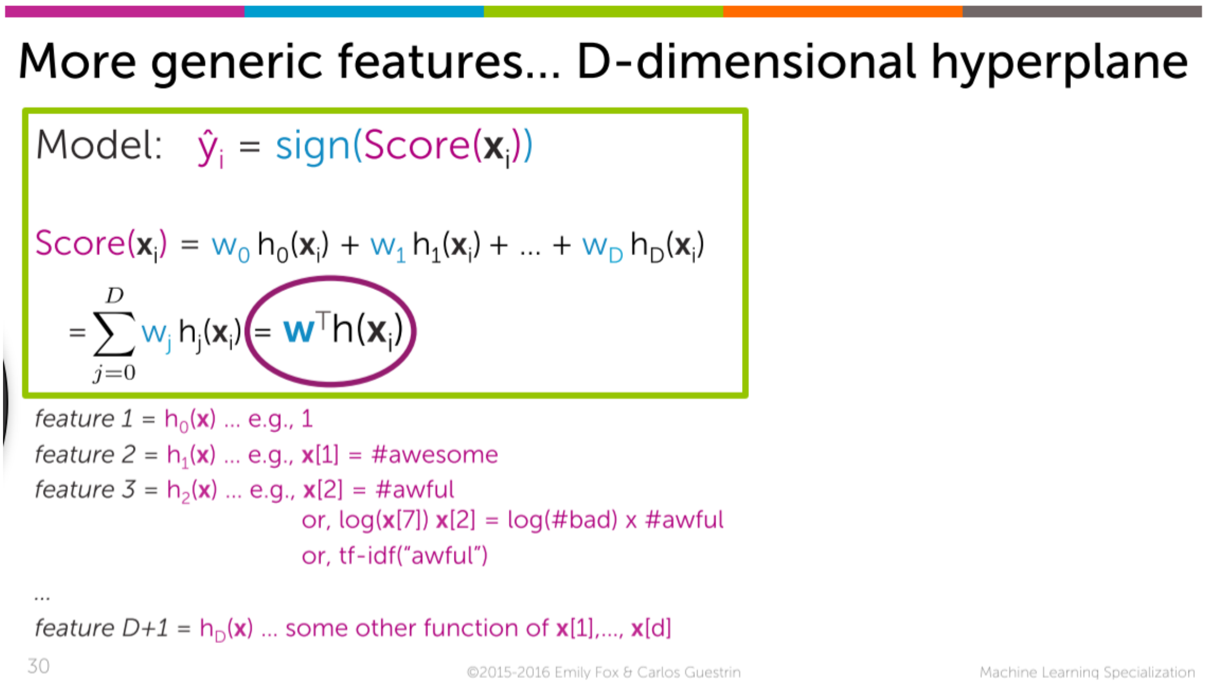


After changing the w2=-3.0, the line gets modified, and the blue point which was +ve Now becomes -ve.



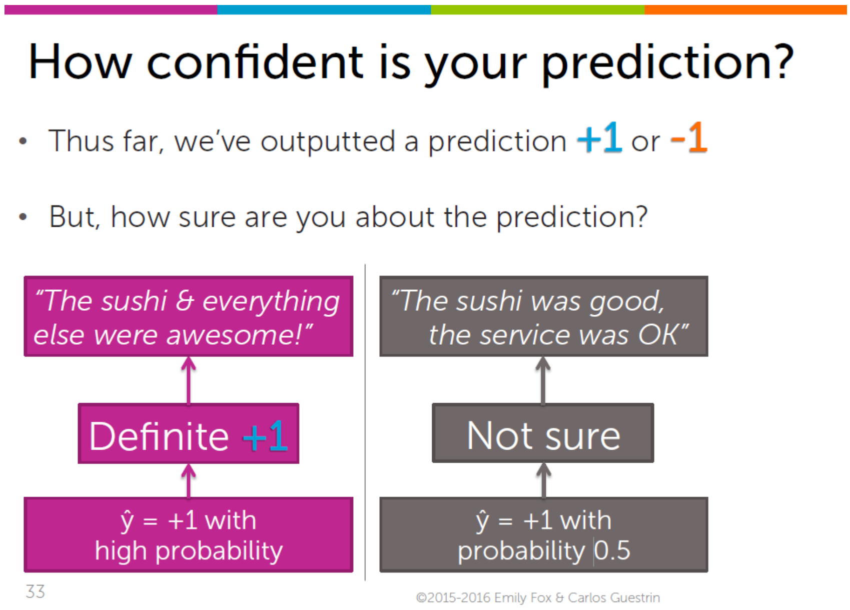
From this we can conclude that the coefficients are playing a very important role in the classification.

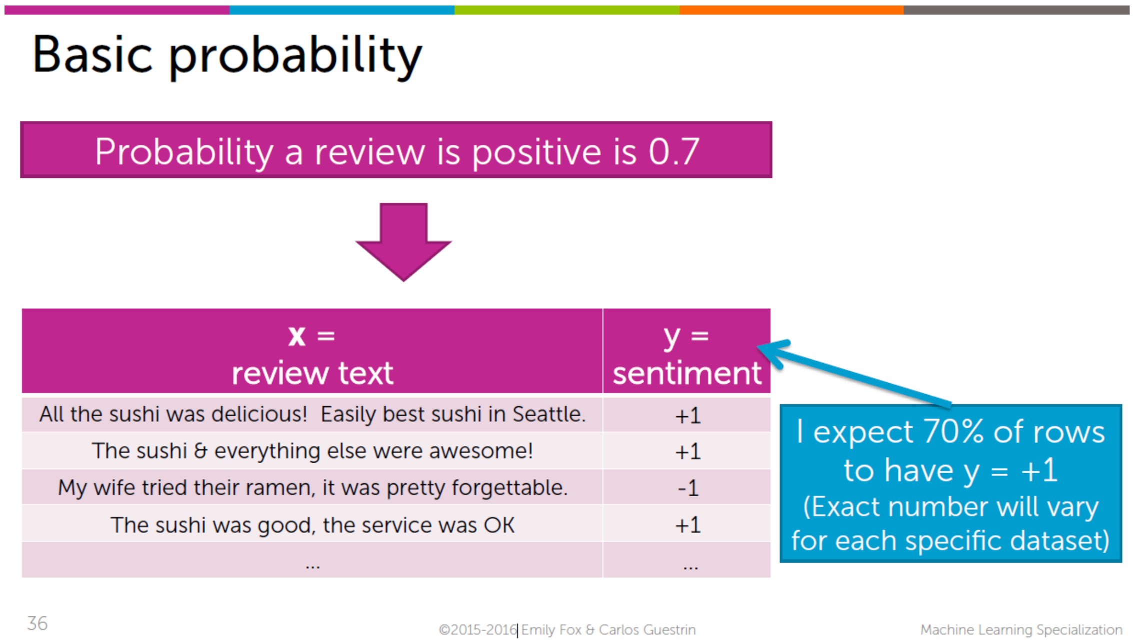
3) Using features of inputs



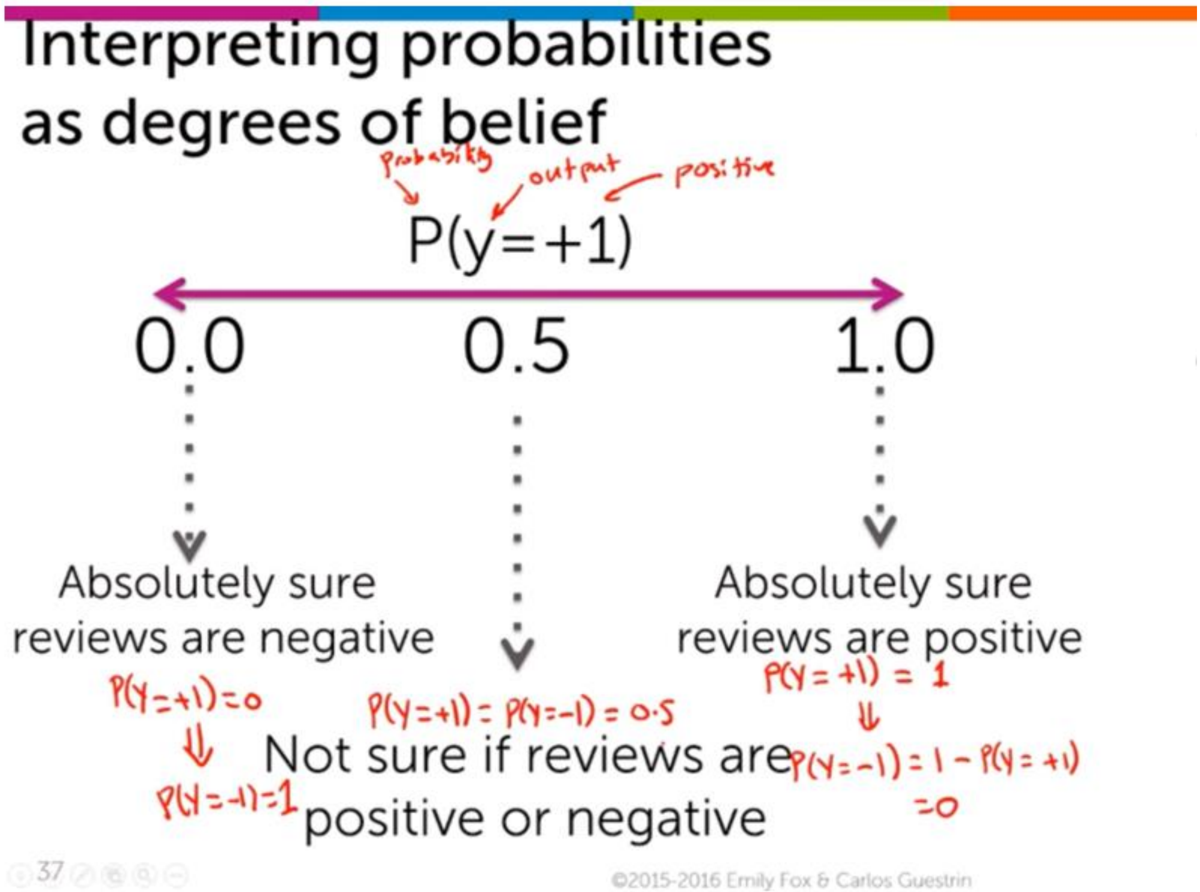
Probabilities and its basics

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)



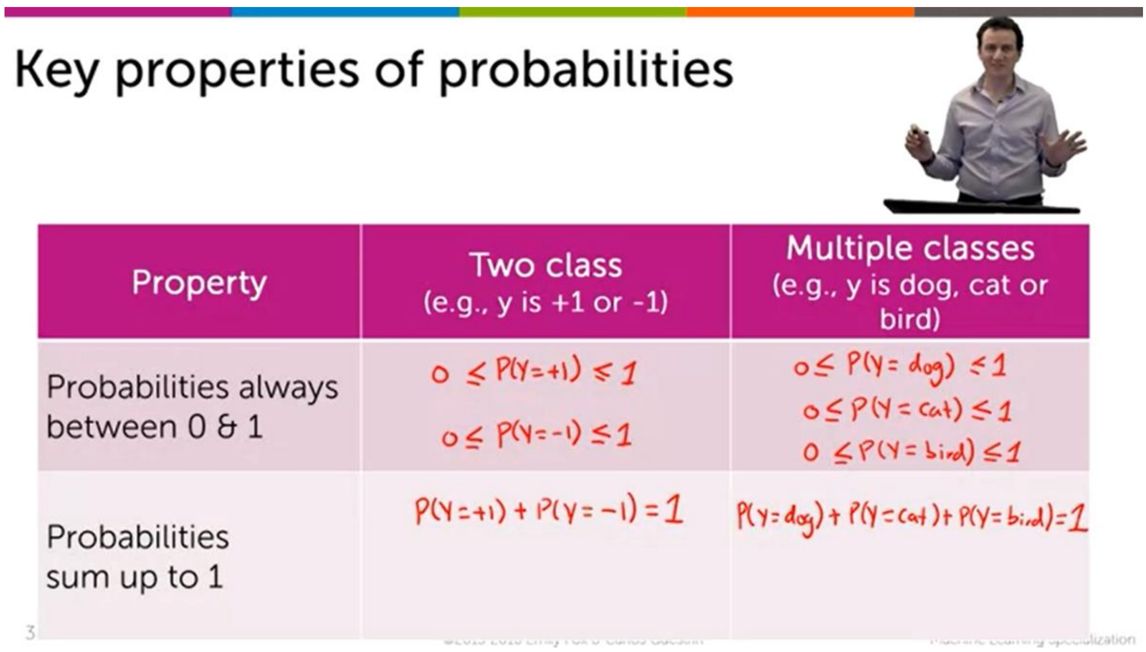


From the above it is assumed on an average,
70% → +ve reviews and the
remaining 30% → -ve reviews



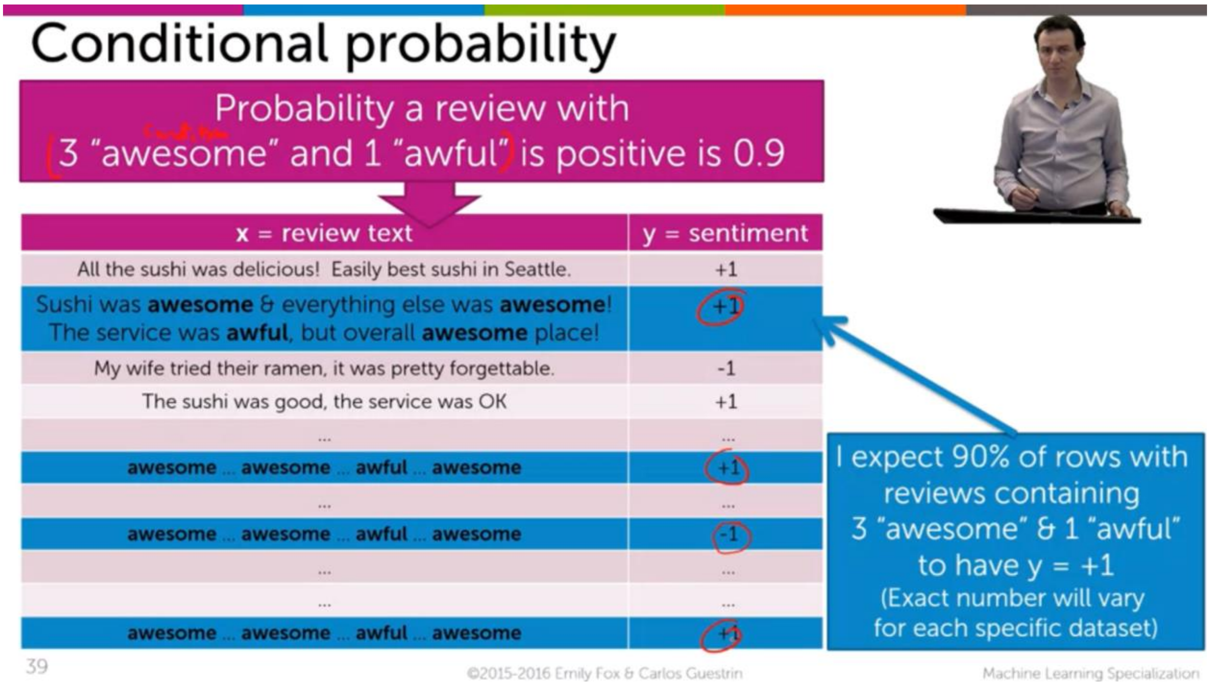
P(y=+1)->1[Complete +ve reviews],P(y=-1)→0 [No negative reviews]

P(y=-1)->1[Complete -ve reviews],P(y=1)→0 [No positive reviews]

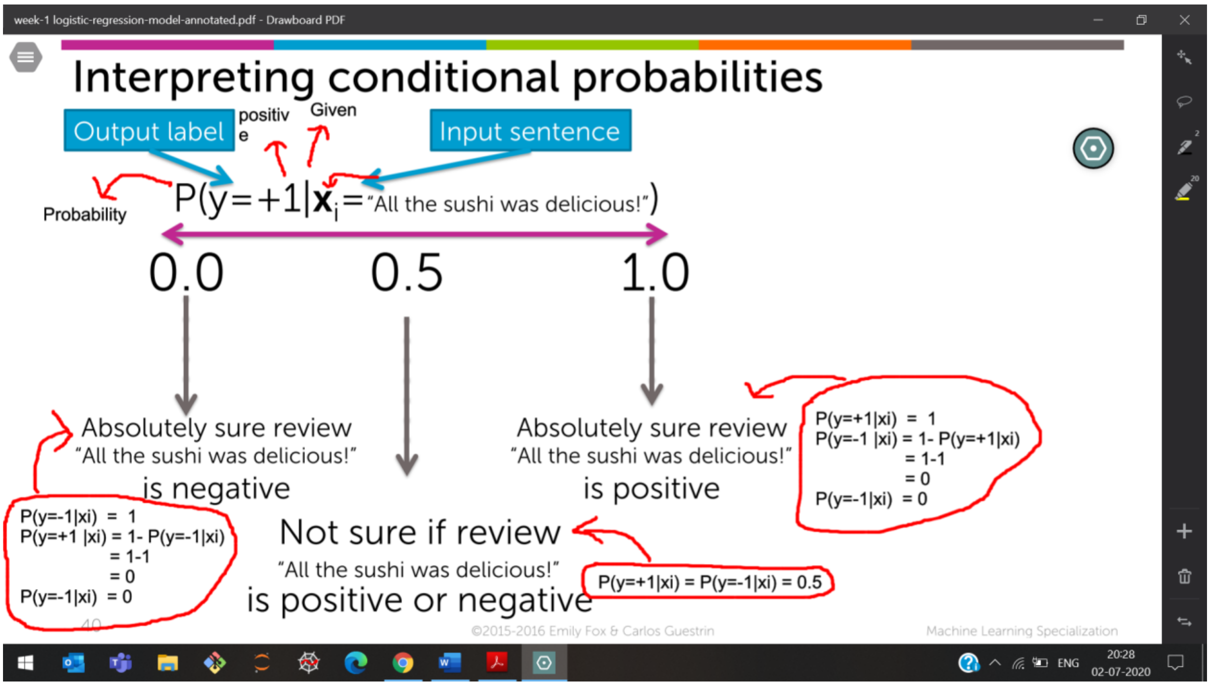


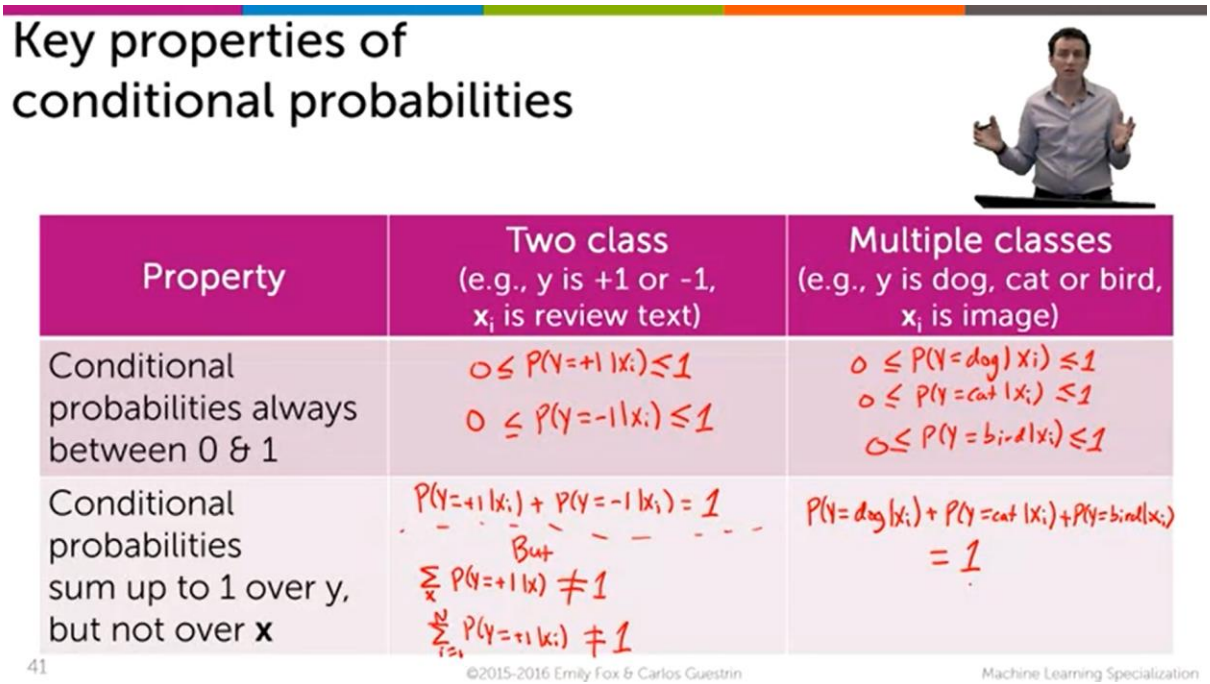
Property	Two class (e.g., y is +1 or -1)	Multiple classes (e.g., y is dog, cat or bird)
Probabilities always between 0 & 1	$0 \le P(y=+1) \le 1$ $0 \le P(y=-1) \le 1$	$0 \le P(y=dog) \le 1$ $0 \le P(y=cat) \le 1$ $0 \le P(y=bird) \le 1$
Probabilities sum up to 1	$P(y=+1) + P(y=-1) = 1$	$P(y=dog) + P(y=cat)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

Conditional Probability



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.





Key properties of conditional probabilities

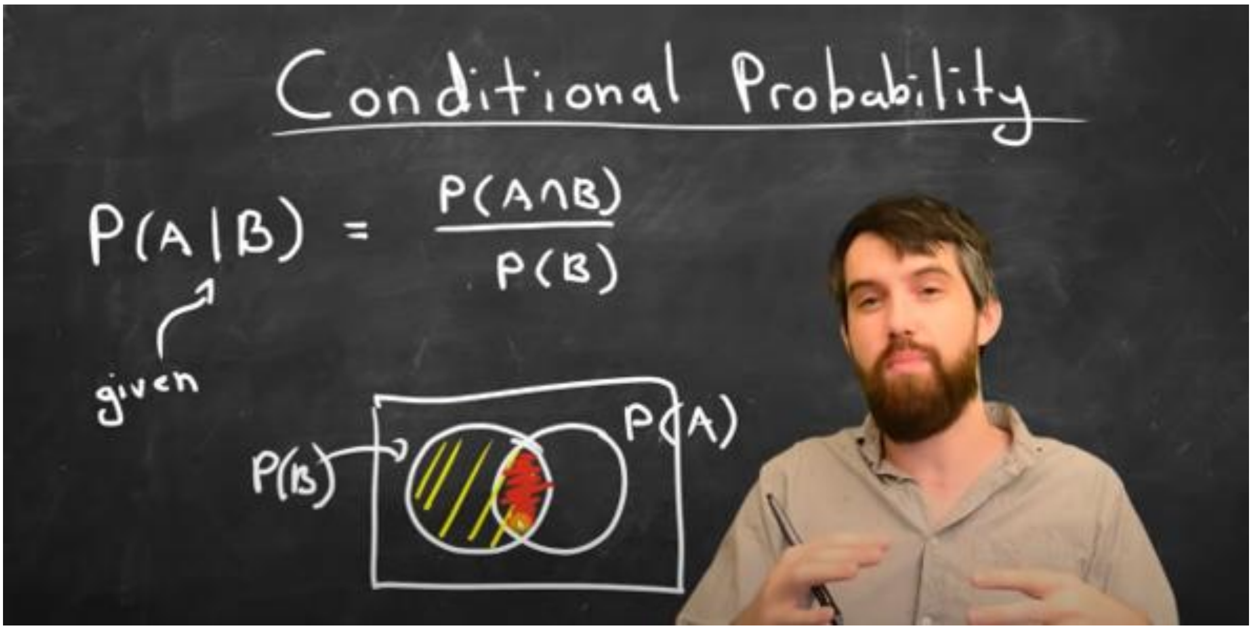
Property	Two class (e.g., y is +1 or -1, $\mathbf{x}_i$ is review text)	Multiple classes (e.g., y is dog, cat or bird, $\mathbf{x}_i$ is image)
Conditional probabilities always between 0 & 1	$0 \le P(y=+1 \mid x_i) \le 1$ $0 \le P(y=-1 \mid x_i) \le 1$	$0 \le P(y=dog \mid x_i) \le 1$ $0 \le P(y=cat \mid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

41

©2015-2016 Emily Fox & Carlos Guestrin

Machine Learning Specialization

Probabilities used in Classification



P(A|B) → The probability of How much A is in B

How confident is your prediction?

"The sushi & everything else were awesome!"

Definite +1

P(y=+1|**x**=*"The sushi & everything else were awesome!"*)
= 0.99

"The sushi was good, the service was OK"

Not sure

P(y=+1|**x**=*"The sushi was good, the service was OK"*)
= 0.55

Many classifiers provide a degree of certainty:

Output label → P(y|**x**) ← Input sentence

Extremely useful in practice

Goal: Learn conditional probabilities from data

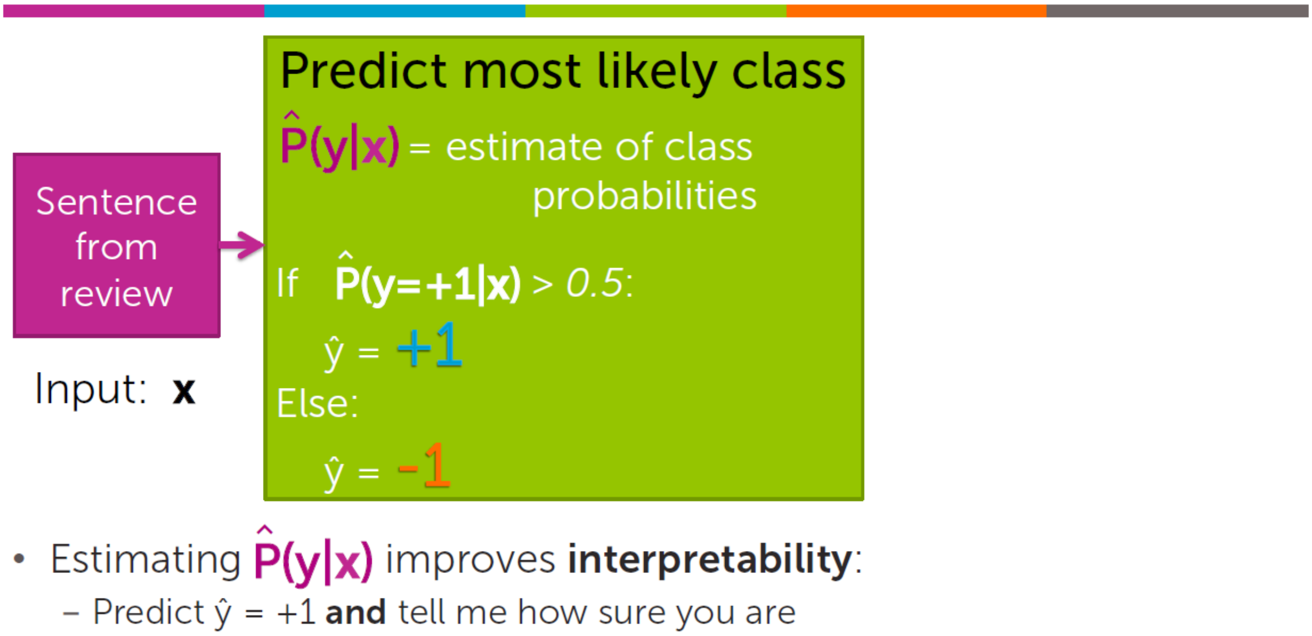
Training data: N observations ($\mathbf{x}_i$,y_i)

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

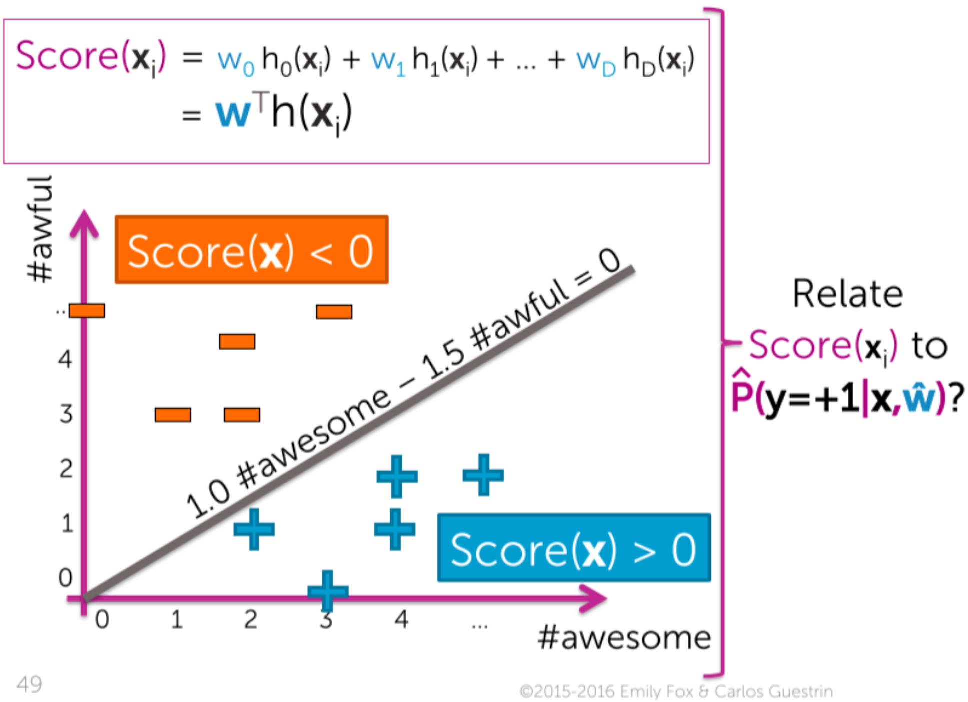
Optimize **quality metric** on training data

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

Useful for predicting $\hat{y}$

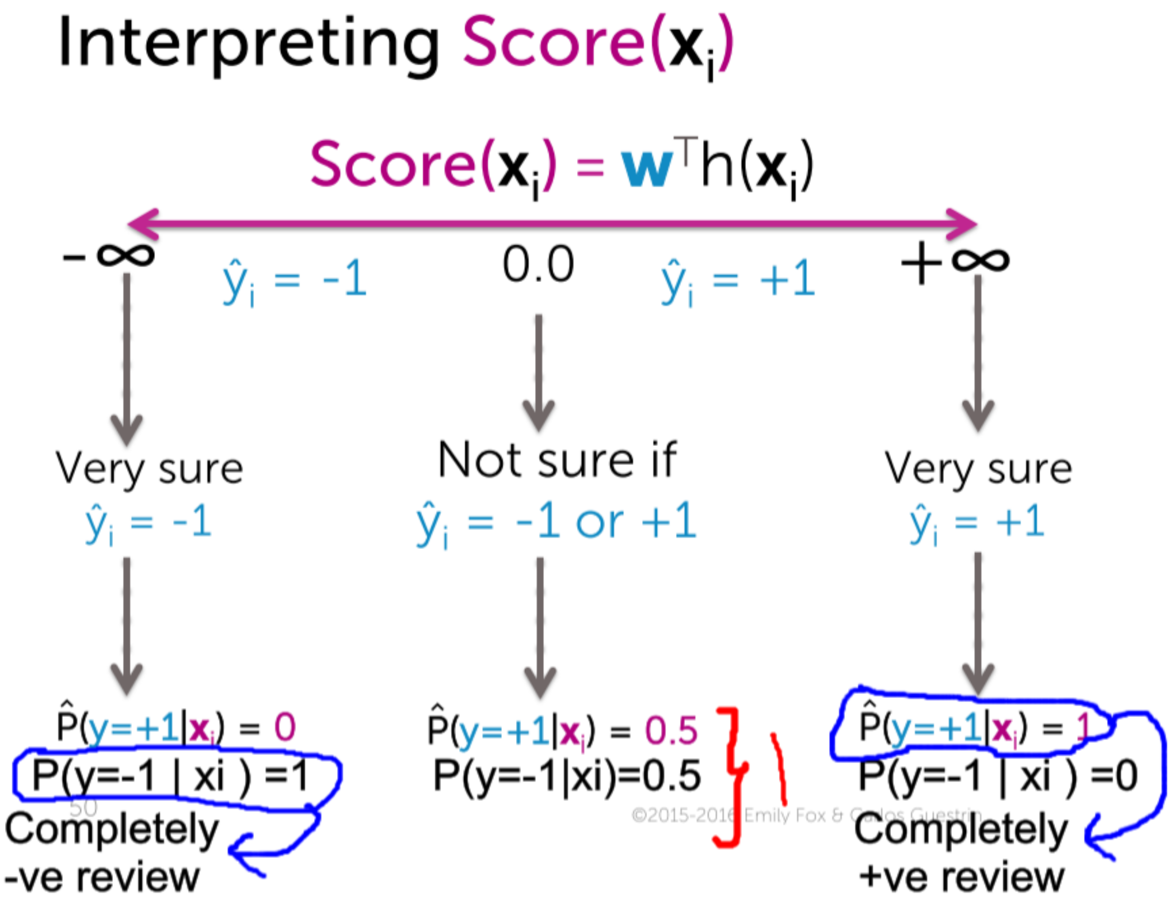


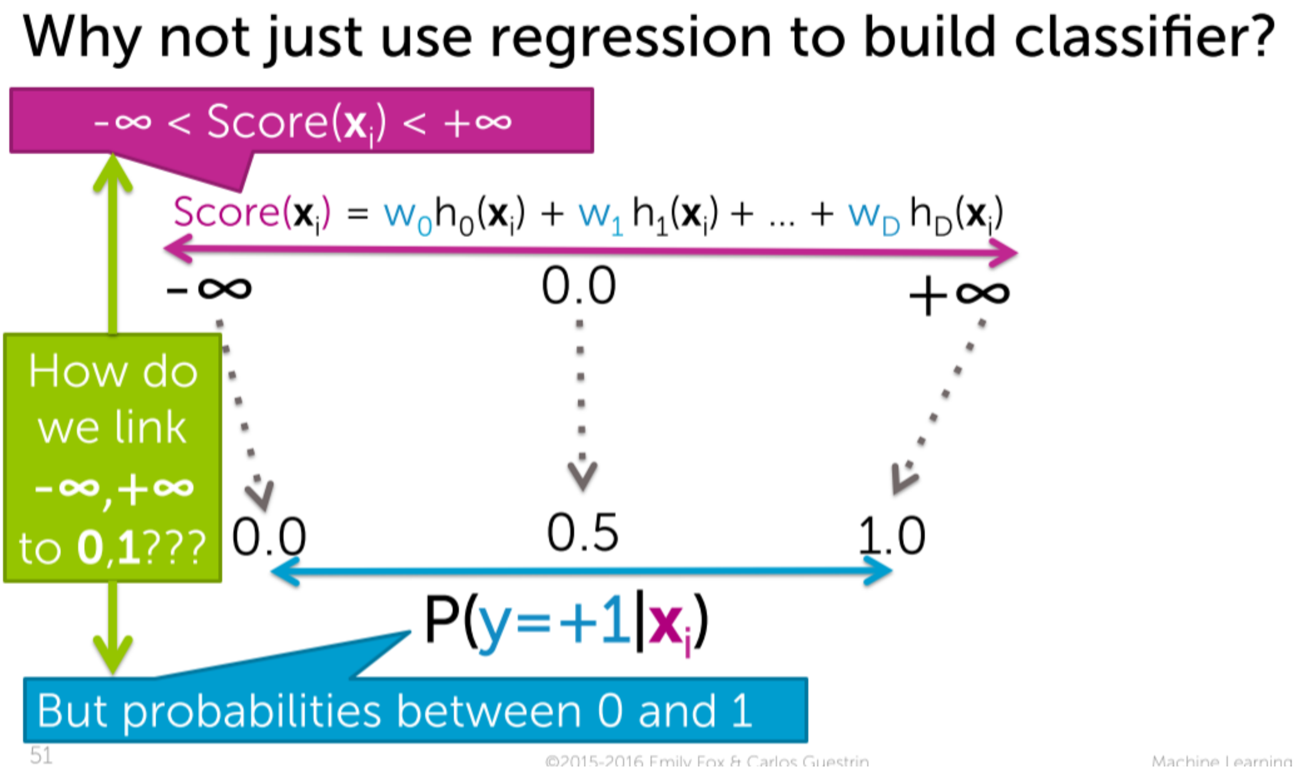
Logistic Regression



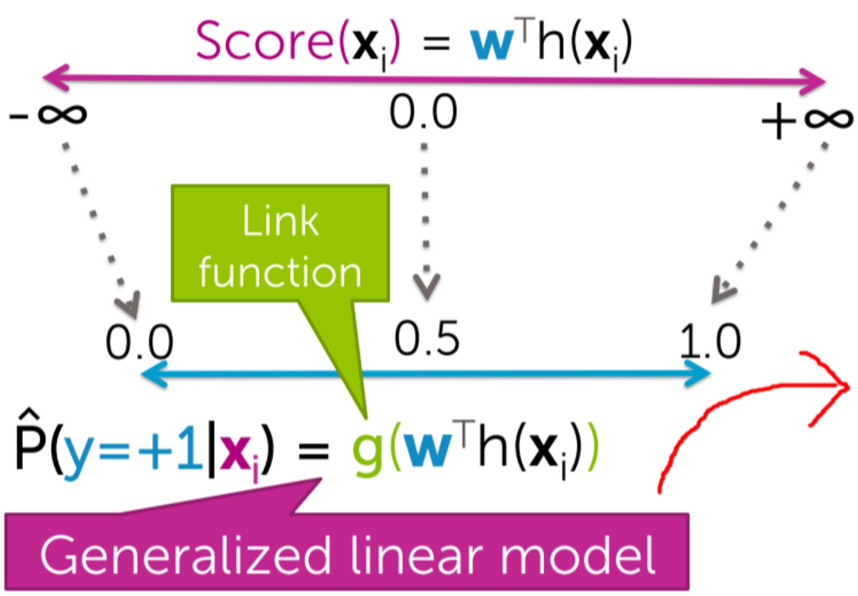
We know below the line the score(x)>0 and
above the line the score(x)<0. But we don’t know how much far is it less/greater than 0

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



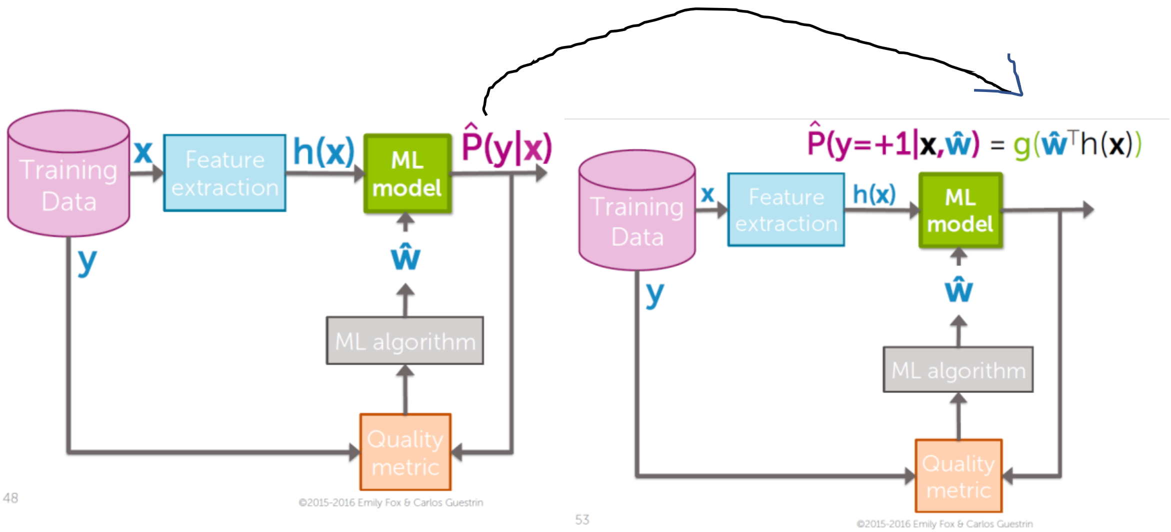


Link function: squeeze real line into [0,1]

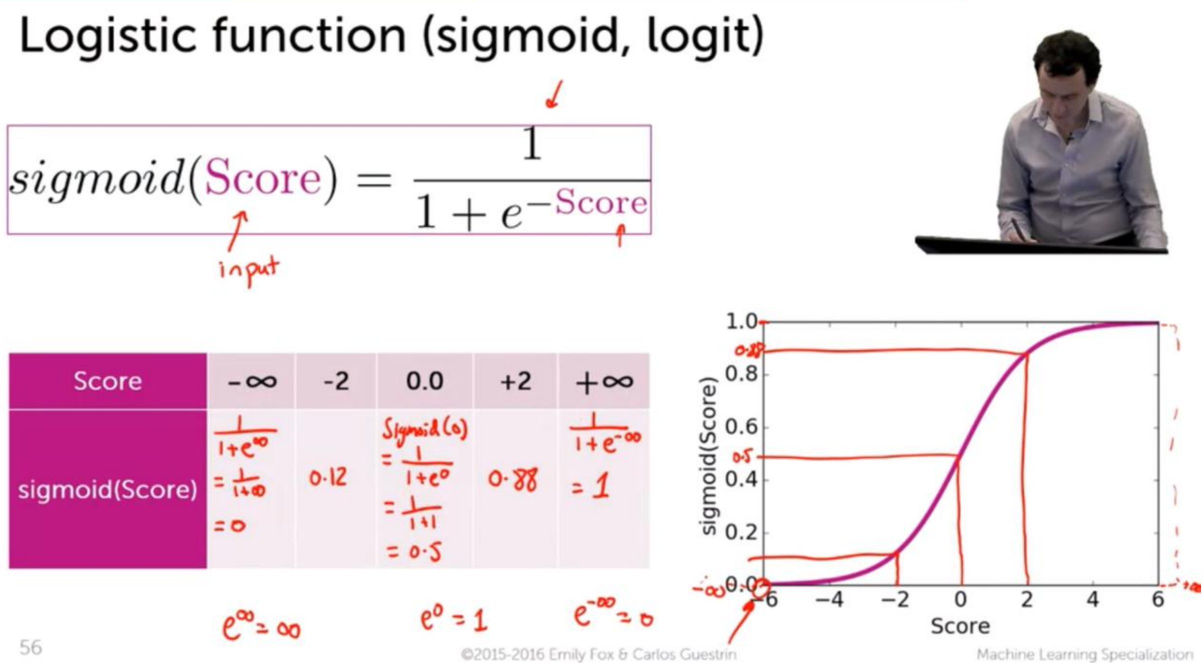


Generally Linear Regression's ouput is between -inf to +inf

Squeezing the Linear regression's output between 0 and 1 is called Generalized Linear Model



2) The sigmoid(logistic) Link function

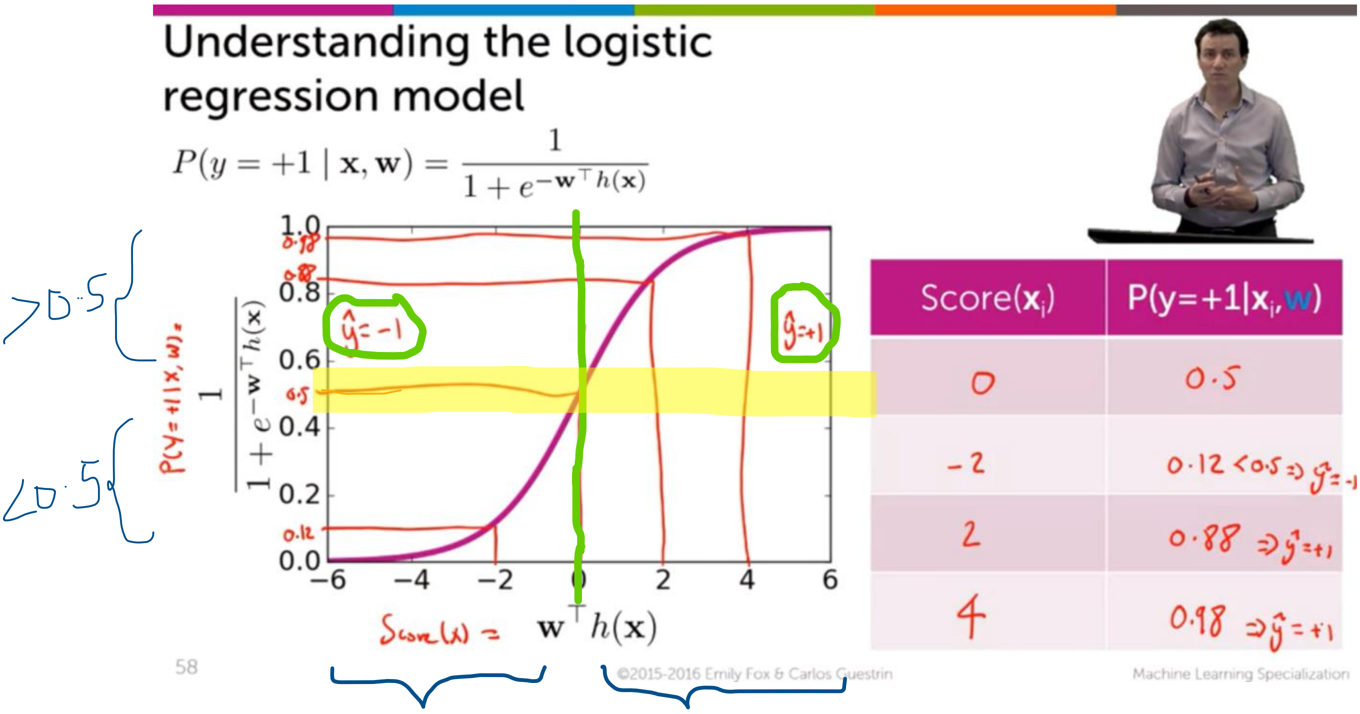


We are giving the score whose output is (-inf to +inf)

Wants to change the output (0 to +inf)

This is done using Link function which is Sigmoid function

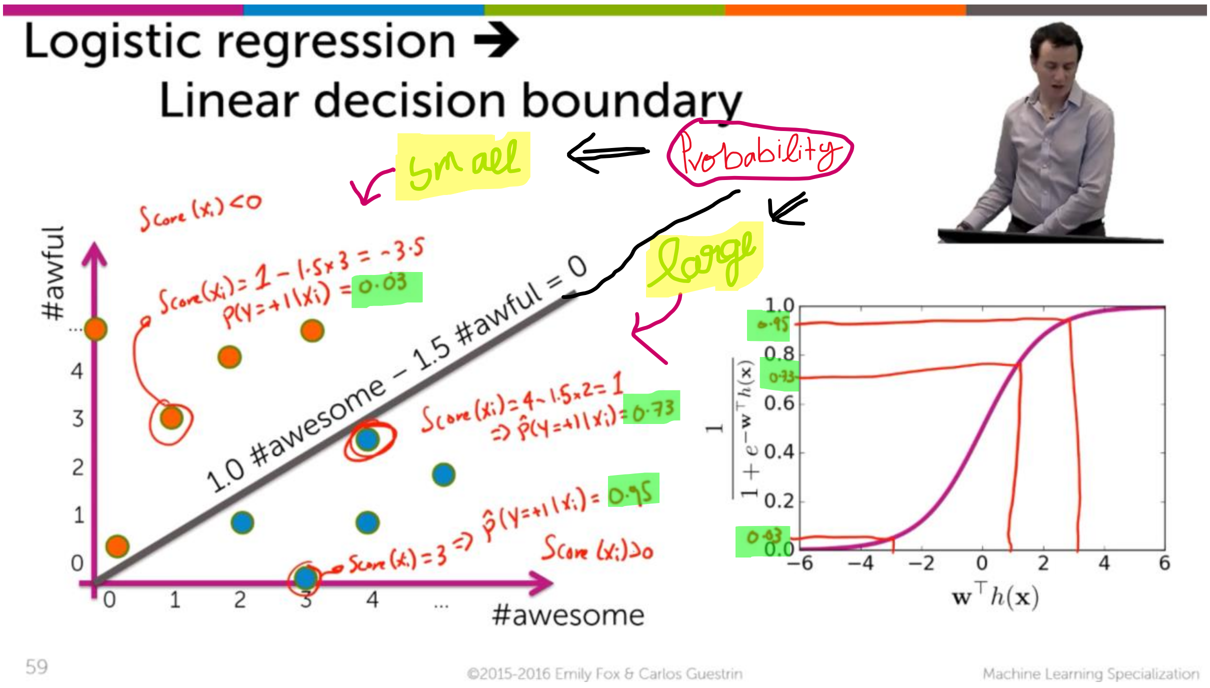
2)Logistic Regression Model



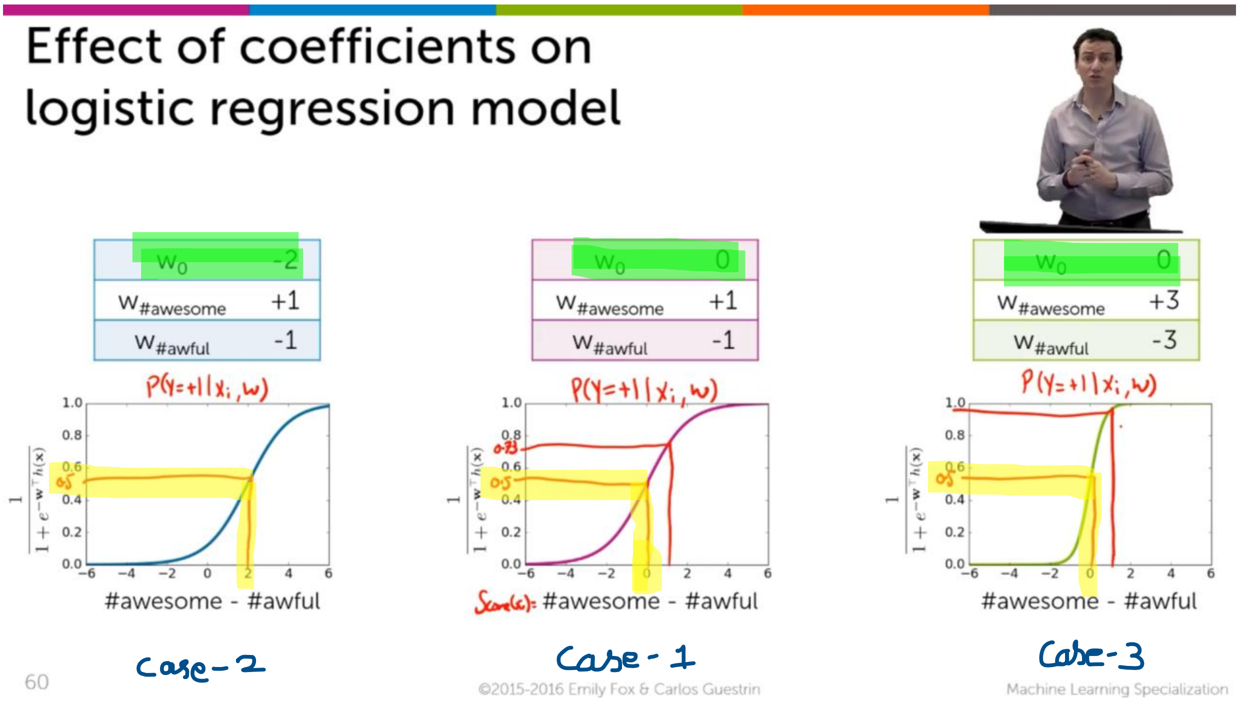
-ve reviews +ve reviews

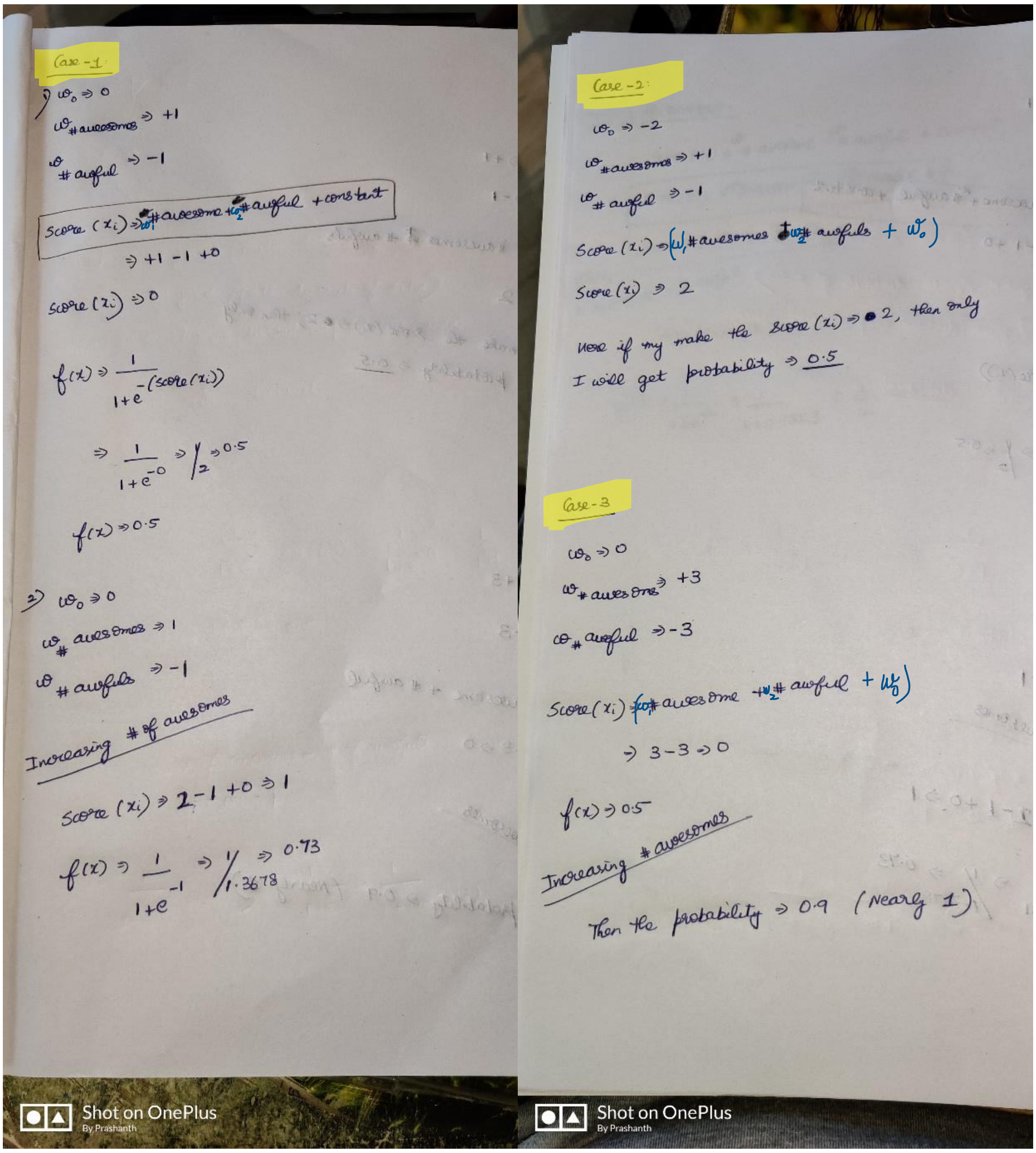
Taking Score(x-axis) and Probability(y-axis)

Effect of coefficient values on predicted probabilities



4) Effect of coefficient on Logistic Regression Model





Conclusion

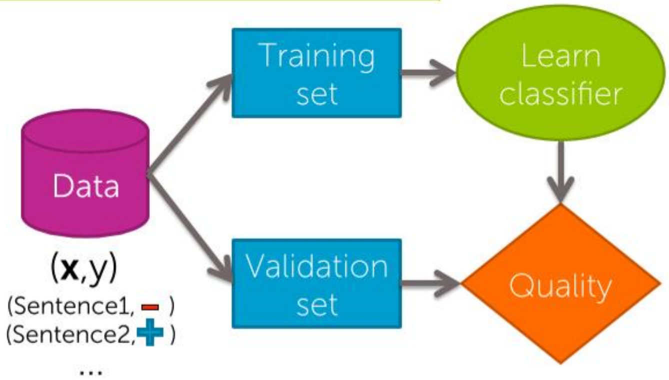
- See the effect of coefficients affecting the probability. We can conclude if the model has some bigger coefficients then the probabilities can be found more quickly.
- Changing the constant will shift the line to the left and to the right.

Now we want to find the coefficients that best fits

Training a classifier = Learning the coefficients

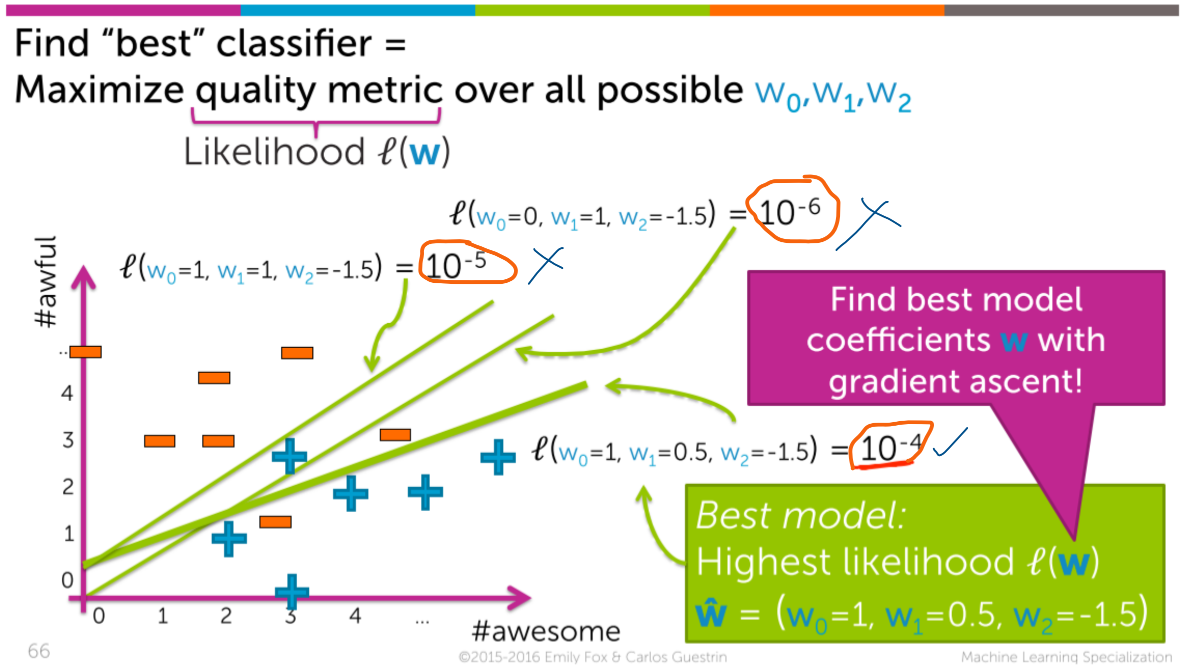
Word	Coefficient	Value
	$\hat{w}_0$	-2.0
good	$\hat{w}_1$	1.0
awesome	$\hat{w}_2$	1.7
bad	$\hat{w}_3$	-1.0
awful	$\hat{w}_4$	-3.3
...	...	...

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



The data is splitted up into training and validation set.
In training set, running a learning algorithm and output the parameter w^.
This w^ 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^ ????

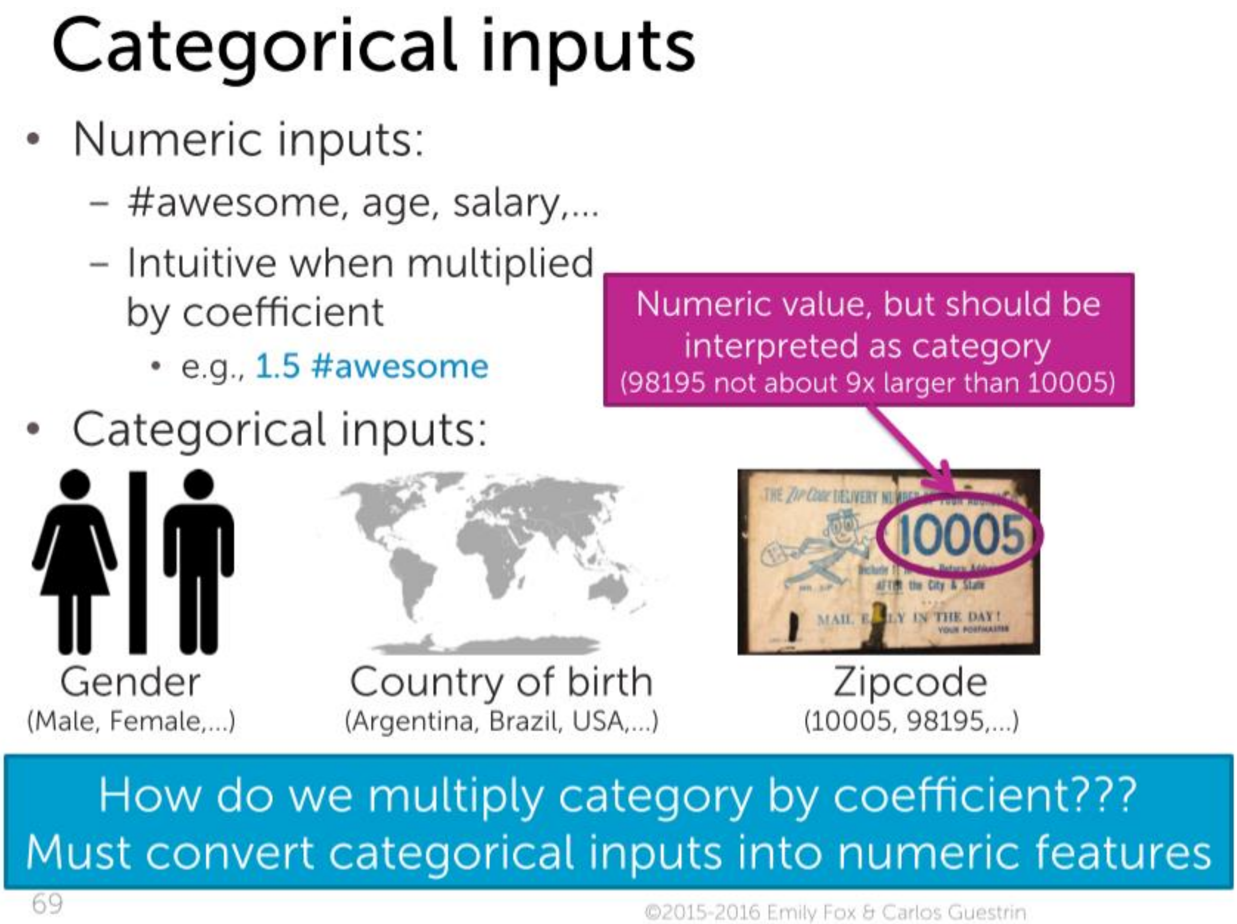


Quality metric → likelihood function l(w)

We are saying among the three 10^-4 → best likelihood

(Since for the best classifier → Maximise the likelihood l(w))

We want the required w^, in which l(w) → maximum. So in-order to choose that w^ gradient ascent comes to the picture.

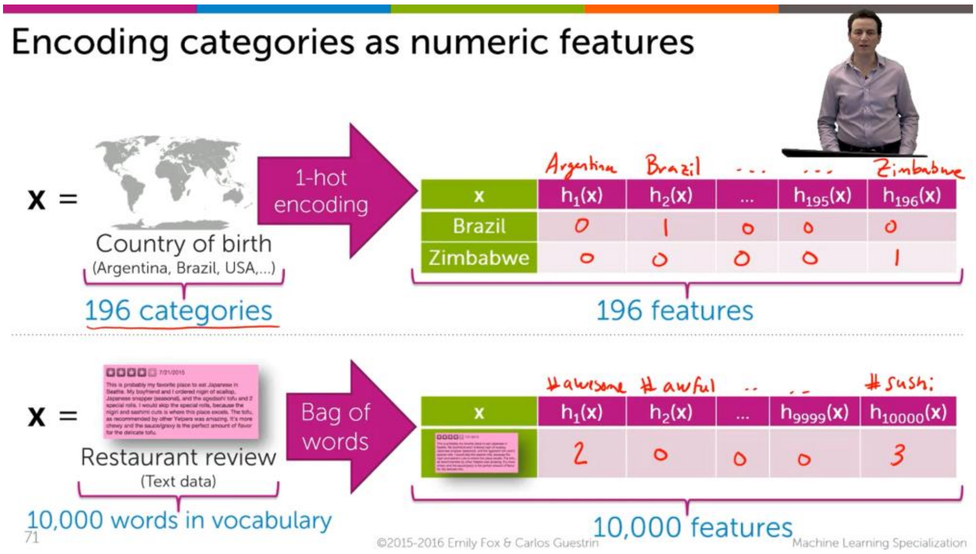


Basically in numerical data,
in score function → we will multiply the numeric with the coefficient.

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



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

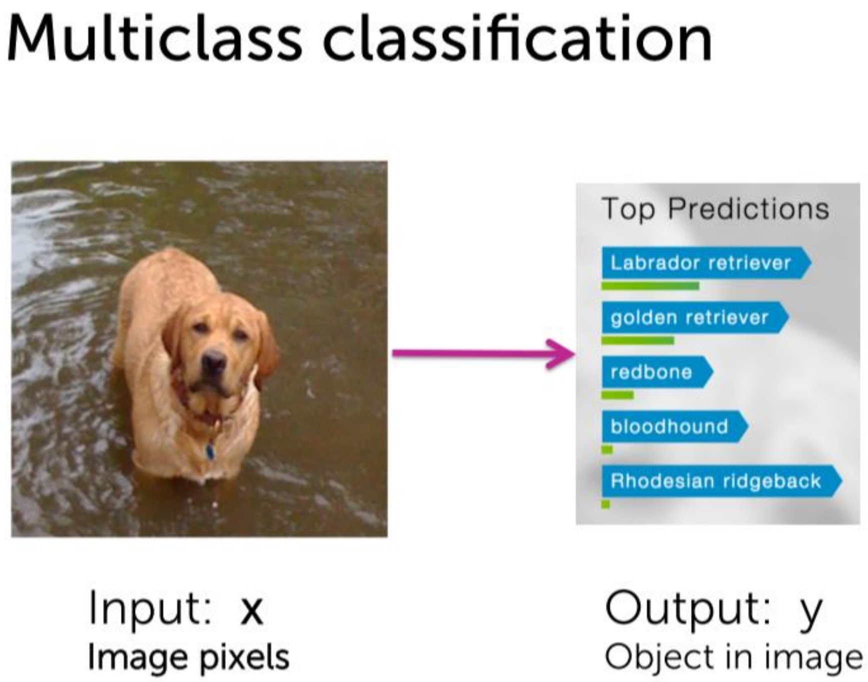
If somebody is born in Zimbabwe → then for Zimbabwe put 1 and for everything put 0

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)

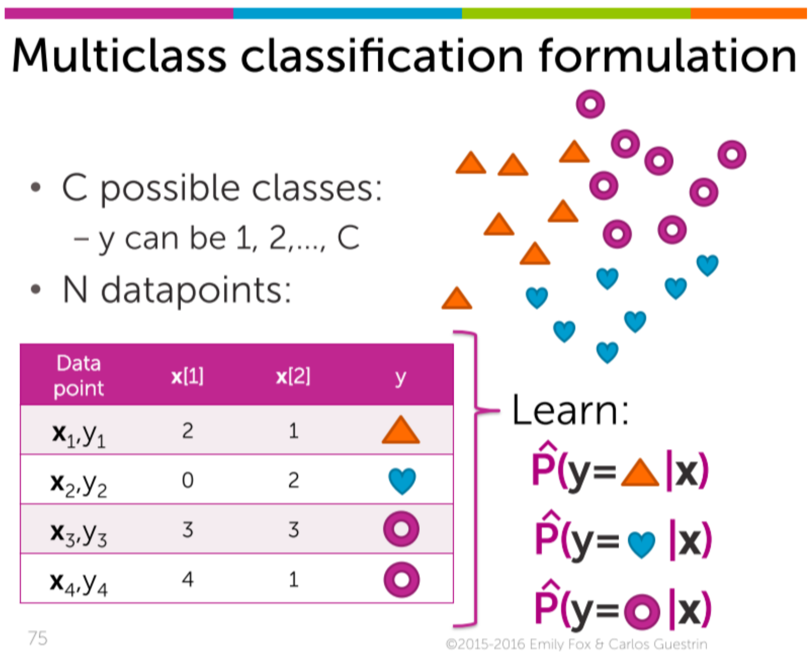
Multiclass classification (1 versus all)



In this image → 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 a nearly 1000's of categories.

How to solve this??? → One versus ALL

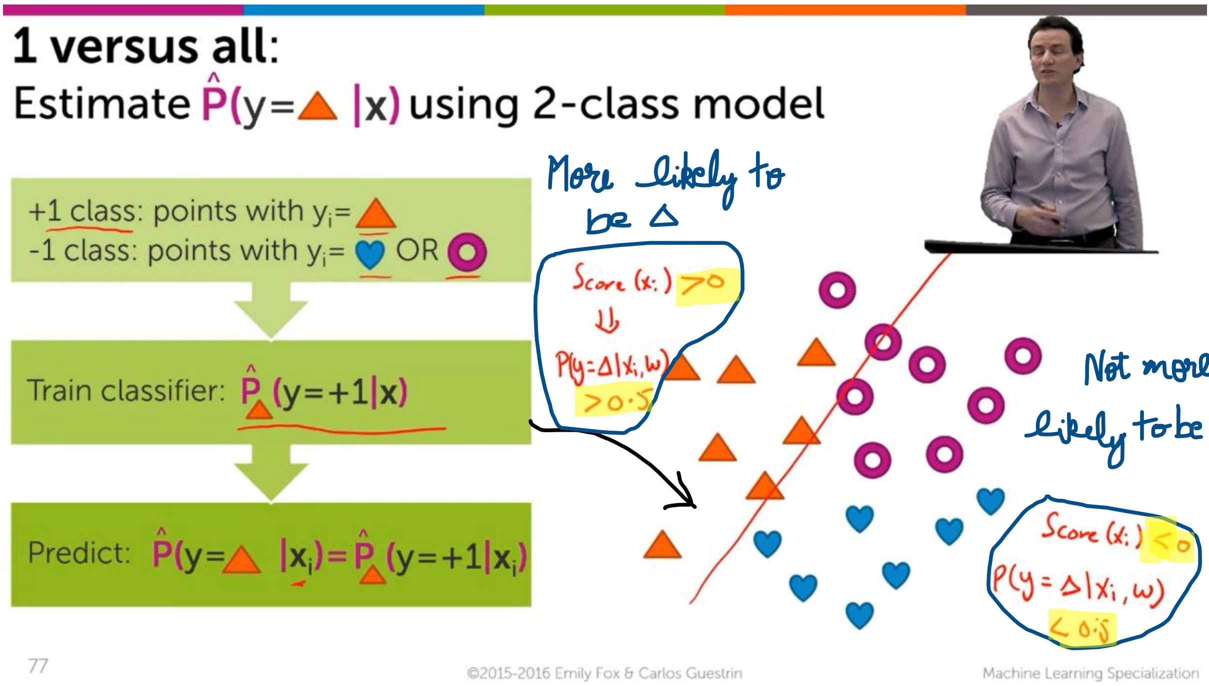
Eg: Triangle, donut and hearts are different classes



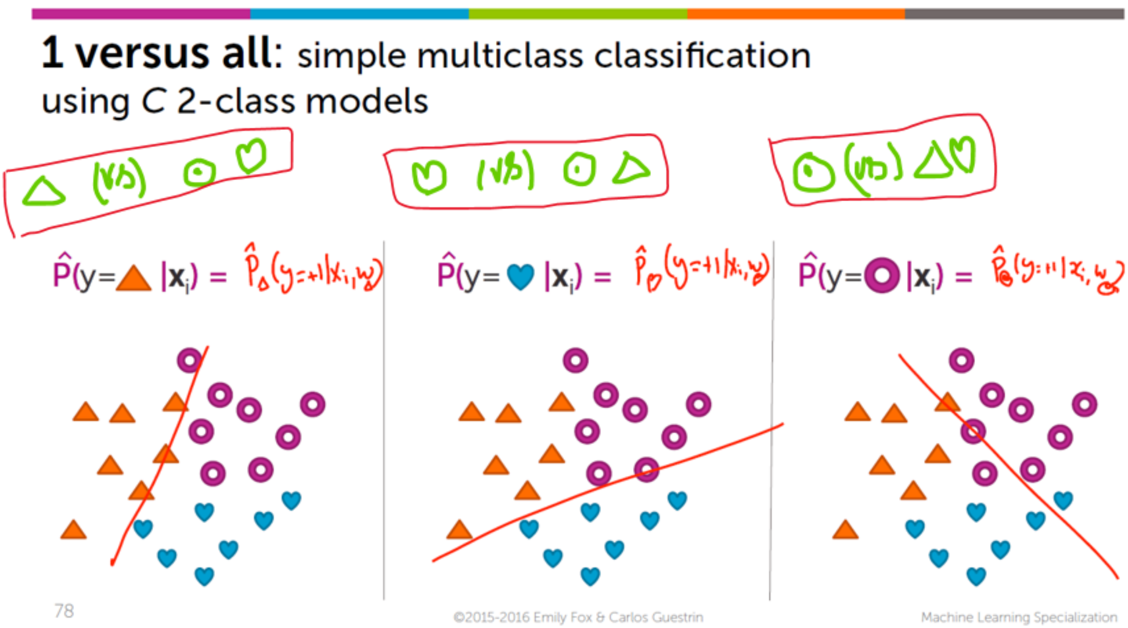
There are 3 classes,

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



Now we are going to make a classifier to learn → 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.



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 classifer.