What is classification?

- **Classification** is the task of ***learning a target function f*** that maps attribute set **x** to one of the predefined class labels **y**

Tid	Refund (categorical)	Marital Status (categorical)	Taxable Income (continuous)	Cheat (class)
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

One of the attributes is the class attribute
In this case: Cheat

Two class labels (or classes): Yes (1), No (0)

Figure 4.2. Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

Examples of Classification Tasks

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Categorizing news stories as finance, weather, entertainment, sports, etc
- Identifying spam email, spam web pages, adult content
- Understanding if a web query has commercial intent or not

Classification is everywhere in data science
Big data have the answers all questions.

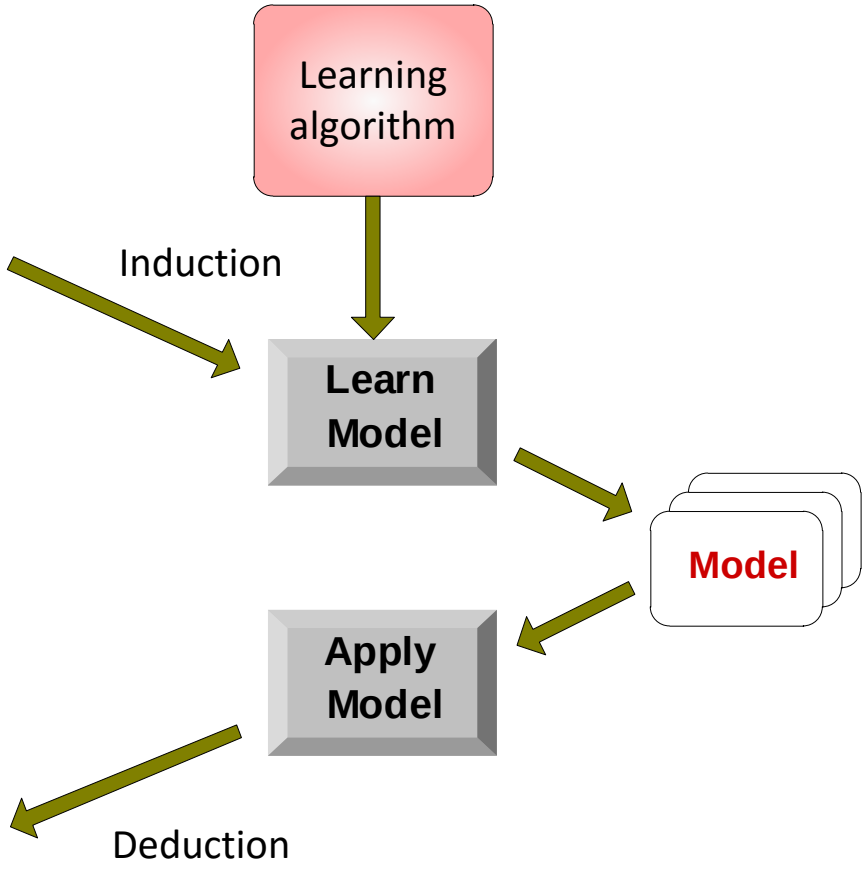
General approach to classification

- Training set consists of records with known class labels
- Training set is used to build a classification model
- A labeled test set of previously unseen data records is used to evaluate the quality of the model.
- The classification model is applied to new records with unknown class labels

Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

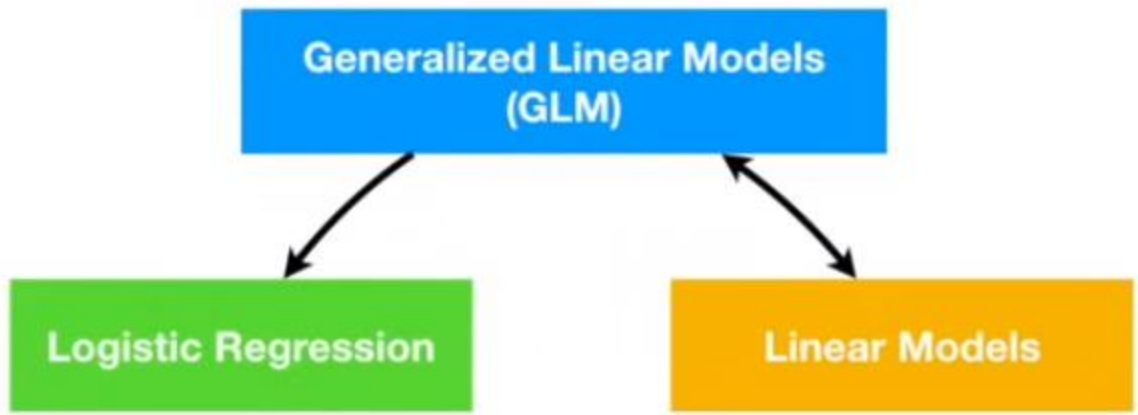
Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Generalized Linear Models are a generalization of the concepts and abilities of regular Linear Models



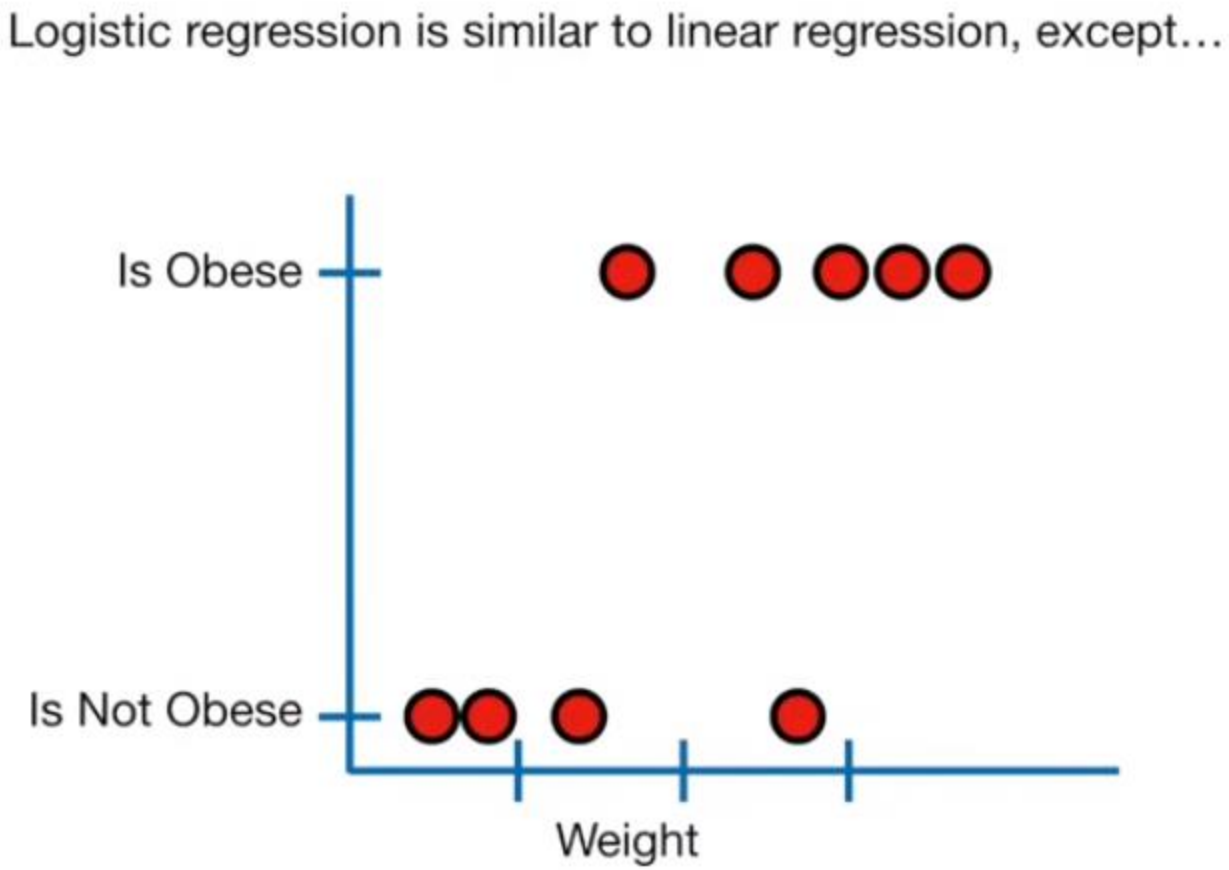
Logistic Regression

- In statistics, the logistic model is a statistical model that is usually taken to apply to a binary dependent variable – Classification problem.

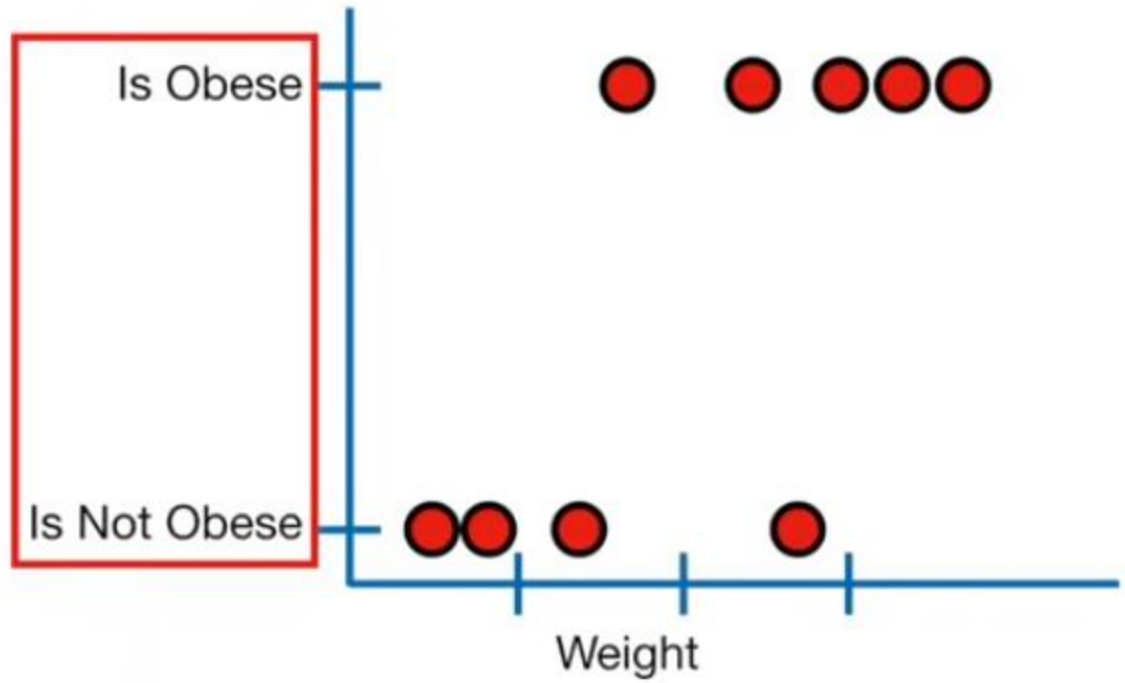
- In regression analysis, logistic regression or logit regression is estimating the parameters of a logistic model.

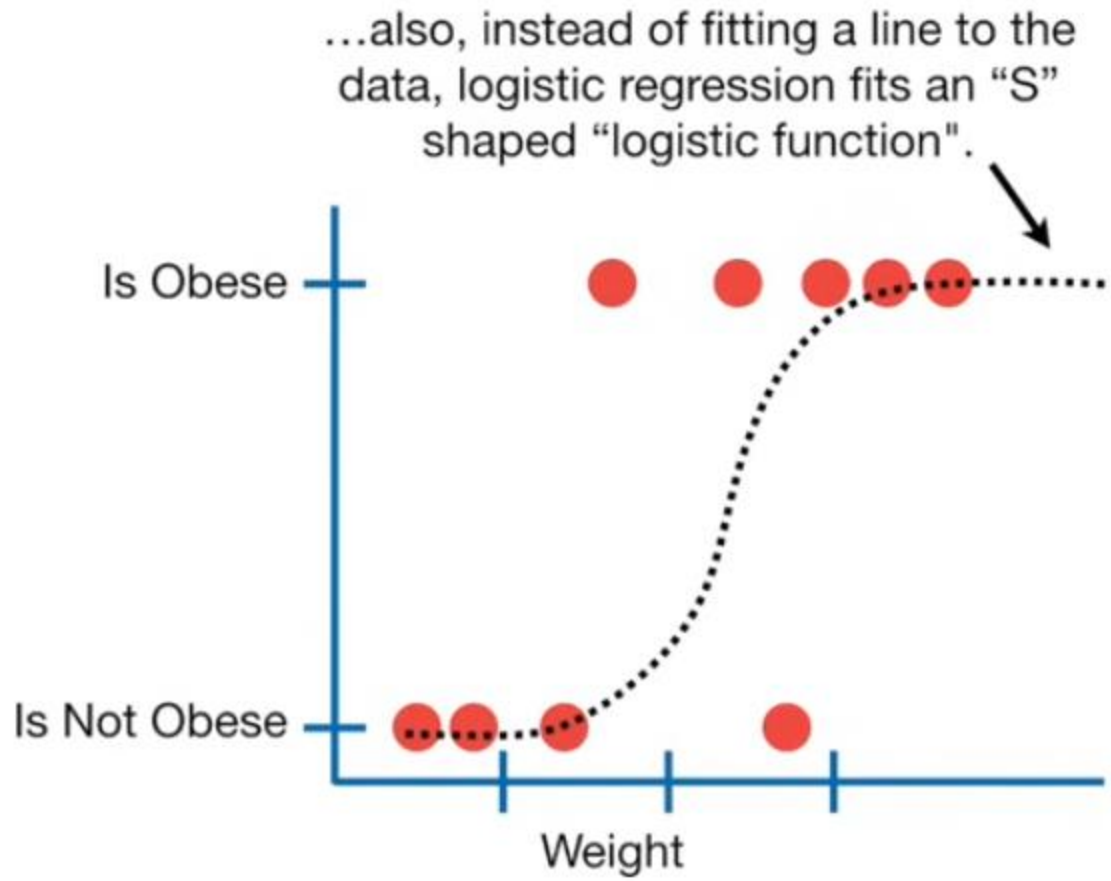
- In Logistic Regression, the dependent variable is binary rather than continuous and it can also be applied to ordered categories (ordinal data).

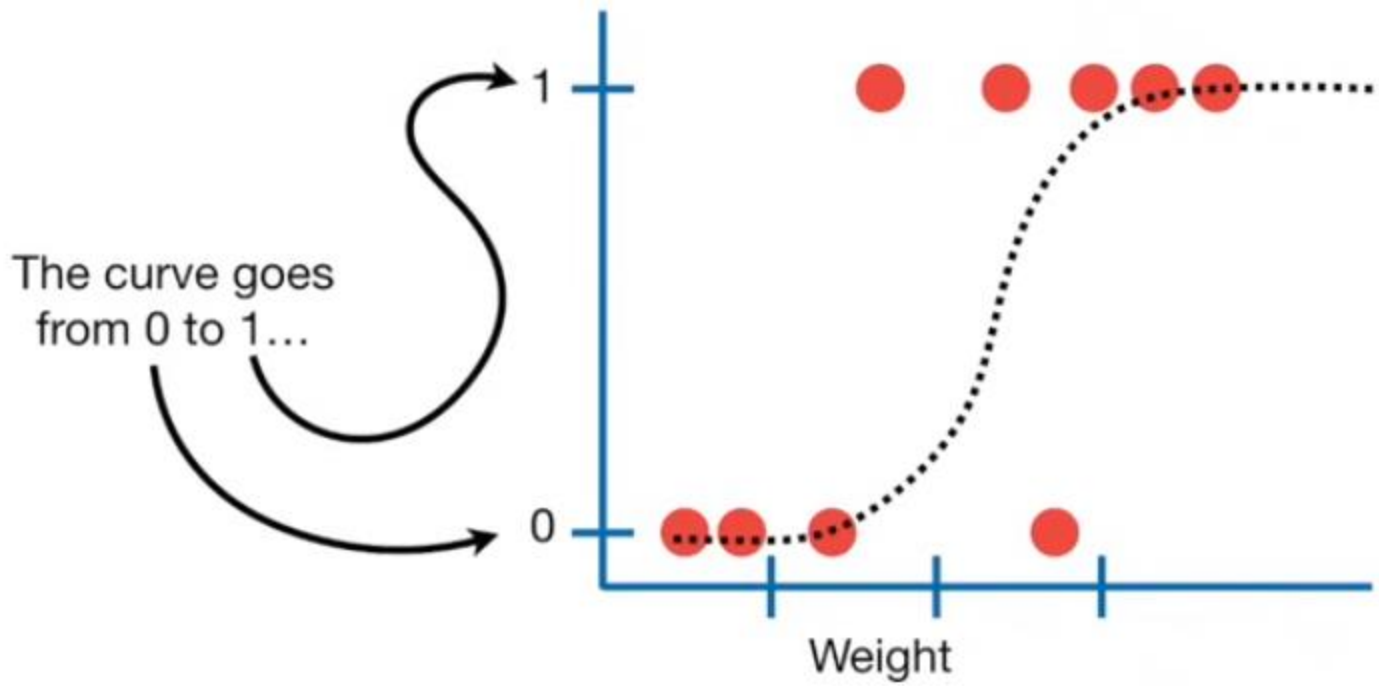
Concept


Logistic regression is similar to linear regression, except…
Is Obese
Is Not Obese
Weight

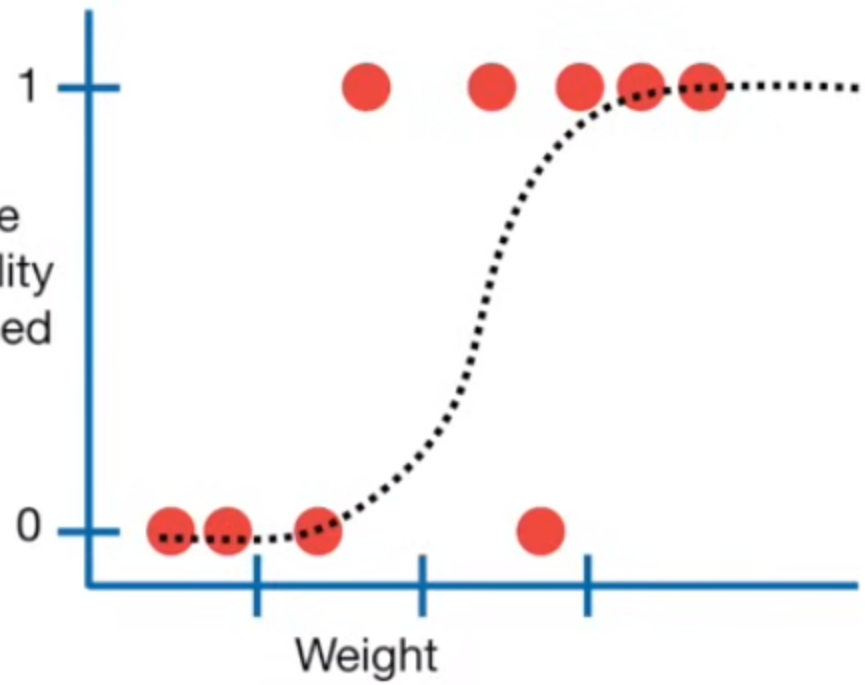
Logistic regression predicts whether something is ***True*** or ***False***, instead of predicting something continuous like ***size***.

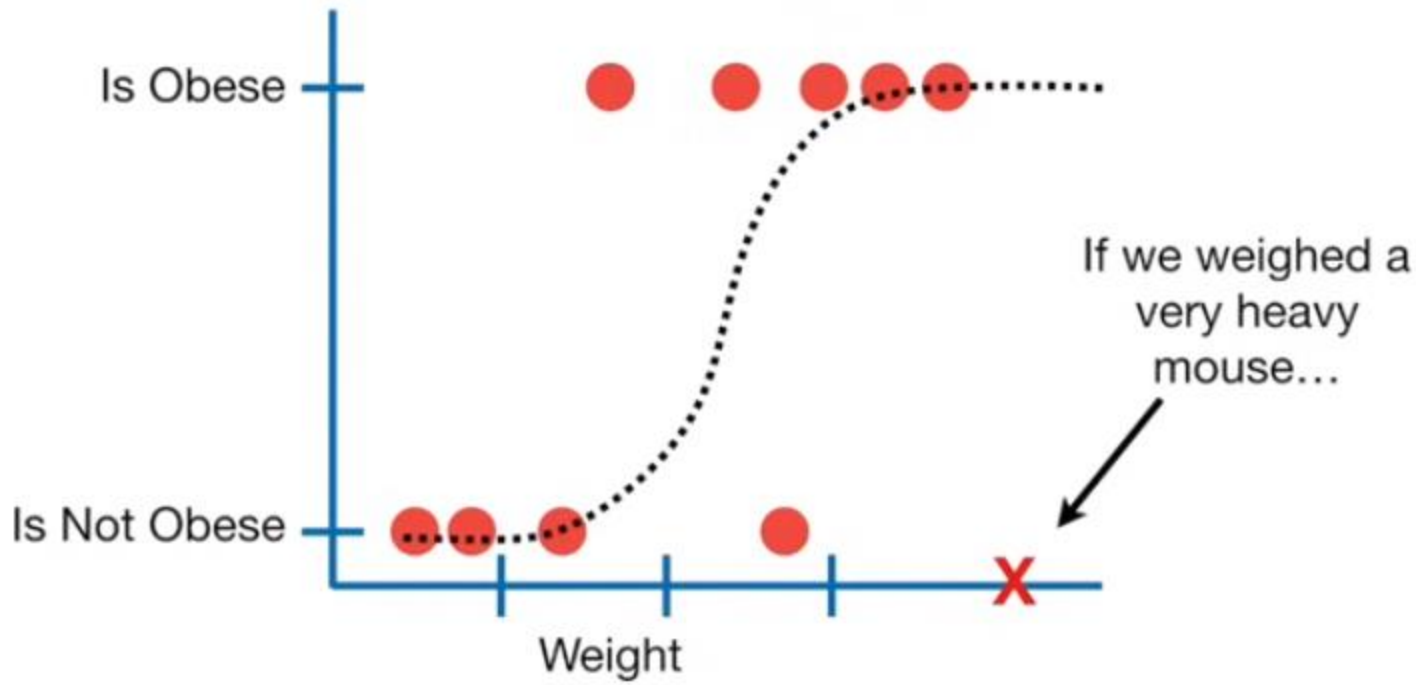


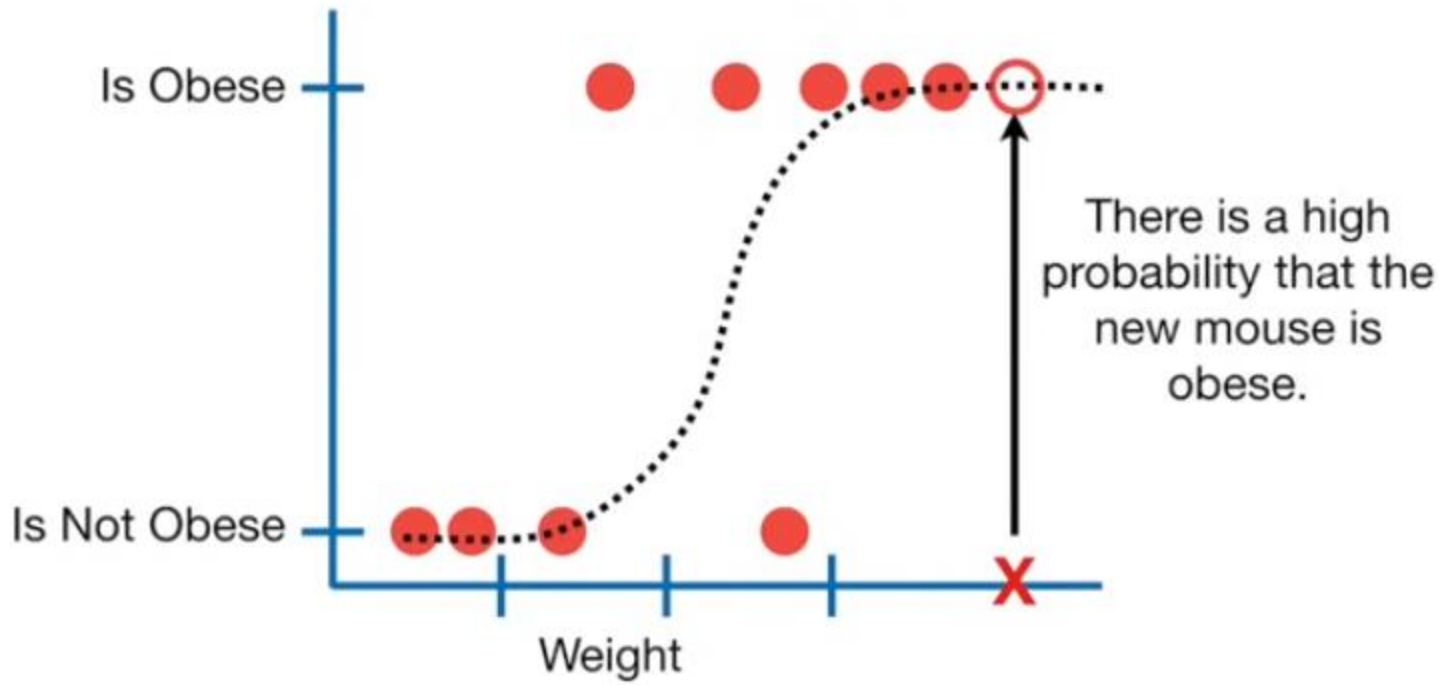

…also, instead of fitting a line to the data, logistic regression fits an “S” shaped “logistic function".
Is Obese
Is Not Obese
Weight

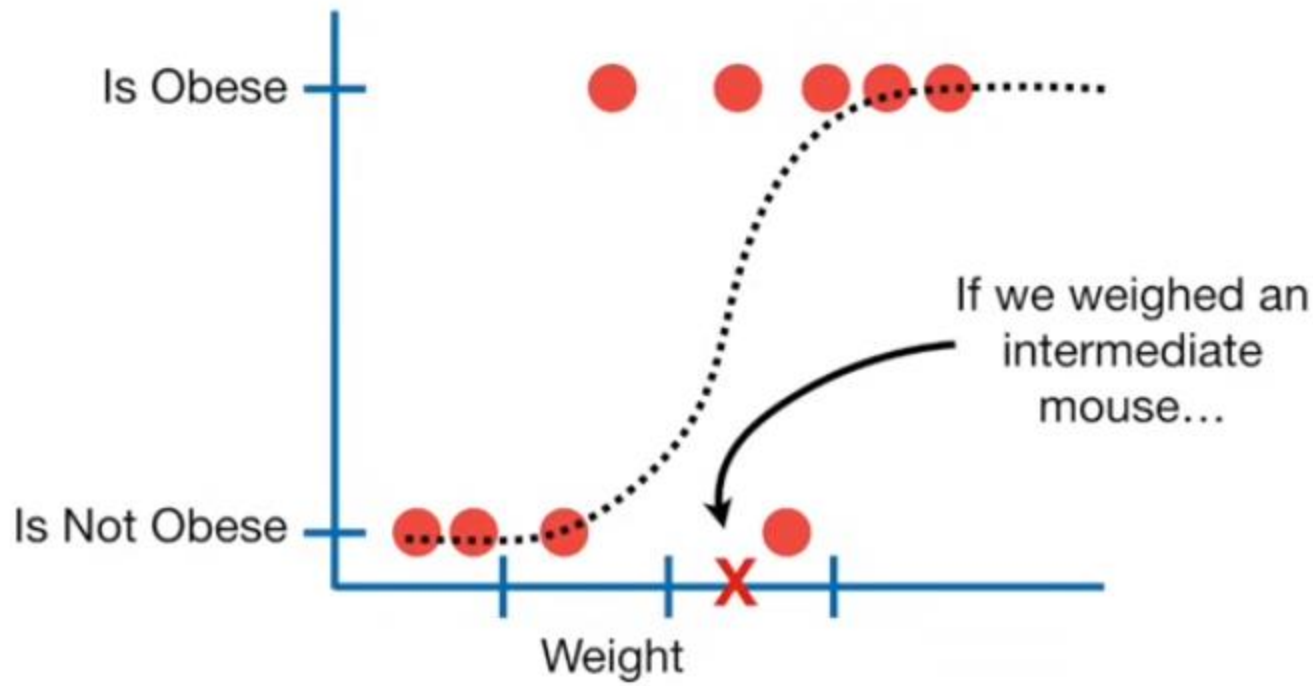

The curve goes
from 0 to 1…
1
0
Weight

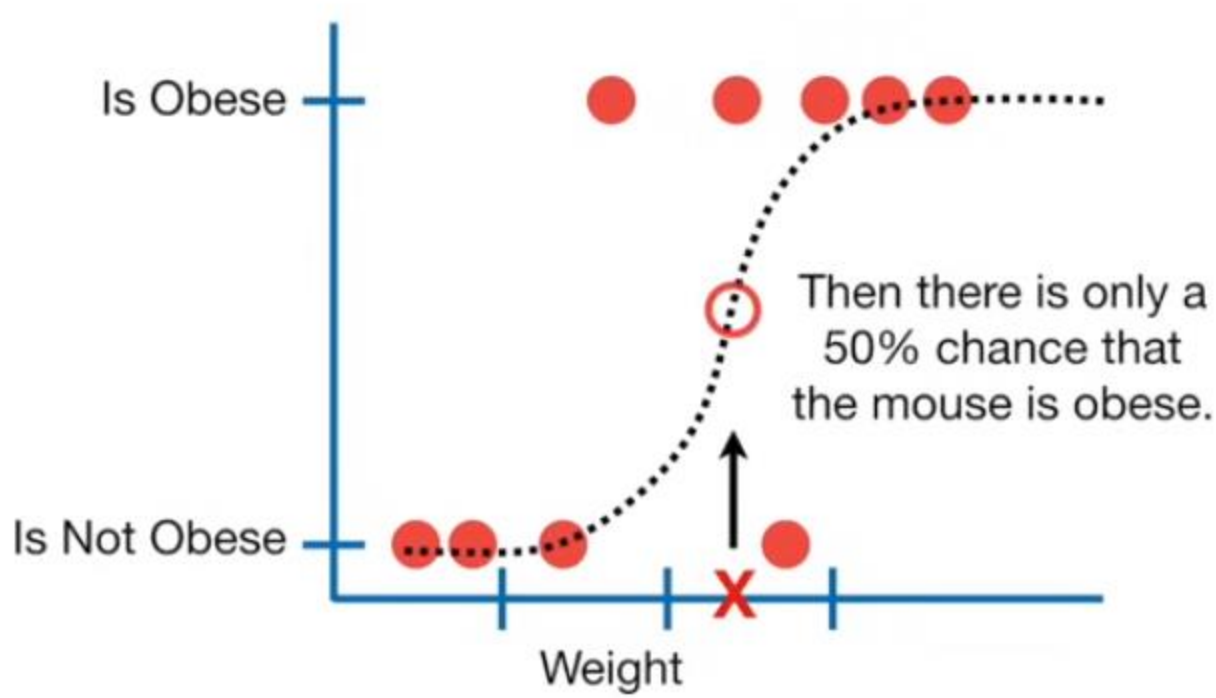
…and that means that the curve tells you the probability that a mouse is ***obese*** based on its ***weight***.

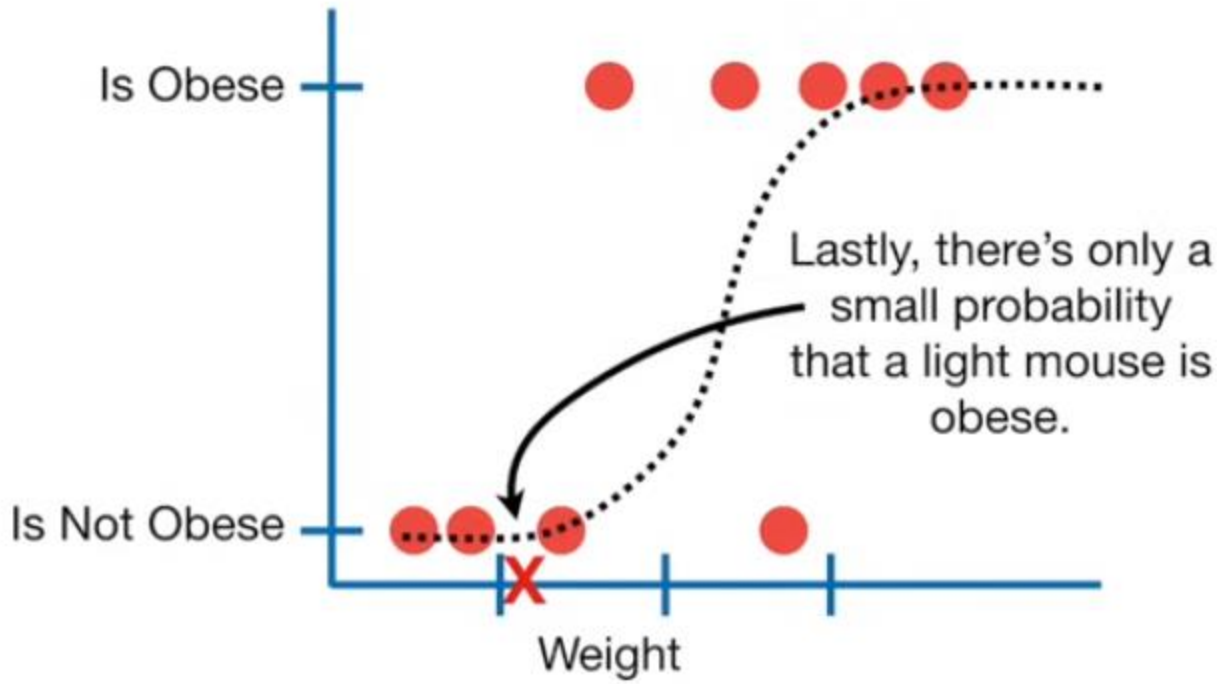



Is Obese
Is Not Obese
Weight
If we weighed a very heavy mouse…

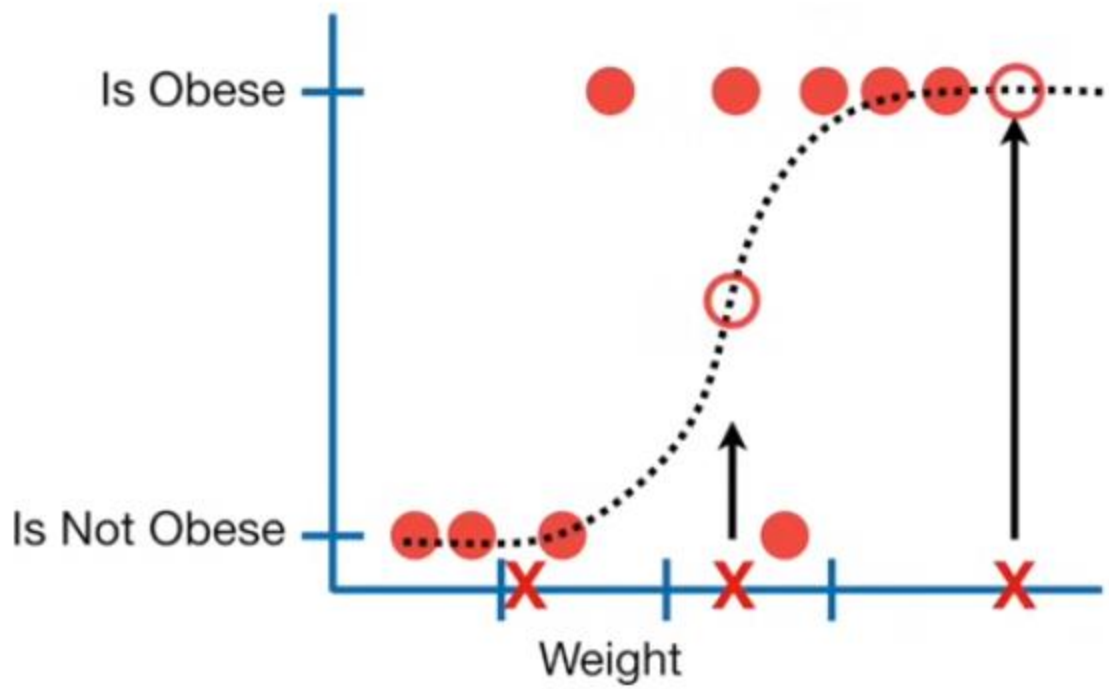

Is Obese
Is Not Obese
Weight
There is a high probability that the new mouse is obese.
X


Is Obese
Is Not Obese
If we weighed an intermediate mouse…
X
Weight

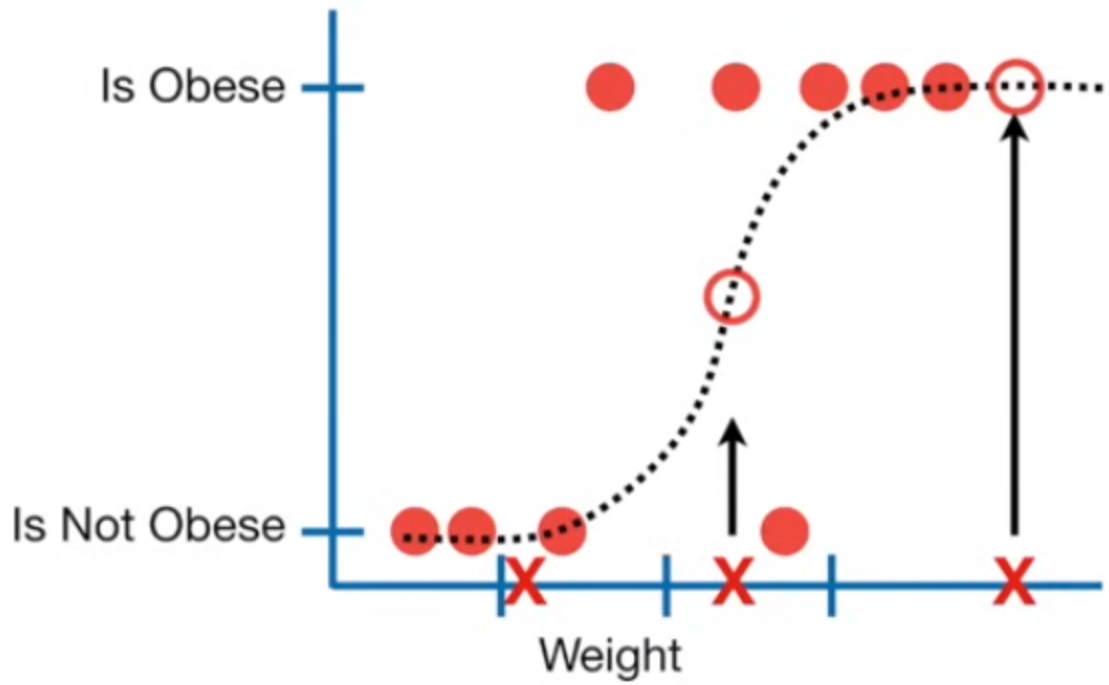

Is Obese
Is Not Obese
Weight
Then there is only a
50% chance that
the mouse is obese.

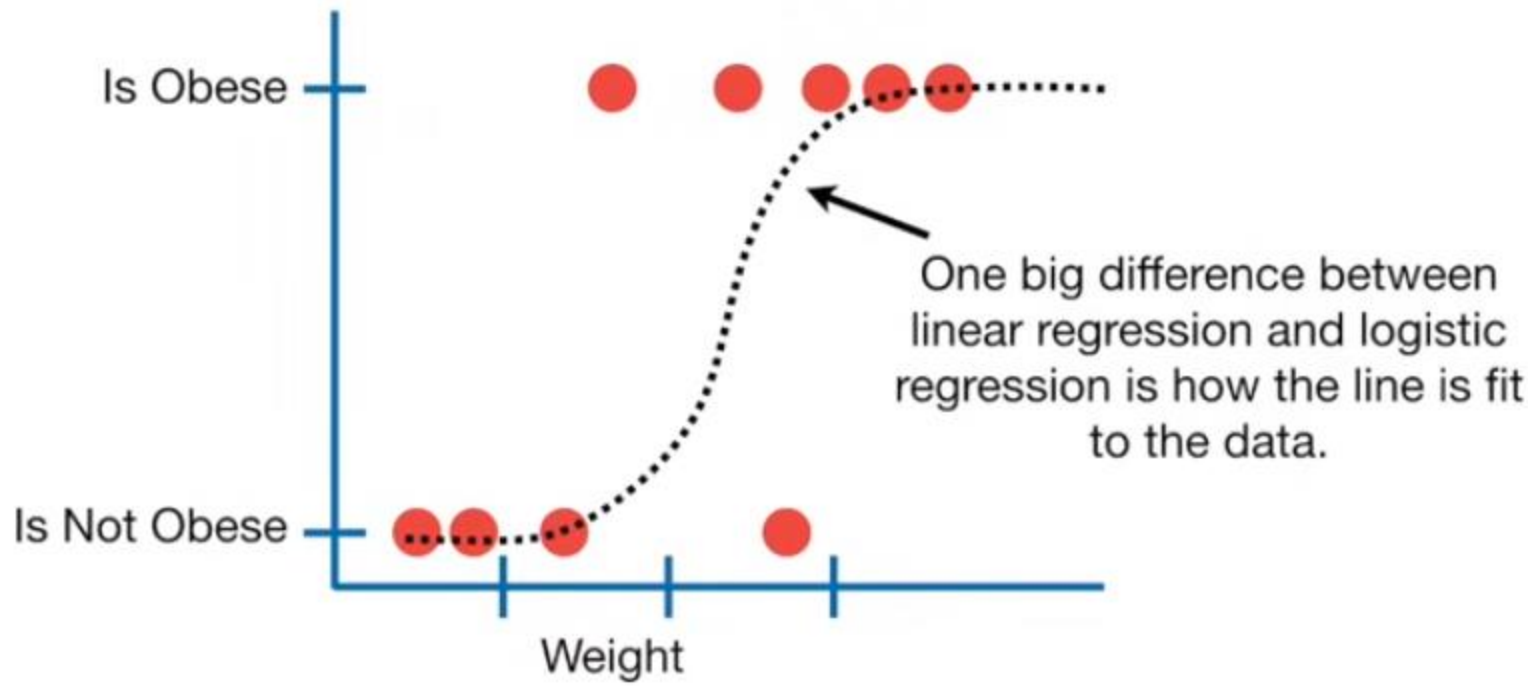

Is Obese
Is Not Obese
Lastly, there's only a small probability that a light mouse is obese.
X
Weight

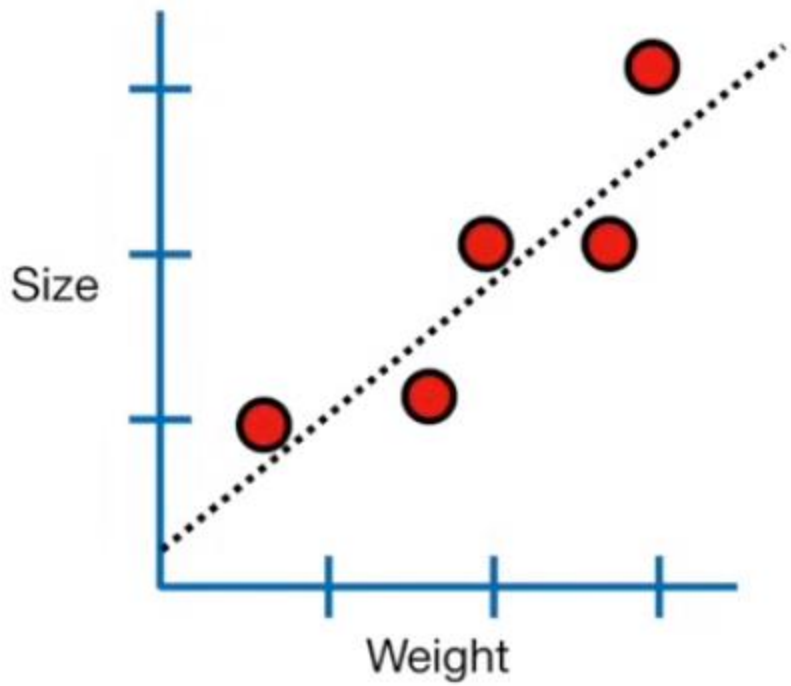
Although logistic regression tells the probability that a mouse is obese or not, it's usually used for classification.



For example, if the probability a mouse is obese is > 50%, then we'll classify it as obese, otherwise we'll classify it as "not obese".

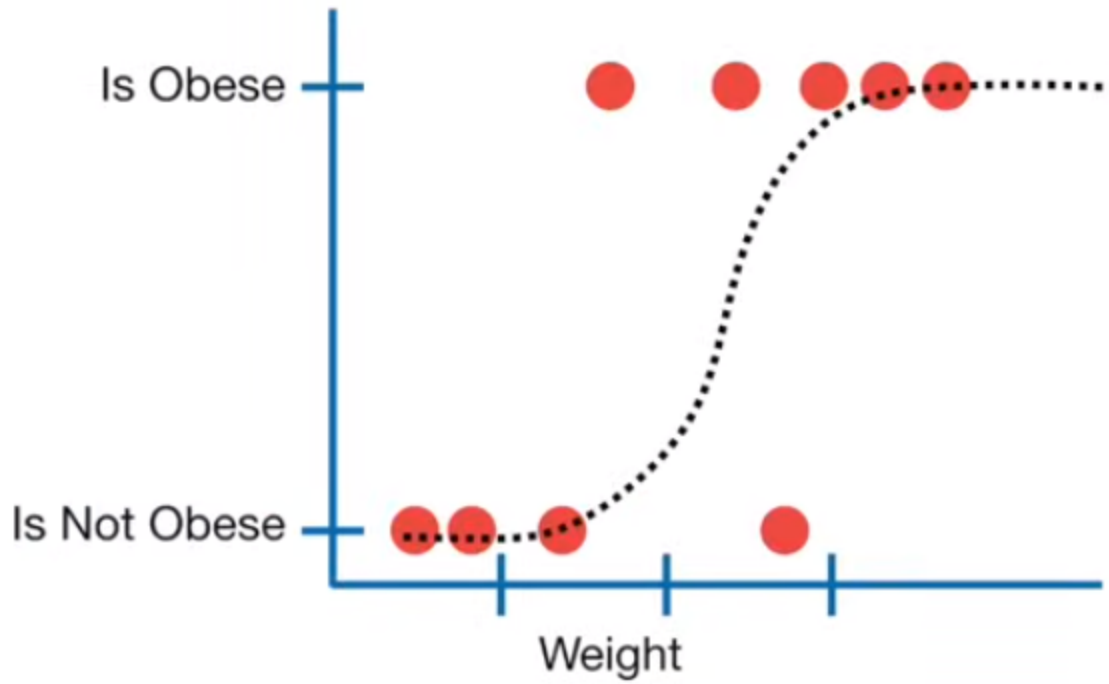



Is Obese
Is Not Obese
Weight
One big difference between linear regression and logistic regression is how the line is fit to the data.

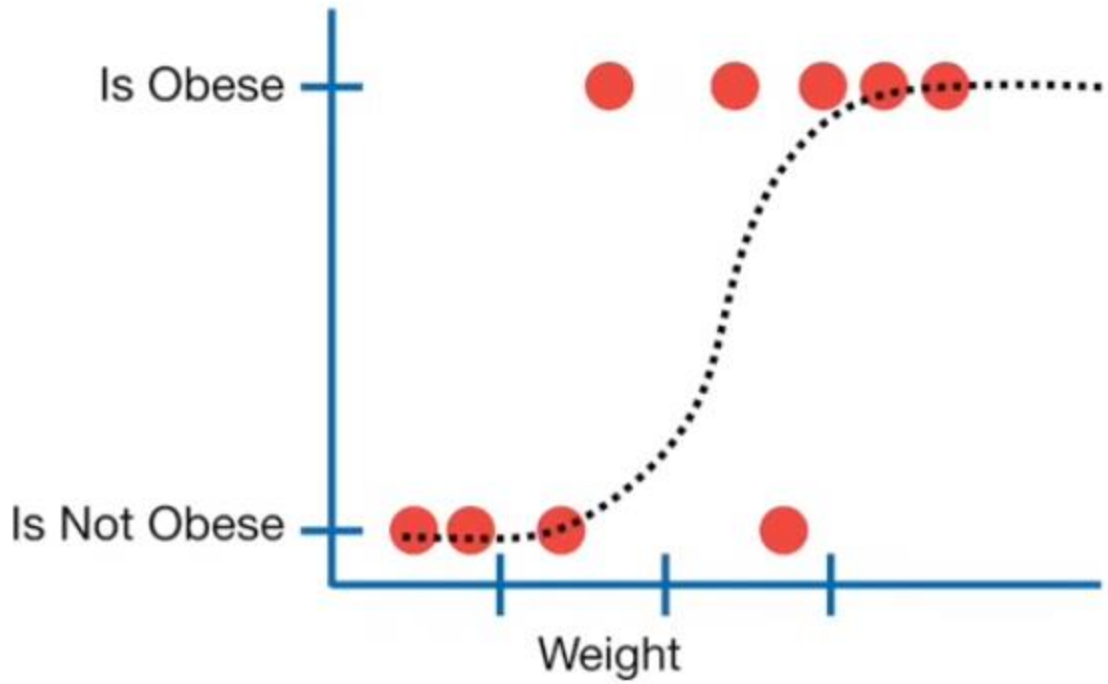


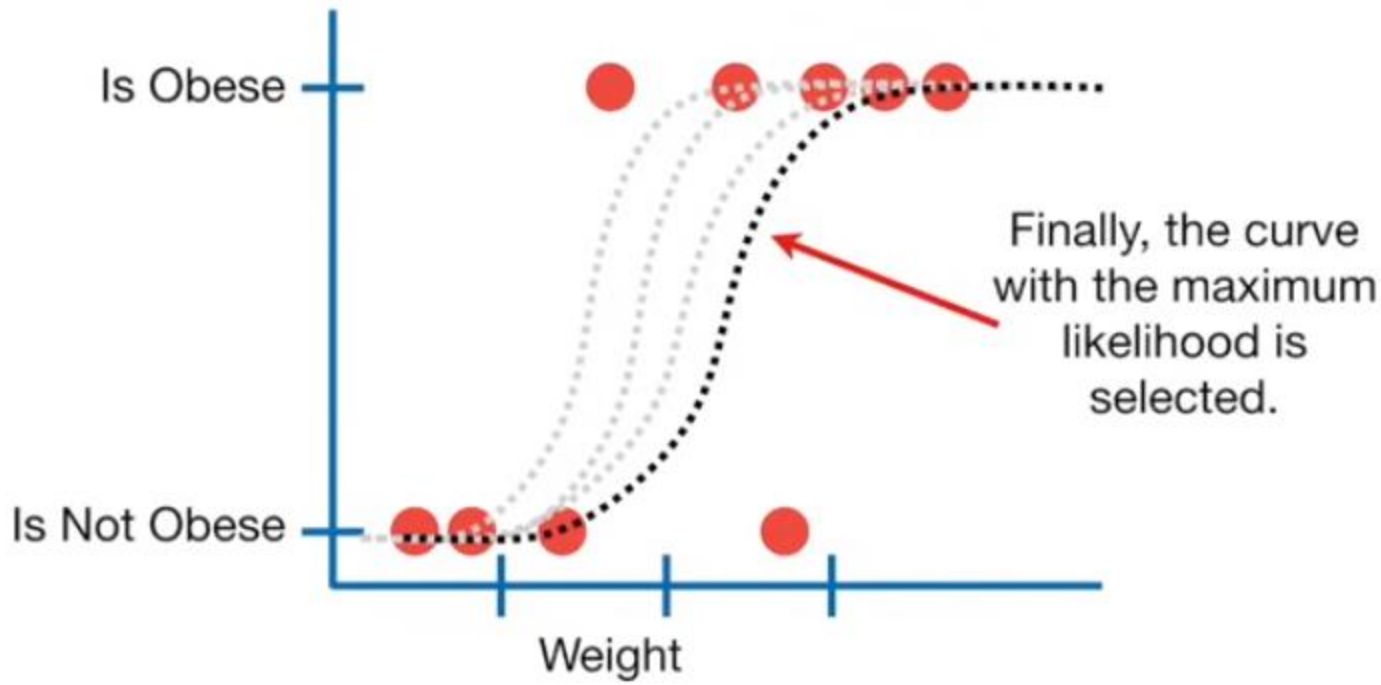
With linear regression, we fit the line using “least squares”.

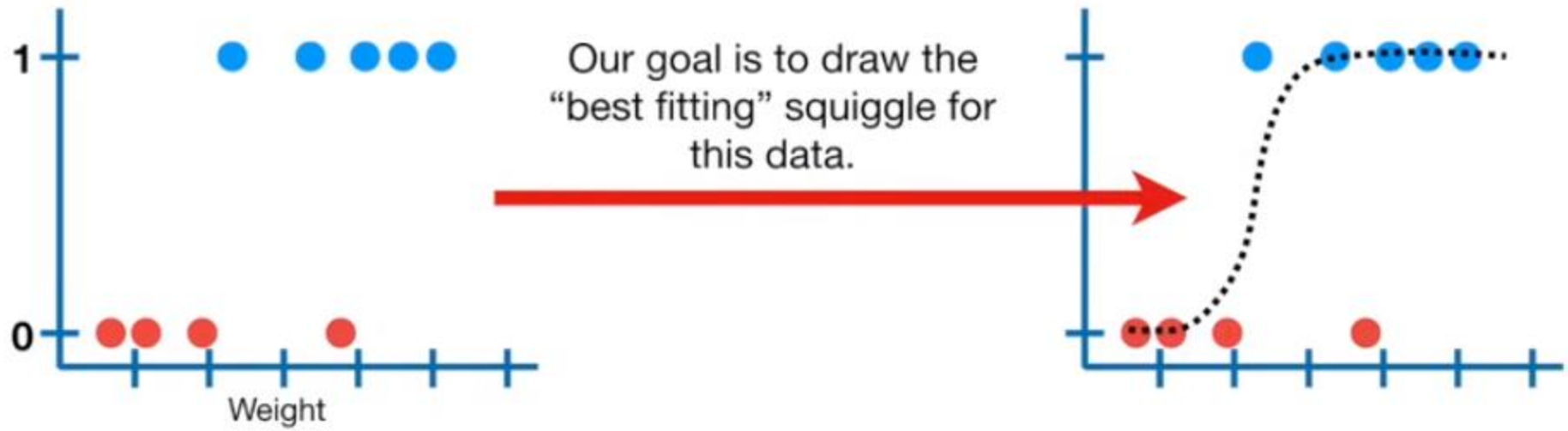
Logistic regression doesn't have the same concept of a "residual", so it can't use least squares and it can't calculate R^2.



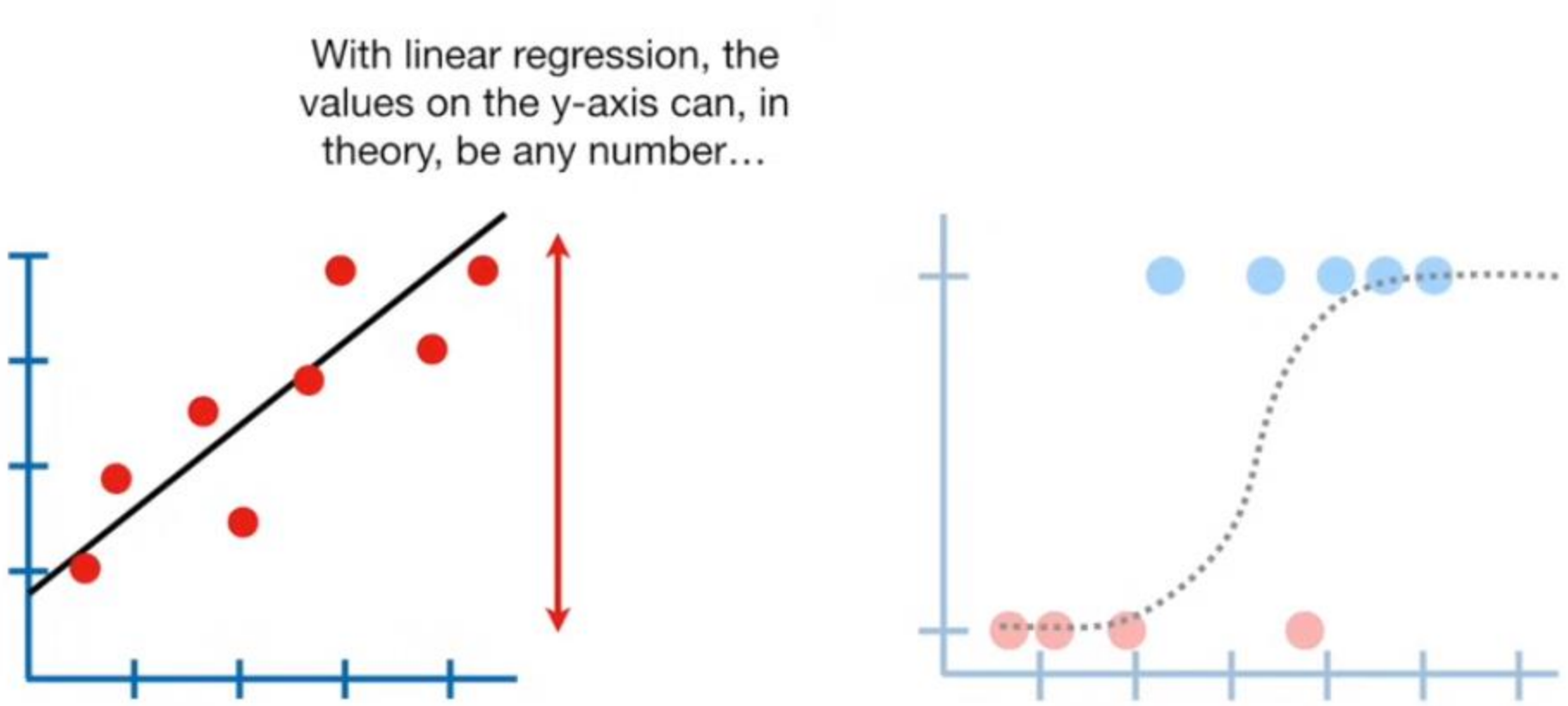
Instead it uses something called
“maximum likelihood”.



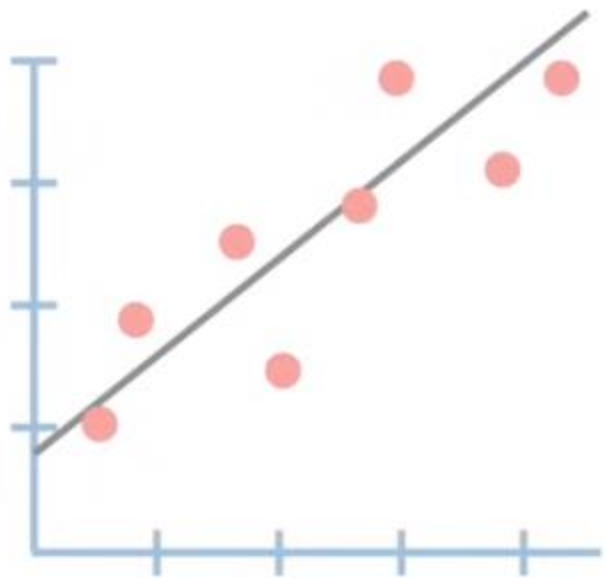

Is Obese
Is Not Obese
Weight
Finally, the curve with the maximum likelihood is selected.

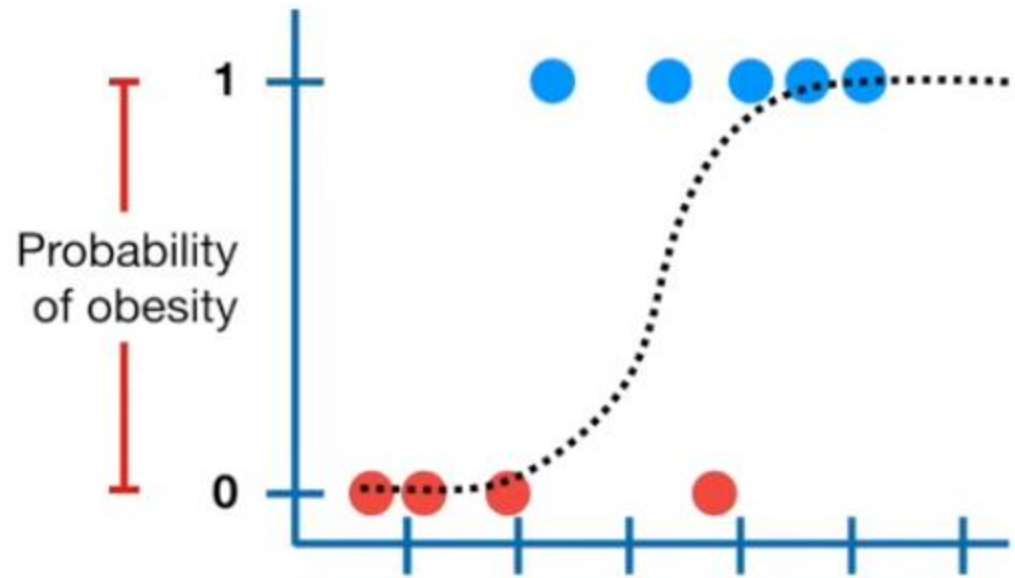

1
0
Weight
Our goal is to draw the
“best fitting” squiggle for
this data.

Transforming axis

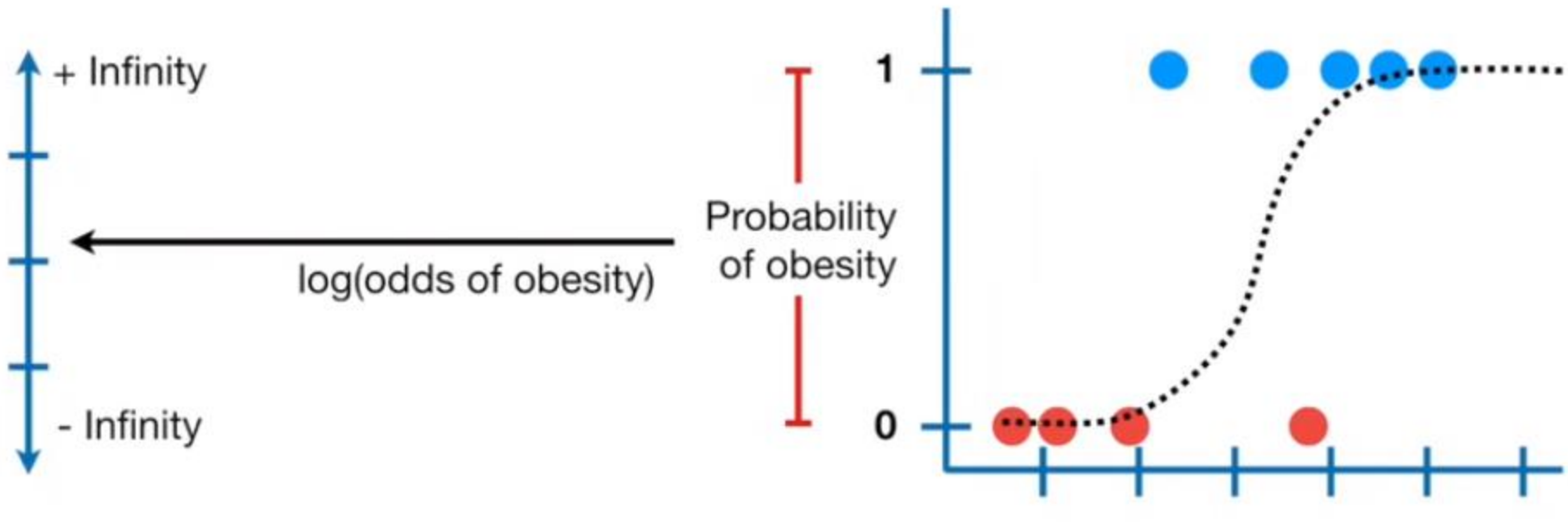


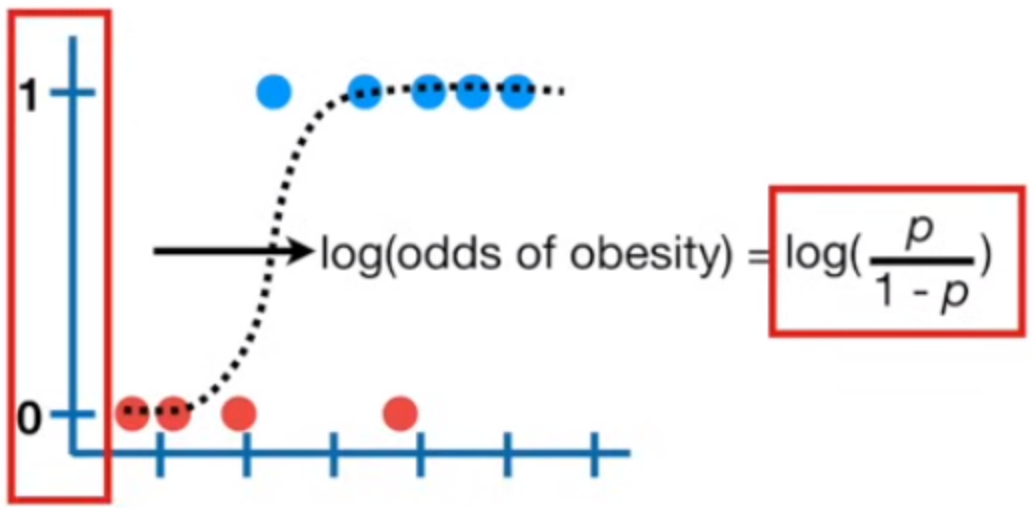
…unfortunately, with logistic regression, the y-axis is confined to probability values between 0 and 1.



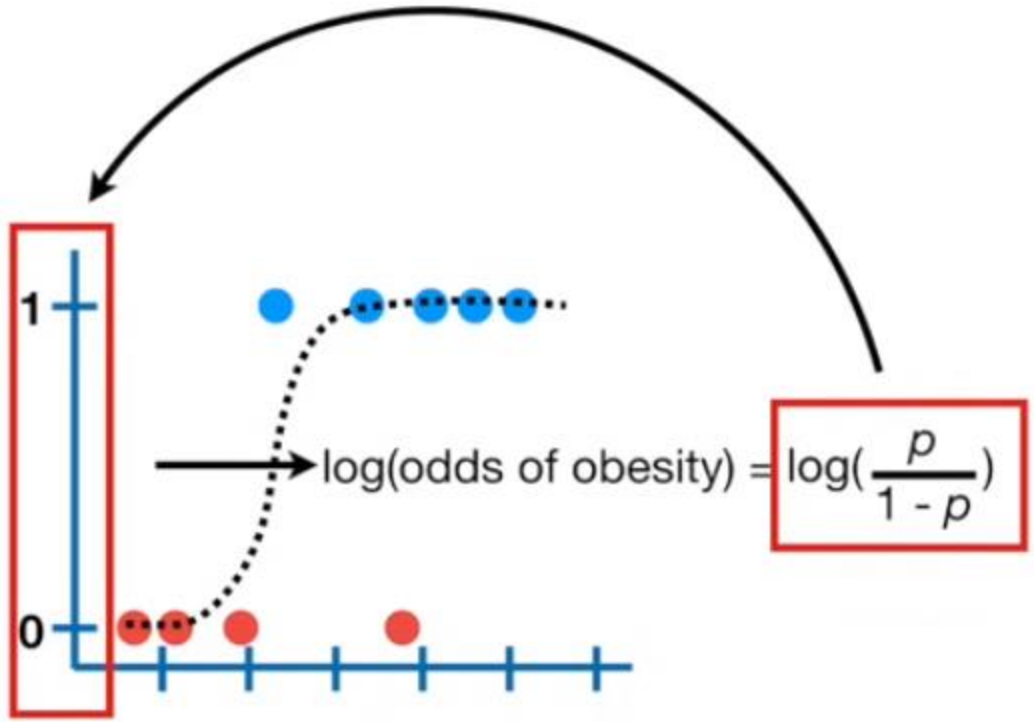


To solve this problem, the y-axis in logistic regression is transformed from the “probability of obesity” to the “log(odds of obesity)” so, just like the y-axis in linear regression, it can go from -infinity to +infinity.

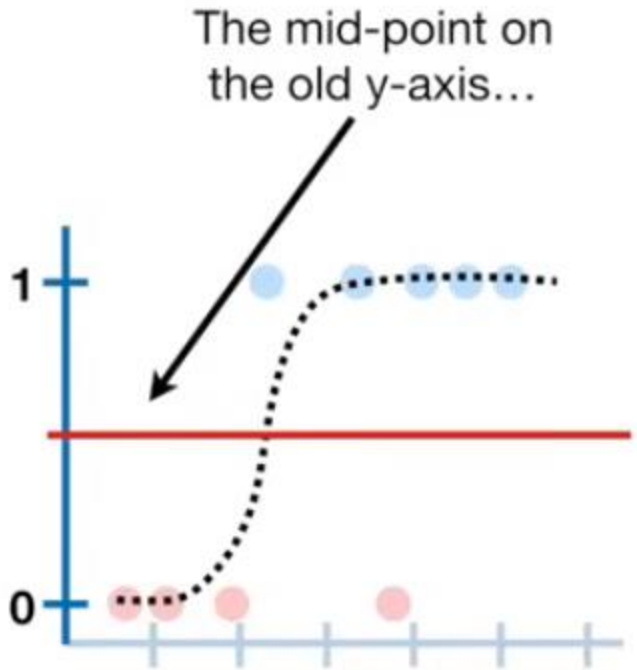


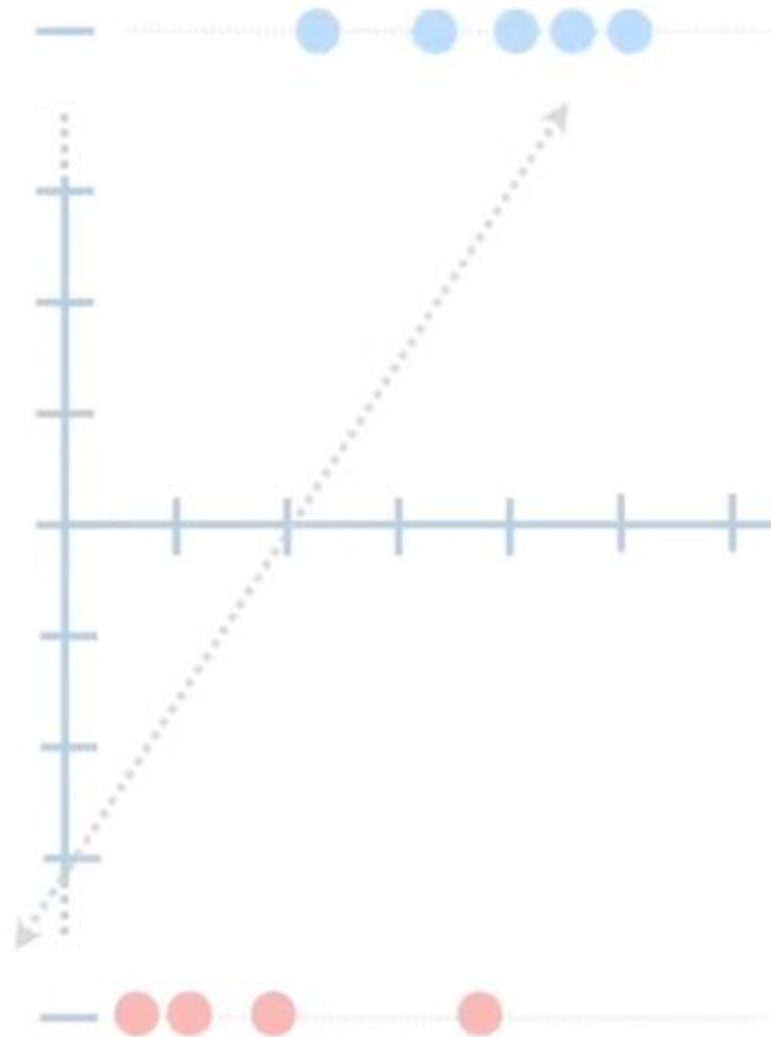


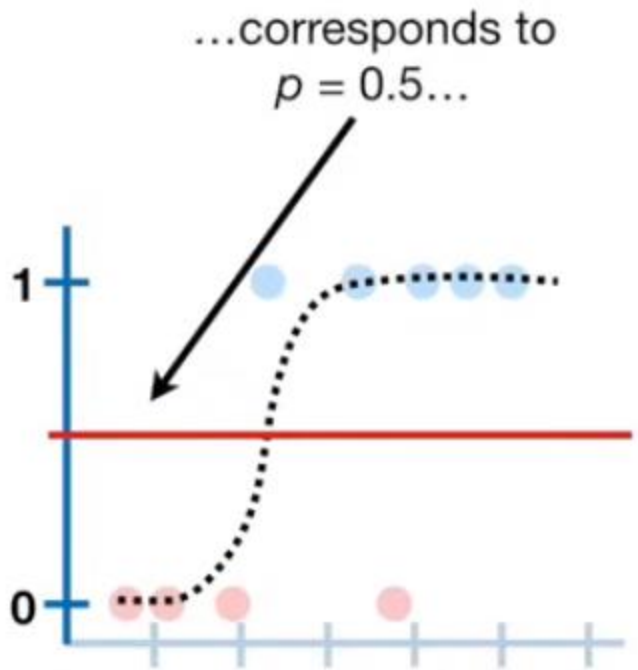
We do that with the **logit** function

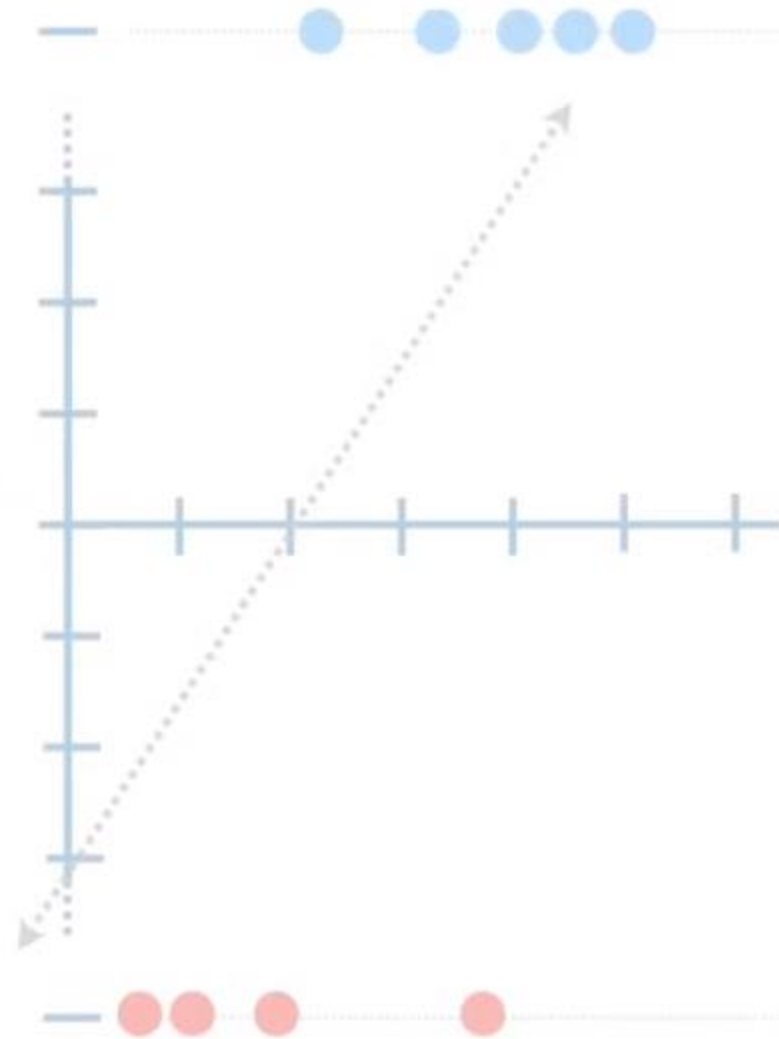


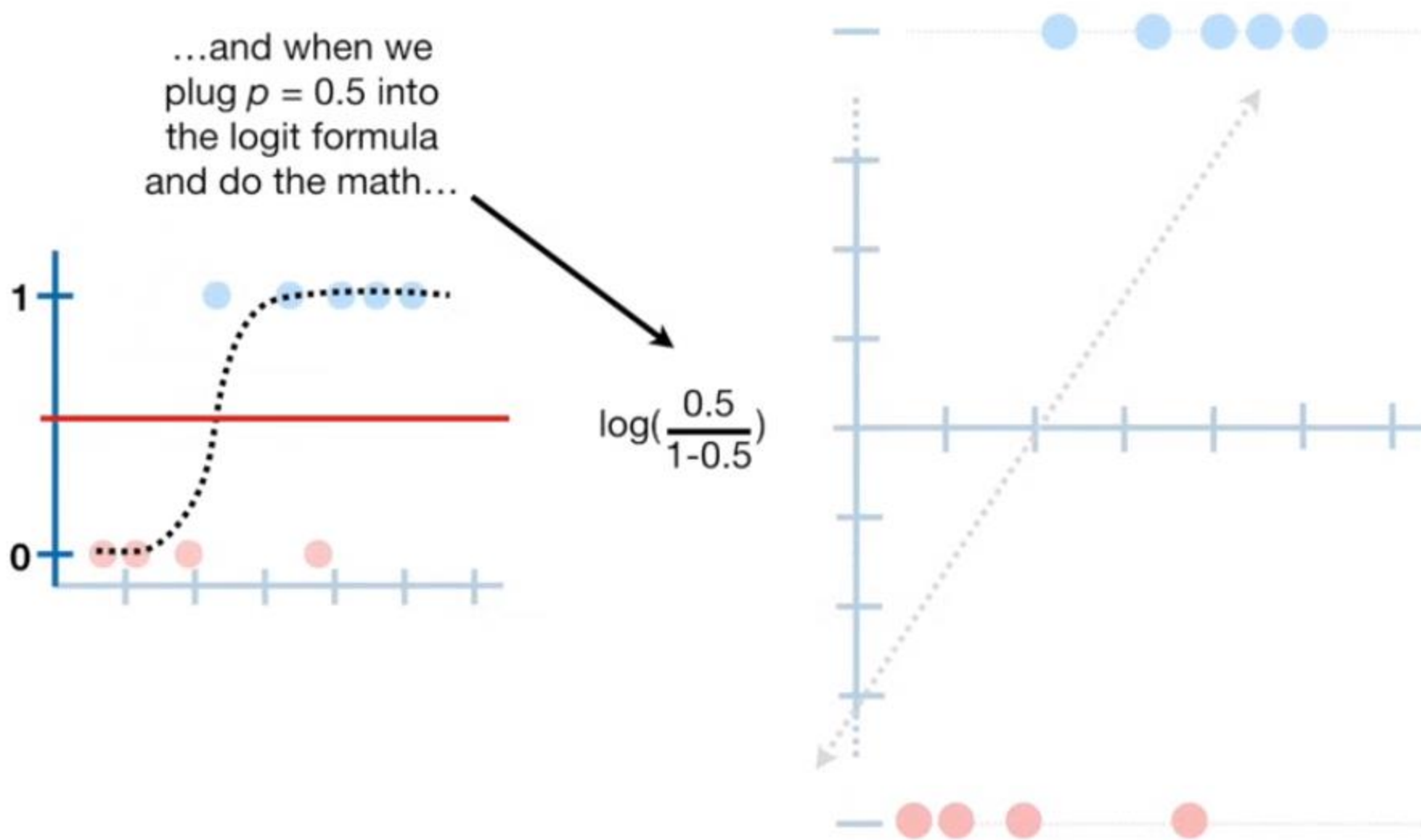
p, in this case, is the probability of a mouse being obese, and corresponds a value on the old y-axis between 0 and 1.

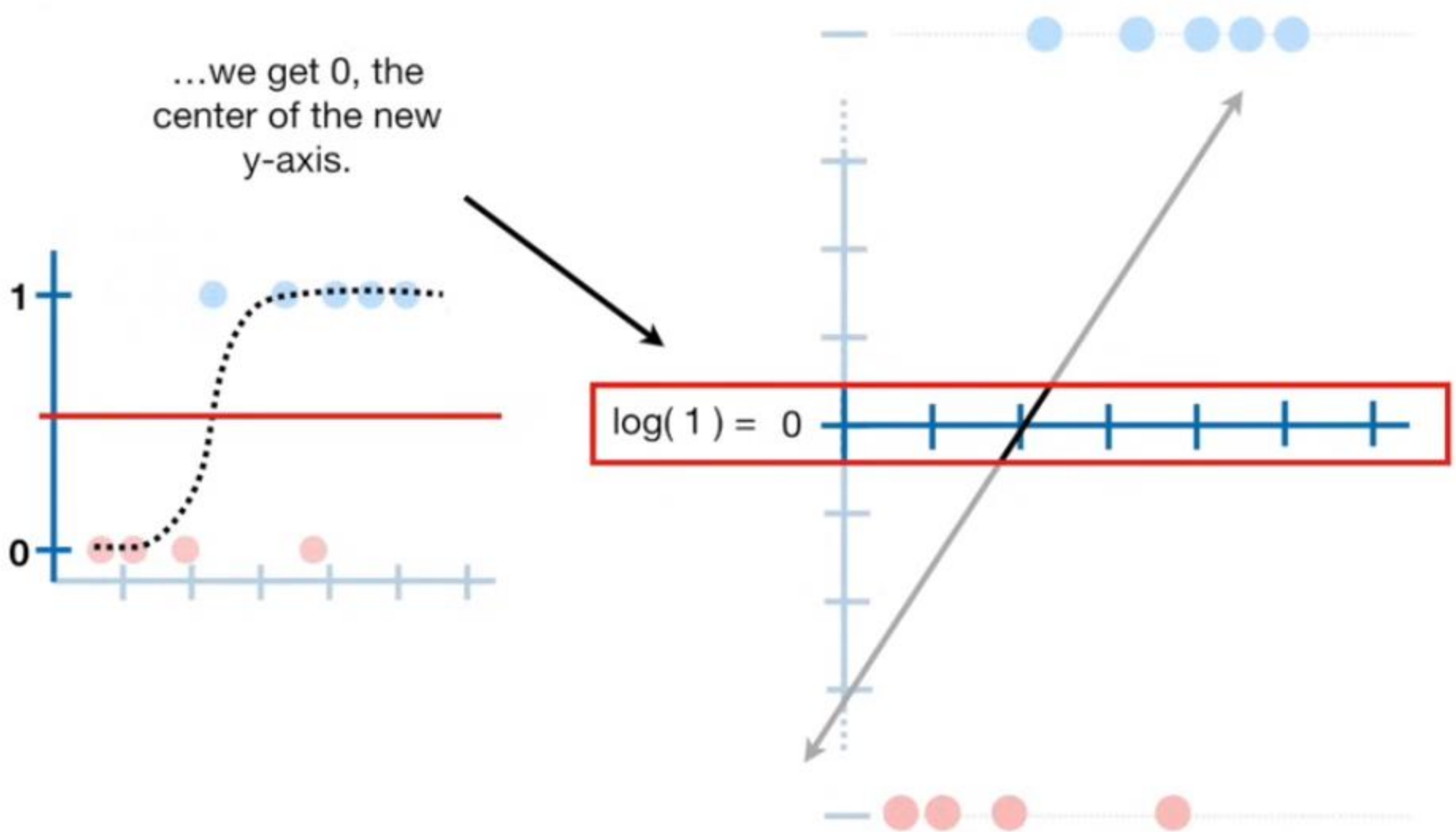

The mid-point on
the old y-axis…
1
0

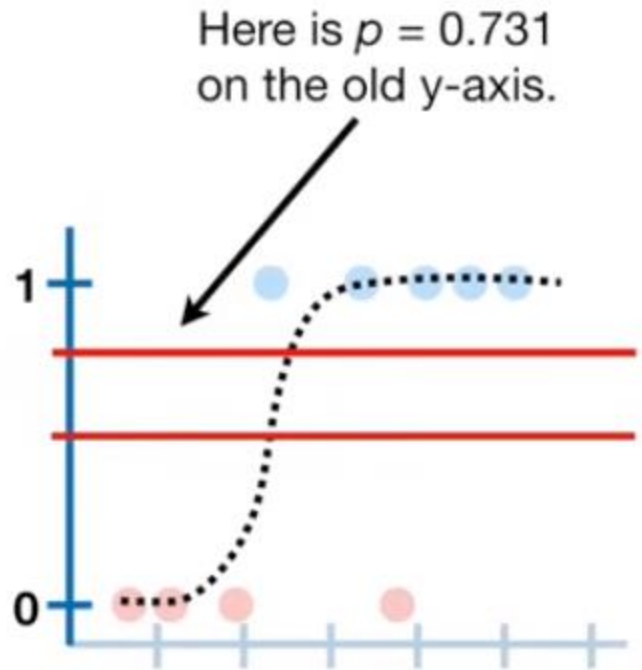


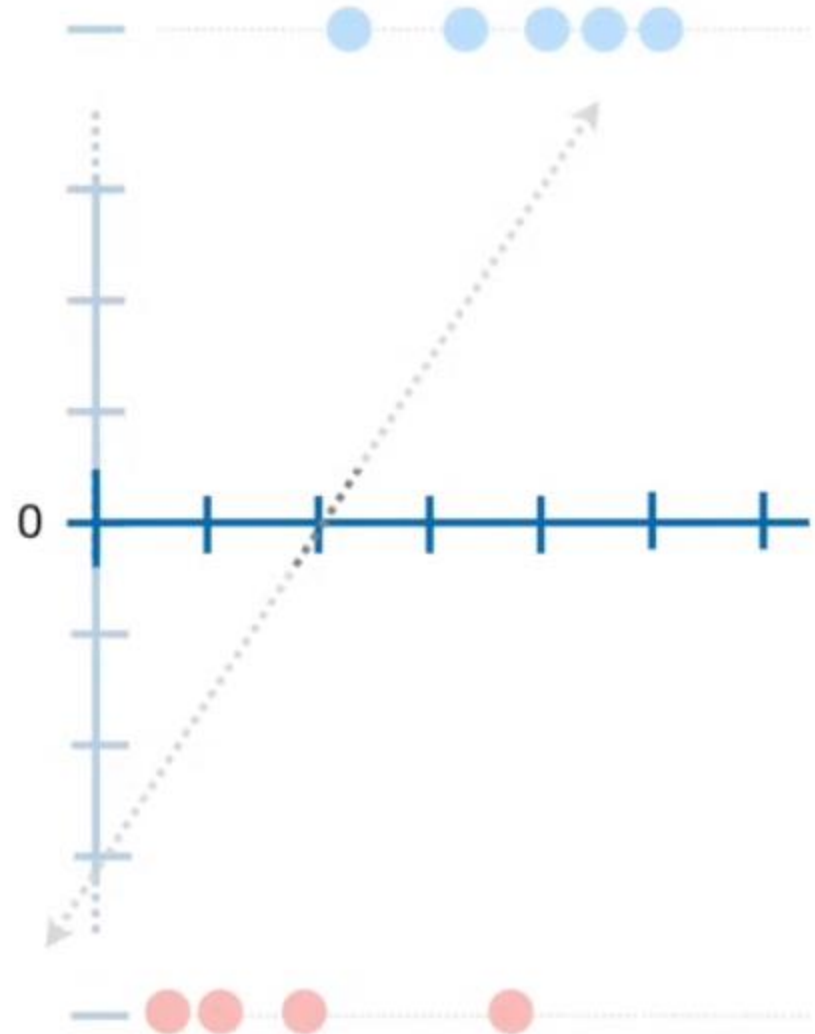

…corresponds to
p = 0.5…
1
0

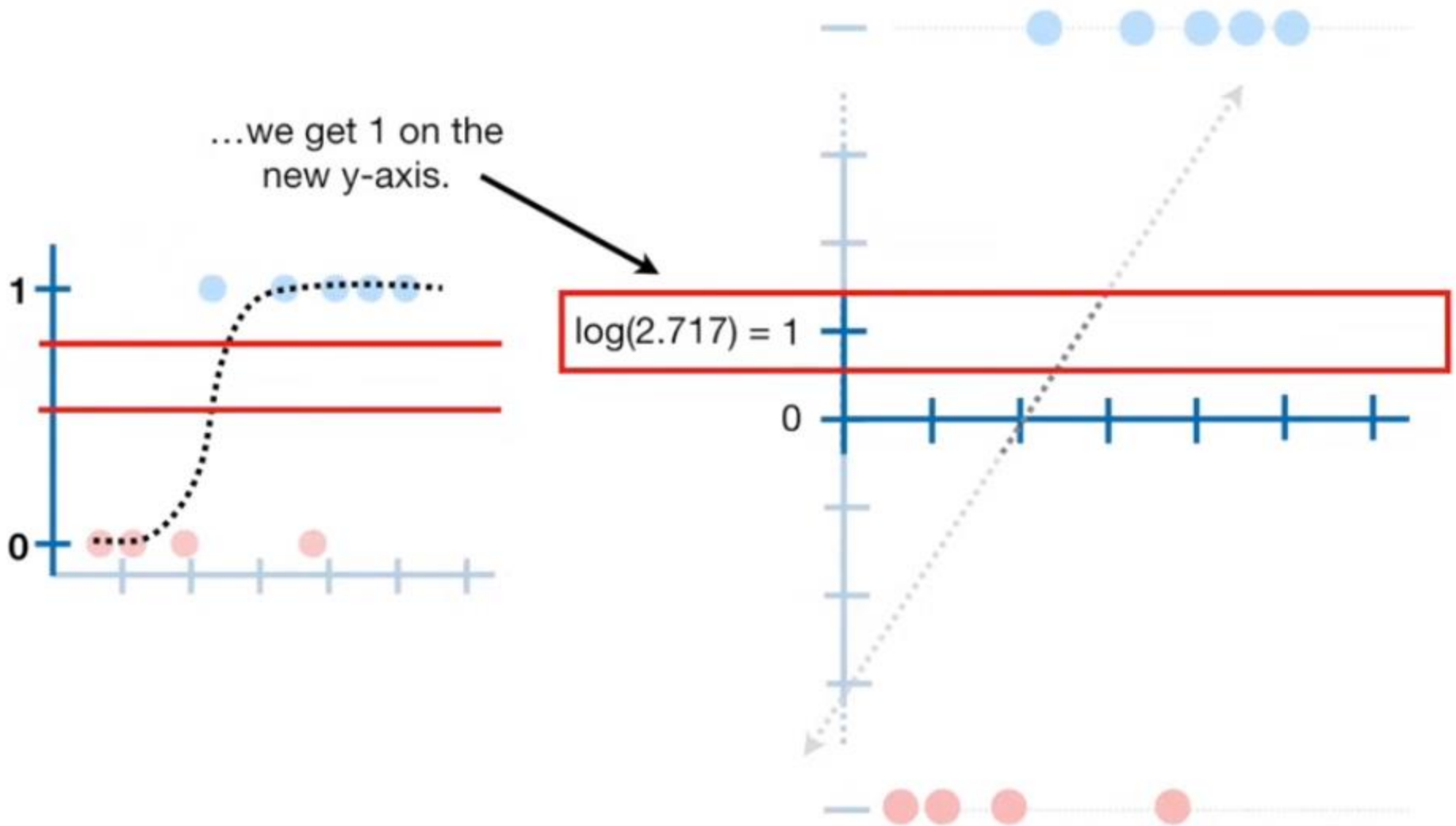



…and when we plug $p = 0.5$ into the logit formula and do the math…
1
0
$\log(\frac{0.5}{1-0.5})$


…we get 0, the center of the new y-axis.
1
0
log(1) = 0

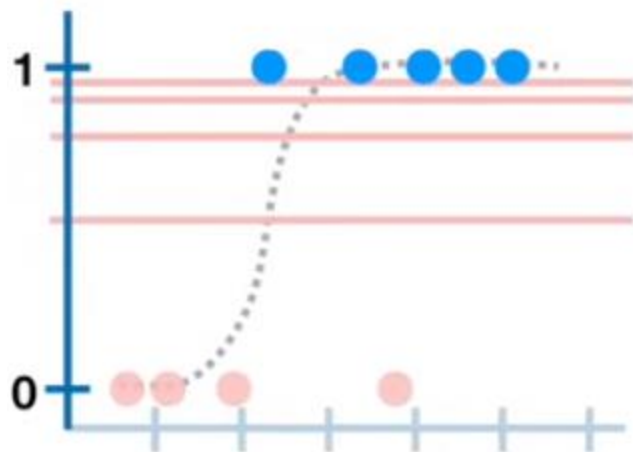

Here is $p = 0.731$
on the old y-axis.
1
0

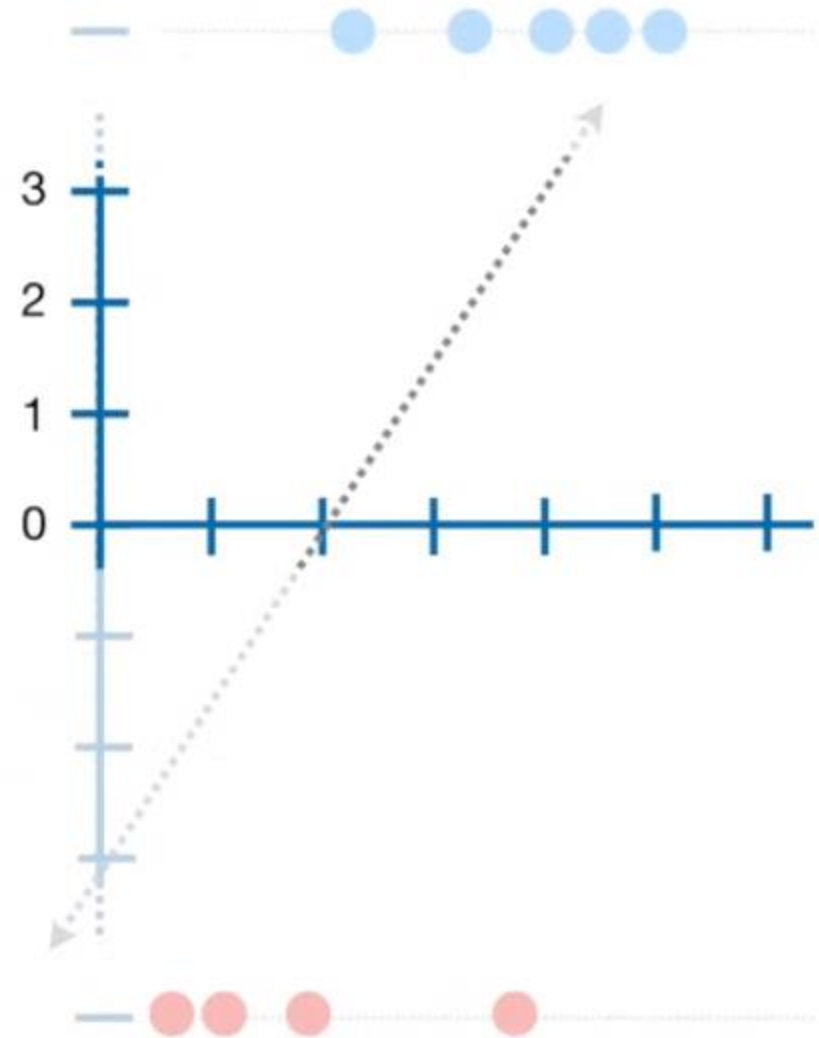

0


…we get 1 on the
new y-axis.
1
0
log(2.717) = 1
0

If we plug $p = 1$ into the logit function and do the math…

$$\log\left(\frac{1}{0}\right)$$



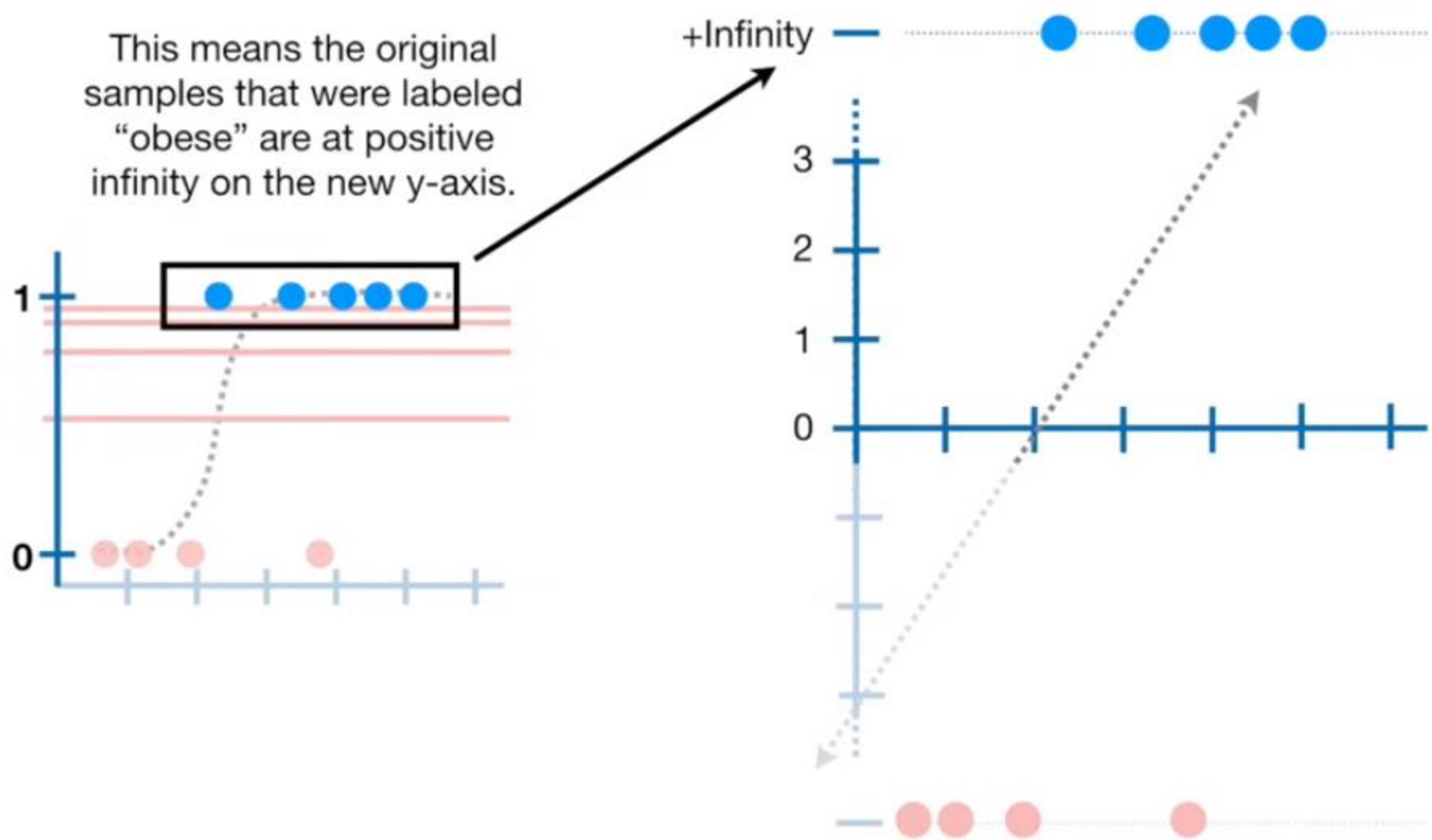


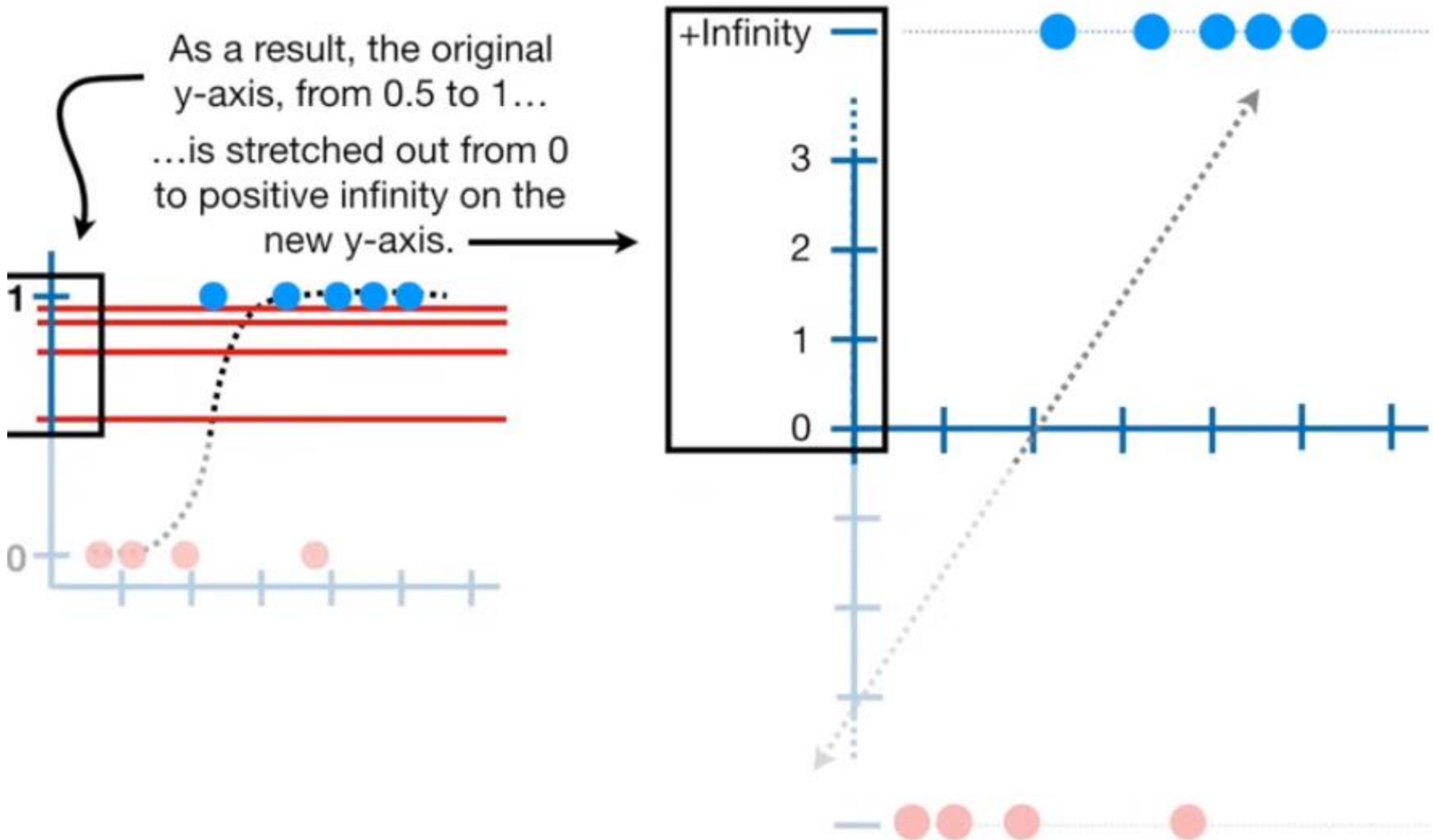
…technically, you can't divide by 0, however… → $\log(\frac{1}{0})$

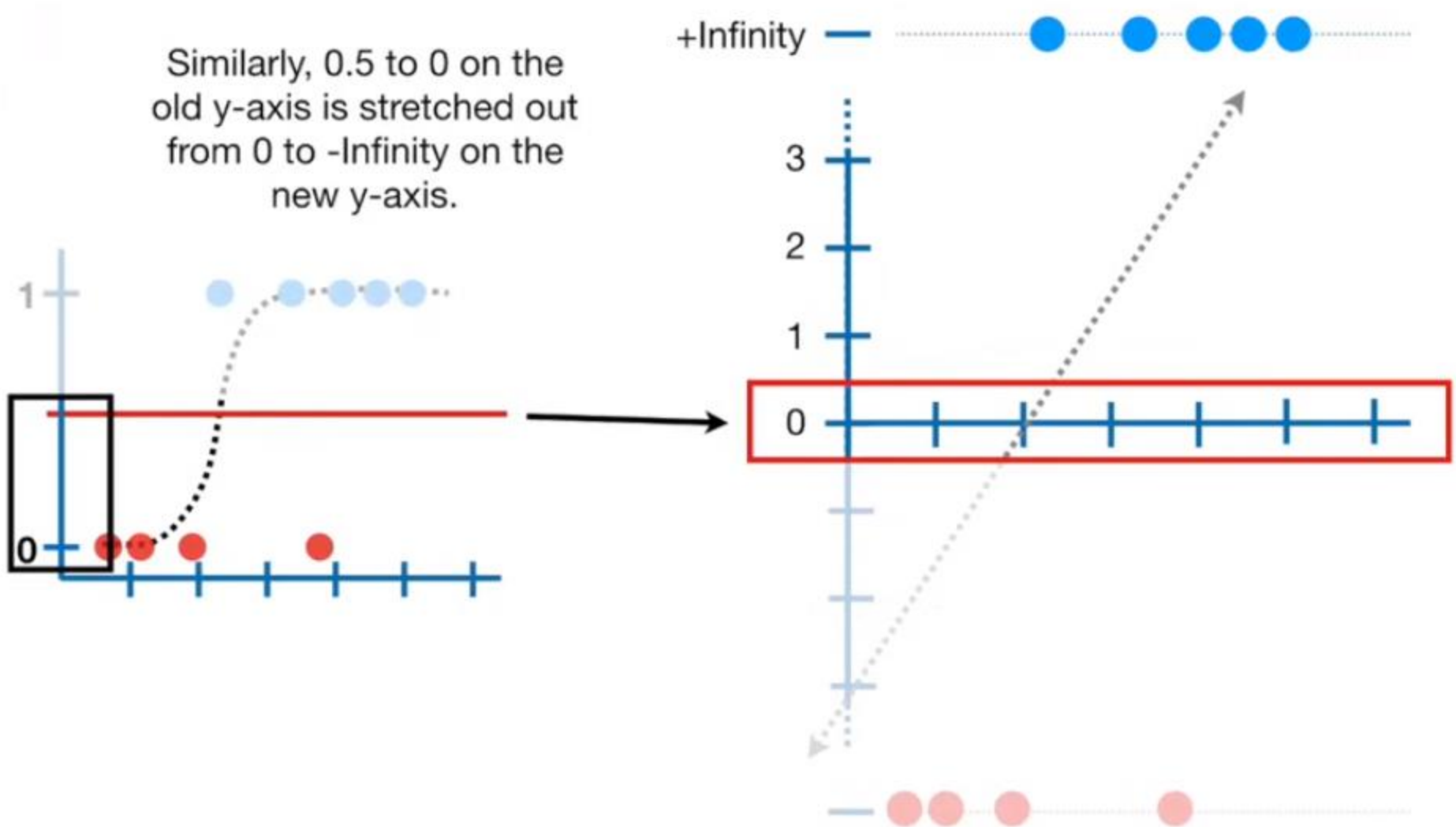
=

$\log(1) - \log(0)$

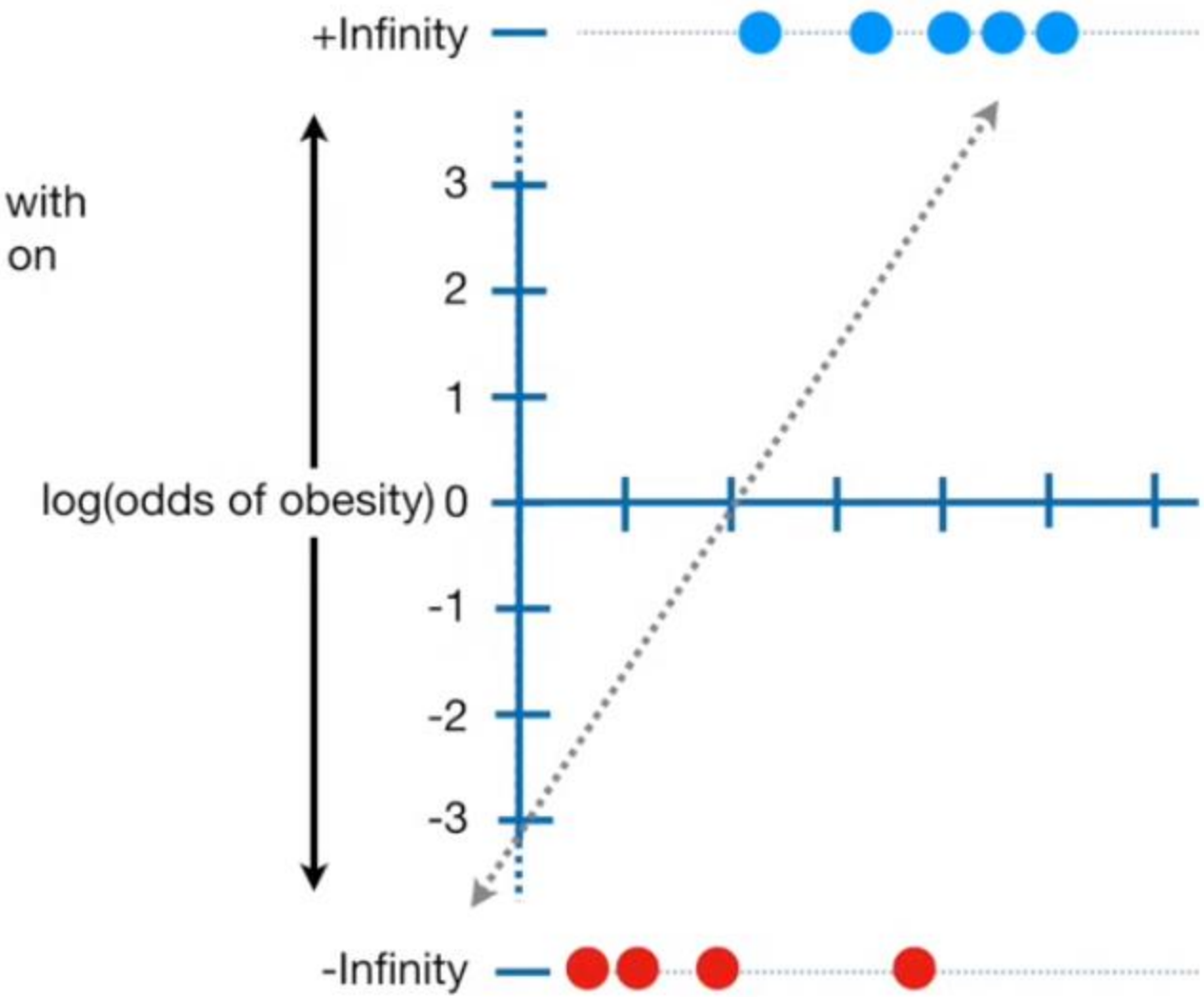
…and since
"something - negative infinity = positive infinity",
this whole thing is
equal to positive infinity.

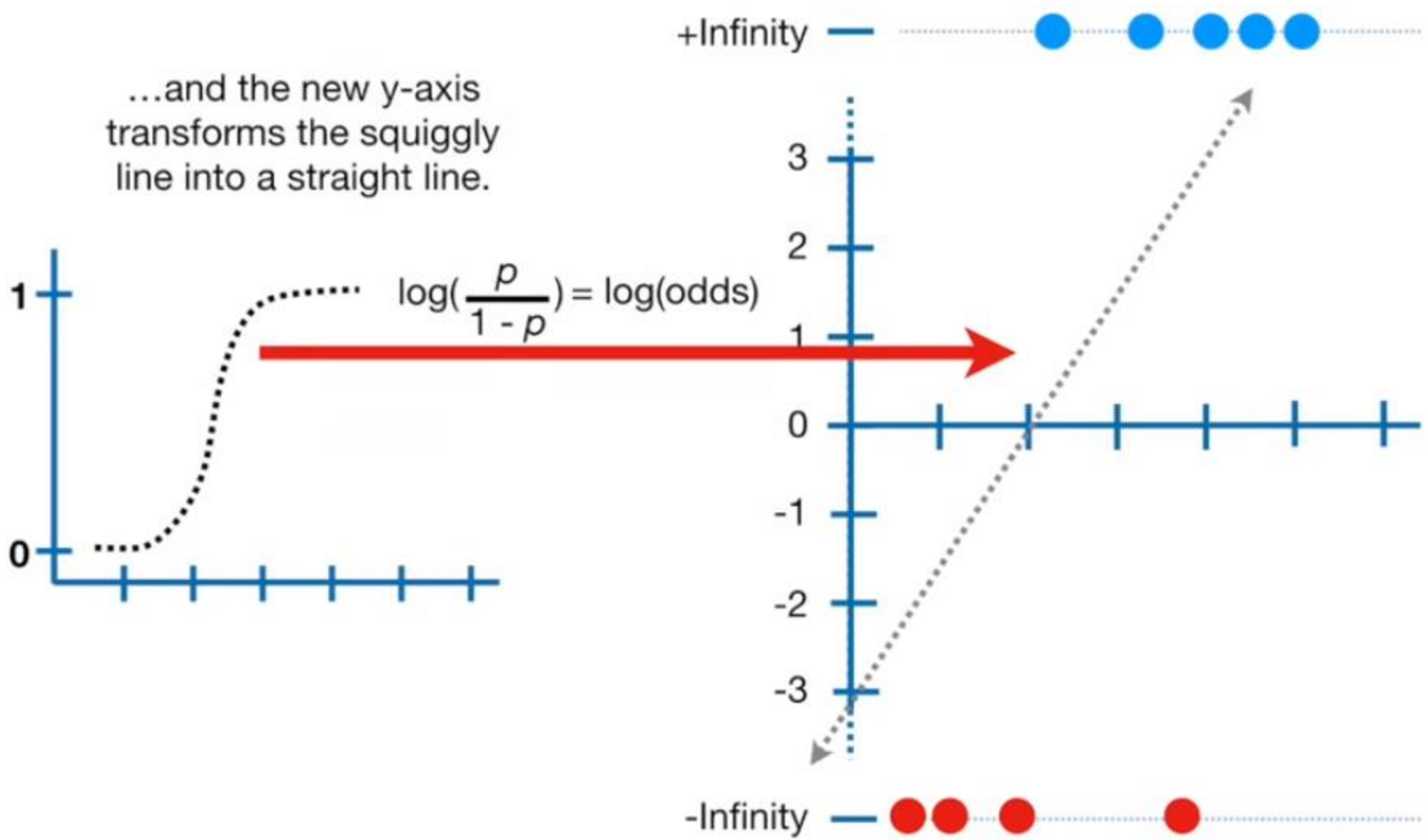

This means the original samples that were labeled “obese” are at positive infinity on the new y-axis.
1
0
+Infinity
3
2
1
0

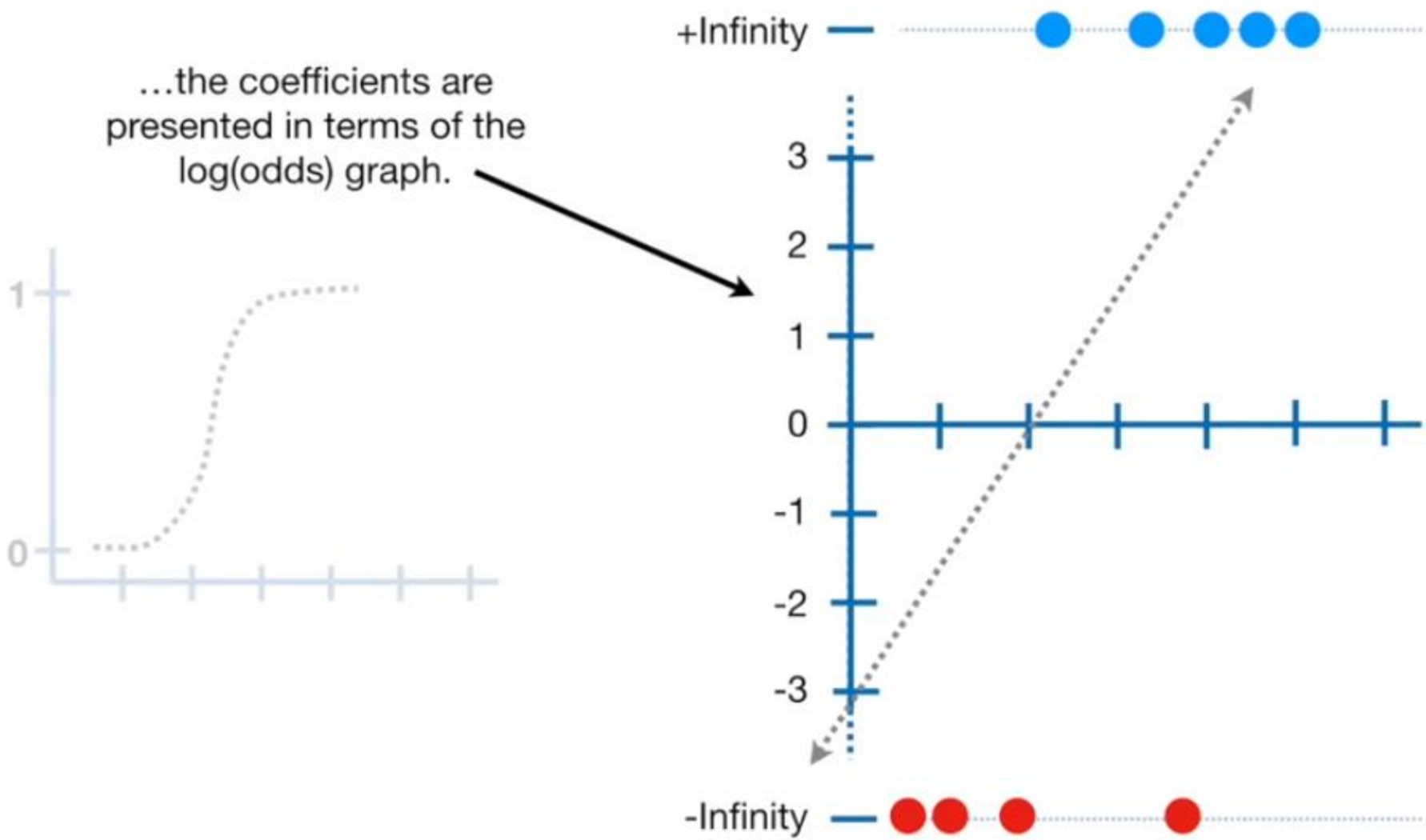

As a result, the original y-axis, from 0.5 to 1…
…is stretched out from 0 to positive infinity on the new y-axis.
1
0
+Infinity
3
2
1
0


Similarly, 0.5 to 0 on the old y-axis is stretched out from 0 to -Infinity on the new y-axis.
1
0
+Infinity
3
2
1
0

Ultimately we end up with log(odds of obesity) on the new y-axis…

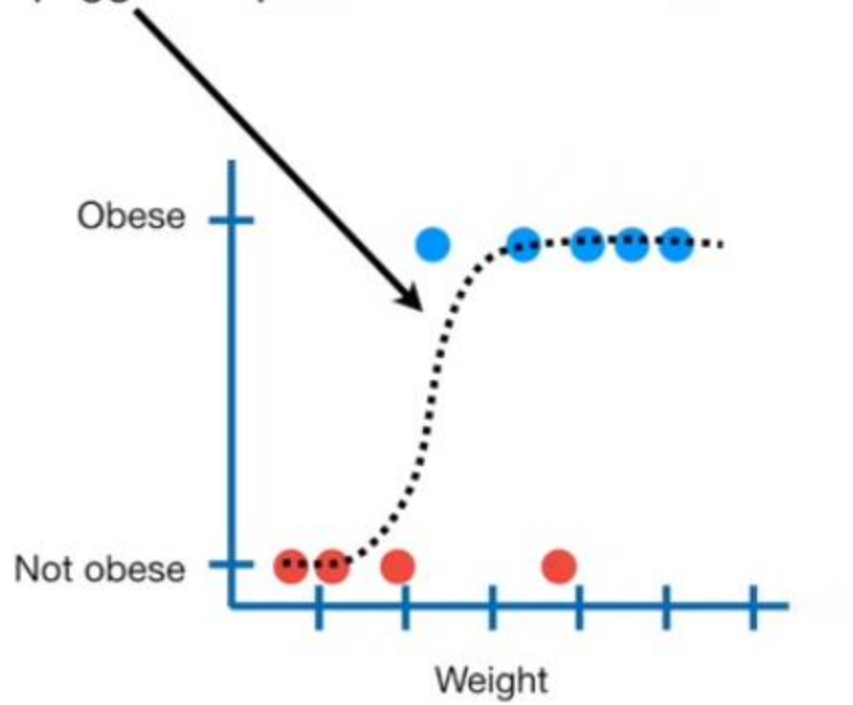


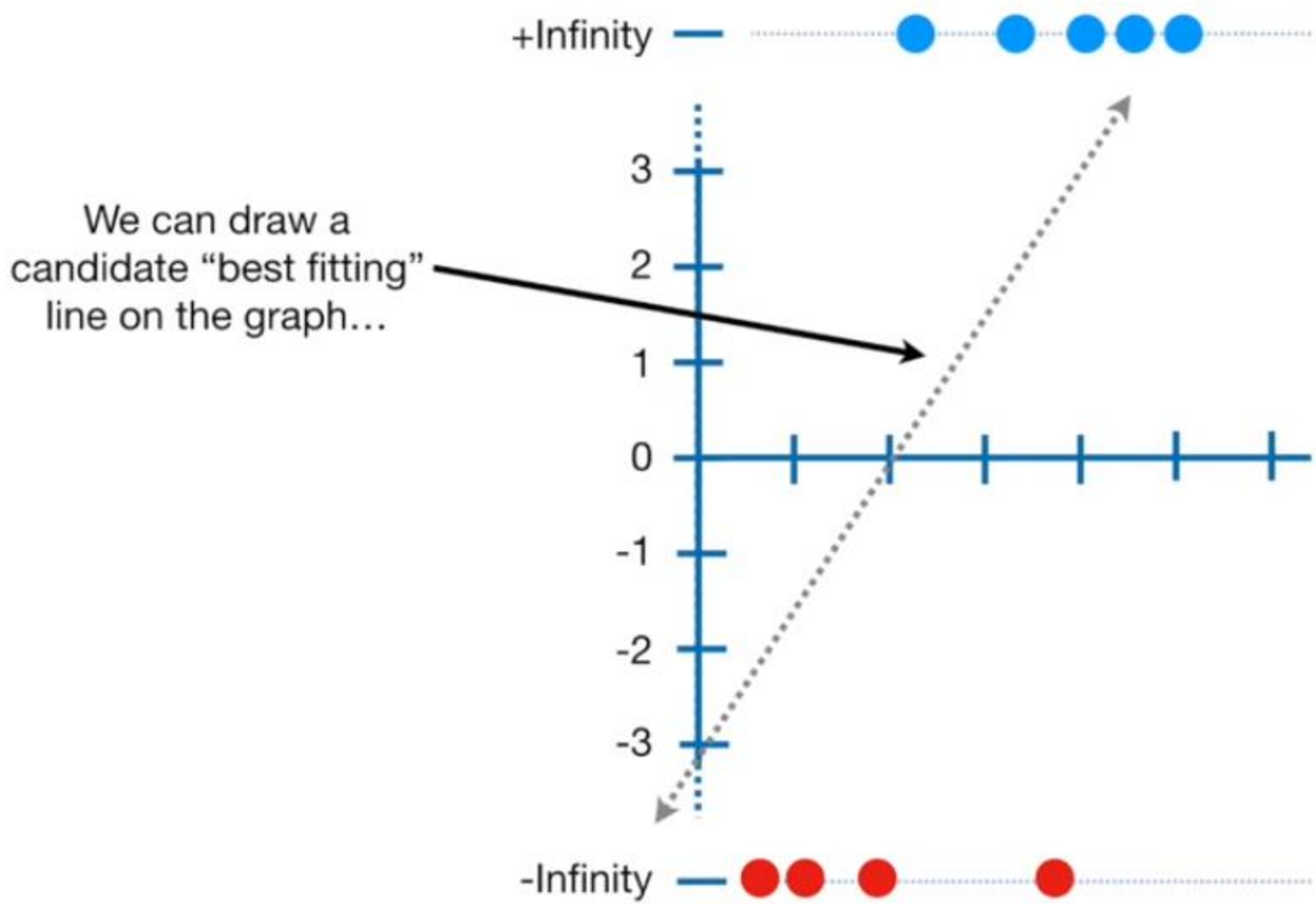

…and the new y-axis
transforms the squiggly
line into a straight line.
1
0
$\log(\frac{p}{1-p}) = \log(odds)$
+Infinity
3
2
1
0
-1
-2
-3
-Infinity


…the coefficients are presented in terms of the log(odds) graph.
1
0
+Infinity
3
2
1
0
-1
-2
-3
-Infinity

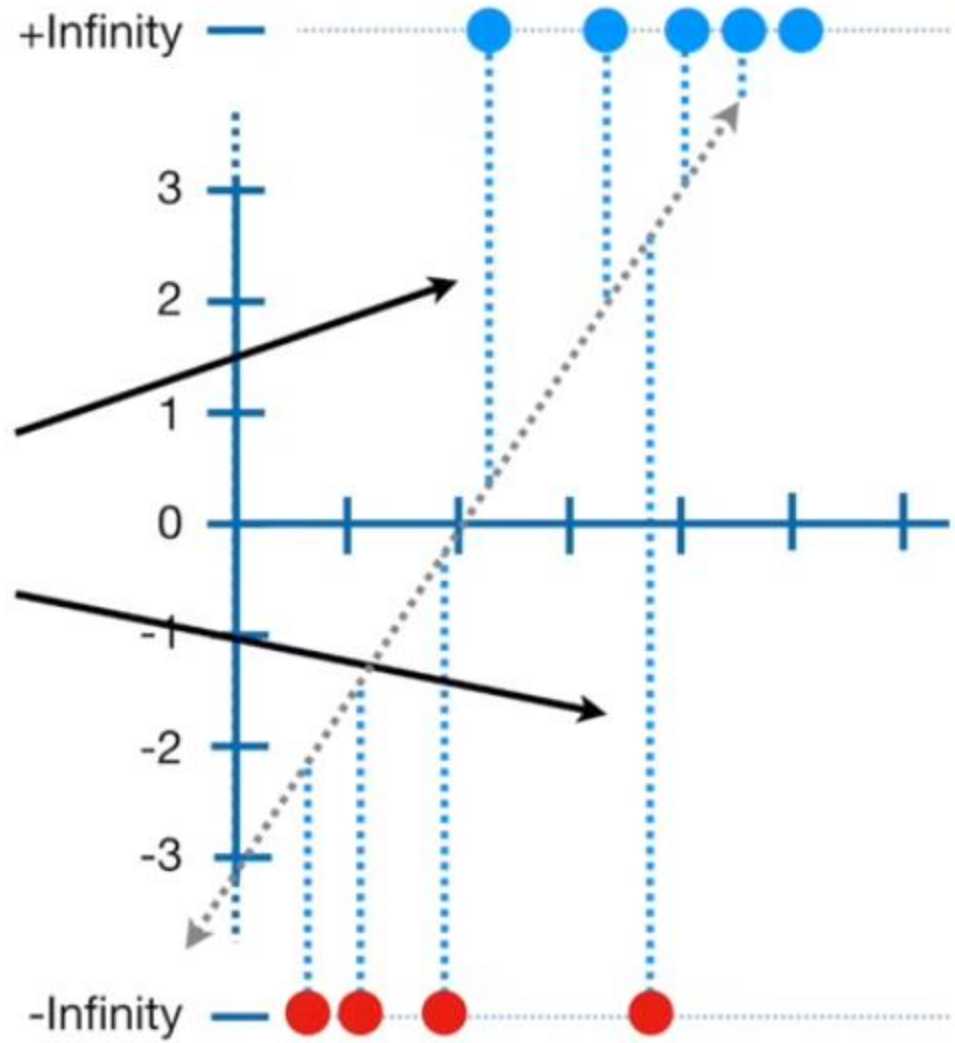
Fitting a Line with Maximum Likelihood

That is to say, we're going to talk about how this squiggle is optimized to fit the data the best.



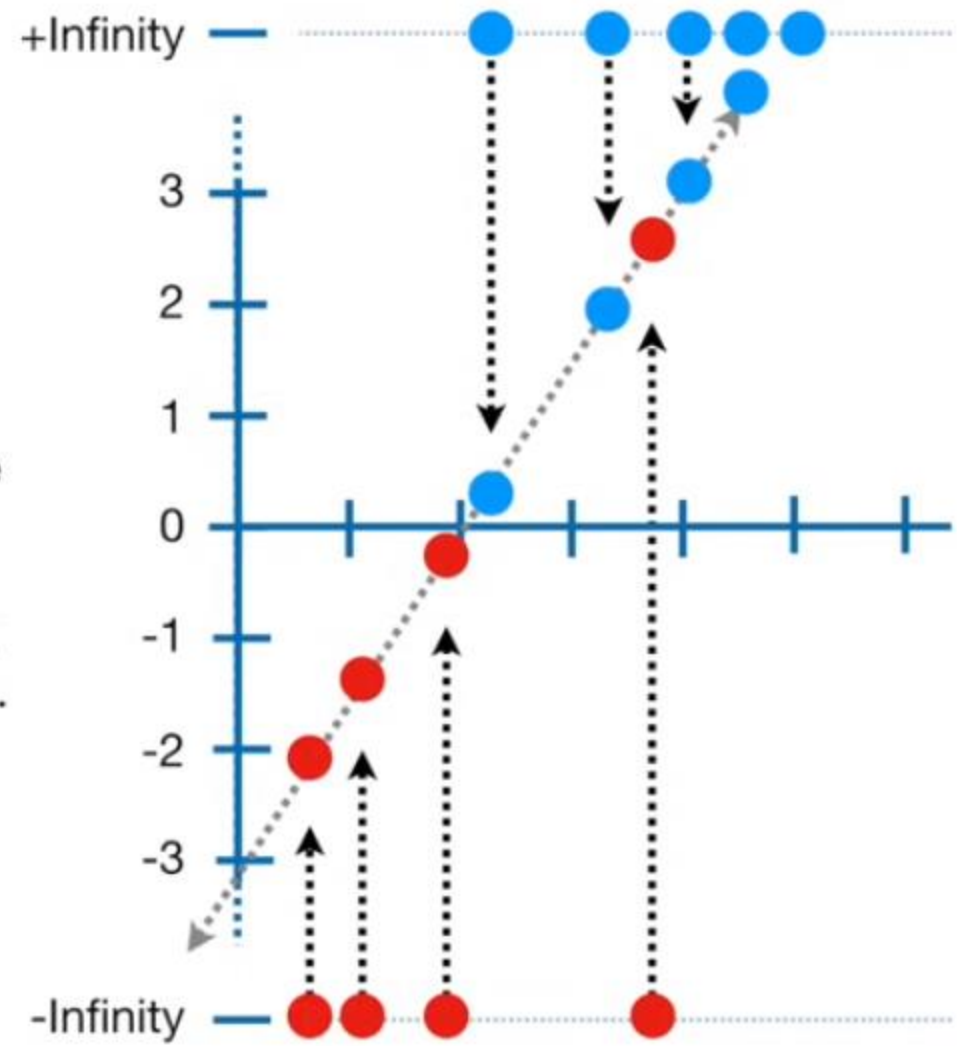

+Infinity
We can draw a candidate "best fitting" line on the graph…
3
2
1
0
-1
-2
-3
-Infinity

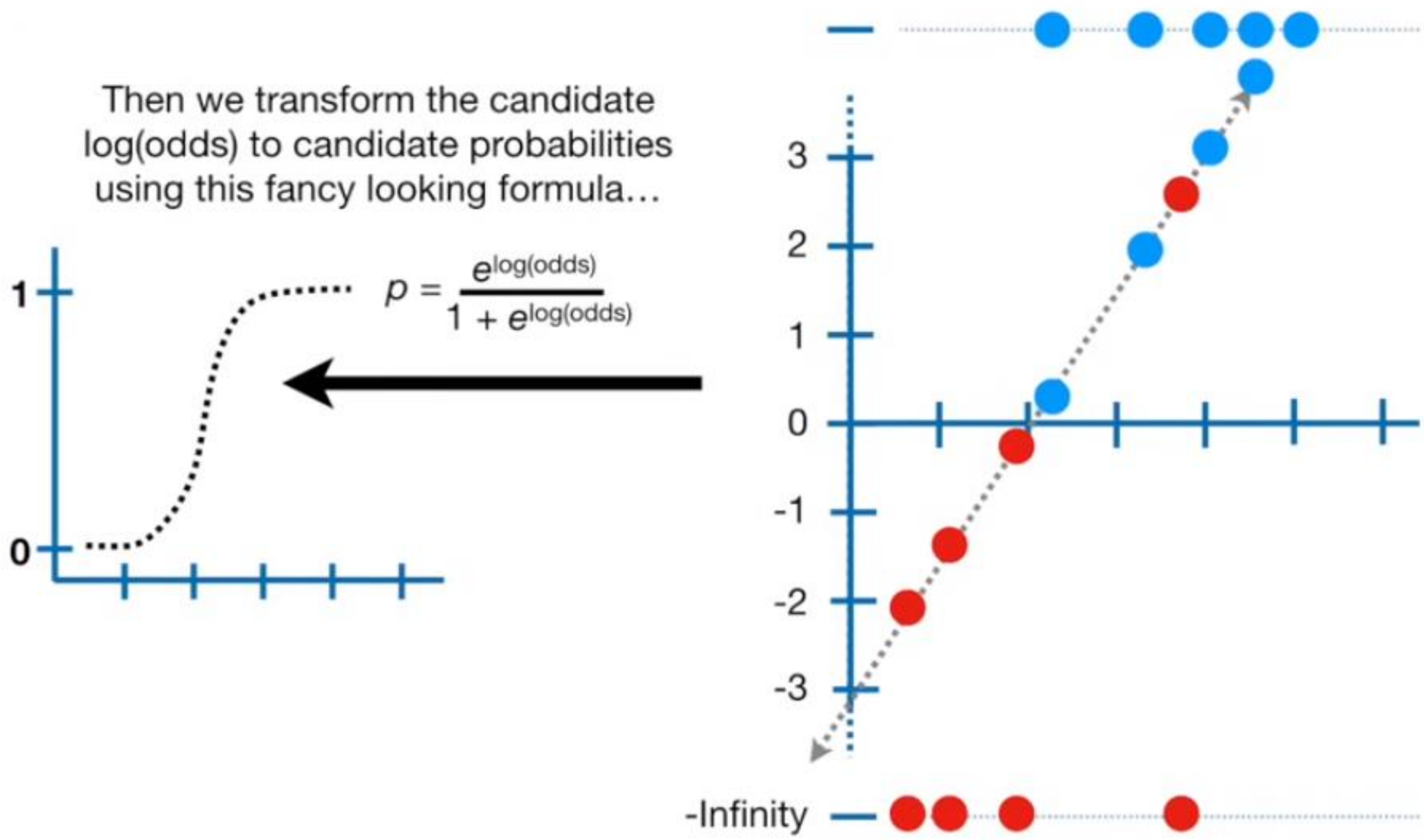
…and this means that the residuals (the distance from the data points to the line) are also equal to positive and negative infinity…

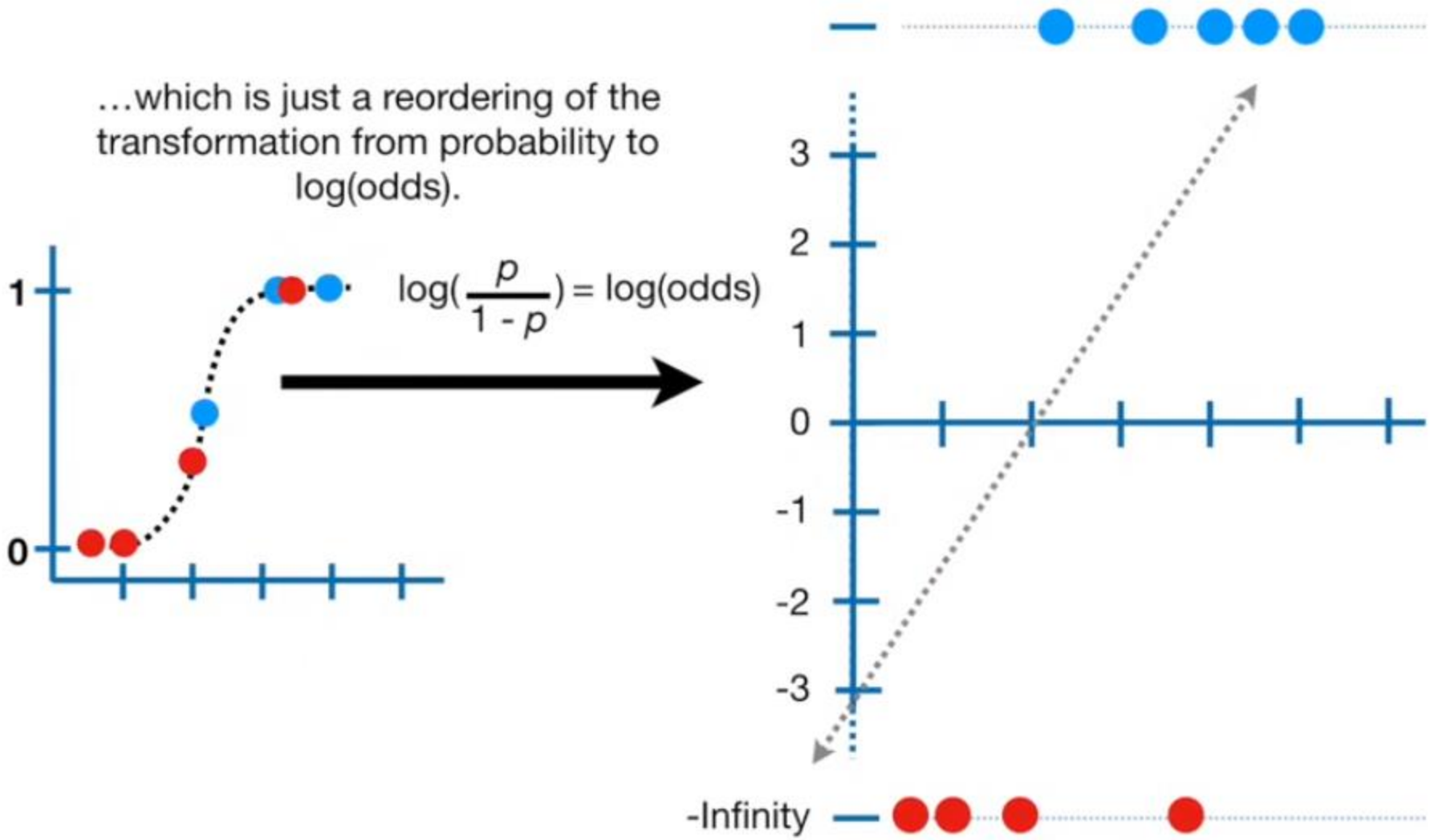


The first thing we do is project the original data points onto the candidate line.

This gives each sample a candidate log(odds) value.

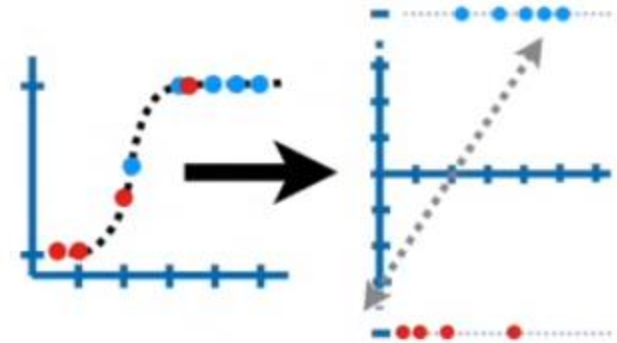



Then we transform the candidate log(odds) to candidate probabilities using this fancy looking formula…
$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$
1
0
3
2
1
0
-1
-2
-3
-Infinity


…which is just a reordering of the transformation from probability to log(odds).
1
0
$\log(\frac{p}{1-p}) = \log(odds)$
3
2
1
0
-1
-2
-3
-Infinity

$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$

Exponentiate both sides… $$\frac{p}{1-p} = e^{\log(\text{odds})}$$



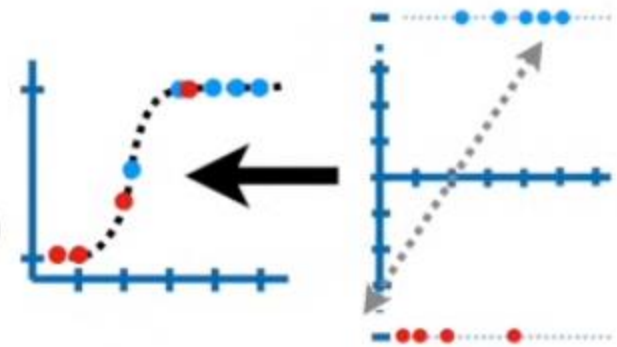
Multiply both sides by $(1 - p)$… $$p = (1-p)e^{\log(\text{odds})}$$

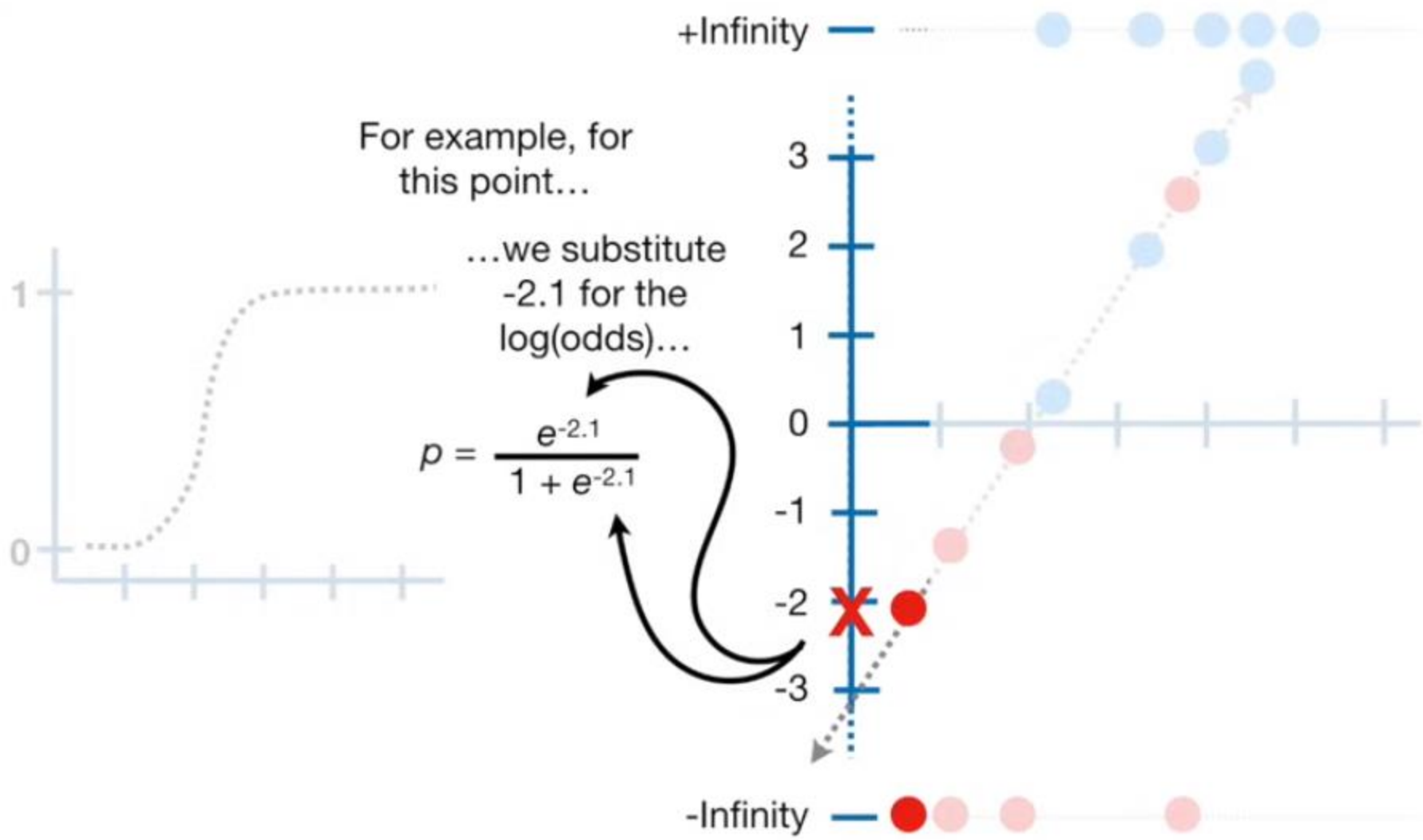
Multiply $(1 - p)$ and $e^{\log(\text{odds})}$… $$p = e^{\log(\text{odds})} - pe^{\log(\text{odds})}$$

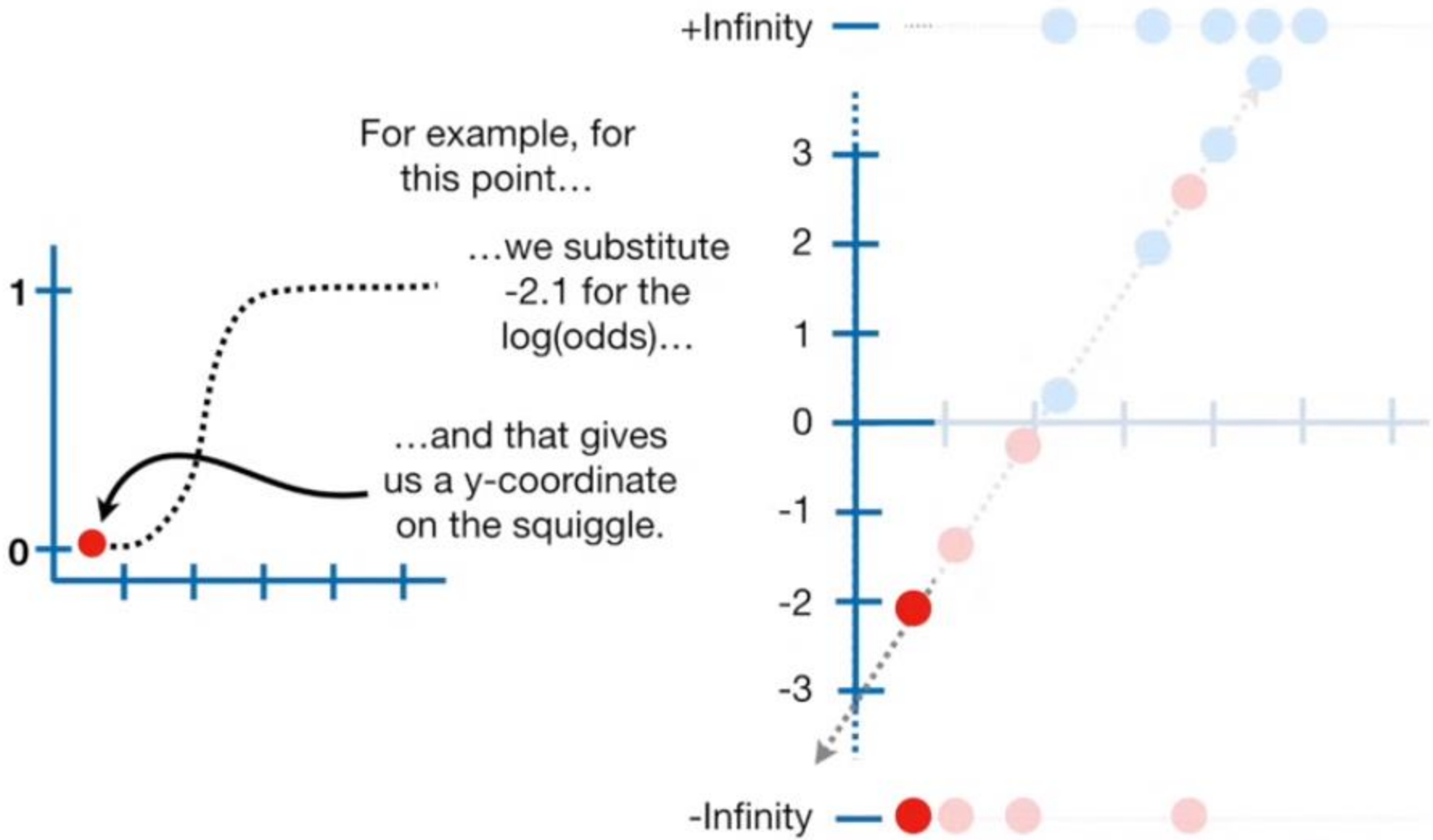
Add $pe^{\log(\text{odds})}$ to both sides… $$p + pe^{\log(\text{odds})} = e^{\log(\text{odds})}$$

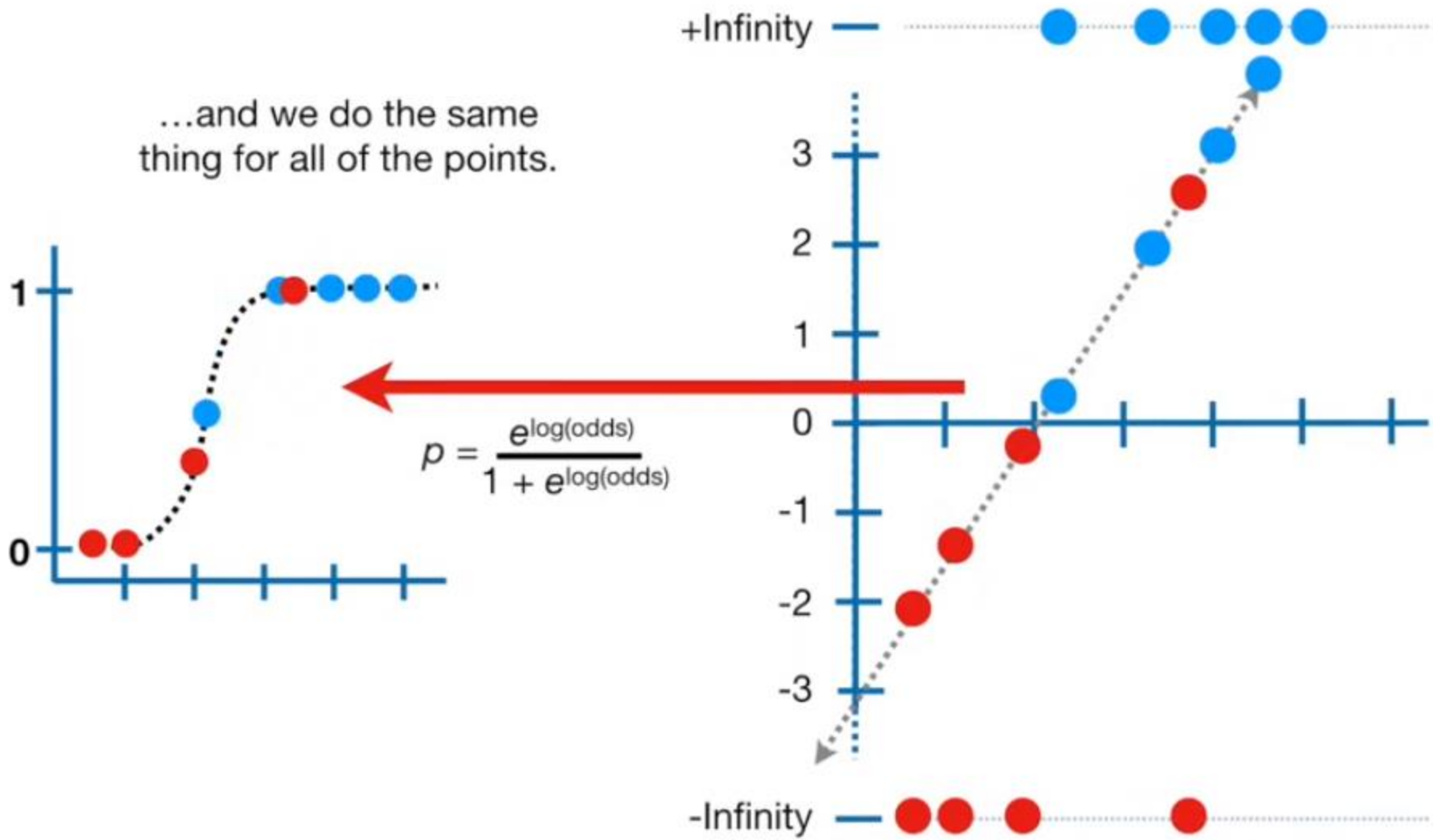
Pull p out… $$p(1 + e^{\log(\text{odds})}) = e^{\log(\text{odds})}$$

Divide both sides by $(1 + e^{\log(\text{odds})})$… $$p = \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}$$

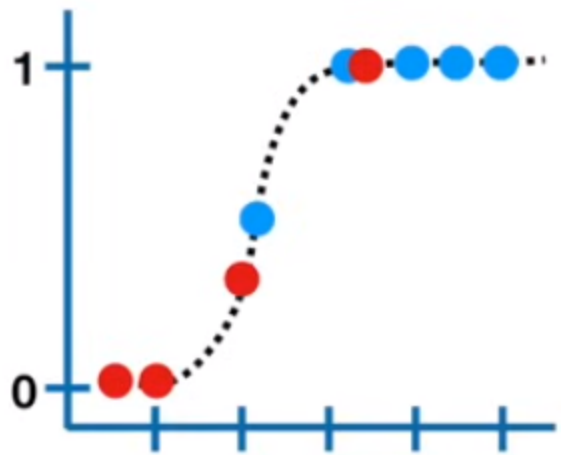


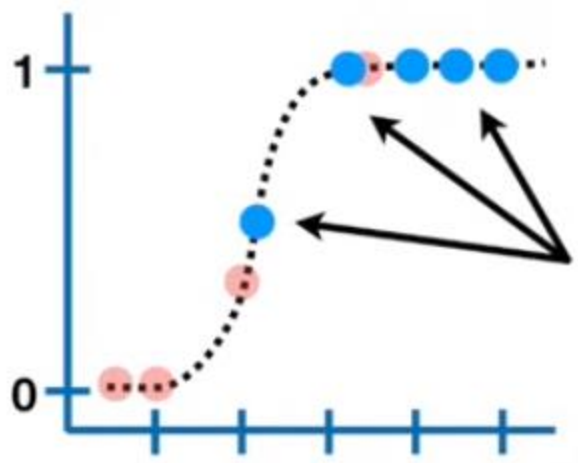

+Infinity
For example, for this point…
…we substitute -2.1 for the log(odds)…
$p = \frac{e^{-2.1}}{1 + e^{-2.1}}$
3
2
1
0
-1
-2
-3
1
0
-Infinity


For example, for this point…
…we substitute -2.1 for the log(odds)…
…and that gives us a y-coordinate on the squiggle.
1
0
+Infinity
3
2
1
0
-1
-2
-3
-Infinity

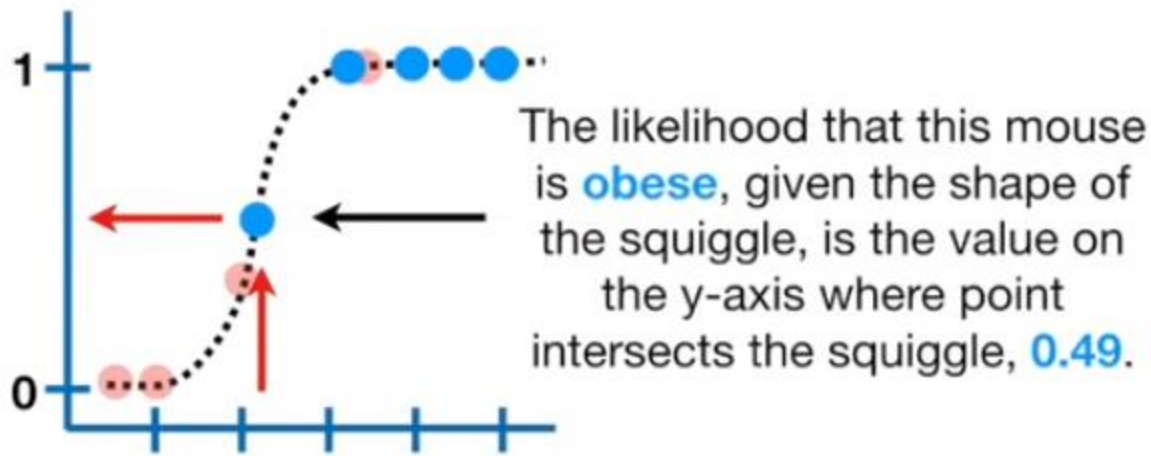

…and we do the same thing for all of the points.
1
0
$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$
+Infinity
3
2
1
0
-1
-2
-3
-Infinity

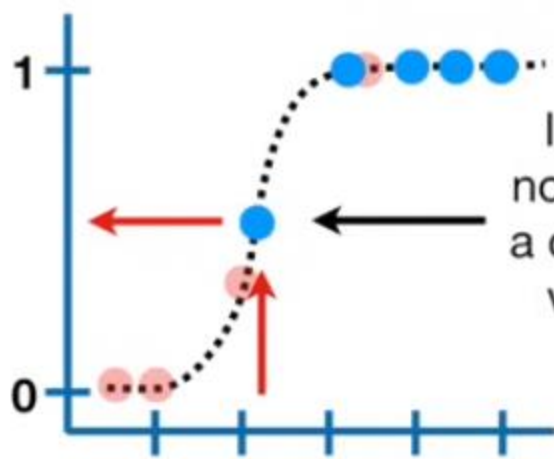
Now we use the observed status (**obese** or **not obese**) to calculate their likelihood given the shape of the squiggly line.



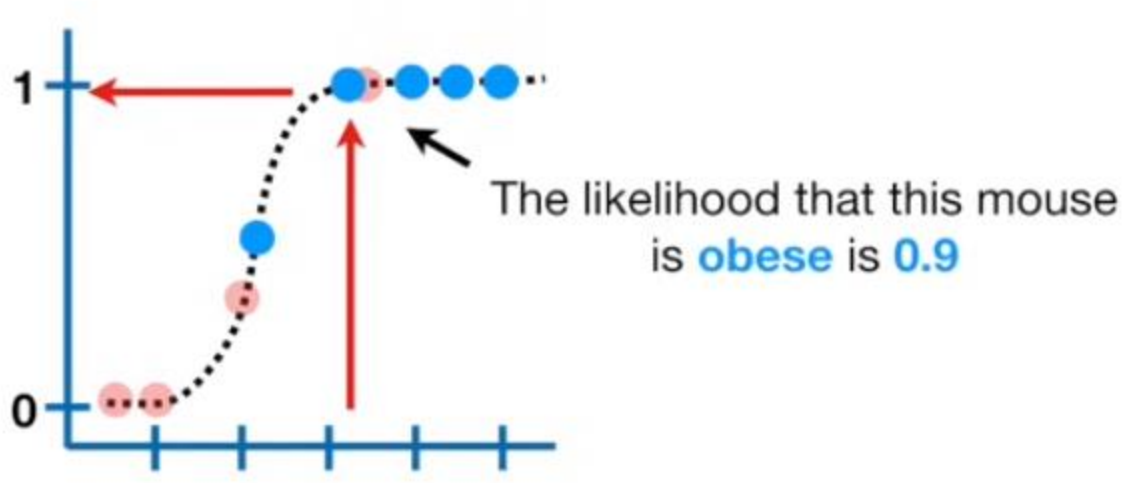


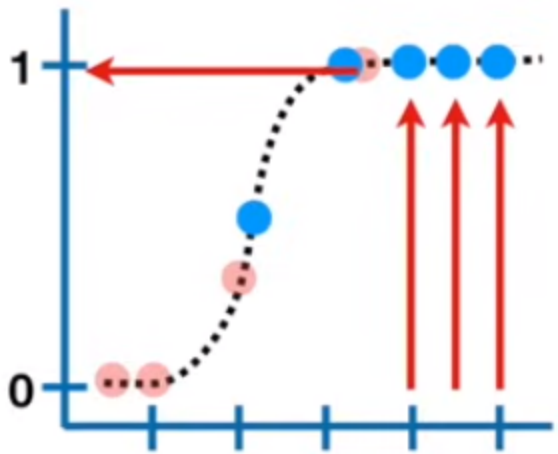
We'll start by calculating the likelihood of the **obese** mice, given the shape of the squiggle.


1
0
The likelihood that this mouse is obese, given the shape of the squiggle, is the value on the y-axis where point intersects the squiggle, 0.49.



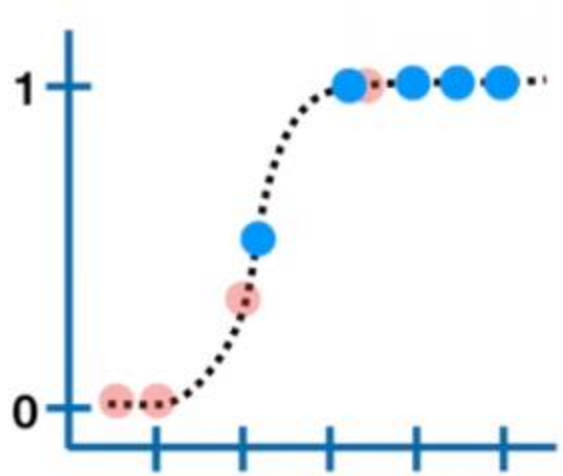
In this case, the probability is not calculated as the area under a curve, but instead is the y-axis value, and that's why it is the same as the likelihood.


1
0
The likelihood that this mouse
is obese is 0.9



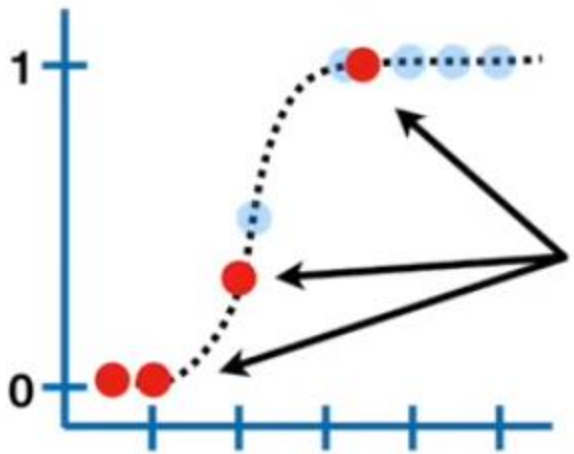
The likelihoods that these mice are obese are 0.91, 0.91 and 0.92

likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 × …



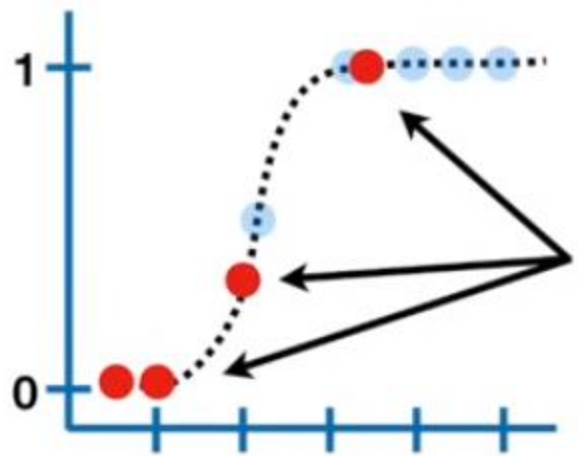
The likelihood for all of the obese mice is just the product of the individual likelihoods.

likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 × …



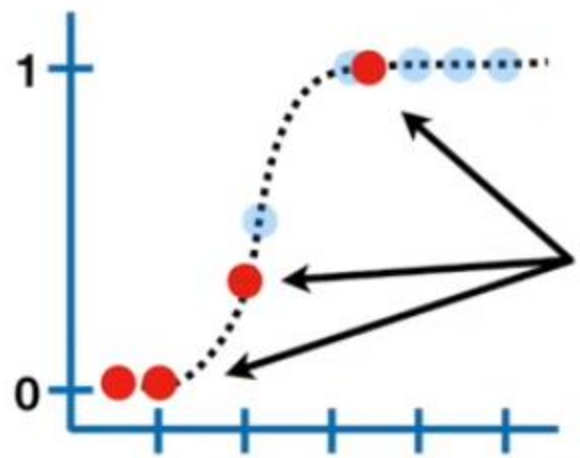
Now we'll figure out the likelihoods for the mice that are **not obese**.

likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 × …



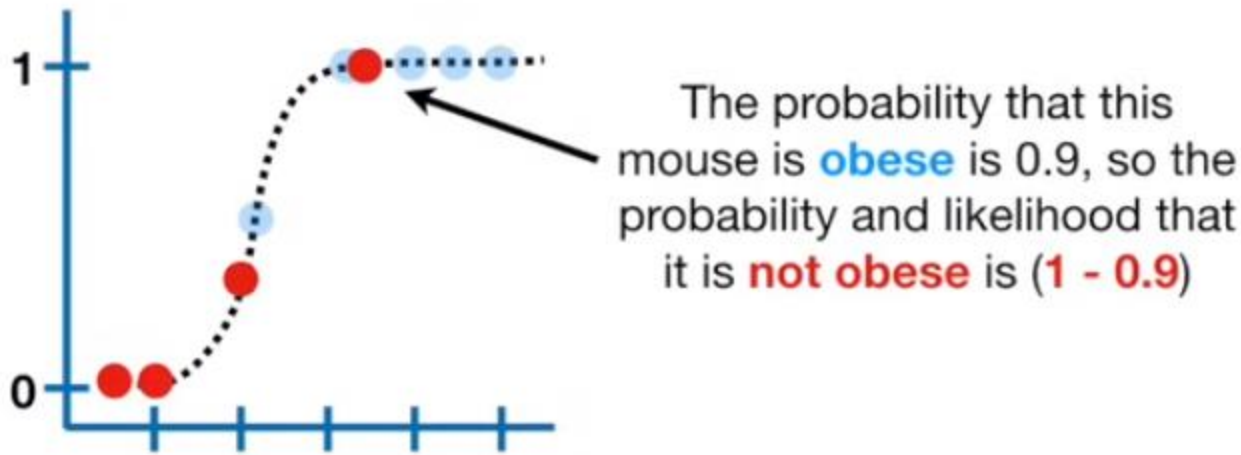
NOTE: The lower the probability of being obese, the higher the probability of not being obese.

likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 × …

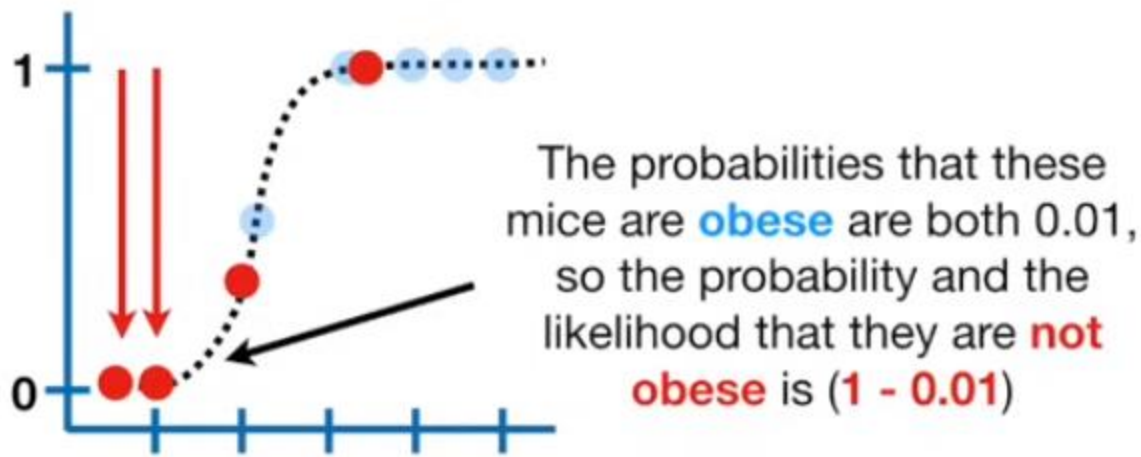


Thus, for these mice, the
likelihood = (1 - probability the mouse is **obese**)

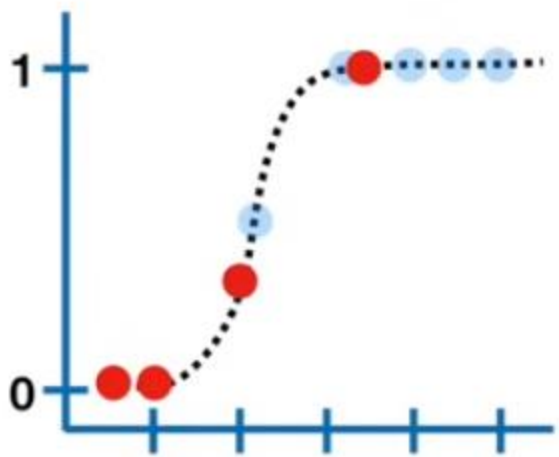
likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 × …



likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 × …

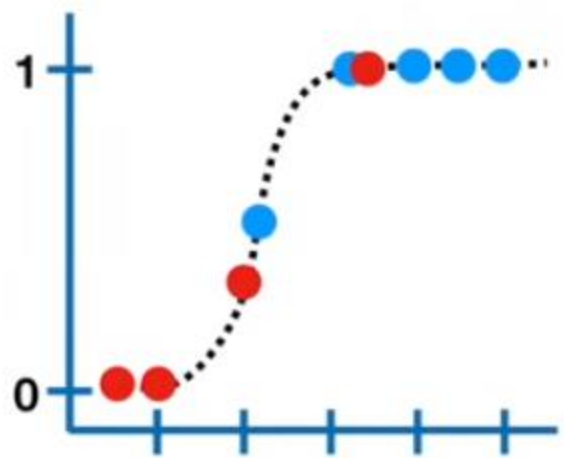


likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 ×
(1 - 0.9) × (1 - 0.3) × (1 - 0.01) × (1 - 0.01)



Now we can include the individual likelihoods for the mice that are **not obese** to the equation for the overall likelihood.

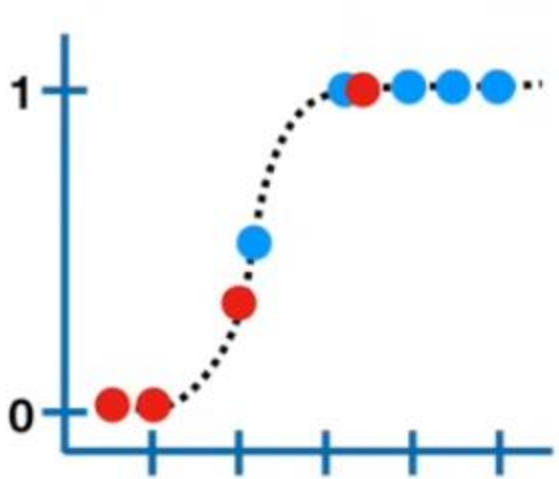
likelihood of data given the squiggle = 0.49 × 0.9 × 0.91 × 0.91 × 0.92 ×
(1 - 0.9) × (1 - 0.3) × (1 - 0.01) × (1 - 0.01)



NOTE: Although it is possible to calculate the likelihood as the product of the individual likelihoods, statisticians prefer to calculate the **log of the likelihood** instead.

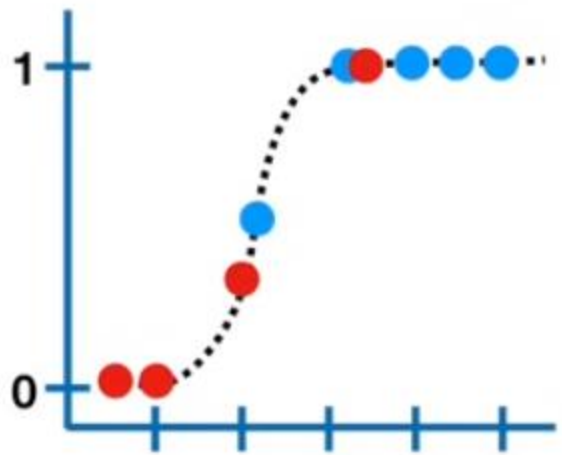
Either way works because the squiggle that maximizes the likelihood is the same one that maximizes the log of the likelihood.

log(likelihood of data given the squiggle) = log(0.49) + log(0.9) + log(0.91) + log(0.91) +
log(0.92) + log(1 - 0.9) + log(1 - 0.3) +
log(1 - 0.01) + log(1 - 0.01)



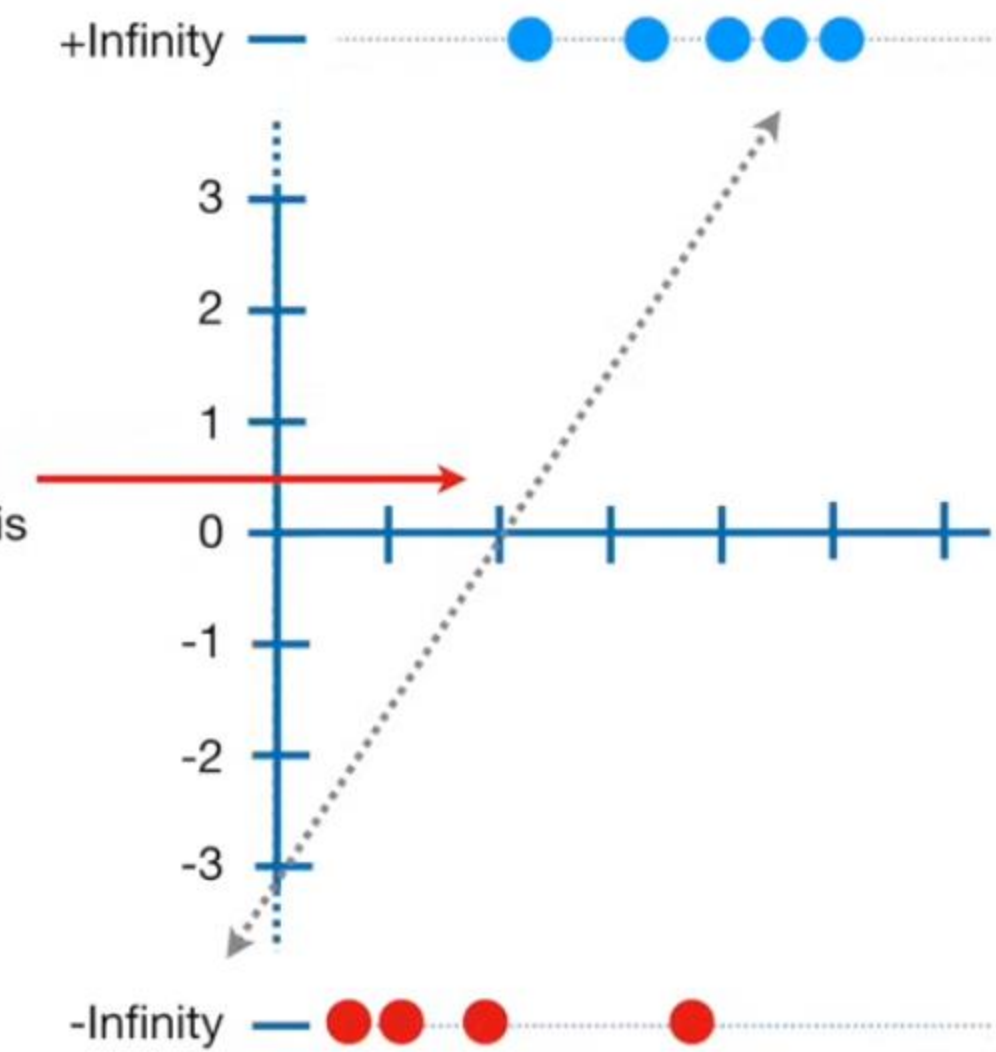
With the log of the likelihood, or "log-likelihood" to those in the know, we **add the logs of the individual likelihoods** instead of multiplying the individual likelihoods…

log(likelihood of data given the squiggle) = -3.77

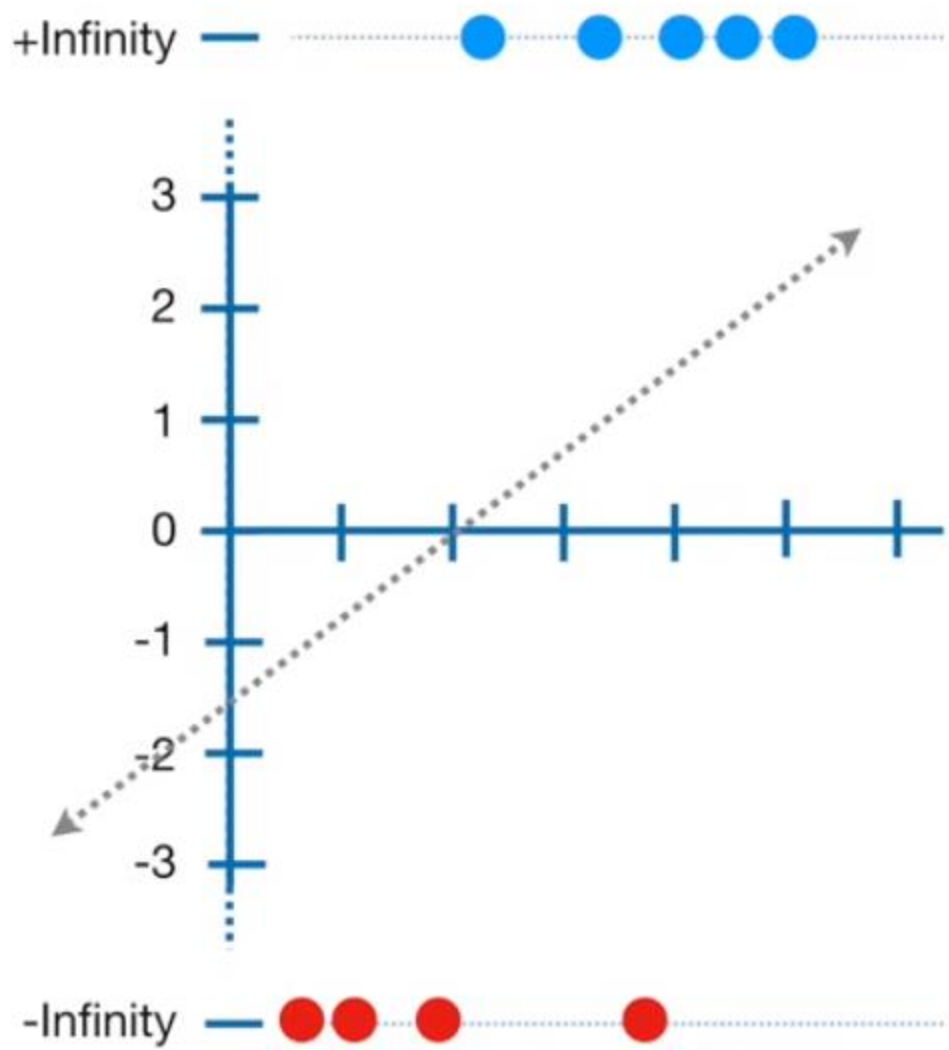


Thus, the log-likelihood of the data given the squiggle is -3.77…

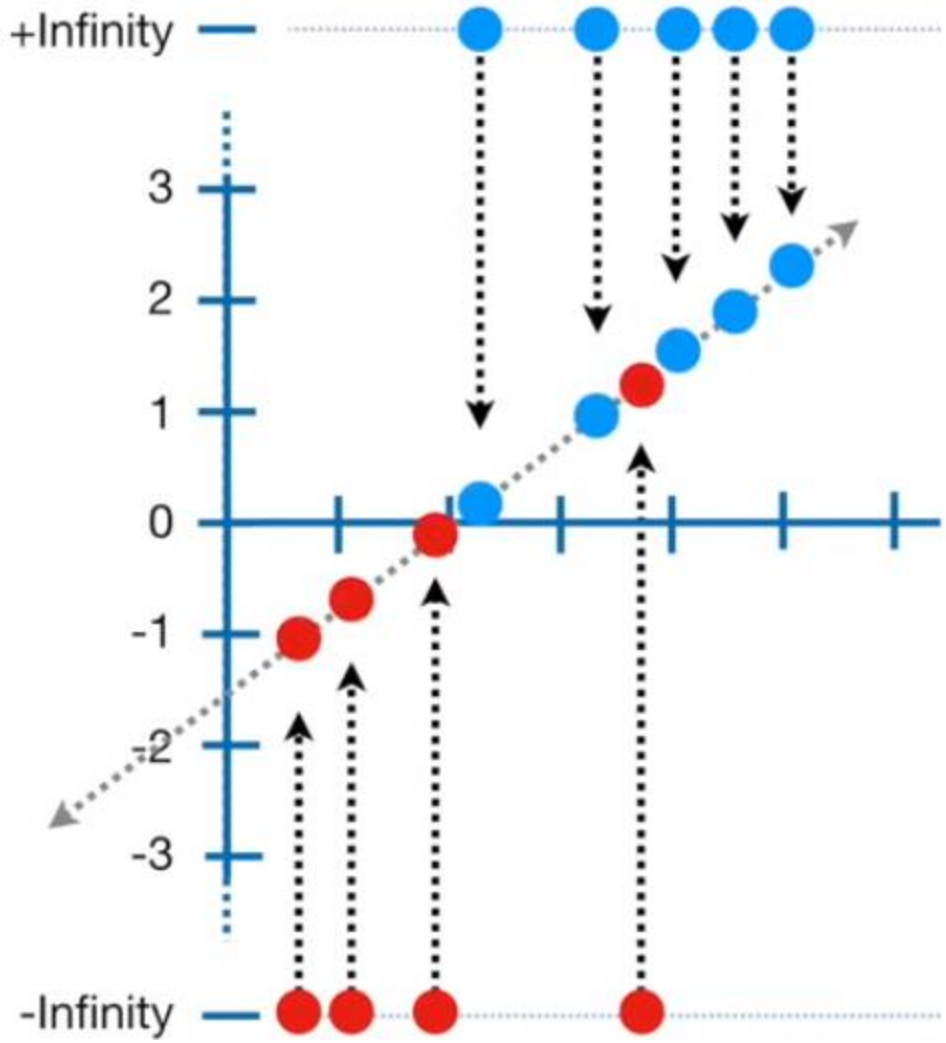
…and this means that the log-likelihood of the original line is -3.77.

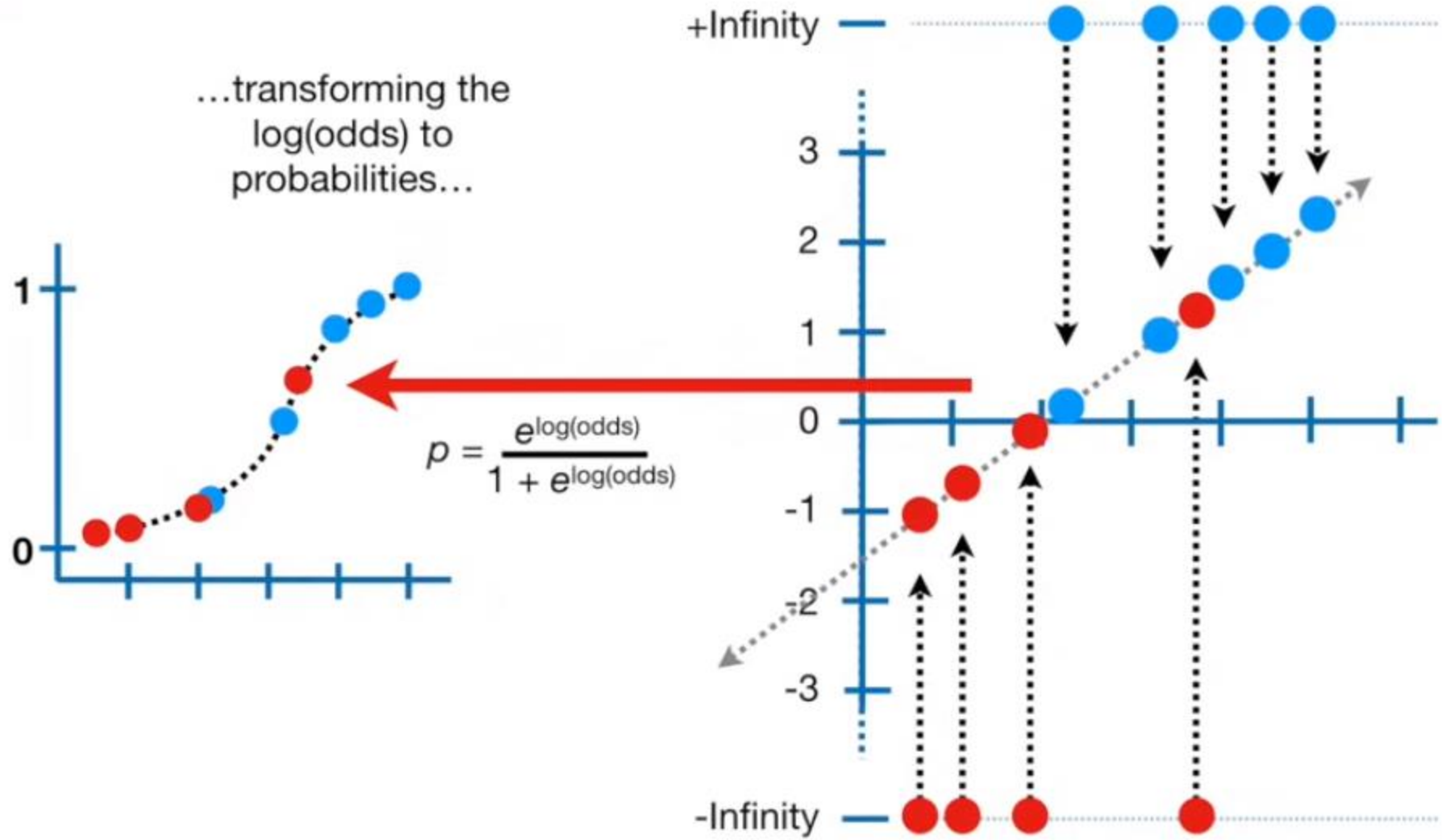


Now we rotate the line…

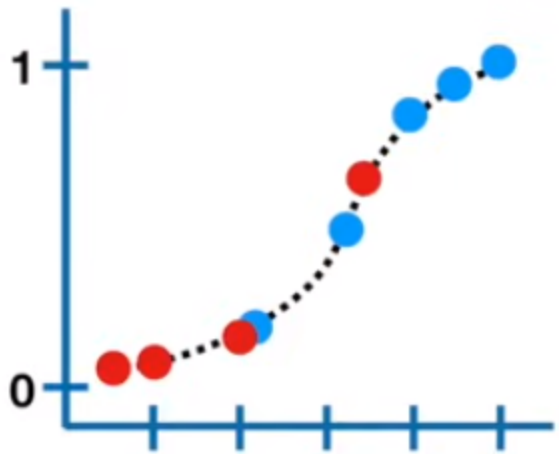


…and calculate its log-likelihood
by projecting the data onto it…



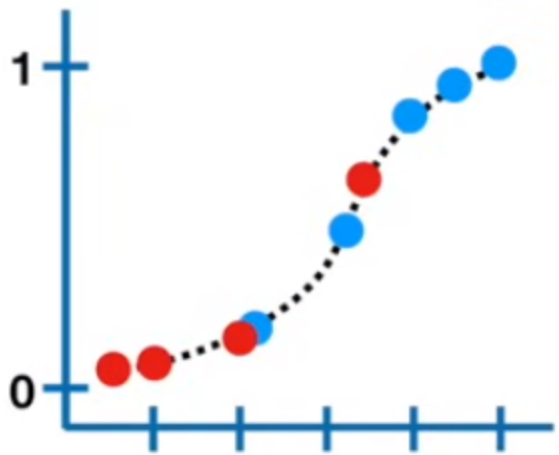

…transforming the log(odds) to probabilities…
1
0
$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$
+Infinity
3
2
1
0
-1
-2
-3
-Infinity

log(likelihood of data given the squiggle) = log(0.22) + log(0.4) + log(0.8) + log(0.89) +
log(0.92) + log(1 - 0.6) + log(1 - 0.2) +
log(1 - 0.1) + log(1 - 0.05)



…and then calculating the log-likelihood…

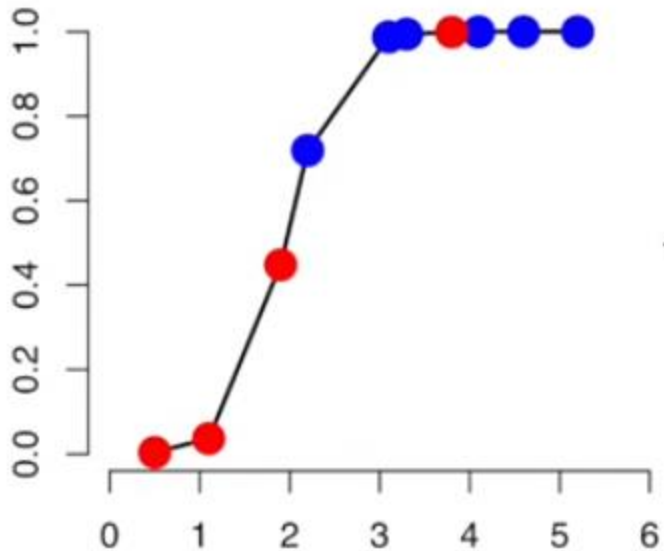
log(likelihood of data given the squiggle) = -4.15



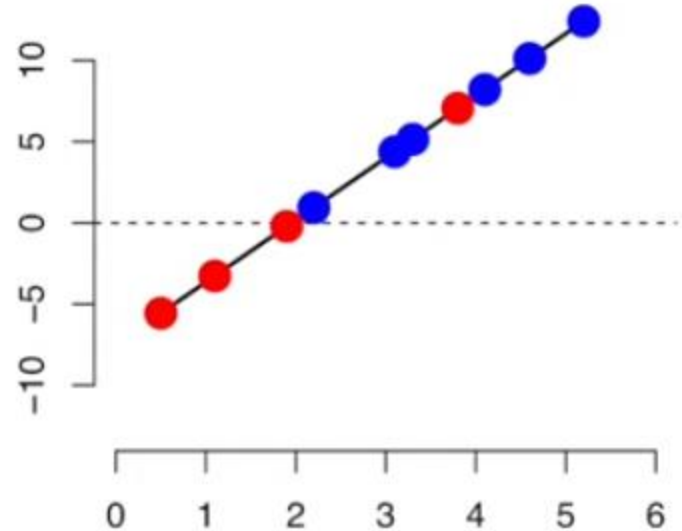
…and the final value for the log-likelihood is -4.15.

(So this one is not as good as the first line.)

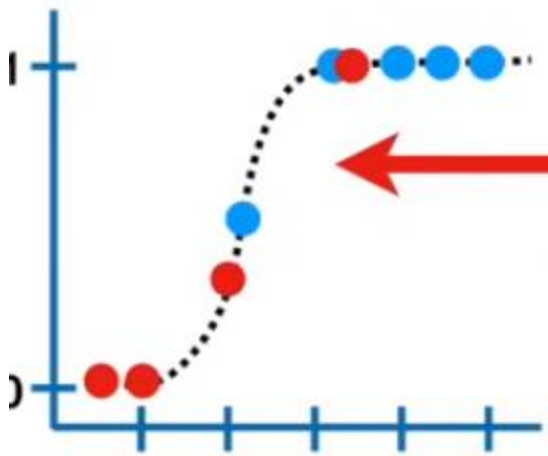
…and transforming it to probabilities and calculating the log-likelihood.

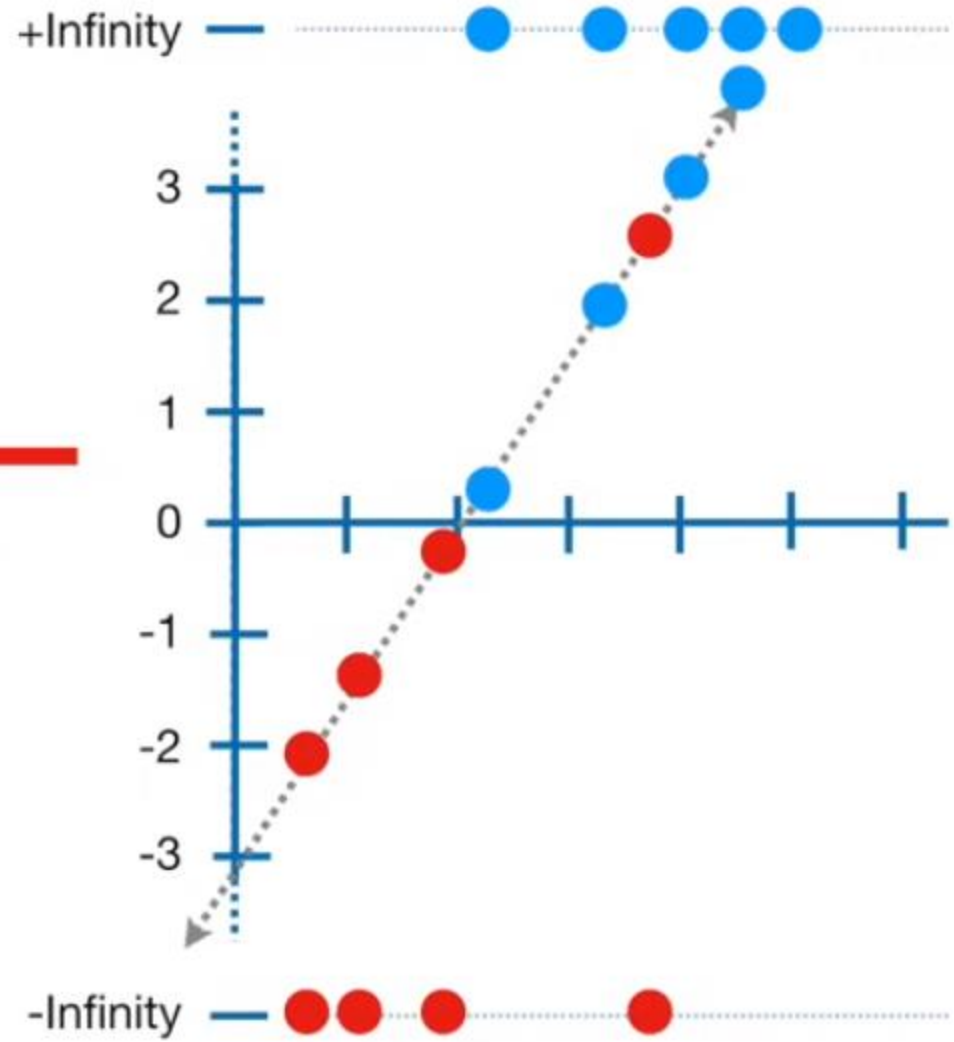


…and we just keep rotating the log(odds) line and projecting the data onto it…



Ultimately we get a line that maximizes the likelihood and that's the one chosen to have the best fit.





Pros and Cons of logistic regression

Pros:

- It is a model that gives probabilities.
- It can be easily scaled to multiple classes.
- It is very quick to train and very fast at classifying unknown records.

Cons:

- The classifies constructs liner boundaries.
- Assumes that the variables are independent.
- Interpretation of coefficients is difficult.

Performance Evaluation Metrics
Confusion Matrix

- Well, it is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.
- A confusion matrix for a two classes (+, -) is shown below.

	C1	C2
C1	True positive	False negative
C2	False positive	True negative

	+	-
+	++	+-
-	-+	--

- There are four quadrants in the confusion matrix, which are symbolized as below.
  - True Positive (TP: f_{++}) : The number of instances that were positive (+) and correctly classified as positive (+).
  - False Negative (FN: f_{+-}): The number of instances that were positive (+) and incorrectly classified as negative (-).
  - False Positive (FP: f_{-+}): The number of instances that were negative (-) and incorrectly classified as (+).
  - True Negative (TN: f--): The number of instances that were negative (-) and correctly classified as (-).

Two Types of Error

False positive (“false alarm”), FP

alarm sounds but person is not carrying metal

False negative (“miss”), FN

alarm doesn’t sound but person is carrying metal

Performance Evaluation Metrics

- We now define a number of metrics for the measurement of a classifier.
  - In our discussion, we shall make the assumptions that there are only two classes: + (positive) and – (negative)
- **True Positive Rate (TPR):** It is defined as the fraction of the positive examples predicted correctly by the classifier.

$$TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{f_{++}}{f_{++}+f_{+-}}$$

  - This metrics is also known as ***Recall***, ***Sensitivity*** or ***Hit rate***.
- **False Positive Rate (FPR):** It is defined as the fraction of negative examples classified as positive class by the classifier.

$$FPR = \frac{FP}{N} = \frac{FP}{FP+TN} = \frac{f_{-+}}{f_{-+}+f_{--}}$$

Accuracy

- It is defined as the fraction of the number of examples that are correctly classified by the classifier to the total number of instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Performance Evaluation Metrics

- **False Negative Rate (FNR):** It is defined as the fraction of positive examples classified as a negative class by the classifier.

$$FNR = \frac{FN}{P} = \frac{FN}{TP + FN} = \frac{f_{+-}}{f_{++} + f_{+-}}$$

- **True Negative Rate (TNR):** It is defined as the fraction of negative examples classified correctly by the classifier

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = \frac{f_{--}}{f_{--} + f_{-+}}$$

  - This metric is also known as ***Specificity***.

Why accuracy is not a good model performance measure?

Total customers – 100
of customers who churn - 8

Churn rate – 8%

Model predicts that no one churns

Accuracy – 92%

Misses out critical customers who churn

		Predicted	
		0	1
Observed	0	TN = 92	FP
	1	FN = 8	TP

Performance Evaluation Metrics

- Both, Precision and Recall are defined by

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

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

Performance Evaluation Metrics

- **F_1 Score (F_1):** Recall (r) and Precision (p) are two widely used metrics employed in analysis..
  - It is defined in terms of (r or Recall) and (p or Precision) as follows.

$$F_1 Score = \frac{2 * Recall\ .Precision}{Recall + Precision} = \frac{2TP}{2TP + FP + FN}$$

Note

- F_1 represents the harmonic mean between recall and precision
- High value of F_1 score ensures that both Precision and Recall are reasonably high.

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- **ROC** curve plots **TPR** (on the **y**-axis) against **FPR** (on the **x**-axis)

$$TPR = \frac{TP}{TP + FN}$$

Fraction of positive instances predicted as positive

$$FPR = \frac{FP}{FP + TN}$$

Fraction of negative instances predicted as positive

		PREDICTED CLASS	
		Yes	No
Actual	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

ROC (Receiver Operating Characteristic)

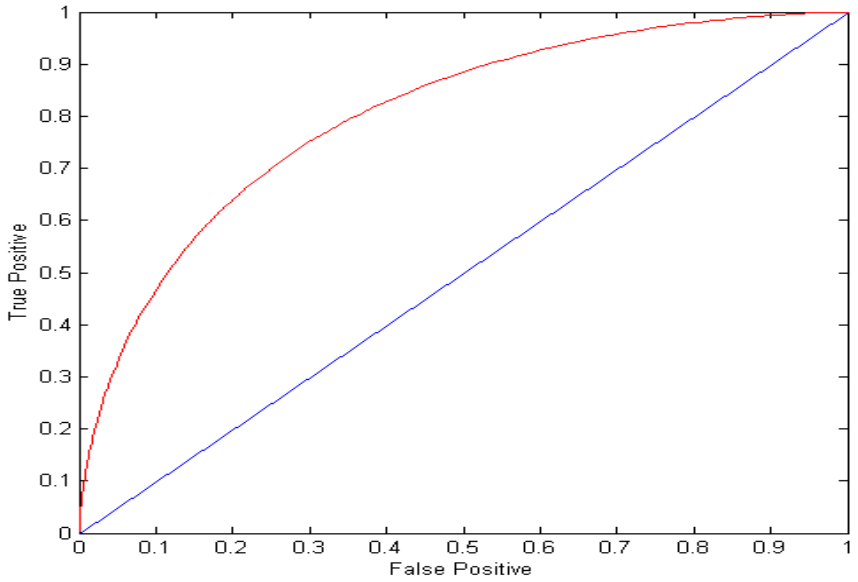
- Performance of a classifier represented as a point on the **ROC** curve
- Changing some parameter of the algorithm, sample distribution, or cost matrix changes the location of the point

ROC Curve

(TP,FP):

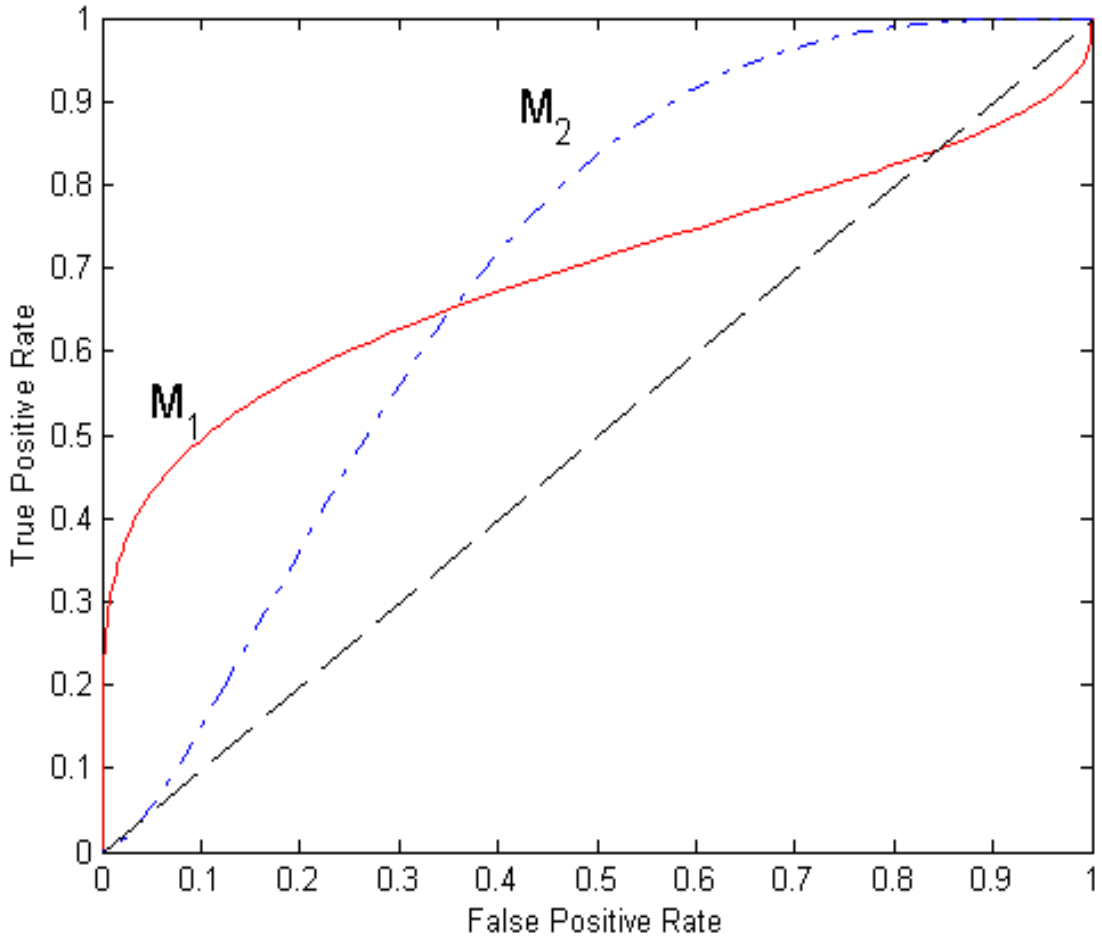
- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class

- Diagonal line:
  - Random guessing
  - Below diagonal line:
    - prediction is opposite of the true class



		PREDICTED CLASS	
		Yes	No
Actual	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

Using ROC for Model Comparison



- No model consistently outperform the other
  - M_1 is better for small FPR
  - M_2 is better for large FPR
- Area Under the ROC curve (AUC)
  - Ideal: Area = 1
  - Random guess:
    - Area = 0.5