Finishing Logistic Regression

15 January 2026

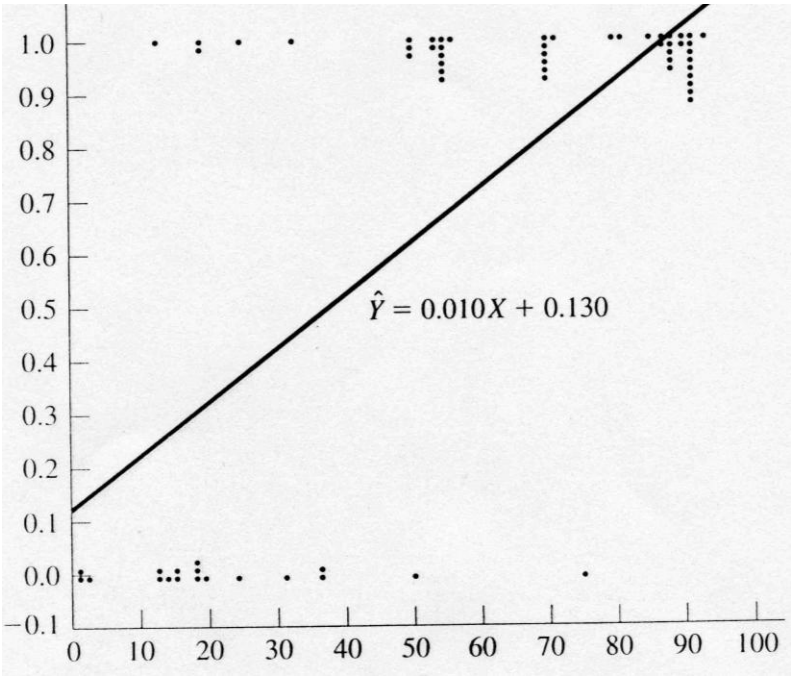
Alex Lyman

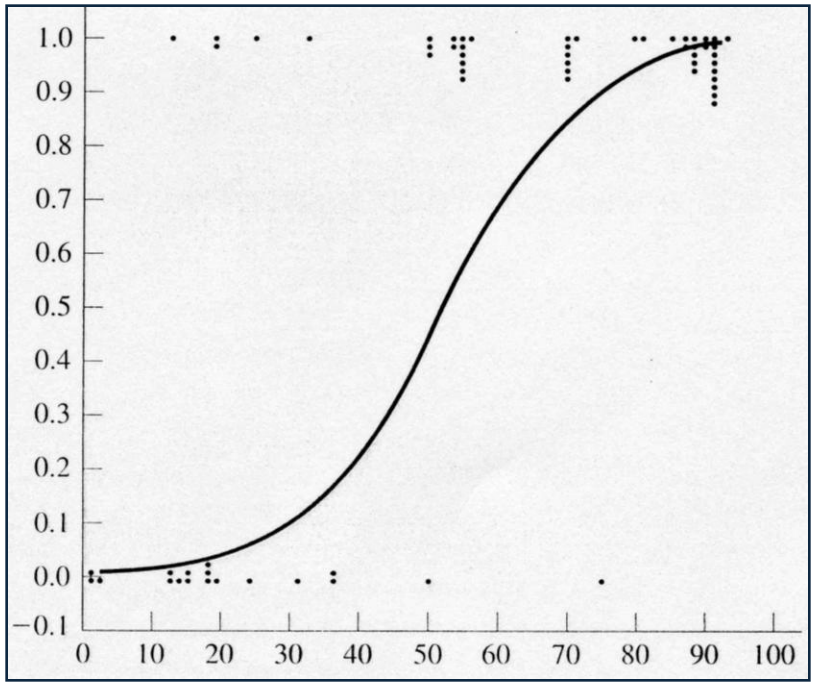
Logistic Regression

- Used for classification
- Produces a **probability estimate** for the **class membership** which is often very useful.
- Uses the **Logistic** or **Sigmoid** function to bound the output space to (0,1)

Logistic Regression Example

- Age (X axis, input variable) – Data is fictional
- Heart Failure (Y axis, 1 or 0, output variable)
- If use value of regression line as a probability approximation
  - Extrapolates outside 0-1 and not as good empirically
- Sigmoidal curve to the right gives empirically good probability approximation and is bounded between 0 and 1

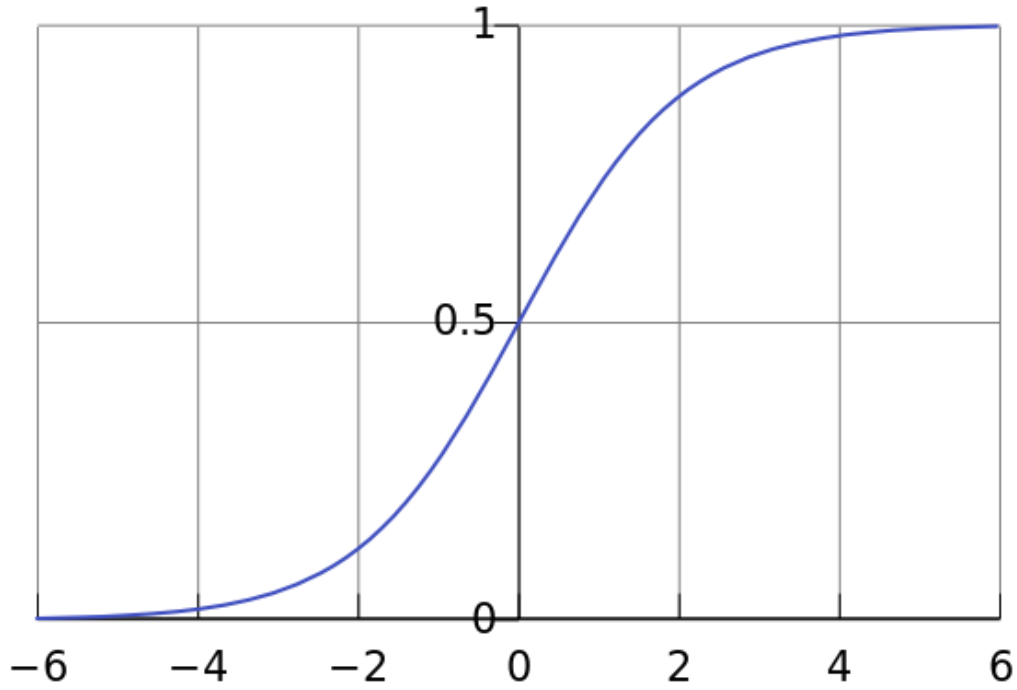




Logistic Regression

The logistic function (sigmoid)

$$f(t) = \frac{1}{1 + e^{-t}}$$



Logistic Regression: Find the vector w that maximizes the probability of the observed data

Logistic Regression Approach

Learning

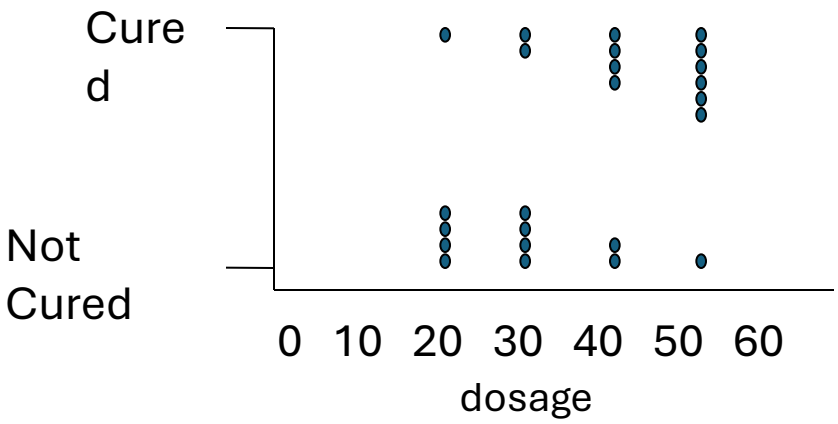
1. Transform initial input probabilities into log odds (logit)
2. Do a standard linear regression using the logit values
   - **This effectively fits a logistic curve to the data, while still just doing a linear regression with the transformed input** (ala quadric machine, etc.)

Generalization

1. Find the value for the new input on the logit line
2. Transform that logit value back into a probability

Logistic Regression Approach

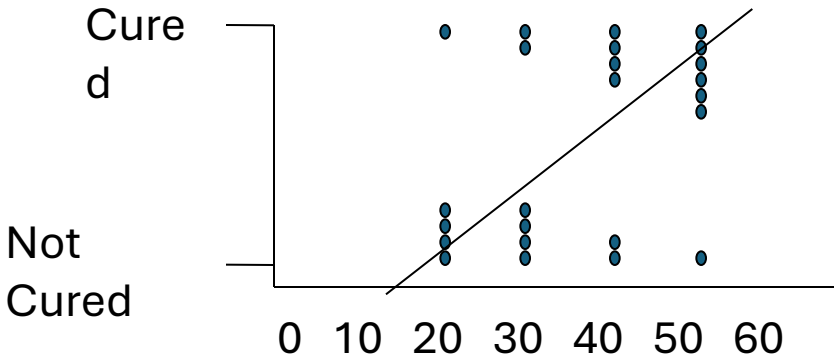
Medication Dosage	# Cured	Total Patients
20	1	5
30	2	6
40	4	6
50	6	7

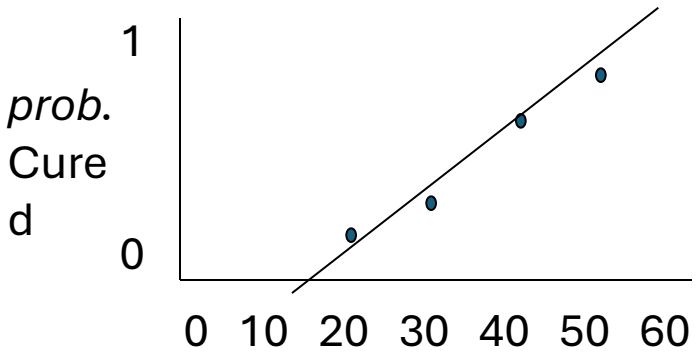


Logistic Regression Approach

Medication Dosage	# Cured	Total Patients	Probability: # Cured/Total Patients
20	1	5	.20
30	2	6	.33
40	4	6	.67
50	6	7	.86

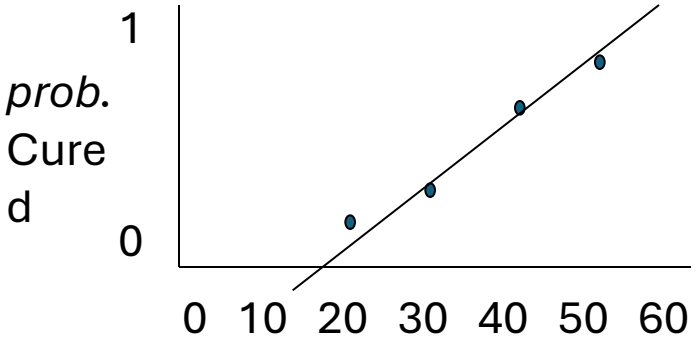
Could just do linear regression on probabilities, but…

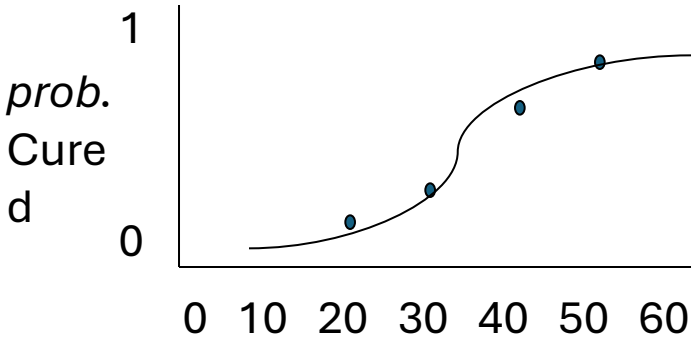




Logistic Regression Approach

- Could use linear regression with the probability points, but that would not extrapolate well
  - Generalized probabilities could be less than 0 or greater than 1
- Logistic version is better but how do we get it?
- We do a non-linear pre-process of the input and then do linear regression on the transformed values – do a linear regression on the log odds - Logit





Log Odds

Log Odds

- The **probability** of an event occurring is a fraction of how many times you would expect to see that event occurring on a set number of occasions.
  - A probability is always expressed as a number between 0 and 1.
  - Example: probability of rolling a 4 on a fair die = 1/6
- **Odds** are expressed as the likelihood of an event occurring divided by the likelihood of it not occurring.
  - Generally a ratio (for example: 2 to 1 odds)
  - Odds = p / (1 – p)
  - 1/6 /(1-1/6) = 1/6 / (5/6) = 1:5

- Logistic regression works in the odds space which converts the model from probabilities to likelihoods
- Then we take the logarithm so we don't have precision issues

Logarithms - Underflow

- Computers express numbers using some number of bits.
- If a number is too small for a computer to represent, that is called **underflow.**
- Sort of like writing digits in a small paper.
- In lots of machine learning contexts, we multiply probabilities together. Small numbers times small numbers make really small numbers.
- We can avoid this by taking the logarithm (inverse of the exponent). Logarithms convert multiplication (dangerous) into addition (safe)

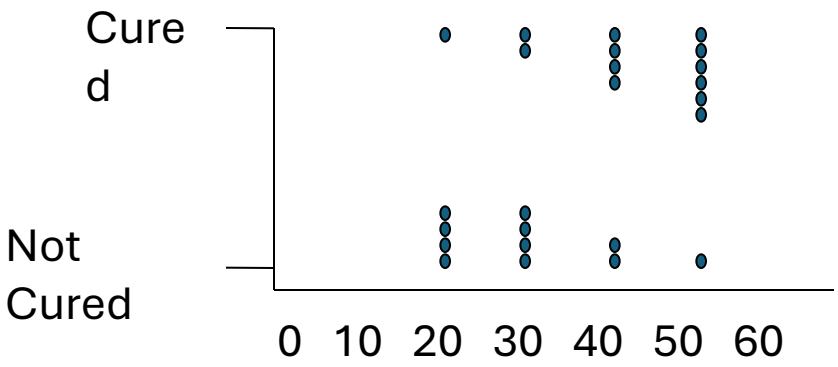
$$\log(a \cdot b) = \log(a) + \log(b)$$

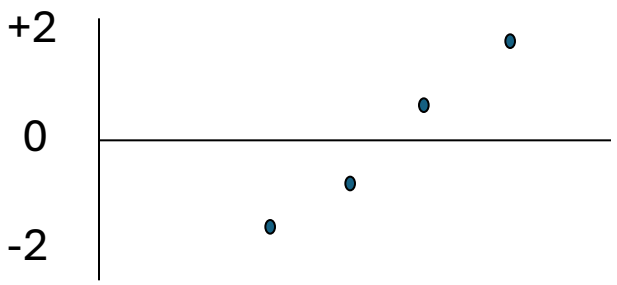
- Often natural log (ln)

```
>>> prob = 0.5 ** 2000
>>> print(prob)
0.0
>>> import math
>>> log_prob = 2000 * math.log(0.5)
>>> print(log_prob)
-1386.2943611198905
>>>
```

Non-Linear Pre-Process to Logit (Log Odds)

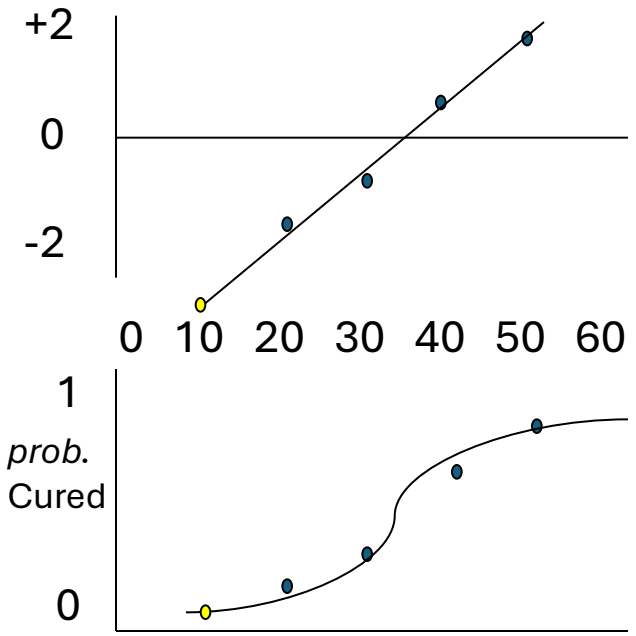
Medication Dosage	# Cured	Total Patients	Probability: # Cured/Total Patients	Odds: $p/(1-p)$ = # cured/ # not cured	Logit Log Odds: ln(Odds)
20	1	5	.20	.25	-1.39
30	2	6	.33	.50	-0.69
40	4	6	.67	2.0	0.69
50	6	7	.86	6.0	1.79





Regression of Log Odds

Medication Dosage	# Cured	Total Patients	Probability: # Cured/Total Patients	Odds: $p/(1-p)$ = # cured/ # not cured	Log Odds: ln(Odds)
20	1	5	.20	.25	-1.39
30	2	6	.33	.50	-0.69
40	4	6	.67	2.0	0.69
50	6	7	.86	6.0	1.79



- $y = .11x - 3.8$ - Logit regression equation
- Now we have a regression line for log odds (logit)
- To generalize, we use the log odds value for the new data point
- Then we transform that log odds point to a probability: $p = e^{\text{logit}(x)}/(1+e^{\text{logit}(x)})$
- For example assume we want p for dosage = 10

$$\text{Logit}(10) = .11(10) - 3.8 = -2.7$$

$p(10) = e^{-2.7}/(1+e^{-2.7}) = .06$ [note that we just work backwards from logit to p]

- These p values make up the sigmoidal regression curve (which we never have to actually plot)

Logistic Regression Approach

Learning

1. Transform initial input probabilities into log odds (logit)
2. Do a standard linear regression using the logit values
   - **This effectively fits a logistic curve to the data, while still just doing a linear regression with the transformed input** (ala quadric machine, etc.)

Generalization

1. Find the value for the new input on the logit line
2. Transform that logit value back into a probability