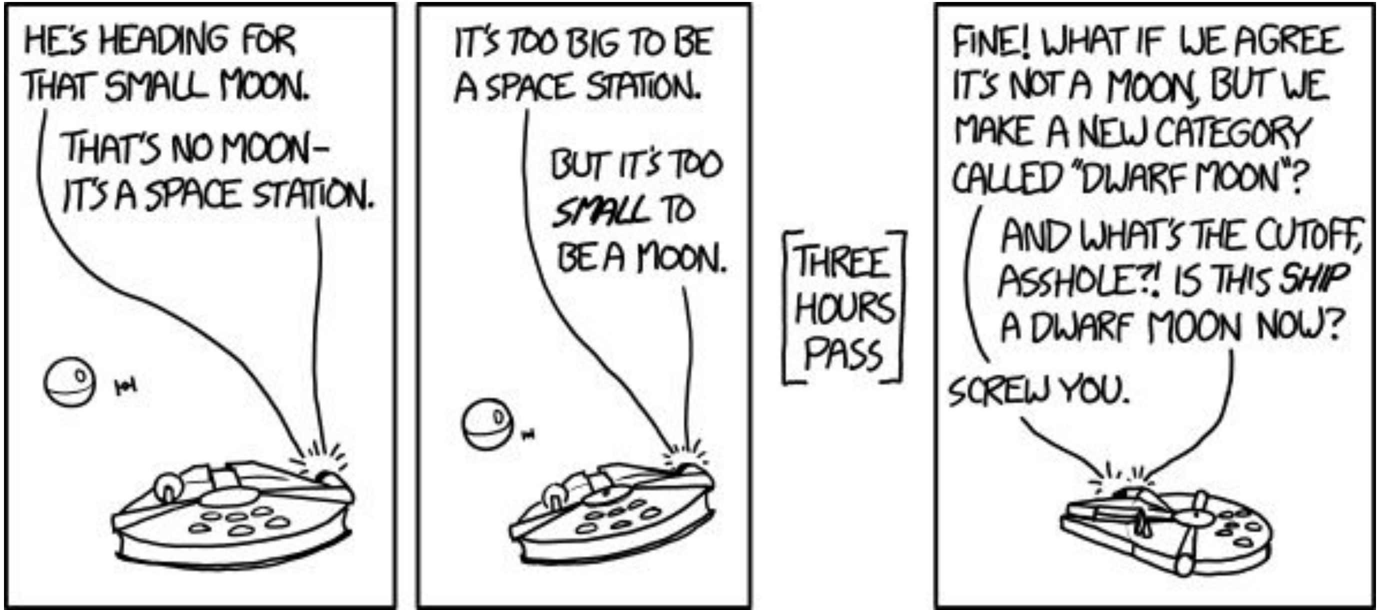
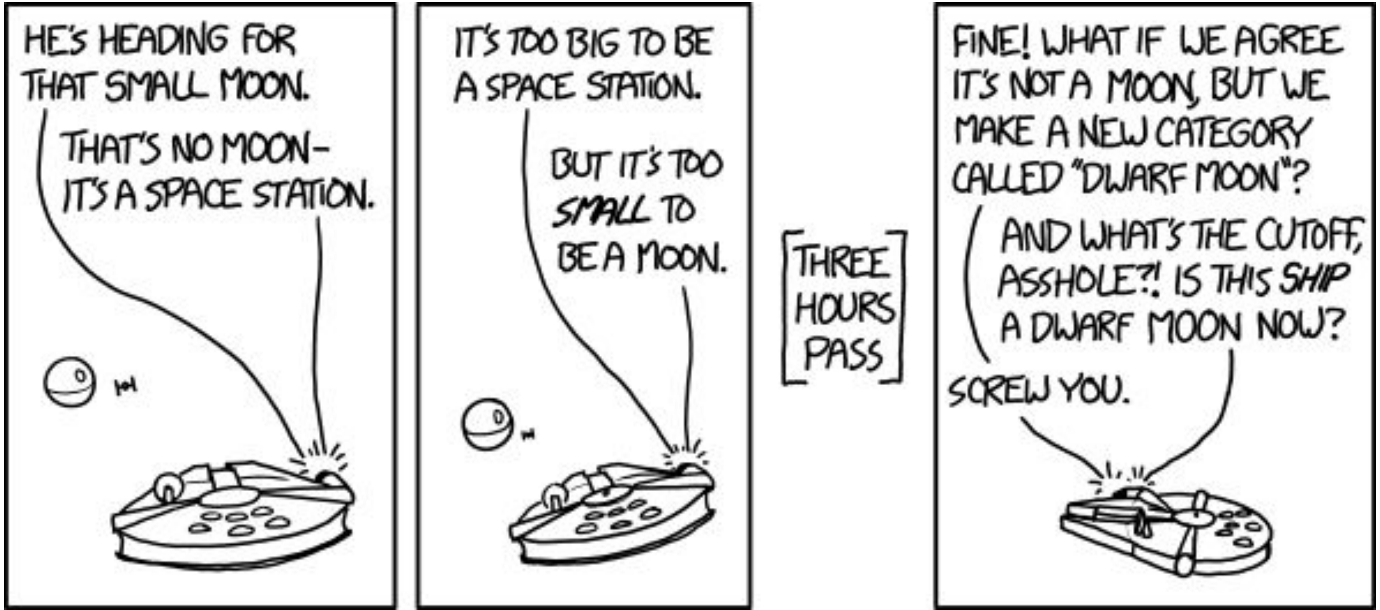
Lecture 25: Classification and Logistic Regression



Announcements and reminders

- **Practicum** is due **11:59 PM on Friday 3 May**
- Tony is out of town, so he won't have office hours this week. **Available by email/Piazza!**
- **FCQs** available until Monday: colorado.campuslabs.com/courseeval



Previously on CSCI 3022…

Given data ($x_{i1}, x_{i2}, \ldots, x_{ip}, y_i$), for $i = 1, 2, \ldots, n$, fit a MLR model of the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i, \quad \text{where each } \epsilon_i \sim N(0, \sigma^2)$$

After learning weights (coefficients), if we want to make a prediction about a new data point:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p$$

Regression as prediction

So far, we’ve learned about various forms of **regression**

We’ve viewed regression in terms of learning a relationship between one or more features and a response:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i$$

We also talked about using regression as a way to make **predictions**

Predicting survival

Based on our previous experience, it might be tempting to use linear regression as a classifier

Example: CSCI 3022 comes full circle -- back to the *Titanic* data!

	age	outcome
0	25	survived
1	30	survived
2	35	survived
3	40	survived
4	45	died
5	50	died
6	55	died
7	60	died

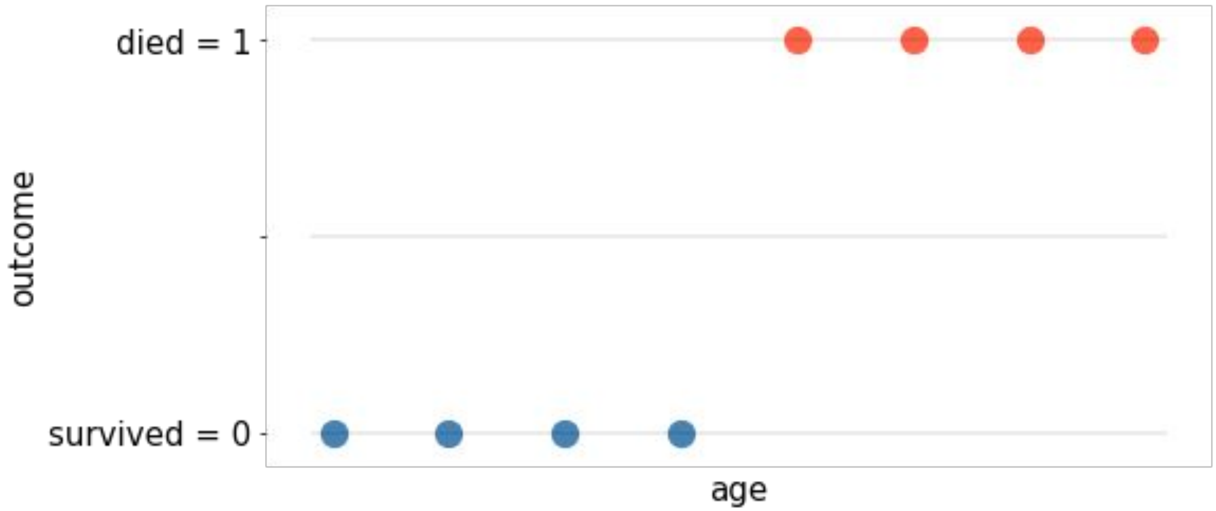
Recode outcomes as y = {0, 1}

	age	outcome
0	25	0
1	30	0
2	35	0
3	40	0
4	45	1
5	50	1
6	55	1
7	60	1

Let's try using linear regression to take feature x = Age and predict the response y = Outcome

Predicting survival

Example: S'pose you want to predict whether a passenger on the Titanic survived or not, based on passenger Age as the sole feature

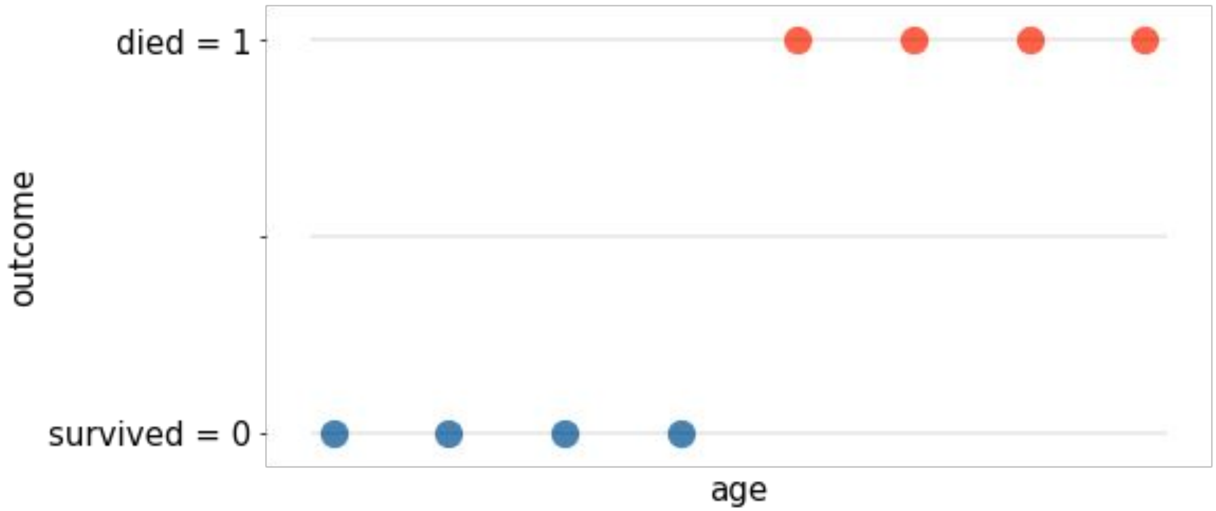


Predicting survival

Example: S’pose you want to predict whether a passenger on the Titanic survived or not, based on passenger Age as the sole feature

Model input: single feature, x_1 = age

Output: prediction, y = {0, 1} = {survived, died}

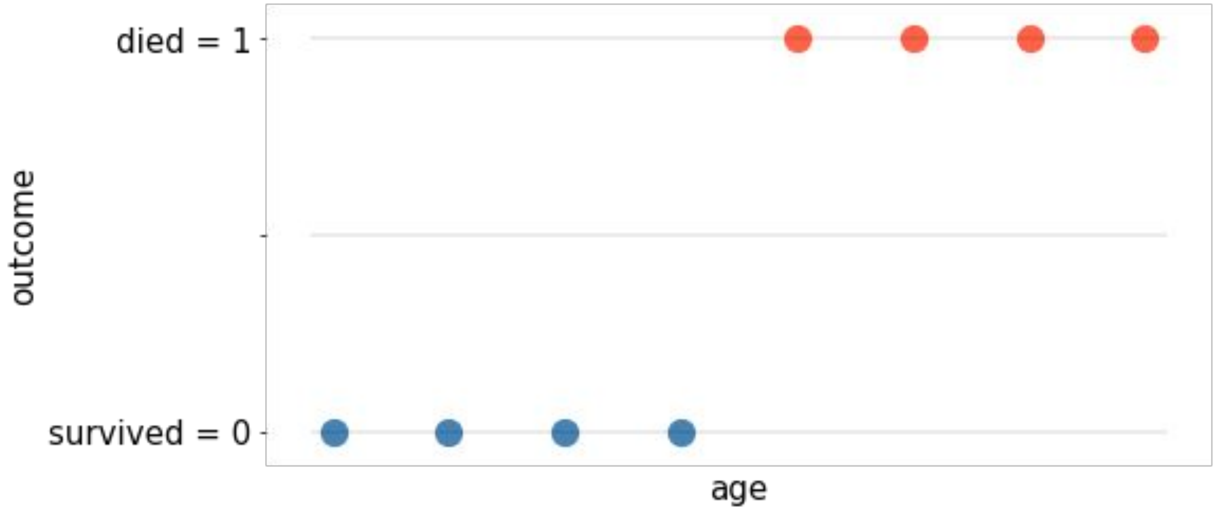


Predicting survival

Example: S'pose you want to predict whether a passenger on the Titanic survived or not, based on passenger Age as the sole feature

Model input: single feature, x_1 = age   Output: prediction, y = {0, 1} = {survived, died}

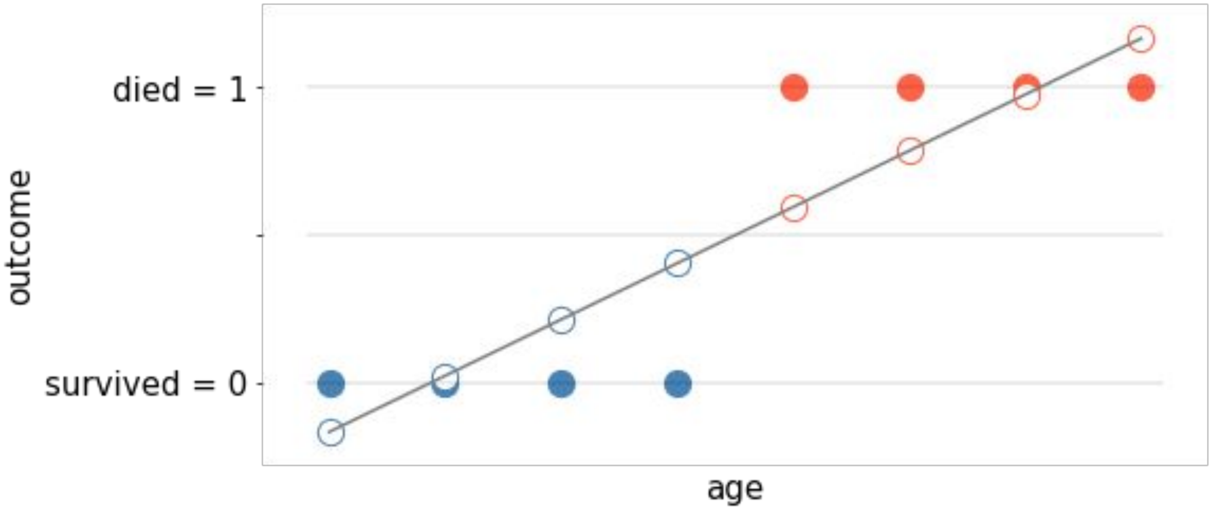
Question: How should we model the relationship between feature and response?



Predicting survival

Example: Model input: single feature, x_1 = age

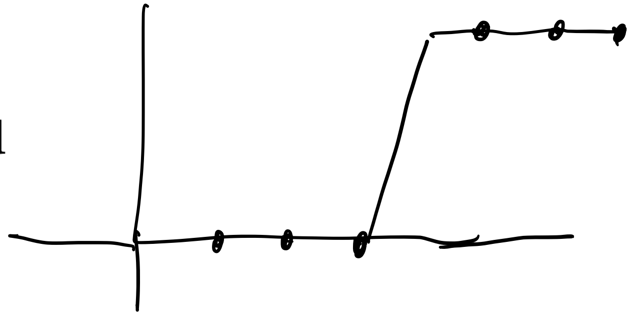
Idea: Linear regression $y = \beta_0 + \beta_1 x_1$

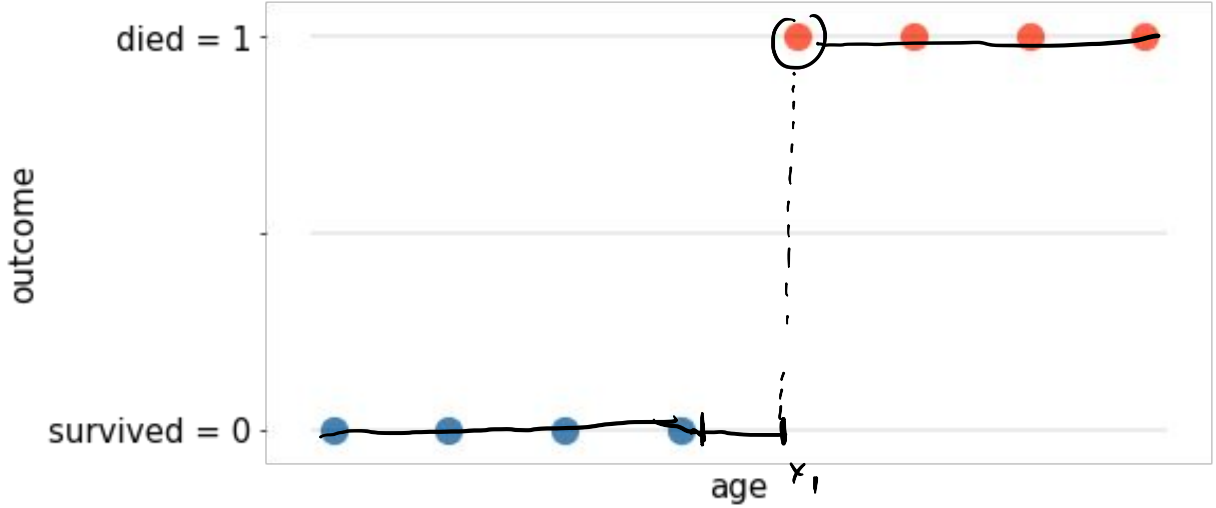


Predicting survival

Example: Model input: single feature, x_1 = age

Idea: Piecewise function $y = \begin{cases} 1 & \text{if } x_1 > \text{some threshold} \\ 0 & \text{otherwise} \end{cases}$

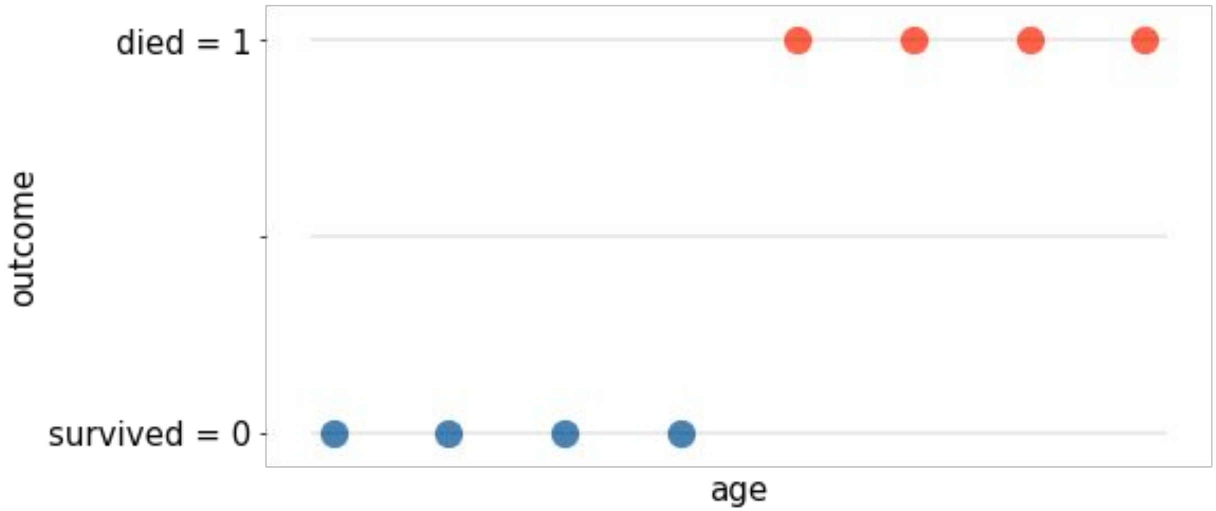




Predicting survival

Example: Model input: single feature, x_1 = age

Idea: Need something that behaves more like a **probability**…

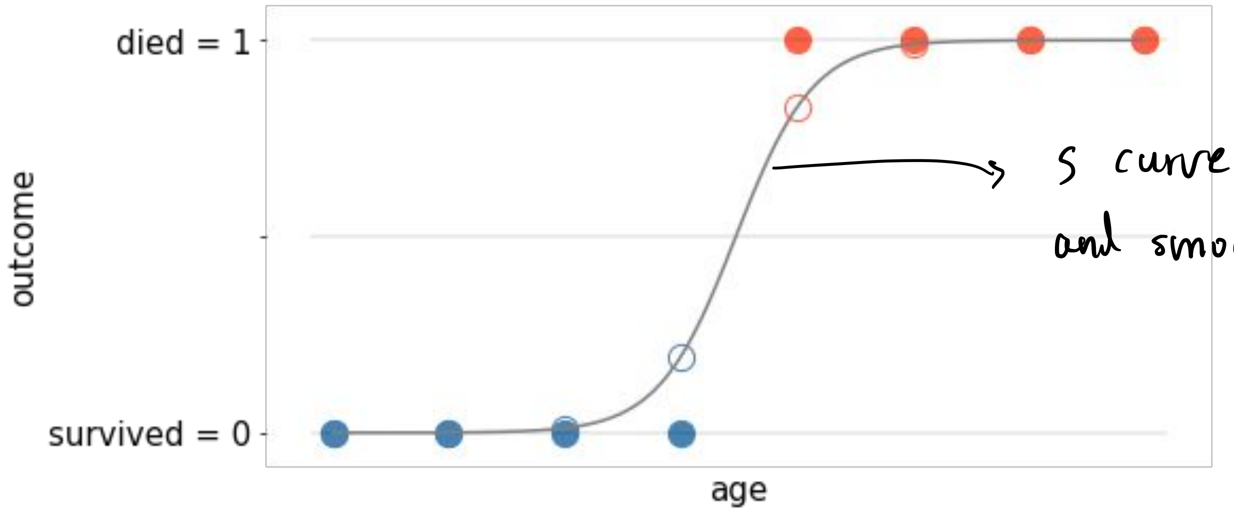


Predicting survival

given the age

Example: Model input: single feature, x_1 = age

Idea: Need something that behaves more like a **probability**…



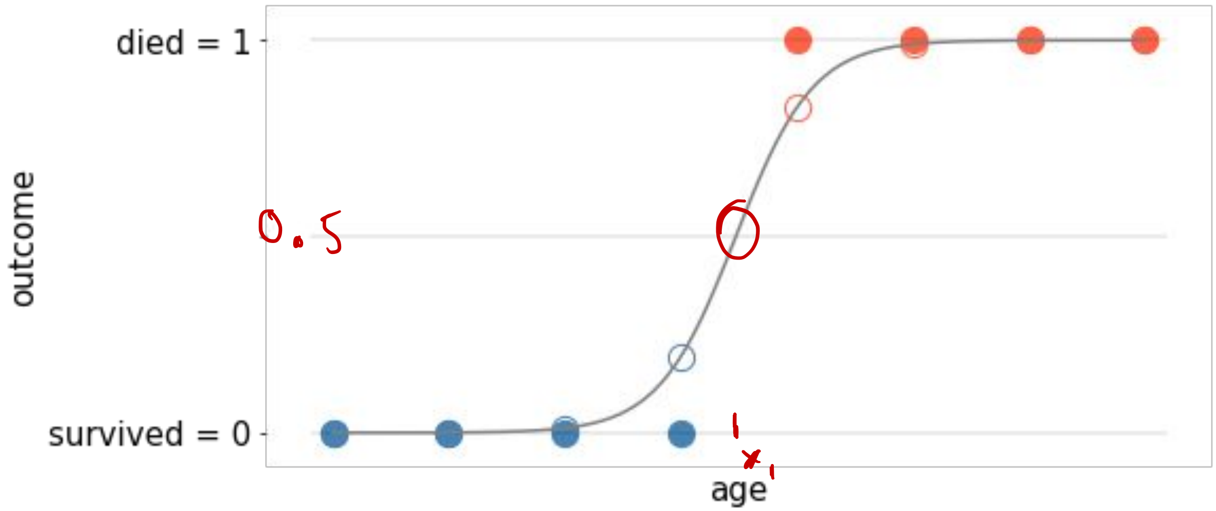
S curve continue and smooth

Predicting survival

Example: Model input: single feature, x_1 = age

$$\hat{y} = \begin{cases} 1 & \text{function value} \geq 0.5 \\ 0 & \text{function} < 0.5 \end{cases}$$

WHOA! That thing is ***perfect!*** What kind of sorcery is this?



The sigmoid function

$$\mathrm{sigm}(z) = \frac{1}{1+e^{-z}}$$

if $z = 0$

$$\mathrm{sigm}(0) = \frac{1}{1+e^{-0}} = \frac{1}{1+1} = \frac{1}{2}$$

Has **awesome** properties:

As $z \to \infty$

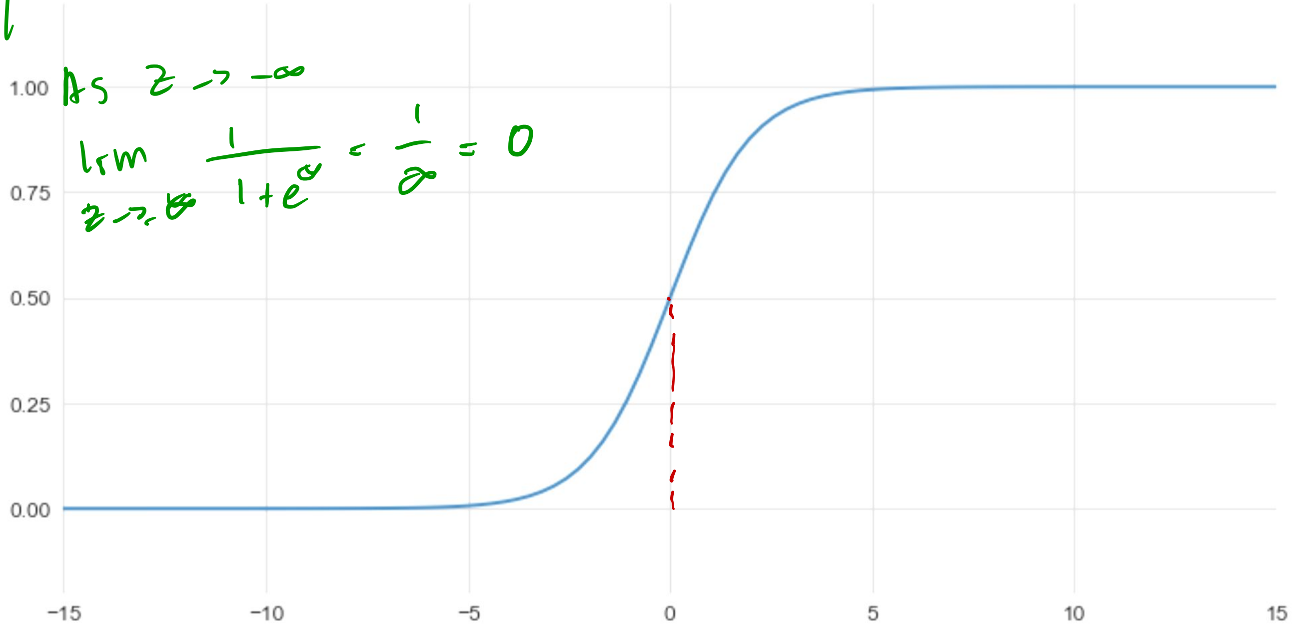
$$\lim_{z \to \infty} \frac{1}{1+e^{-z}} = \frac{1}{1+0} = 1$$

$$\hat{y} = \begin{cases} 1 & \text{if } \mathrm{sigm}(z) \geq 0.5 \\ 0 & \text{if } \mathrm{sigm}(z) < 0.5 \end{cases}$$

As $z \to -\infty$

$$\lim_{z \to -\infty} \frac{1}{1+e^{\infty}} = \frac{1}{\infty} = 0$$

- Behaves like a probability
- Distinguishes between points
- Really smooth



Logistic regression

when does my model = 0.5

$$\frac{1}{1+e^{-(\beta_0+\beta_1 x)}} = \frac{1}{2}$$

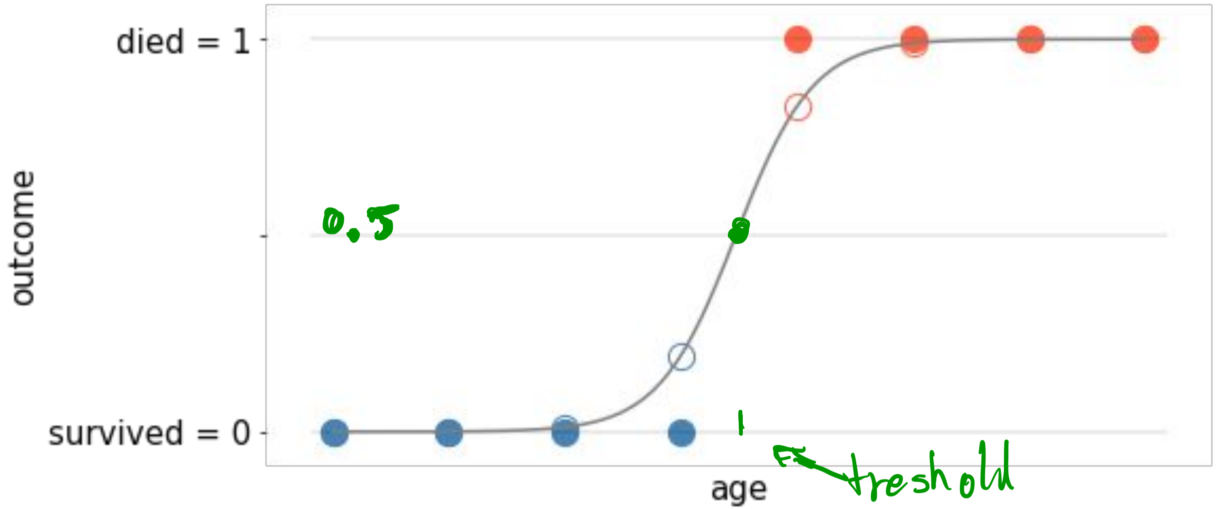
$$2 = 1+e^{-(\beta_0+\beta_1 x)}$$

$$1 = e^{-(\beta_0+\beta_1 x)}$$

The model: $p(y=1 \mid x) = \mathrm{sigm}(\beta_0 + \beta_1 x)$

→ Learn weights β_0 and β_1 from the data

→ Classify data point *x* according to: $\hat{y} = \begin{cases} 1 & \text{if } \mathrm{sigm}(\hat{\beta}_0 + \hat{\beta}_1 x) \geq 0.5 \\ 0 & \text{if } \mathrm{sigm}(\hat{\beta}_0 + \hat{\beta}_1 x) < 0.5 \end{cases}$



An odd(s) view of logistic regression

Our inevitable path to **logistic regression** and the **sigmoid function** began with our insistence on modeling the relationship between features and the response as a legit **probability**.

It turns out that through some basic algebra, we can arrive at an interpretation of logistic regression that is very regression-like

But first we have to put on our gambling hats and talk about **odds**

An odd(s) view of logistic regression

In statistics, the **odds** of an event is the ratio of the probability that the event occurs, divided by the probability that the event does not occur, and then generally flipped to get a value bigger than 1

odds =

Example 1: If p = 0.75, then odds = $\frac{.75}{1-.75} = \frac{.75}{.25} = 3$

We would say that the odds are 3 to 1 in favors

Example 2: If p = 0.1, then odds = $\frac{0.1}{1-0.1} = \frac{0.1}{0.9} = \frac{1}{9}$

We would say that the odds are 9 to 1 against

An odd(s) view of logistic regression

model sigm(z)

In logistic regression, we model $p = p(y=1 \mid x) = \text{sigm}(\beta_0 + \beta_1 x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$

What if we calculate the **odds** that y=1, given the data x?

$$\text{odds} = \frac{p}{1-p} = \frac{\frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}}{1 - \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}}$$

$$= \frac{\frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}}{\frac{\cancel{1} + e^{-(\beta_0 + \beta_1 x)} \cancel{-1}}{1 + e^{-(\beta_0 + \beta_1 x)}}} = \frac{1}{\cancel{1 + e^{-(\beta_0 + \beta_1 x)}}} \cdot \frac{\cancel{1 + e^{-(\beta_0 + \beta_1 x)}}}{e^{-(\beta_0 + \beta_1 x)}}$$

$$= \frac{1}{e^{-(\beta_0 + \beta_1 x)}}$$

An odd(s) view of logistic regression

Taking the natural log of both sides, we get:

$$\log(\text{odds}) = \beta_0 + \beta_1 x$$

→ We ***have*** been doing linear regression all along, but for the **log-odds** instead of probability!

Let's look at that coefficient β_1: odds $= \exp(\beta_0 + \beta_1 x)$

With a unit increase in x, we get: odds $= \exp(\beta_0 + \beta_1 (x+1))$

So we have a new interpretation of the Logistic Regression weight β_1:

An odd(s) view of logistic regression

Taking the natural log of both sides, we get:

$$\log(\text{odds}) = \beta_0 + \beta_1 x$$

→ We ***have*** been doing linear regression all along, but for the **log-odds** instead of probability!

Let's look at that coefficient β_1: $\text{odds} = \exp(\beta_0 + \beta_1 x)$

With a unit increase in x, we get: $\text{odds} = \exp(\beta_0 + \beta_1 (x+1))$

So we have a new interpretation of the Logistic Regression weight β_1:

An odd(s) view of logistic regression

Taking the natural log of both sides, we get:

$$\log(\text{odds}) = \beta_0 + \beta_1 x$$

→ We ***have*** been doing linear regression all along, but for the **log-odds** instead of probability!

Let's look at that coefficient β_1: odds = $\exp(\beta_0 + \beta_1 x)$

With a unit increase in x, we get: odds = $\exp(\beta_0 + \beta_1(x+1))$

$= e^{\beta_0 + \beta_1 (x+1)} = e^{\beta_0 + \beta_1 x + \beta_1} = e^{\beta_0 + \beta_1 x} e^{\beta_1}$

So we have a new interpretation of the Logistic Regression weight β_1:

Logistic Regression with many features

The LogReg model with a single feature looks like: $p(y=1 \mid x) = \text{sigm}(\beta_0 + \beta_1 x)$

But in real life we typically have many features

Example:

- **Predict** the probability of precipitation
- **Features:** temperature, pressure, humidity, wind speed, whether it rained yesterday…

Multiple features LogReg model:

$$p(y=1 \mid x) = \text{sigm}(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p)}}$$

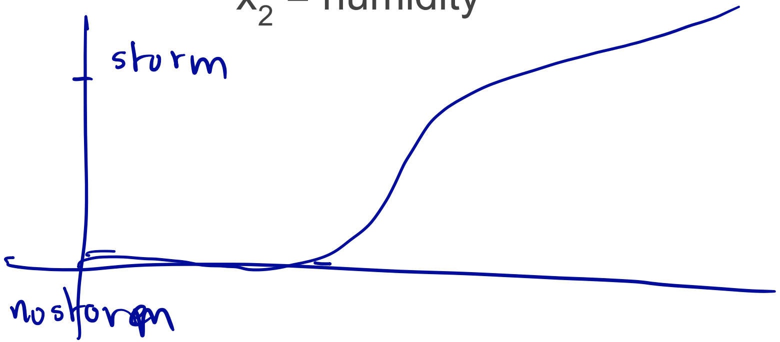
Logistic Regression with many features

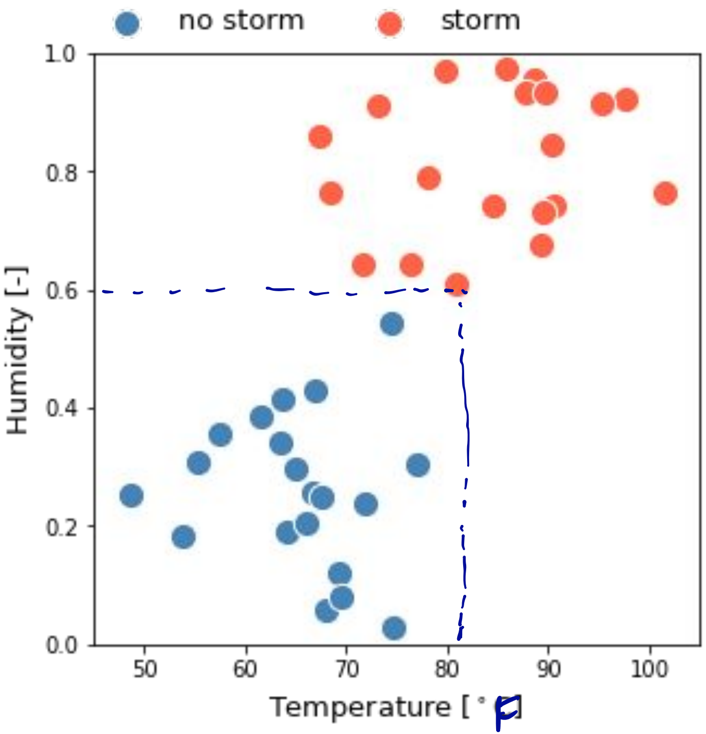
Multiple features LogReg model:

$$p(y=1 \mid x) = \text{sigm}(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$$

Predict: $y = 1$ = storm
$y = 0$ = no storm

Features: x_1 = temperature
x_2 = humidity





Logistic Regression with many features

Above the decision boundary predict storm

Below the decision boundary predict no storm

Multiple features LogReg model:

$$p(y=1 \mid x) = \text{sigm}(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$$

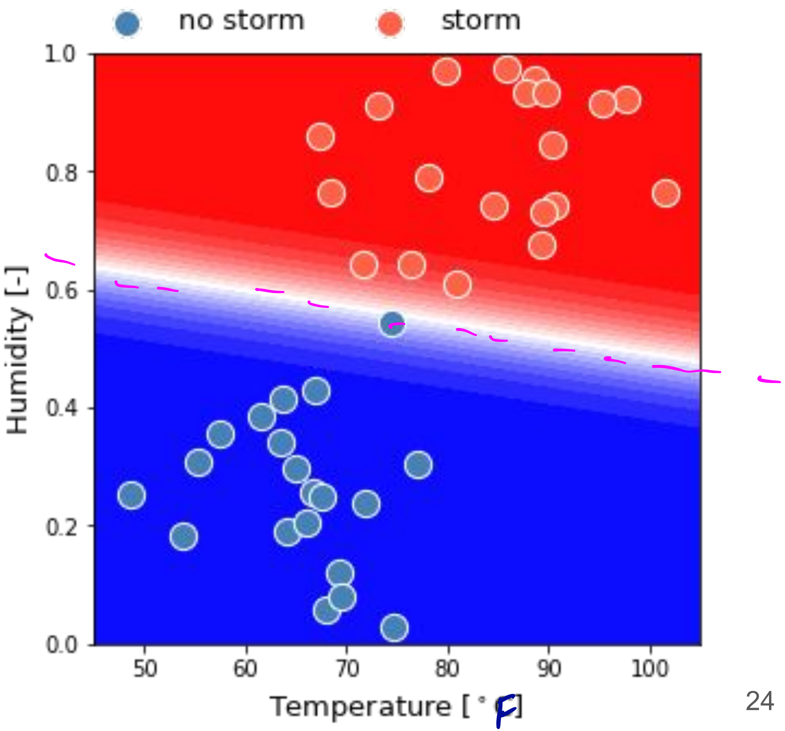
Predict: $y = 1$ = storm

$y = 0$ = no storm

Features: x_1 = temperature

x_2 = humidity

Decision Boundary is line



The Decision Boundary

The **decision boundary** is the line/surface that separate predictions into Class 0 and Class 1

For a 2-feature model, it is described by:

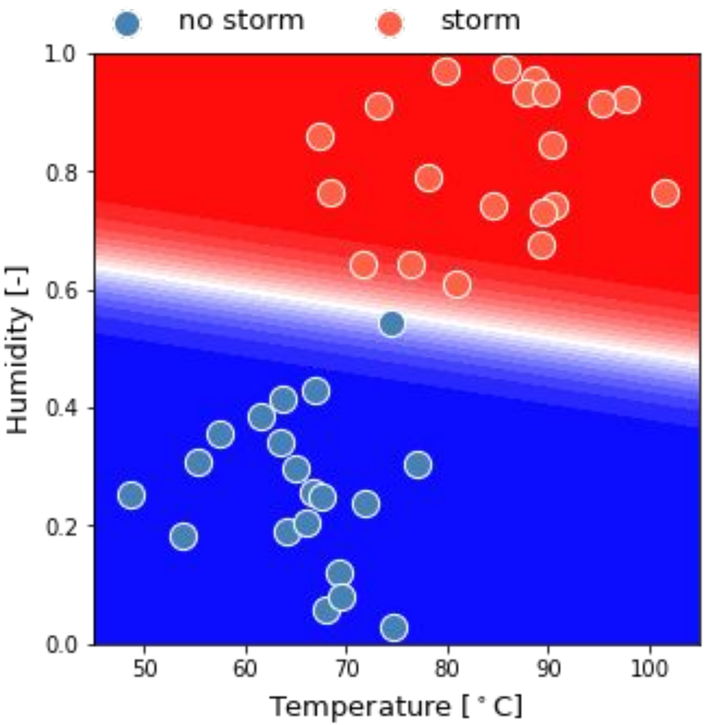
$$p(y=1 \mid x) = \mathrm{sigm}(\beta_0 + \beta_1 x_1 + \beta_2 x_2) = 0.5$$

Which is just a line in 2D space:

$$\frac{1}{1+e^{-(\beta_0+\beta_1 x+\beta_2 x_2)}} = \frac{1}{2}$$

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$$

$$\Rightarrow x_2 = -\frac{\beta_1 x_2}{\beta_2} - \frac{\beta_0}{\beta_1}$$

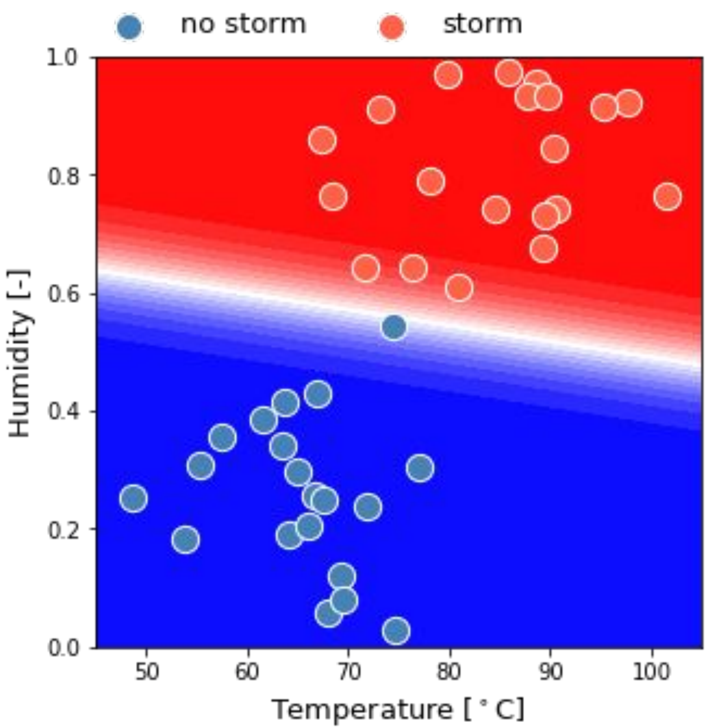


Neat property of the sigmoid function

The Sigmoid function has some nice differential properties that we'll explore next time.

The most important of these is that…

$$\text{If } f(z) = \text{sigm}(z),$$
$$\text{then } f'(z) = \text{sigm}(z)(1 - \text{sigm}(z))$$



What just happened?!

… logistic regression just happened!

→ a ***binary classification*** algorithm

→ Probability of thing with features (x_1, x_2, …) being in class 1 is:

$$p(y=1 \mid x) = \mathrm{sigm}(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots)$$

→ Can incorporate all kinds of features!