# Logistic Regression

# Logistic Regression

- We’ve explored how to use Linear Regression and its many variations to predict a continuous label.
- But how can we predict a categorical label?

# Logistic Regression

- We’ve explored how to use Linear Regression and its many variations to predict a continuous label.
- But how can we predict a categorical label?
  - Logistic Regression

# Logistic Regression

- Logistic Regression
  - Don’t be confused by the use of the term “regression” in its name!
  - Logistic Regression is a **classification** algorithm designed to predict **categorical** target labels.

# Logistic Regression

- Logistic Regression Section Overview
  - Mathematical Theory behind Logistic Regression
  - Simple Implementation of Logistic Regression for Classification Problem
  - Multiclass Classification with Logistic Regression

# Logistic Regression

- Logistic Regression Section Overview
    - Interpreting Results
        - Odds Ratio and Coefficients
        - Classification Metrics
            - Accuracy
            - Precision
            - Recall
        - ROC Curves

# Logistic Regression

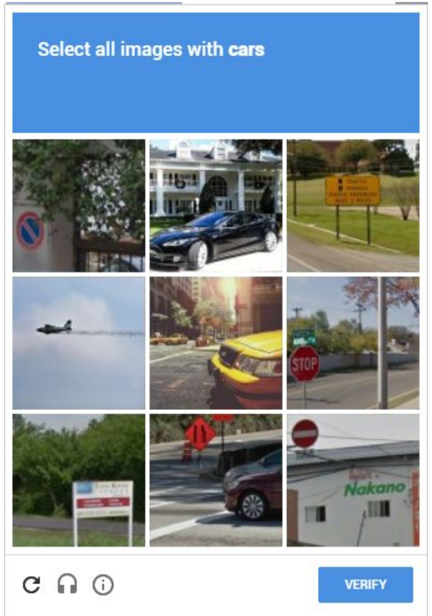
- Logistic Regression will allow us to predict a categorical label based on historical feature data.
- The categorical target column is two or more discrete class labels.

# Logistic Regression

- Classification algorithms predict a class or category label:
    - Class 0: Car Image
    - Class 1: Street Image
    - Class 2: Bridge Image

# Logistic Regression

- You may not have realized you are helping Google label class data!



# Logistic Regression

- Keep in mind, any continuous target can be converted into categories through discretization.
  - Class 0: House Price $0-100k
  - Class 1: House Price $100k-200k
  - Class 2: House Price <$200k

# Logistic Regression

- Classification algorithms also often produce a **probability** prediction of belonging to a class:
    - Class 0: 10% Probability
    - Class 1: 85% Probability
    - Class 2: 5% Probability

# Logistic Regression

- Classification algorithms also often produce a **probability** prediction of belonging to a class:
    - Class 0: 10% Probability - Car Image
    - Class 1: 85% Probability - Street Image
    - Class 2: 5% Probability - Bridge Image
        - Model reports back prediction of Class 1, image is a street.

# Logistic Regression

- Also note our prediction **ŷ** will be a category, meaning we won't be able to calculate a difference based on **y-ŷ.**
  - **Car Image - Street Image** does not make sense.
- We will need to discover a completely different set of error metrics and performance evaluation!

# Logistic Regression Theory and Intuition

## Part One: The Logistic Function

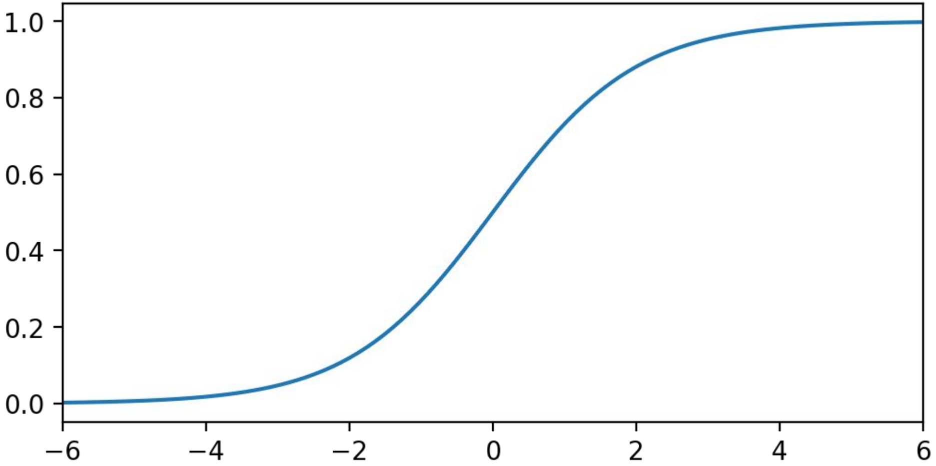
# Logistic Regression

- Logistic Regression works by transforming a Linear Regression into a classification model through the use of the logistic function:

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

# Logistic Regression

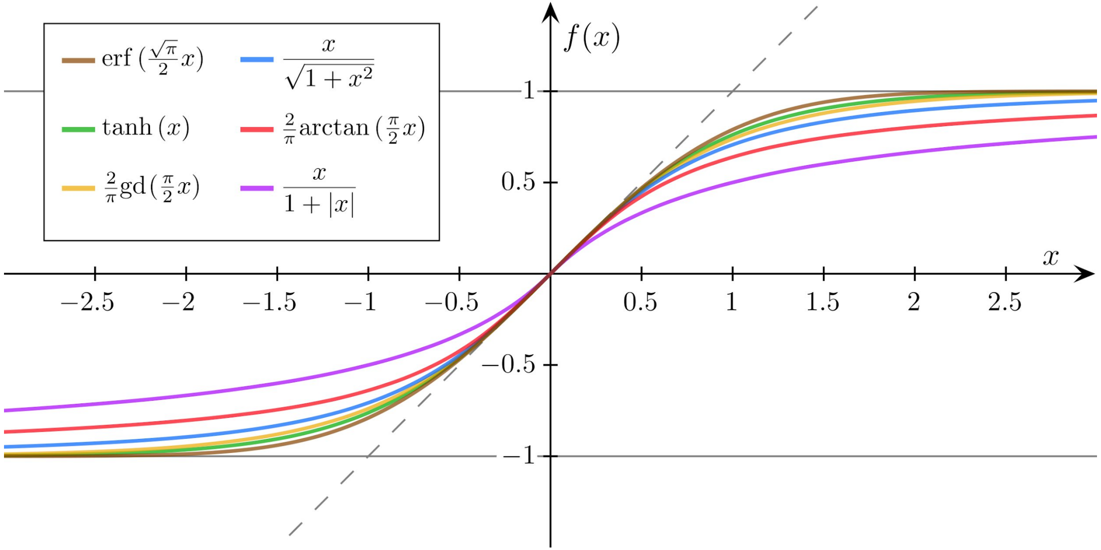
- Why the need for a logistic function versus a logarithmic function?




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

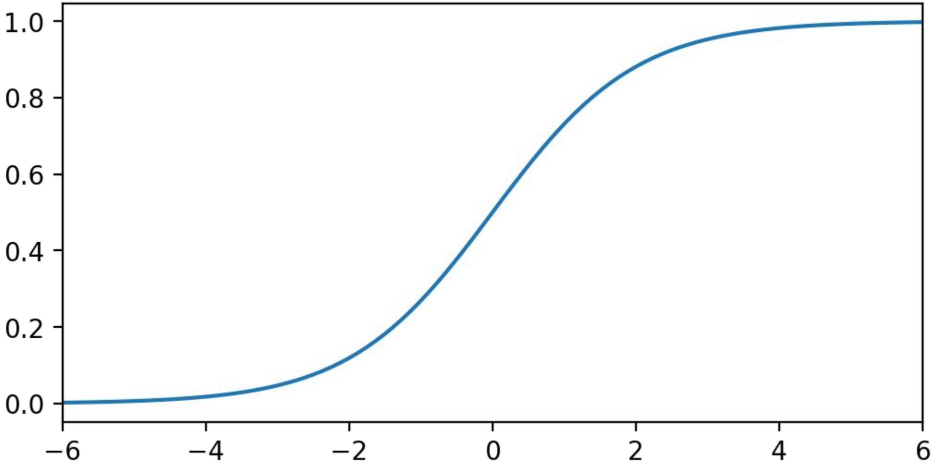
# Logistic Regression

- Note: There is a "family" of logistic functions.



# Logistic Regression

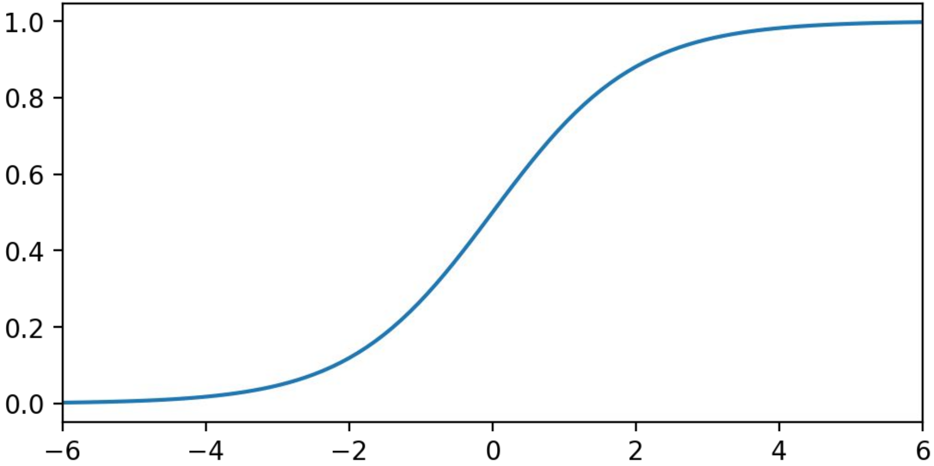
- Notice the “leveling off” behavior of the curve.




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

# Logistic Regression

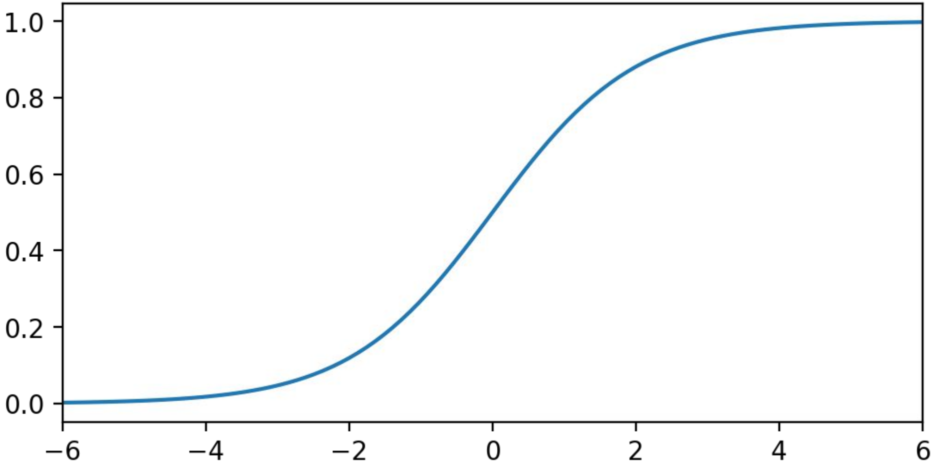
- Also notice **any** value of **x** will have an output range between 0 and 1.




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

# Logistic Regression

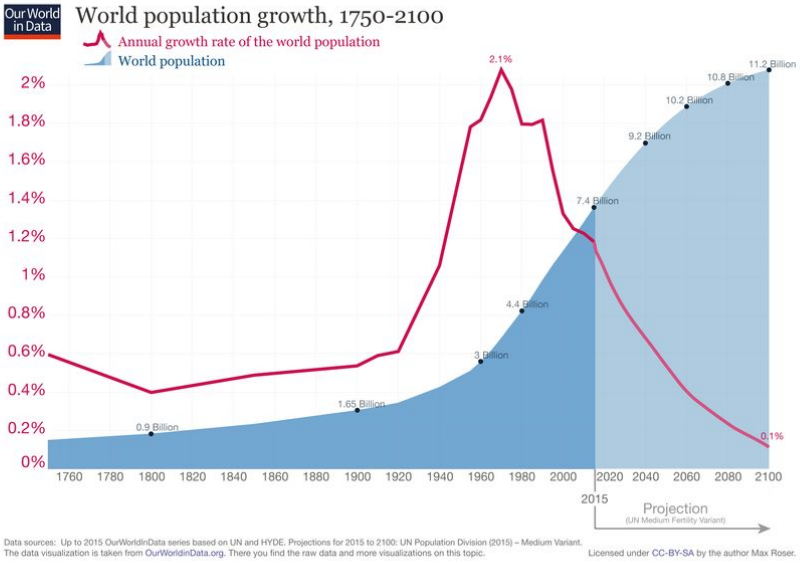
- Many natural real world systems have a “carrying capacity” or a natural limiting factor.




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

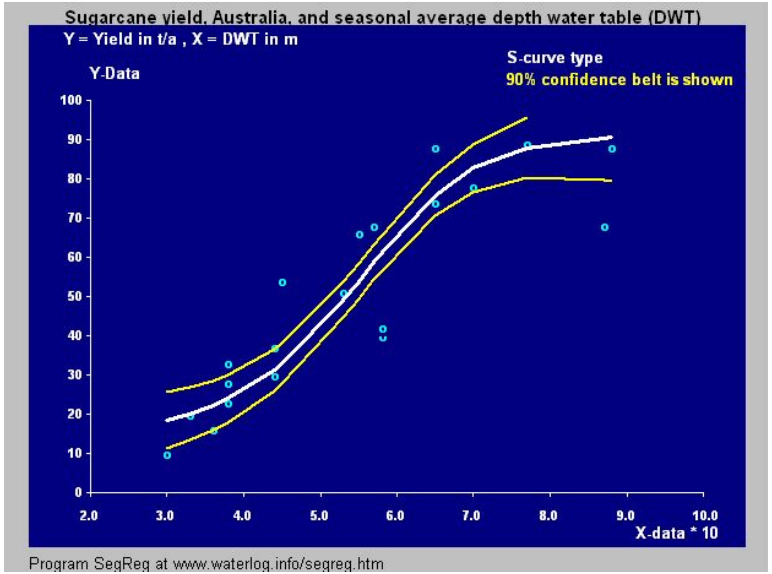
# Logistic Regression

- Many natural real world systems have a “carrying capacity” or a natural limiting factor.



# Logistic Regression

- Many natural real world systems have a “carrying capacity” or a natural limiting factor.



# Logistic Regression Theory and Intuition

## Part Two:
## Linear to Logistic Intuition

# Logistic Regression

- Let’s explore how to convert a Linear Regression model used for a **regression task** into a Logistic Regression model used for a **classification task.**
- Imagine a dataset with a single feature (previous year’s income) and a single target label (loan default)

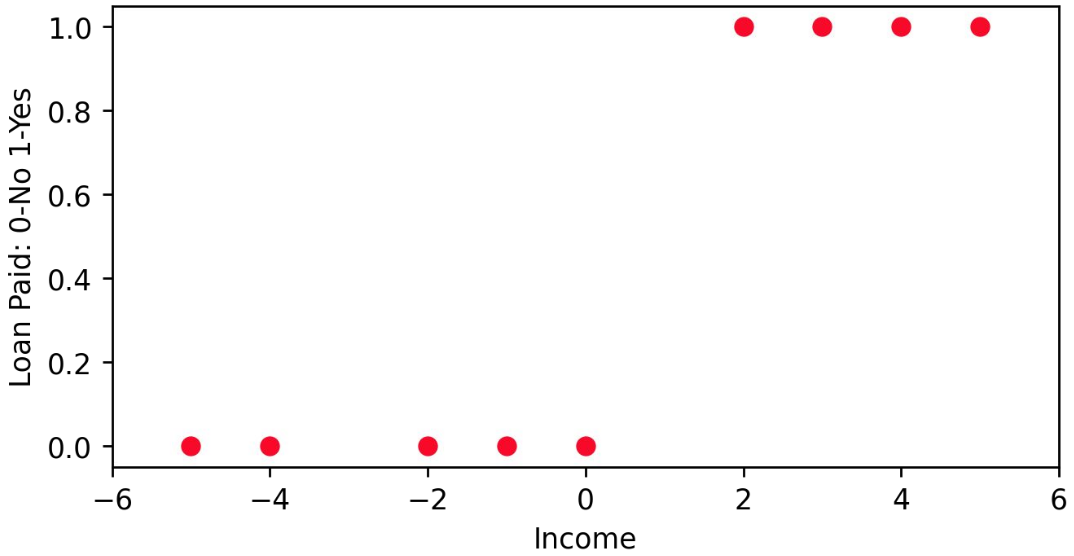
# Logistic Regression

- Our data set:

| Income | Loan Paid |
|---|---|
| -5 | 0 |
| -4 | 0 |
| -2 | 0 |
| -1 | 0 |
| 0 | 0 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |

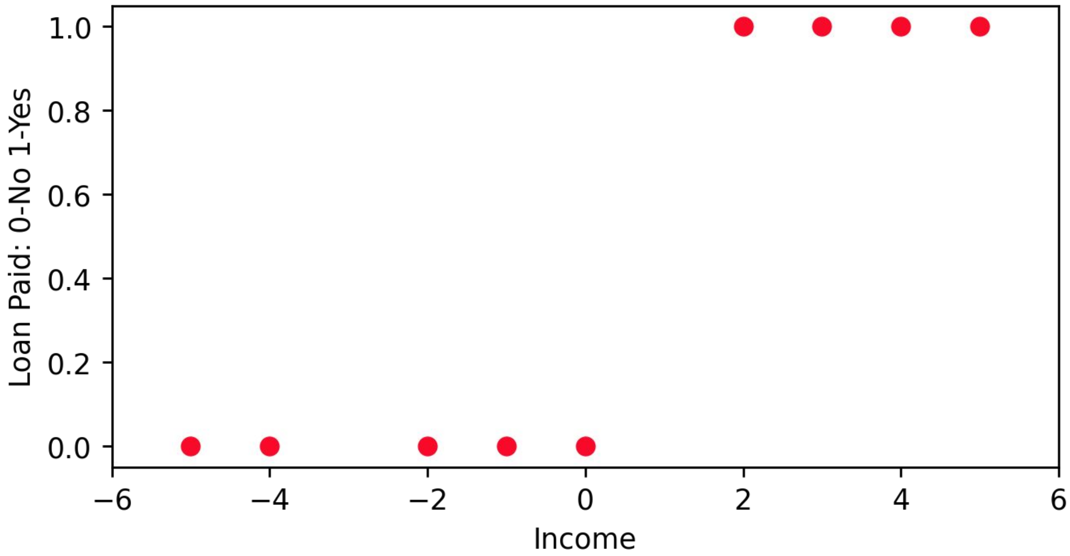
# Logistic Regression

- Let's begin by plotting income versus default:



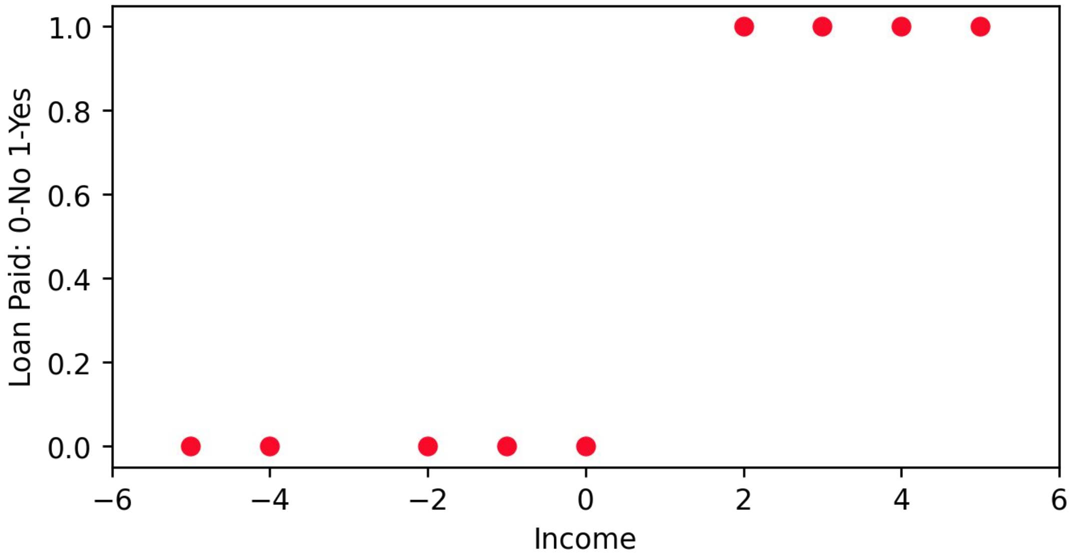
# Logistic Regression

- Notice that people with negative income tend to default on their loans.



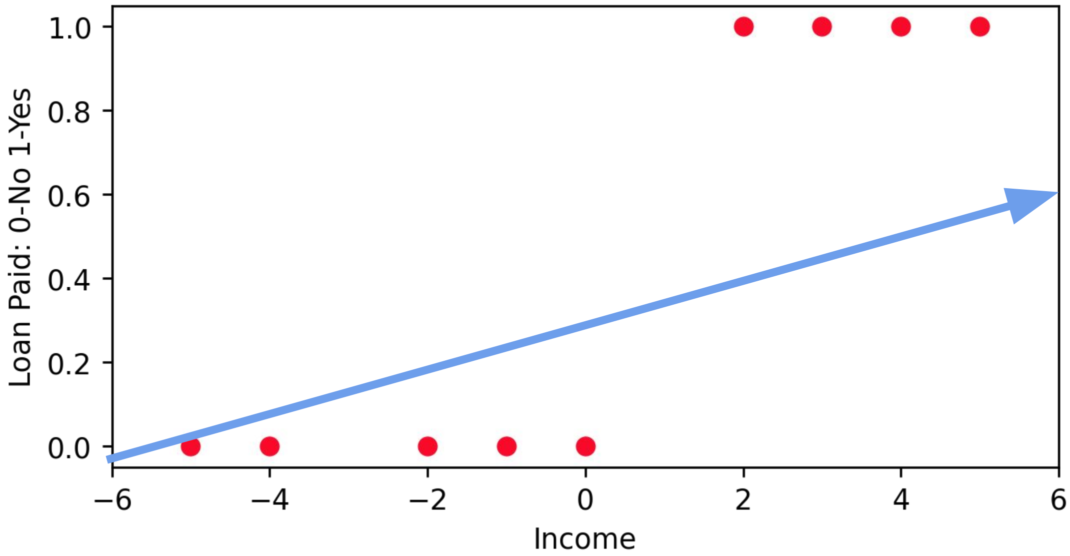
# Logistic Regression

- What if we had to predict default status given someone's income?



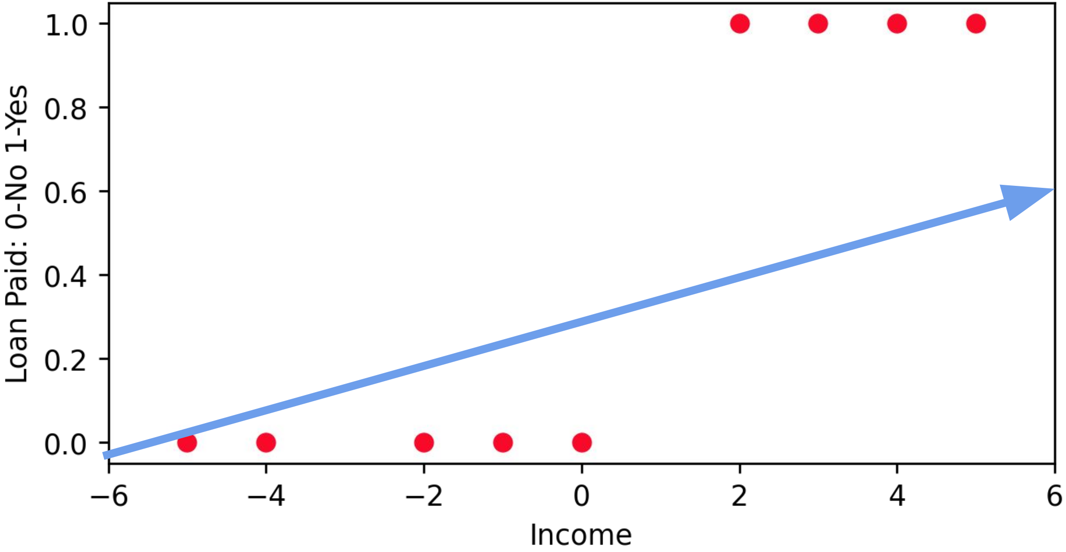
# Logistic Regression

- Fitting a Linear Regression would not work (recall Anscombe's quartet):



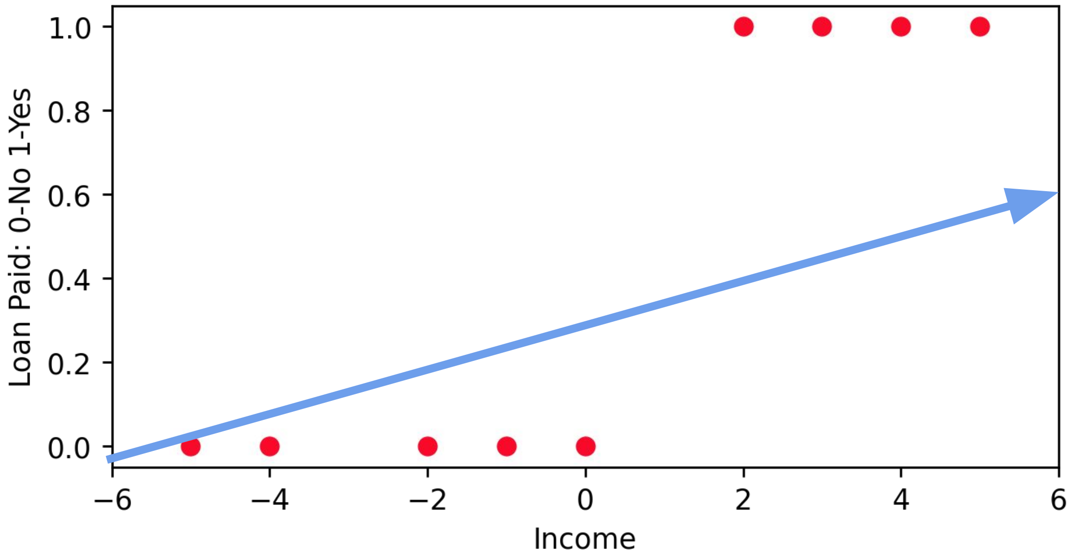
# Logistic Regression

- Linear Regression easily distorted by only having 0 and 1 as possible y training values.



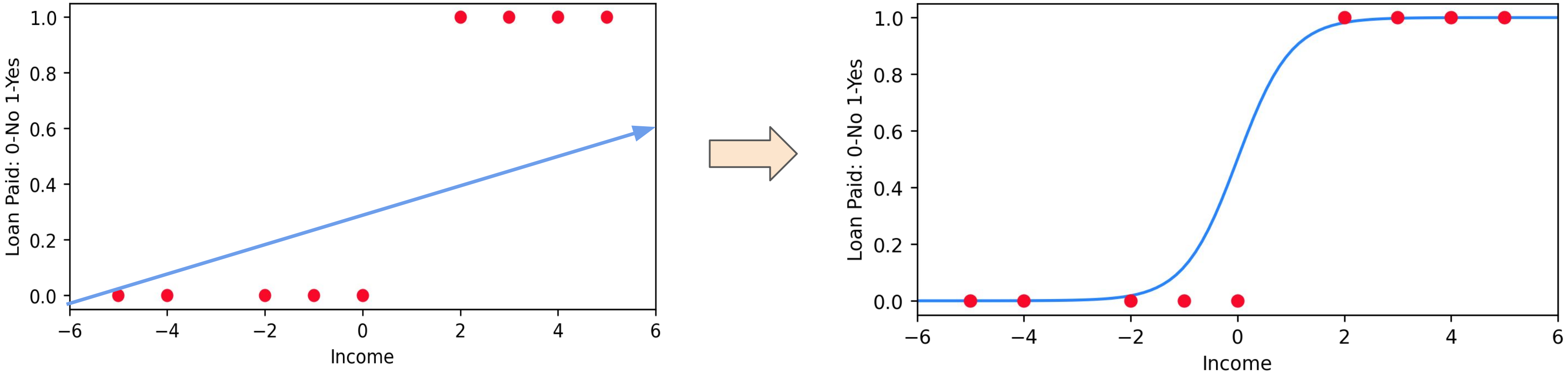
# Logistic Regression

- Also would be unclear how to interpret predicted y values between 0 and 1.



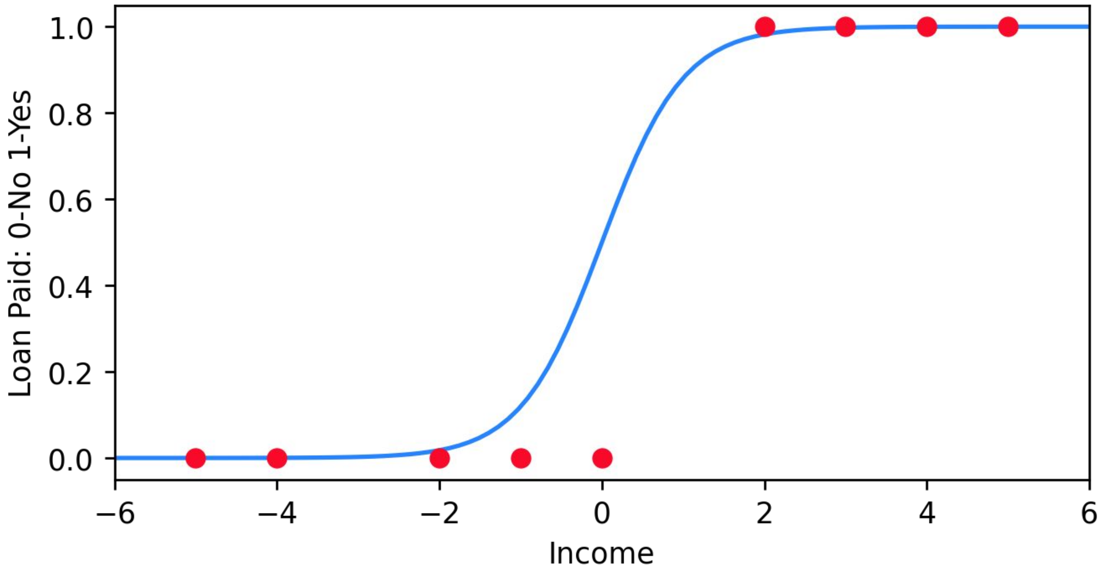
# Logistic Regression

- We could make use of the Logistic Function for a conversion!



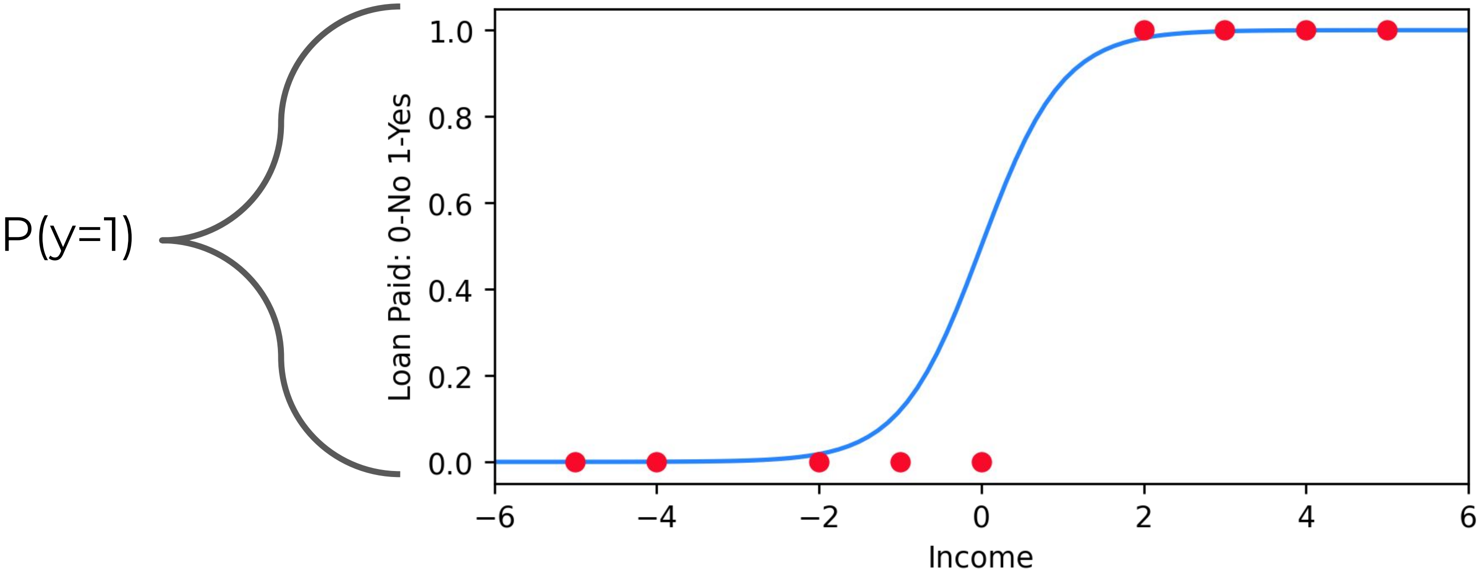
# Logistic Regression

- Let's first focus on what this Logistic Regression would look like.



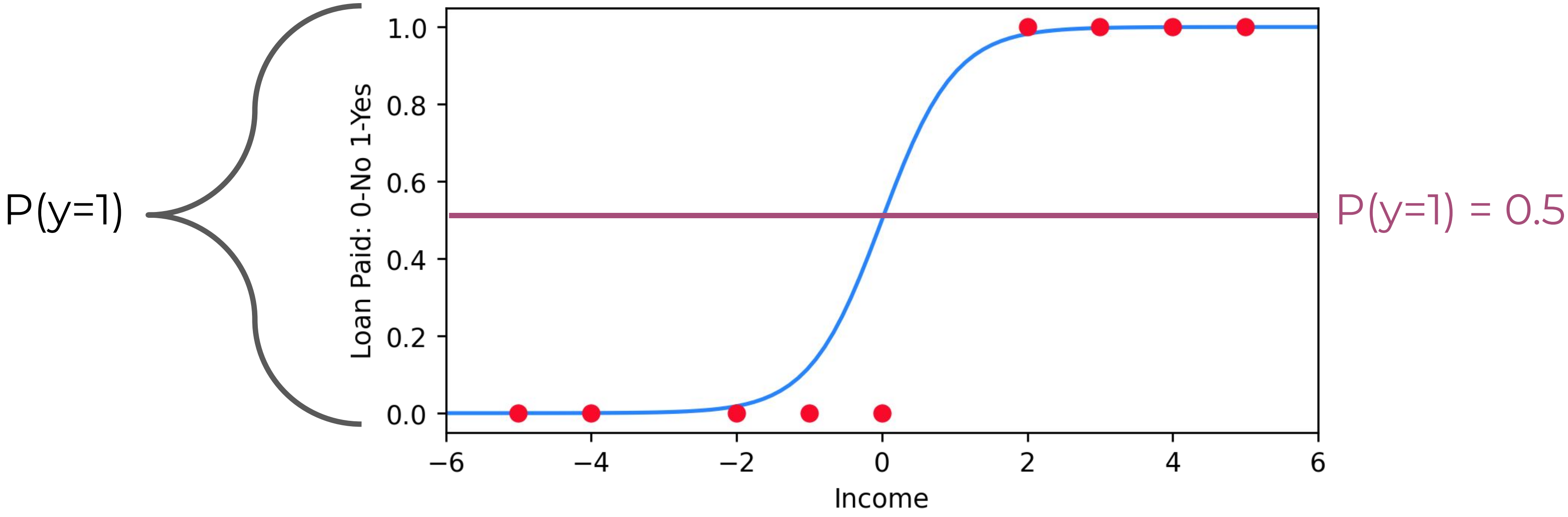
# Logistic Regression

- Treat the y-axis as a probability of belonging to a class:



# Logistic Regression

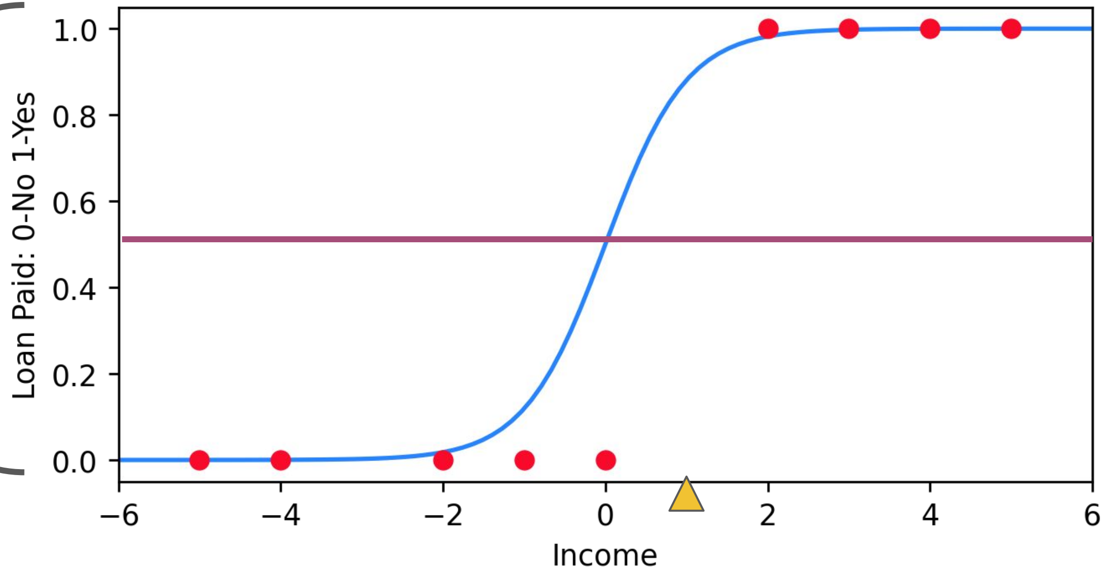
- Treating P(y=1) >= 0.5 as a cut-off for classification:



# Logistic Regression

- For example, a new person with an income of 1:

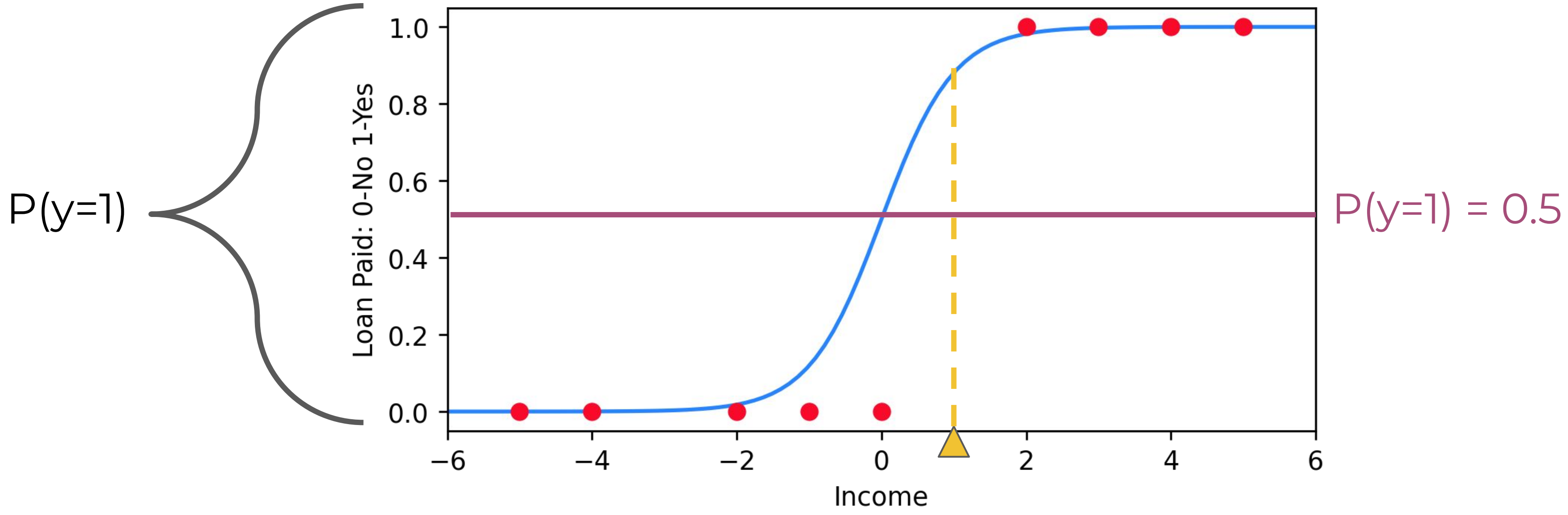
P(y=1)




P(y=1) = 0.5

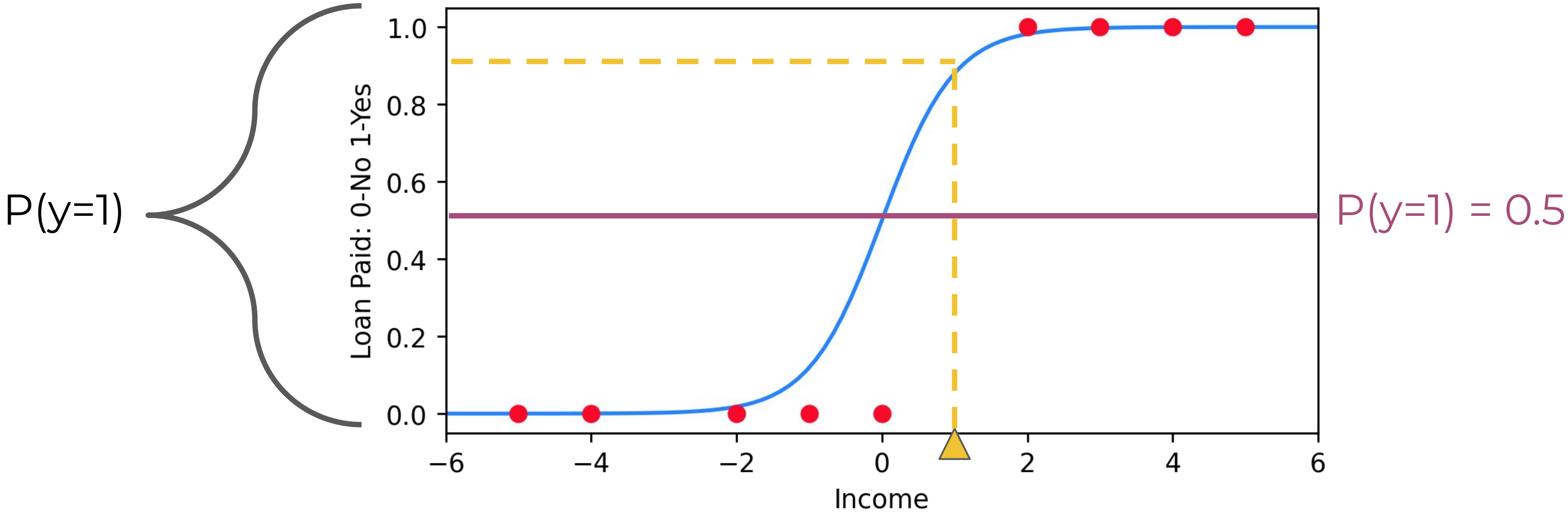
# Logistic Regression

- For example, a new person with an income of 1:



# Logistic Regression

- Predict a 90% probability of paying off loan, return prediction of Loan Paid = 1.

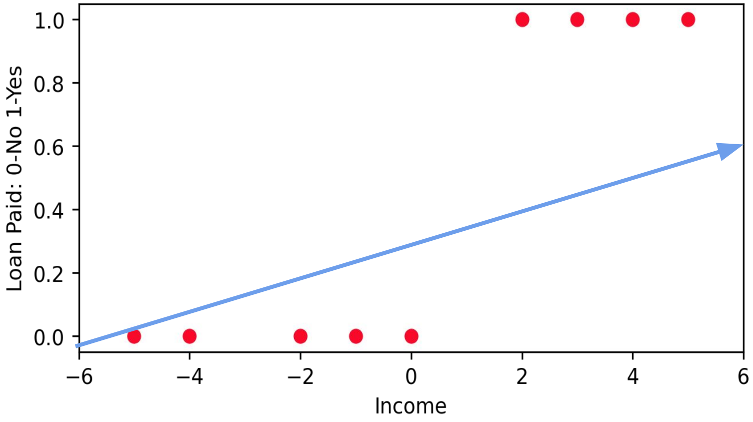


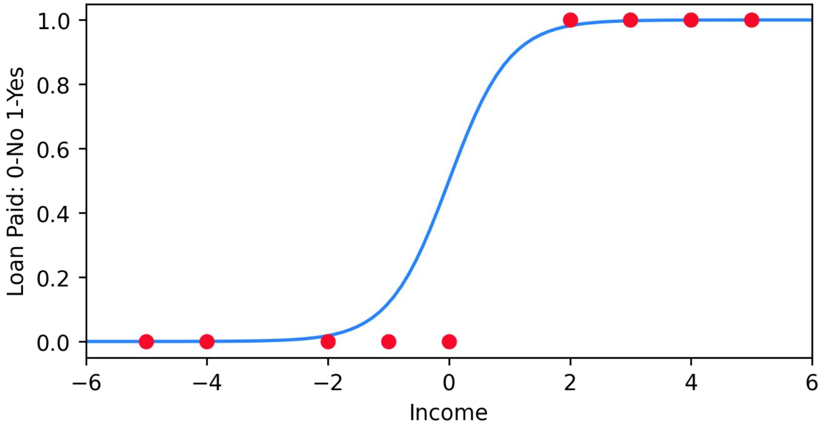
# Logistic Regression Theory and Intuition

## Part Two: Linear to Logistic Math

# Logistic Regression

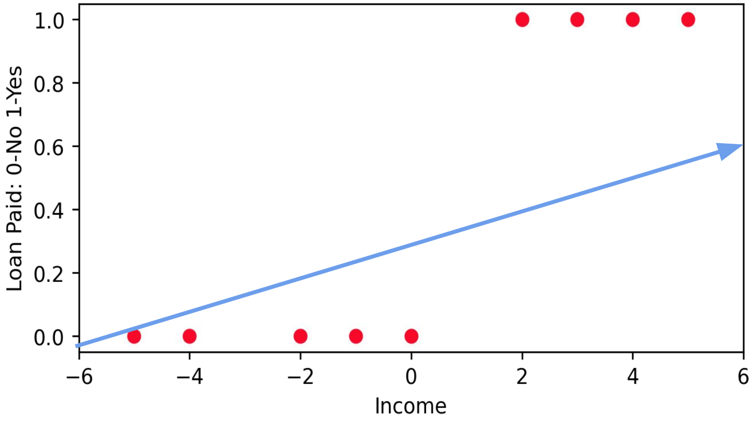
- Let’s go through the math of converting Linear Regression to Logistic Regression.

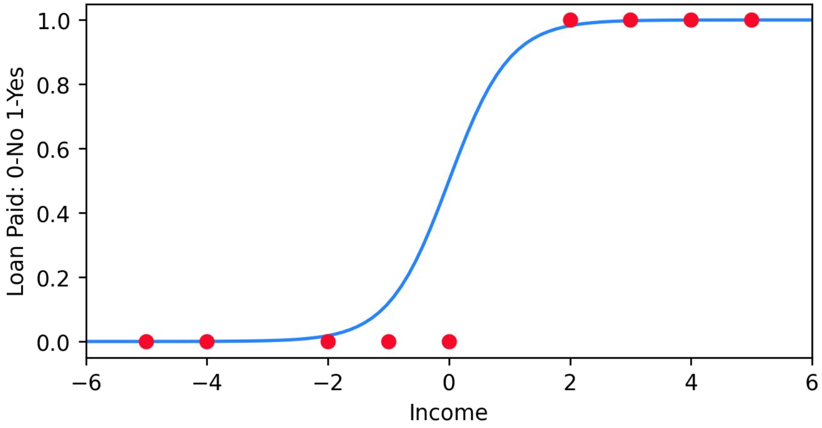


# Logistic Regression

- Relevant ISLR Reading:
    - Section 4.3 Logistic Regression



# Logistic Regression

- We already know the Linear Regression equation:

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

# Logistic Regression

- We also know the Logistic function transforms any input to be between 0 and 1

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

# Logistic Regression

- All we need to do is plug the Linear Regression equation into the Logistic function to create a Logistic Regression!

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

$$\hat{y} = \boxed{\sum_{i=0}^{n} \beta_i x_i}$$

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

# Logistic Regression

- Simply put in terms of the logistic function:

$$\hat{y} = \sigma(\beta_0 x_0 + \cdots + \beta_n x_n)$$

$$\hat{y} = \sigma\left(\sum_{i=0}^{n} \beta_i x_i\right)$$

# Logistic Regression

- Writing it out fully:

$$\hat{y} = \frac{1}{1 + e^{-\sum_{i=0}^{n} \beta_i x_i}}$$

# Logistic Regression

- Solving for **log odds:**

$$\hat{y} + \hat{y}e^{-\sum_{i=0}^{n}\beta_i x_i} = 1$$

$$\hat{y}e^{-\sum_{i=0}^{n}\beta_i x_i} = 1 - \hat{y}$$

$$\frac{\hat{y}}{1-\hat{y}} = e^{\sum_{i=0}^{n}\beta_i x_i}$$

# Logistic Regression

- Solving for **log odds:**

$$\frac{\hat{y}}{1-\hat{y}} = e^{\sum_{i=0}^{n} \beta_i x_i}$$

$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i$$

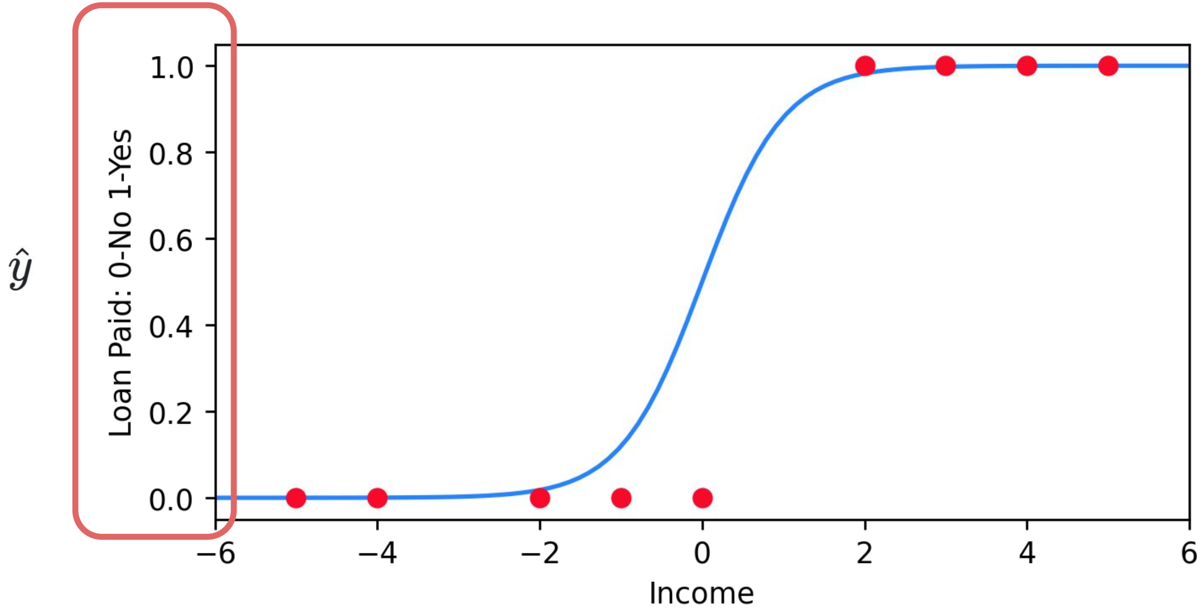
# Logistic Regression

- What would the function curve look like in terms of log odds?

$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i$$

# Logistic Regression

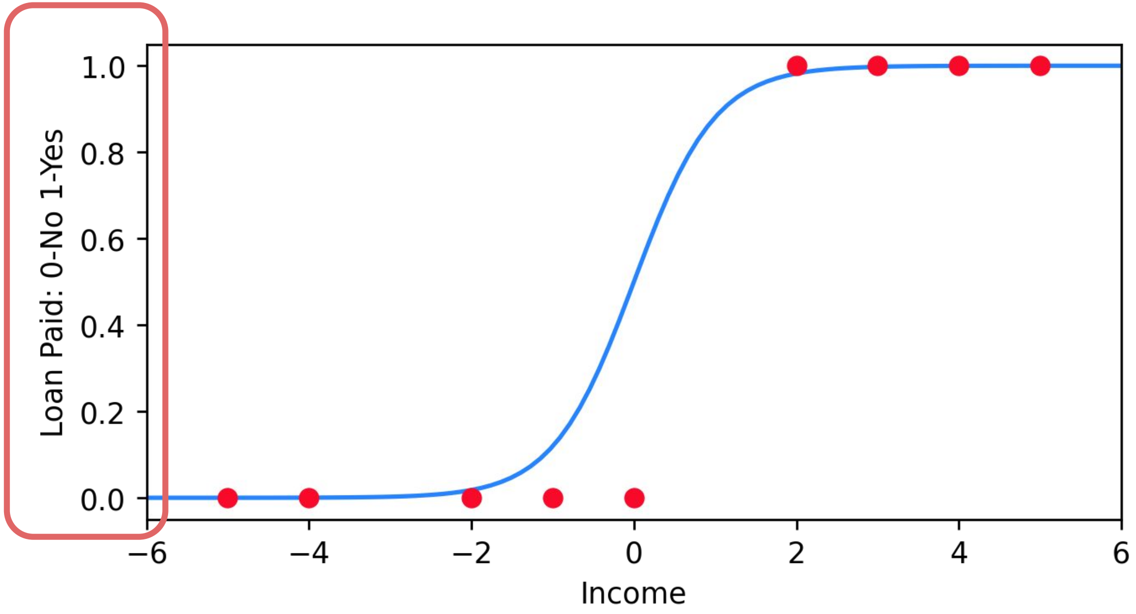
- What would the function curve look like in terms of log odds?



# Logistic Regression

- What would the function curve look like in terms of log odds?

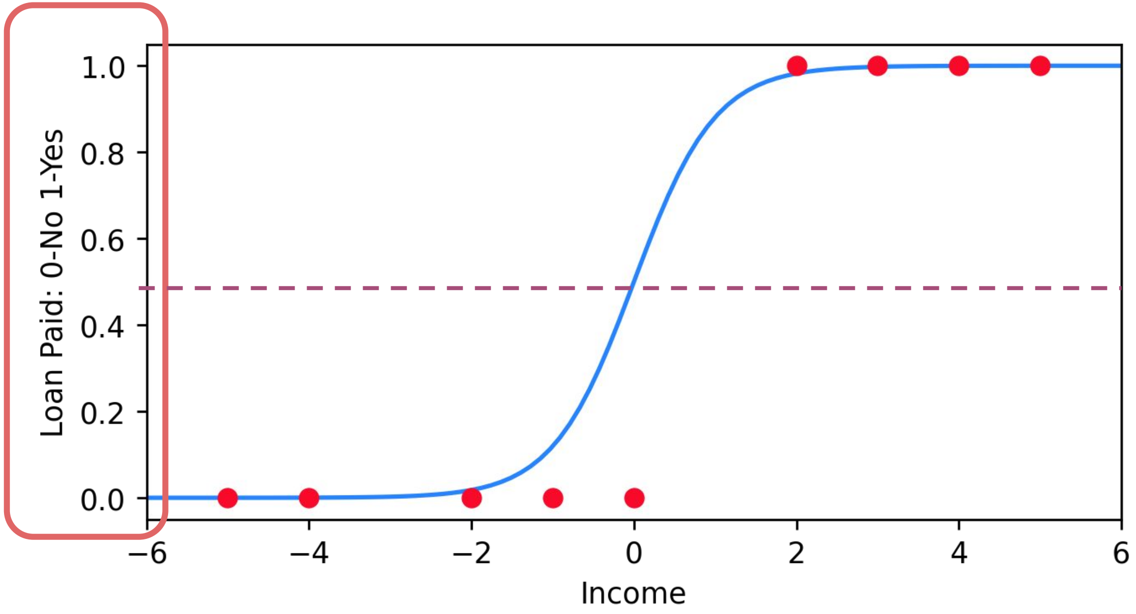
$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) \Leftarrow \hat{y}$$



# Logistic Regression

- Consider p=0.5

$$ln(\frac{0.5}{1-0.5}) = 0$$



# Logistic Regression

- Consider p=0.5, halfway point now at 0.

$$ln(\frac{0.5}{1-0.5}) = 0$$

# Logistic Regression

- As p goes to 1 then log odds becomes ∞

$$\lim_{p \to 1} ln\left(\frac{p}{1-p}\right) = \infty$$

$$ln\left(\frac{0.5}{1-0.5}\right) = 0$$

# Logistic Regression

- As p goes to 0 then log odds becomes -∞

$$\lim_{p \to 1} ln(\frac{p}{1-p}) = \infty$$

$$ln(\frac{0.5}{1-0.5}) = 0$$

$$\lim_{p \to 0} ln(\frac{p}{1-p}) = -\infty$$

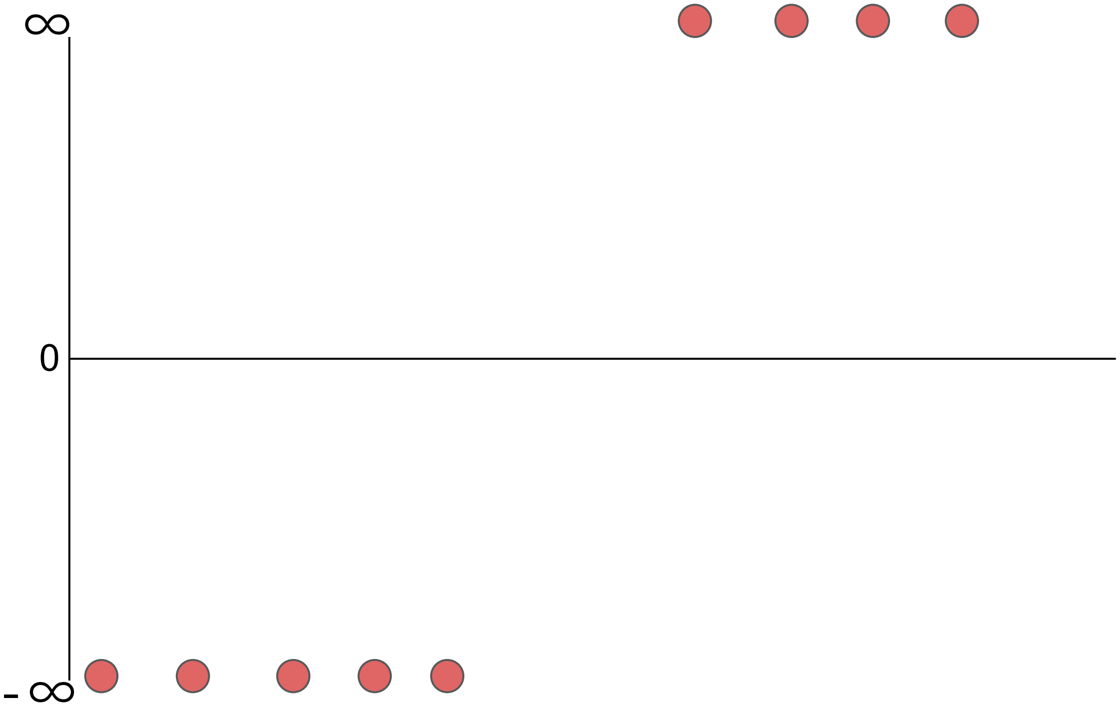
# Logistic Regression

- Class points now at infinity

$$\lim_{p \to 1} ln(\frac{p}{1-p}) = \infty$$

$$ln(\frac{0.5}{1-0.5}) = 0$$

$$\lim_{p \to 0} ln(\frac{p}{1-p}) = -\infty$$



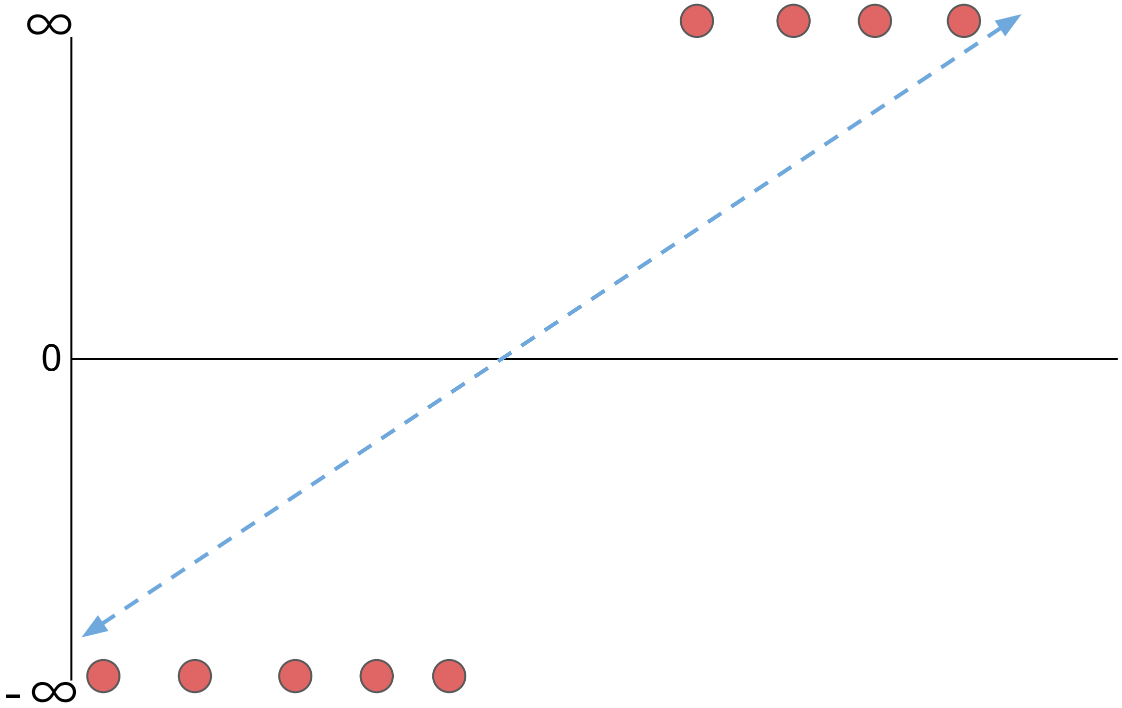
# Logistic Regression

- On log scale logistic function is straight line

$$\lim_{p \to 1} ln(\frac{p}{1-p}) = \infty$$

$$ln(\frac{0.5}{1-0.5}) = 0$$

$$\lim_{p \to 0} ln(\frac{p}{1-p}) = -\infty$$



# Logistic Regression

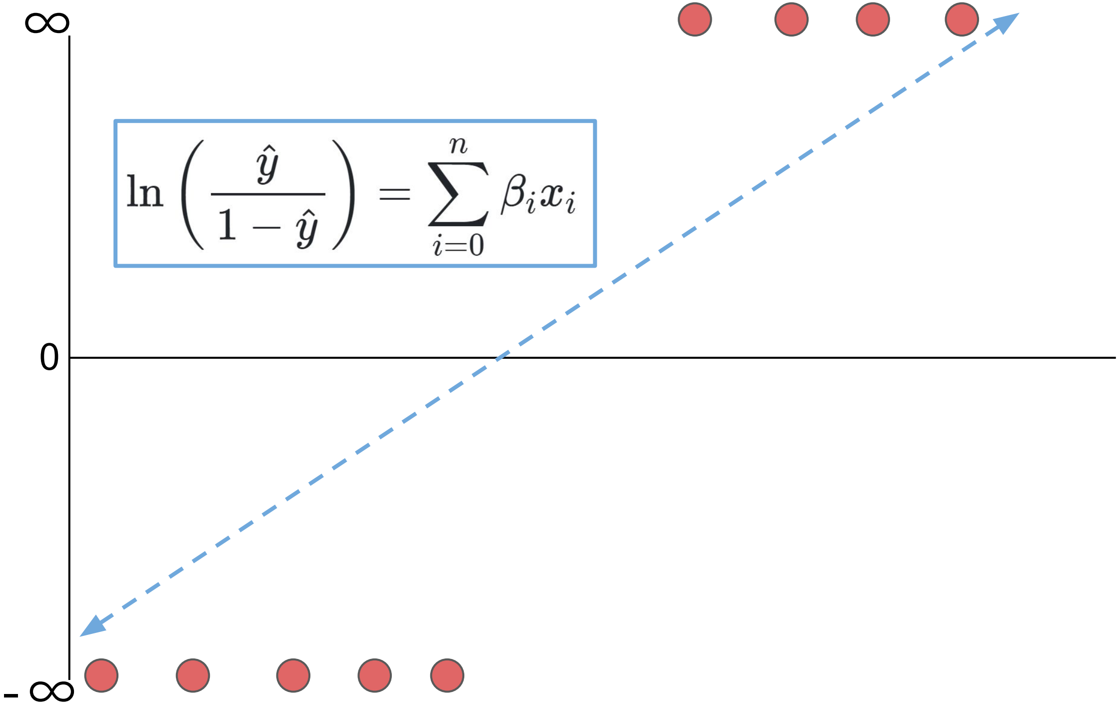
- Coefficients in terms of change in log odds.

$$\lim_{p \to 1} ln\left(\frac{p}{1-p}\right) = \infty$$

$$ln\left(\frac{0.5}{1-0.5}\right) = 0$$

$$\lim_{p \to 0} ln\left(\frac{p}{1-p}\right) = -\infty$$

$$\boxed{\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i}$$



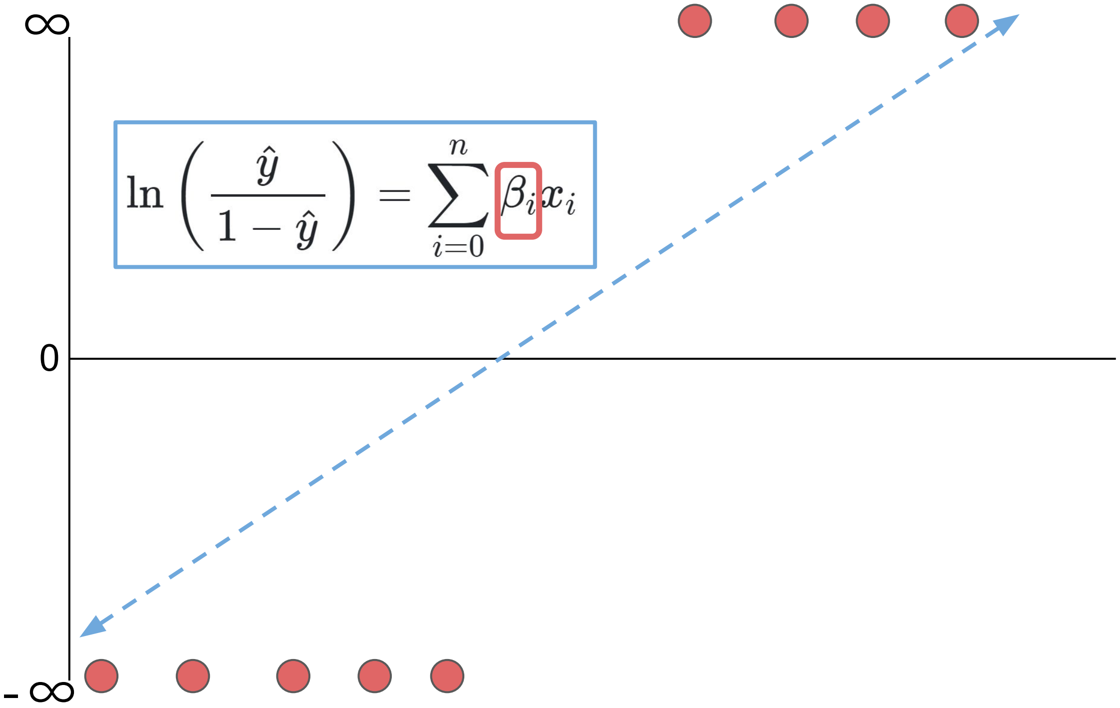
# Logistic Regression

- Is **β** simple to interpret? Not really...

$$\lim_{p \to 1} ln(\frac{p}{1-p}) = \infty$$

$$ln(\frac{0.5}{1-0.5}) = 0$$

$$\lim_{p \to 0} ln(\frac{p}{1-p}) = -\infty$$




$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \boxed{\beta_i} x_i$$

# Logistic Regression

- Since the log odds scale is nonlinear, a **β** value can not be directly linked to “one unit increase” as it could in Linear Regression.

$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i$$

# Logistic Regression

- There are some straightforward insights we can gain however...

$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i$$

# Logistic Regression

- Sign of Coefficient
  - Positive **β** indicates an increase in likelihood of belonging to 1 class with increase in associated **x** feature.
  - Negative **β** indicates an decrease in likelihood of belonging to 1 class with increase in associated **x** feature.

# Logistic Regression

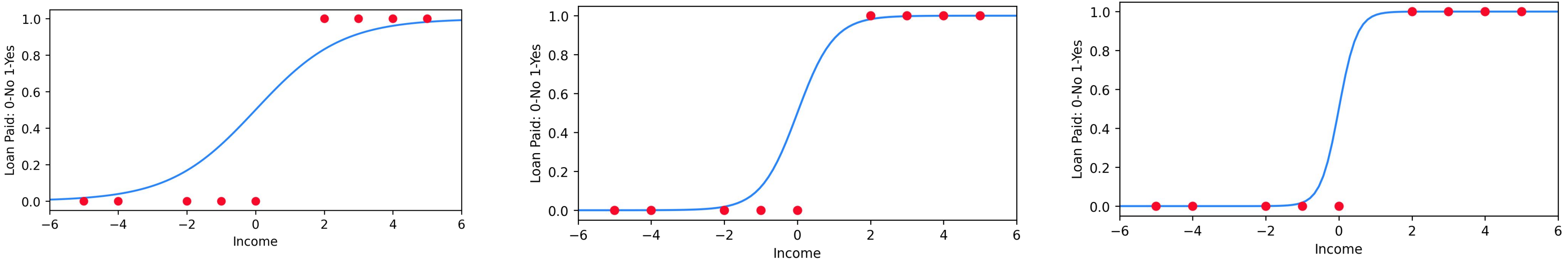
- Magnitude of Coefficient
  - Harder to directly interpret magnitude of **β** directly, especially when we could have discrete and continuous x feature values.
  - We can however begin to use **odds ratio**, essentially comparing magnitudes against each other.

# Logistic Regression

- Magnitude of Coefficient
  - Comparing magnitudes of coefficients against each other can lead to insight over which features have the strongest effect on prediction output.

# Logistic Regression

- The last mathematical topic we need to discuss concerning Logistic Regression is how we actually fit this curve!



# Logistic Regression Theory and Intuition

## Part Three: Finding the Best Fit

# Logistic Regression

- Logistic Regression uses Maximum Likelihood to find the best fitting model.
- This lecture will give you an intuition of how this method works.
- We’ll also then display the cost function and gradient descent that is solved for by the computer.

# Logistic Regression

- Quick Note: ISLR Section 4.3.2

default status. In other words, we try to find $\hat{\beta}_0$ and $\hat{\beta}_1$ such that plugging these estimates into the model for $p(X)$, given in (4.2), yields a number close to one for all individuals who defaulted, and a number close to zero for all individuals who did not. This intuition can be formalized using a mathematical equation called a *likelihood function*:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})). \tag{4.5}$$

The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to *maximize* this likelihood function.

Maximum likelihood is a very general approach that is used to fit many of the non-linear models that we examine throughout this book. In the linear regression setting, the least squares approach is in fact a special case of maximum likelihood. The mathematical details of maximum likelihood are beyond the scope of this book. However, in general, logistic regression

# Logistic Regression

- Quick Note: ISLR Section 4.3.2

default status. In other words, we try to find $\hat{\beta}_0$ and $\hat{\beta}_1$ such that plugging these estimates into the model for $p(X)$, given in (4.2), yields a number close to one for all individuals who defaulted, and a number close to zero for all individuals who did not. This intuition can be formalized using a mathematical equation called a *likelihood function*:

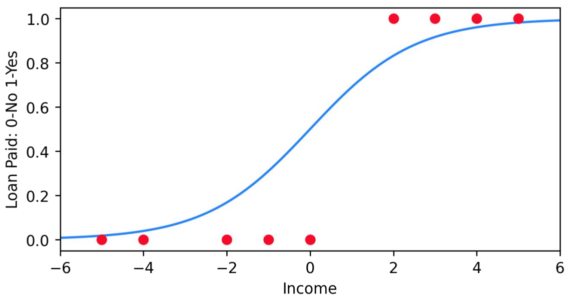
$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})). \tag{4.5}$$

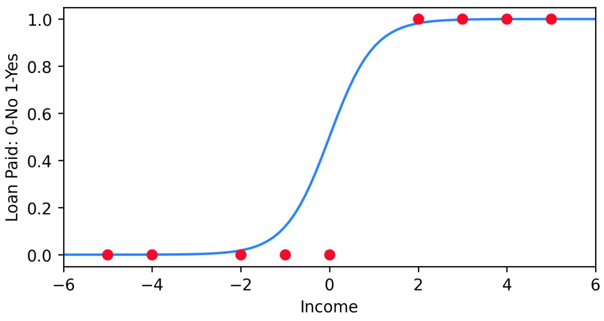
The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to *maximize* this likelihood function.

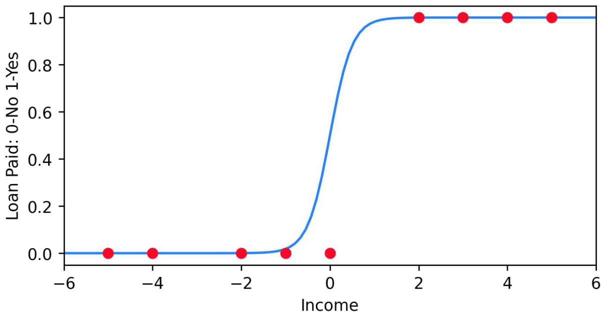
Maximum likelihood is a very general approach that is used to fit many of the non-linear models that we examine throughout this book. In the linear regression setting, the least squares approach is in fact a special case of maximum likelihood. The mathematical details of maximum likelihood are beyond the scope of this book. However, in general, logistic regression

# Logistic Regression

- Here we see three different Logistic Regression curves with different **β** values.
- How do we measure which is the best fit?

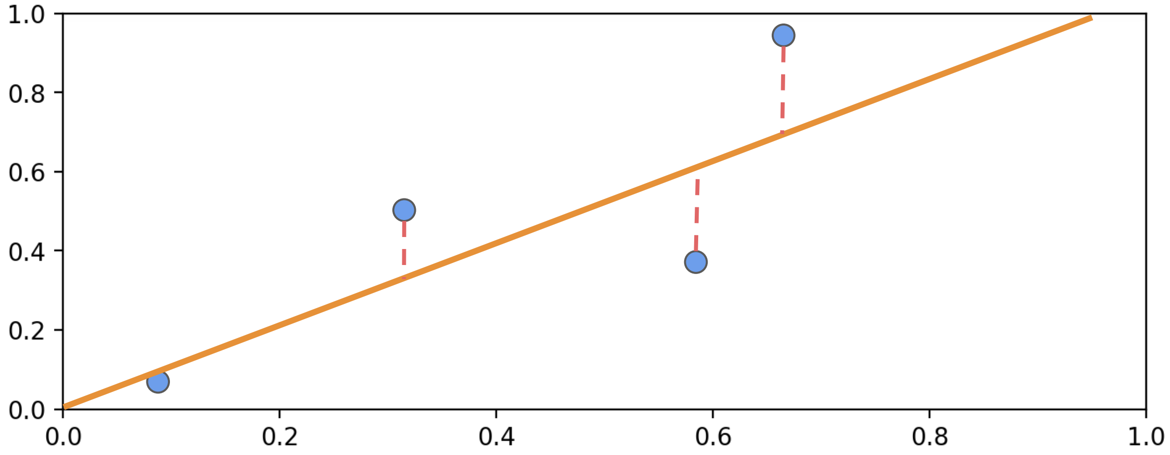


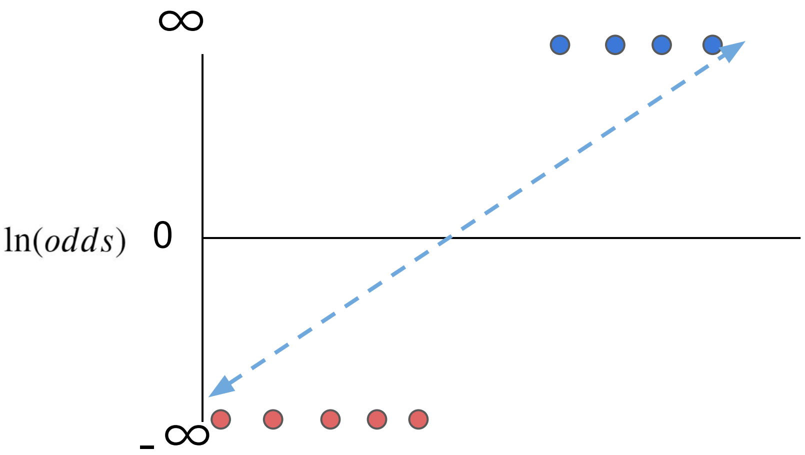
# Logistic Regression

- Recall in Linear Regression we seek to minimize the Residual Sum of Squares (RSS).



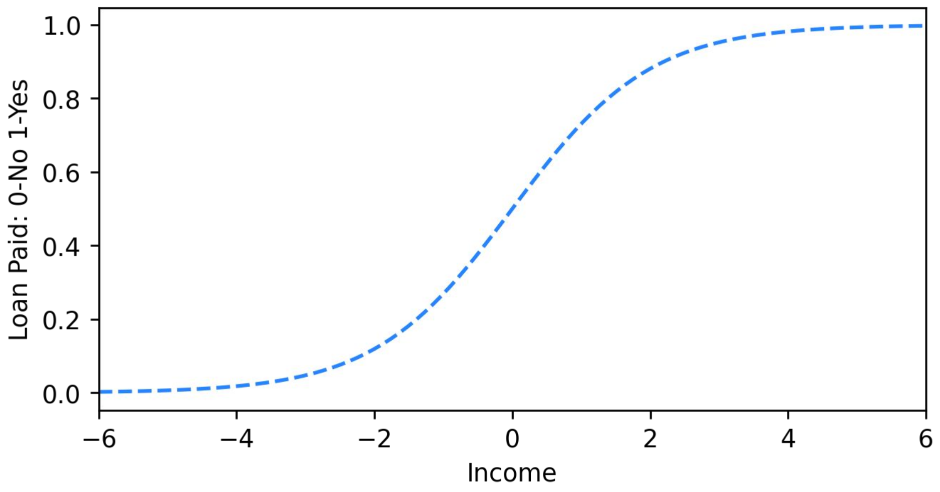
# Logistic Regression

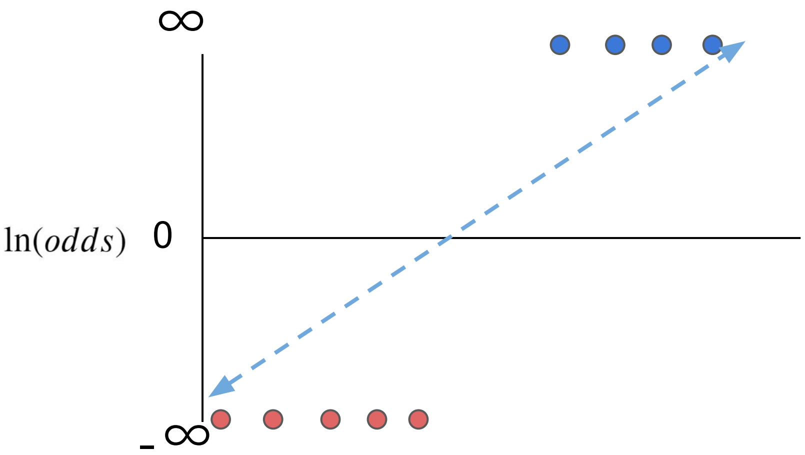
- We choose a line in the log(odds) axis and project the points on to the line:



# Logistic Regression

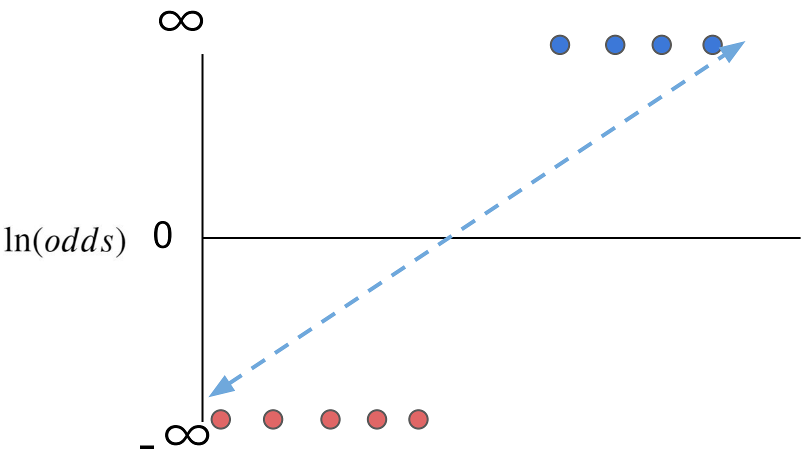
- We also know this line has a form on the probability y-axis.



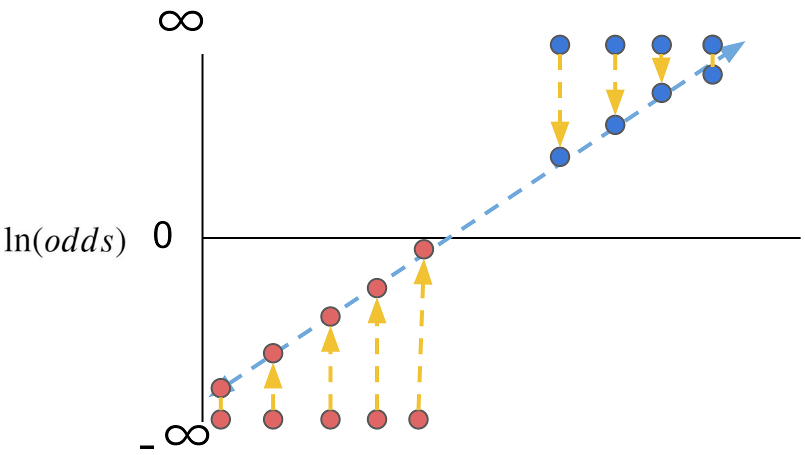
# Logistic Regression

- We choose a line in the log(odds) axis and project the points on to the line:



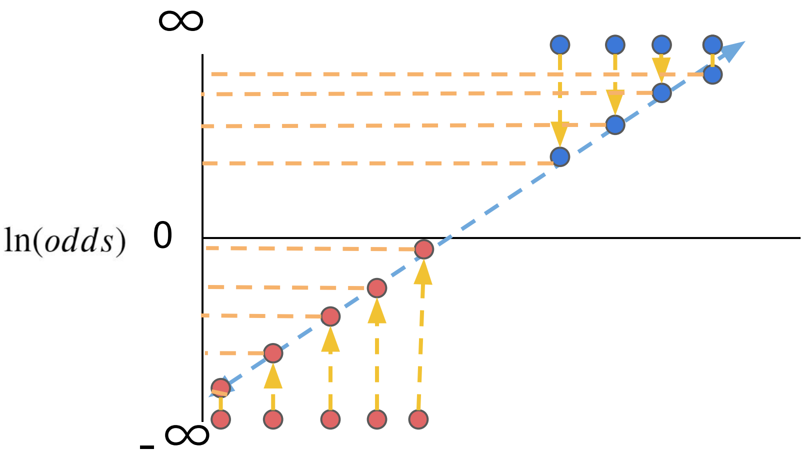
# Logistic Regression

- We choose a line in the log(odds) axis and project the points on to the line:



# Logistic Regression

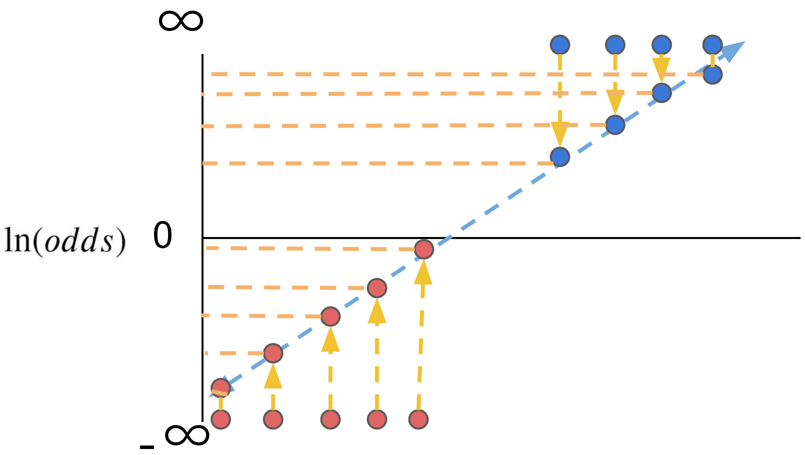
- Calculate the log odds for the projected points on this line.



# Logistic Regression

- Plot these values as probabilities on the logistic regression model.

$$p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}$$

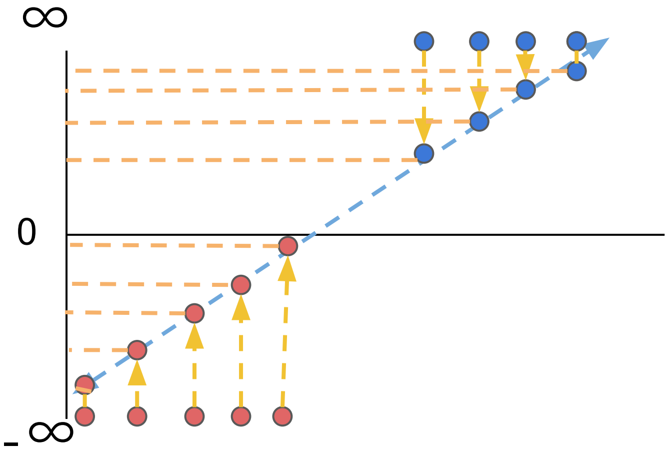


# Logistic Regression

- Plot these values as probabilities on the logistic regression model.

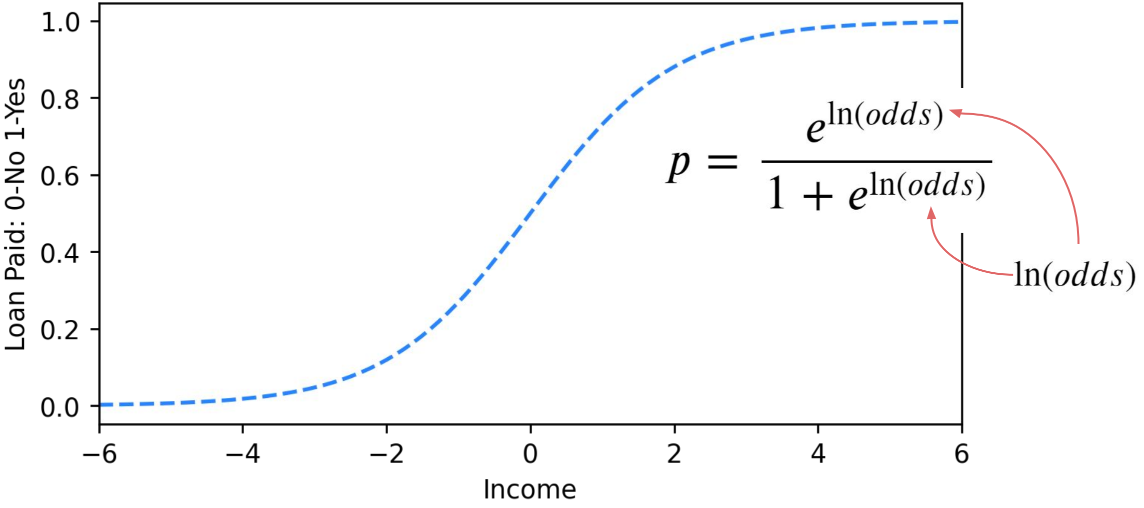
$$p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}$$

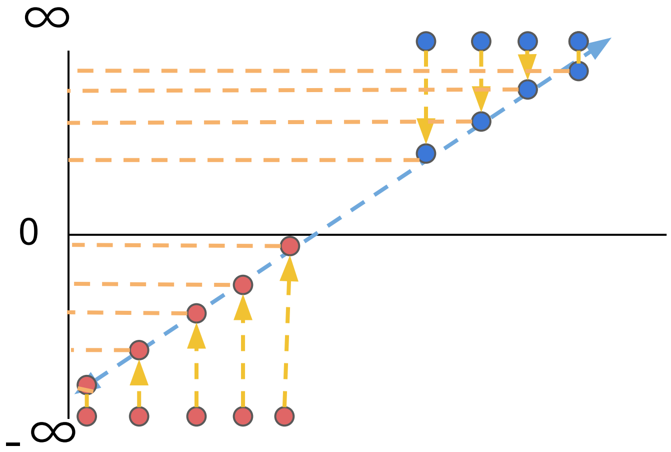
$\ln(odds)$



# Logistic Regression

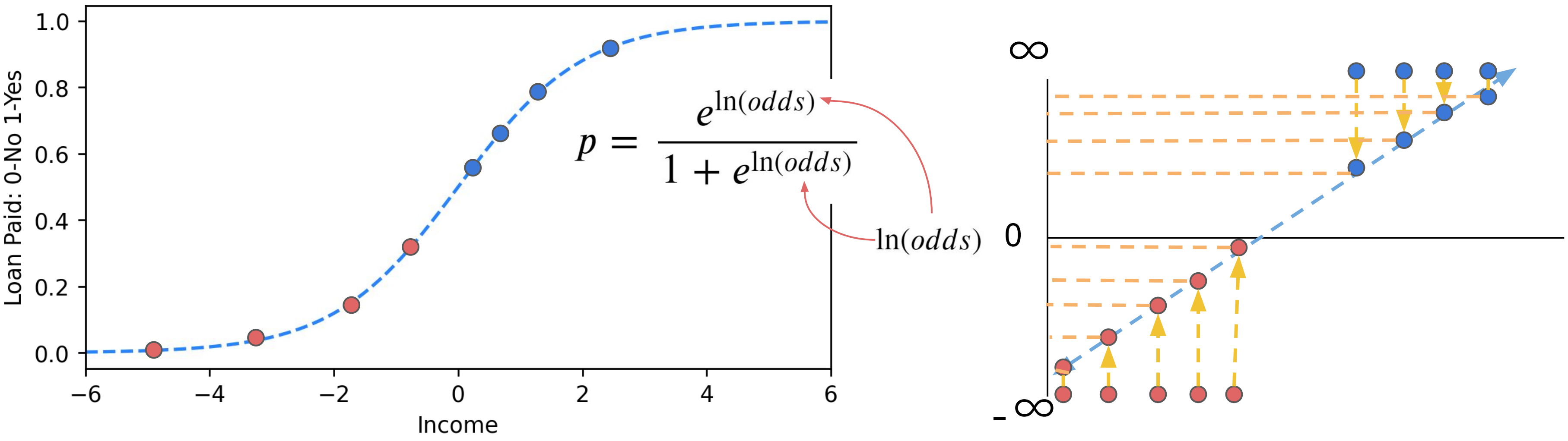
- Plot these values as probabilities on the logistic regression model.



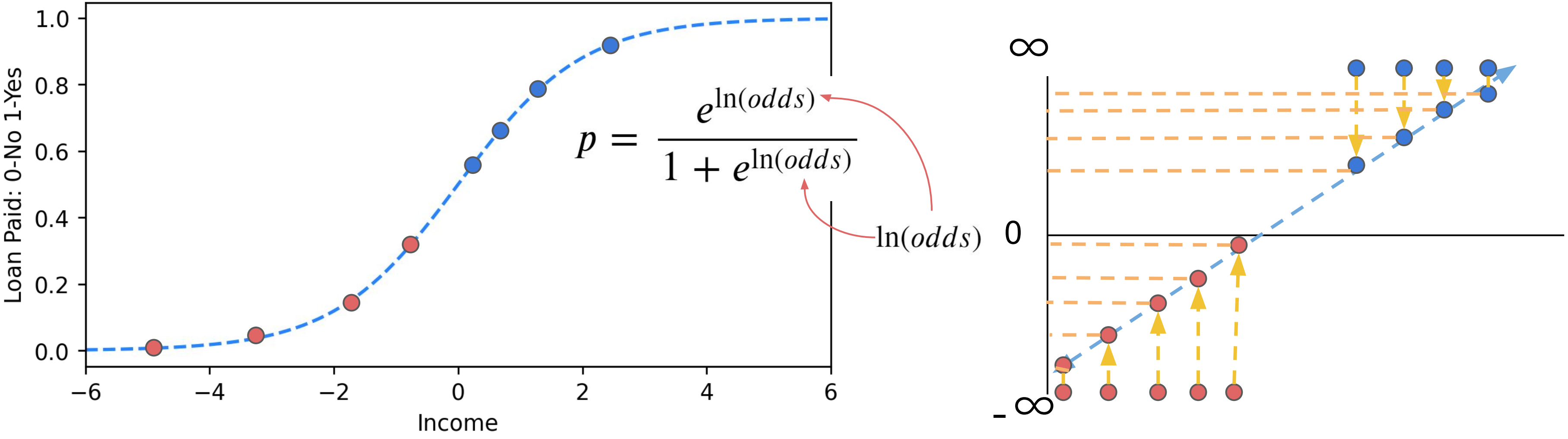
# Logistic Regression

- Plot these values as probabilities on the logistic regression model.



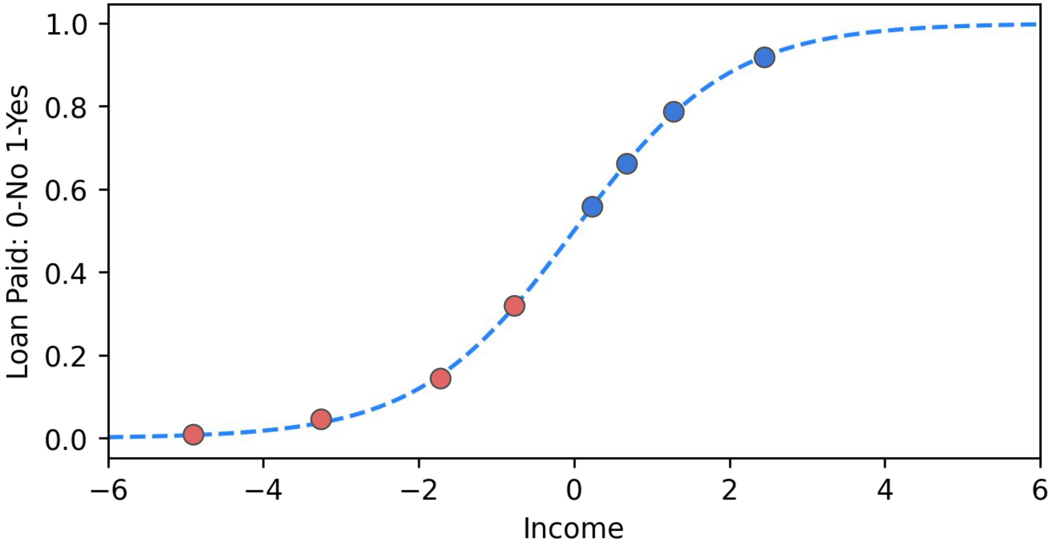
# Logistic Regression

- We now measure the likelihood of these probabilities.



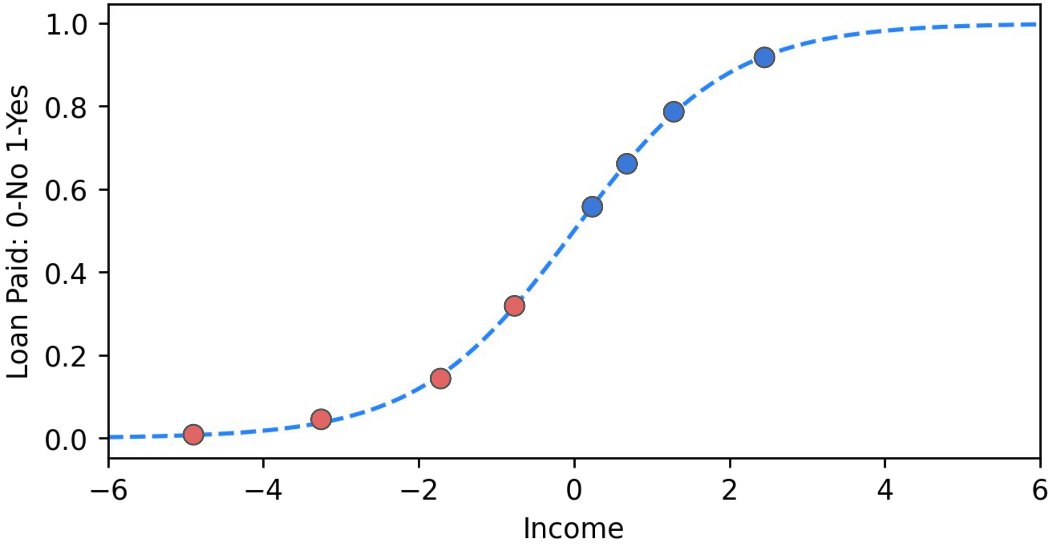
# Logistic Regression

- We now measure the likelihood of these probabilities.



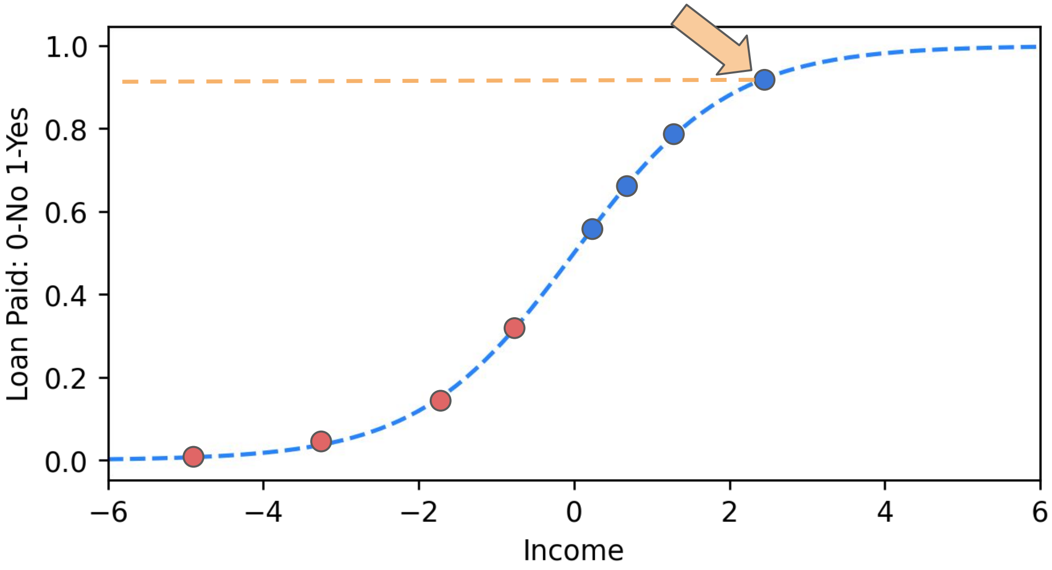
# Logistic Regression

- Likelihood = Product of probabilities of belonging to class 1.



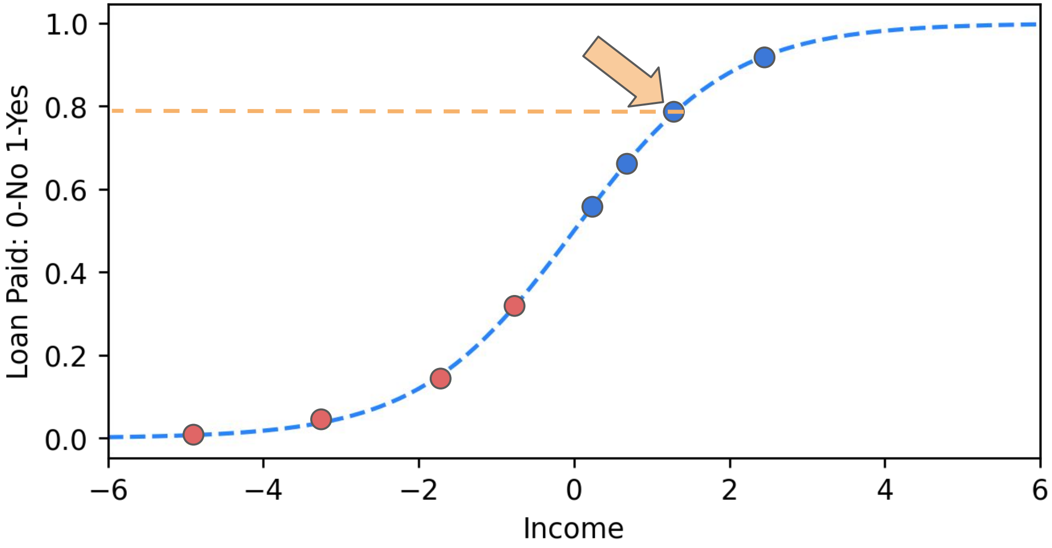
# Logistic Regression

- Likelihood = 0.9 ...



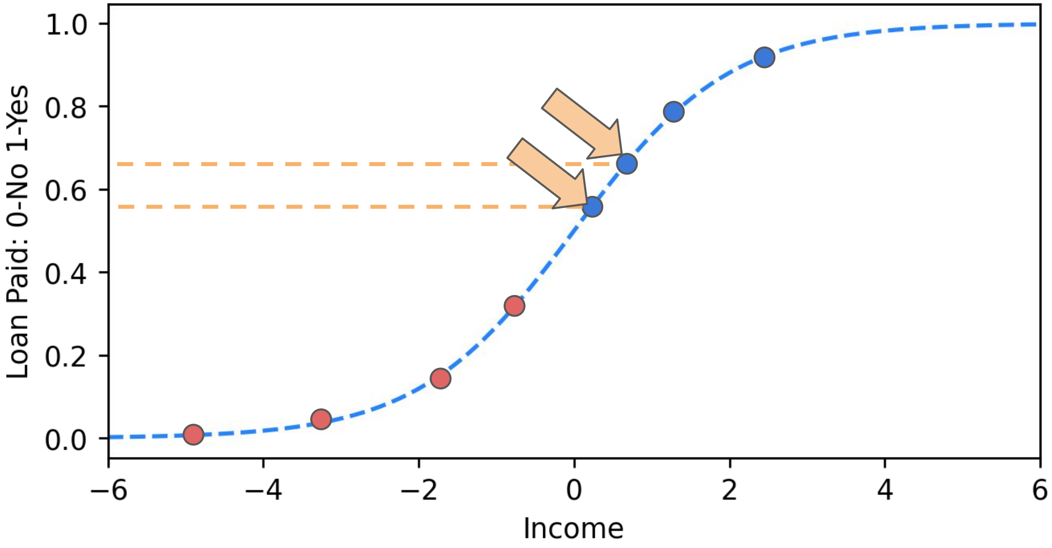
# Logistic Regression

- Likelihood = 0.9 × 0.8 × ...



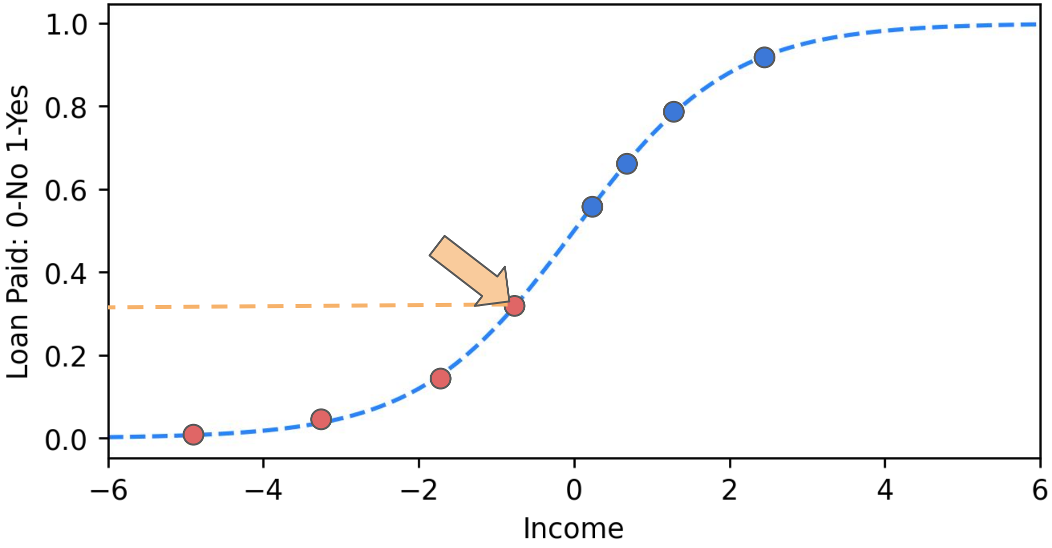
# Logistic Regression

- Likelihood = 0.9 × 0.8 × 0.65 × 0.55 × ...



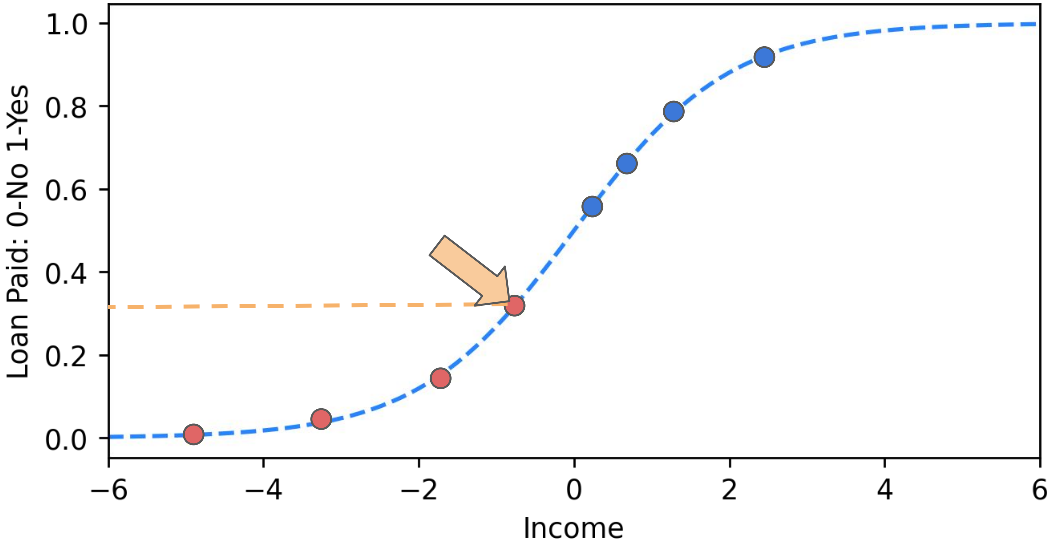
# Logistic Regression

- Likelihood = 0.9 × 0.8 × 0.65 × 0.55 × (1-p) ×…



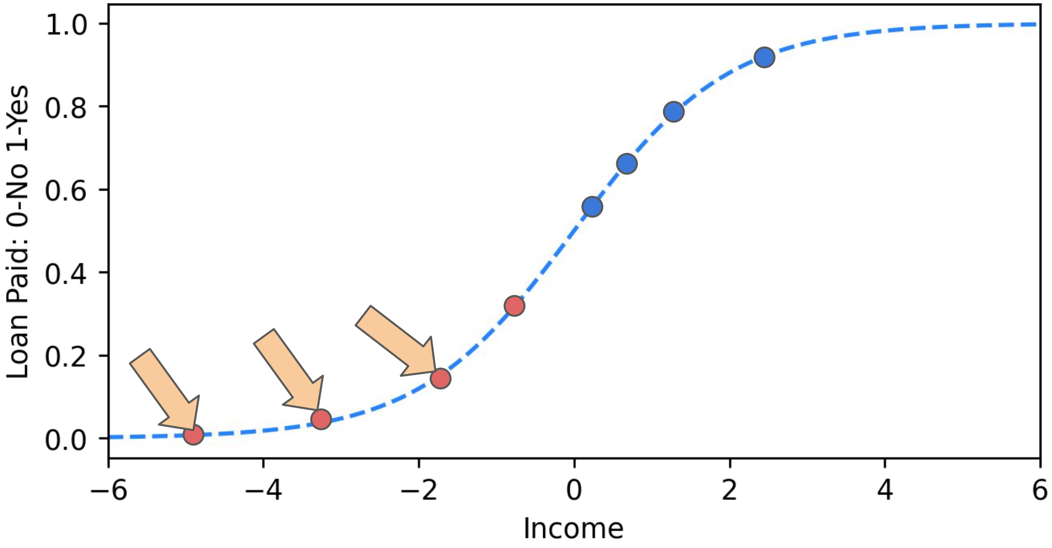
# Logistic Regression

- Likelihood = 0.9 × 0.8 × 0.65 × 0.55 × (1-0.3) ×…



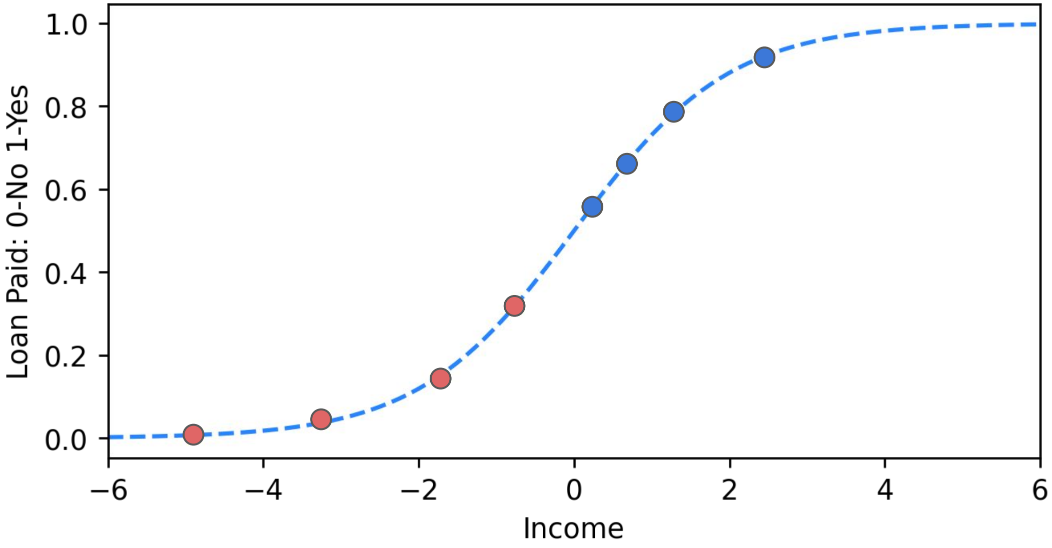
# Logistic Regression

- Likelihood = 0.9 × 0.8 × 0.65 × 0.55 × (1-0.3) × (1-0.2) × (1-0.08) × (1-0.02)



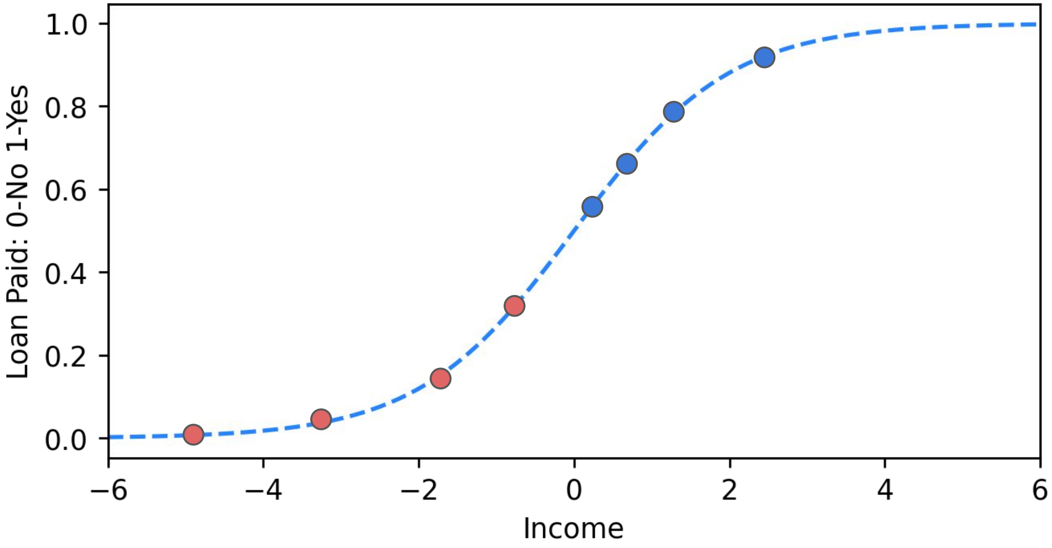
# Logistic Regression

- Likelihood = 0.129



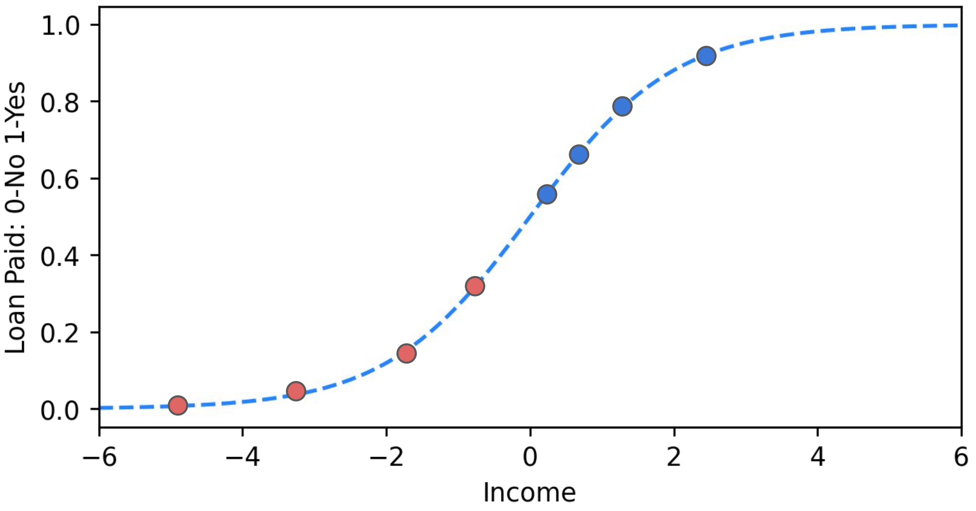
# Logistic Regression

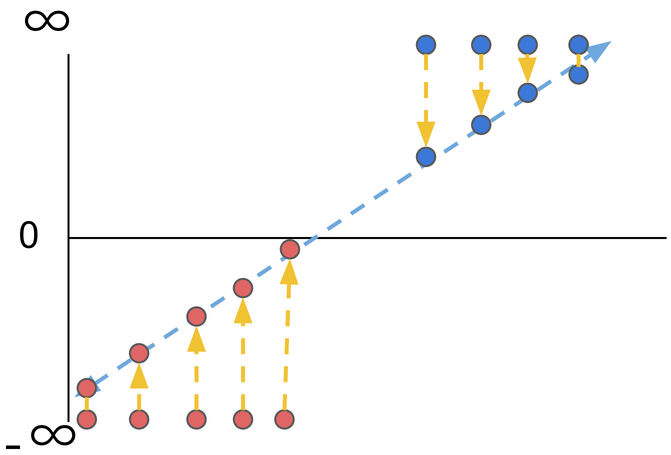
- Note in practice we actually maximize the **log** of the likelihoods. (e.g. ln(0.9)×ln(0.8)×...)



# Logistic Regression

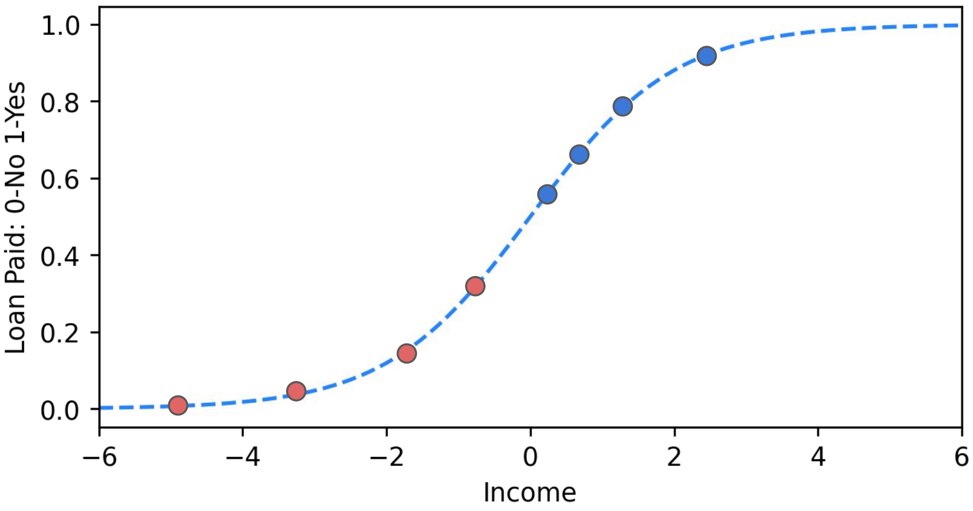
- There is some set of coefficients that will maximize these log likelihoods.

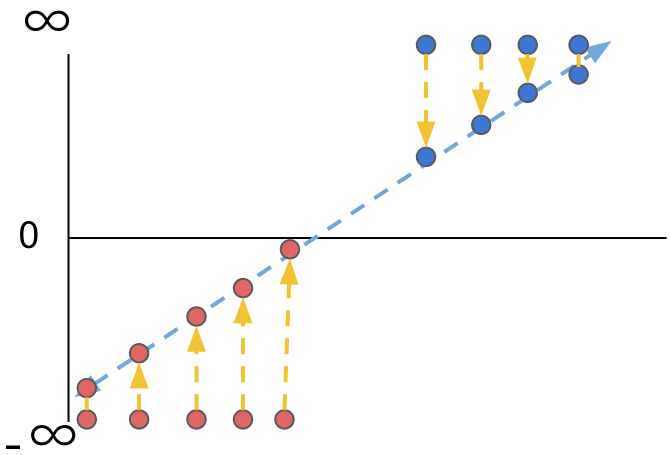


# Logistic Regression

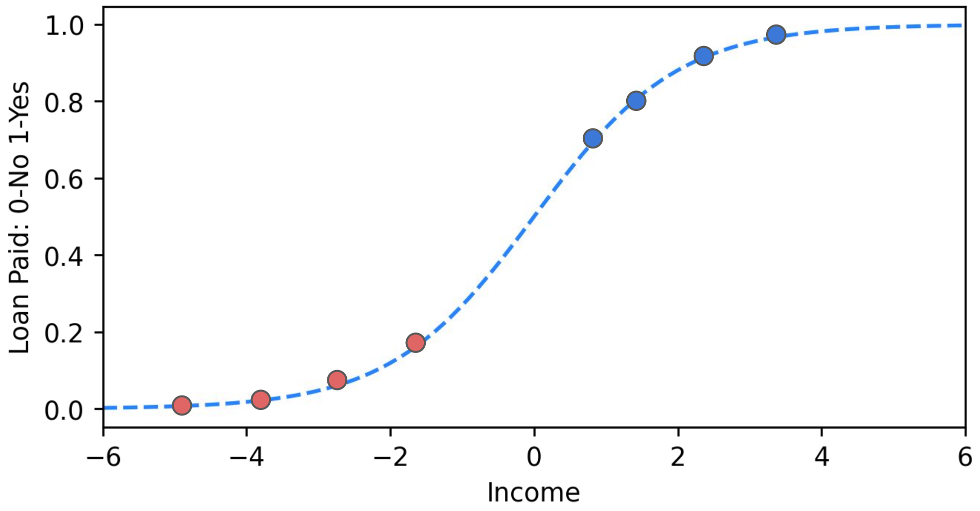
- Choose best coefficient values in log odds terms that creates maximum likelihood.

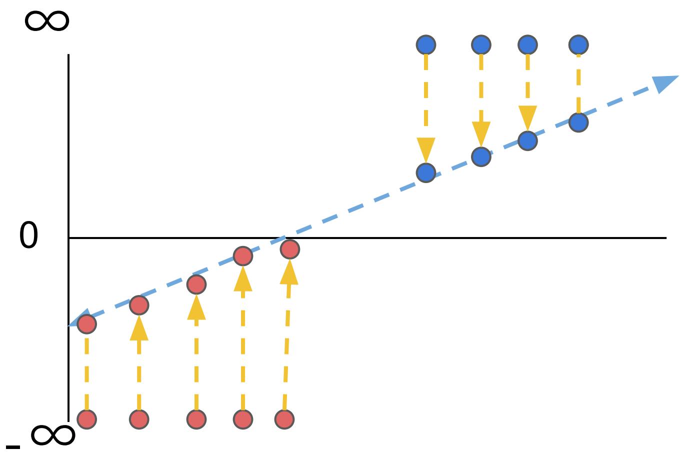


# Logistic Regression

- Choose best coefficient values in log odds terms that creates maximum likelihood.



# Logistic Regression

- While we are trying to **maximize** the likelihood, we still need something to **minimize**, since the computer's gradient descent methods can only search for minimums.

# Logistic Regression

- In terms of a cost function, we seek to minimize the following (log loss):

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m} y^j \log\left(\hat{y}^j\right) + (1-y^j)\log\left(1-\hat{y}^j\right)$$

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m}\left(y^j \log\left(\frac{1}{1+e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right) + (1-y^j)\log\left(1-\frac{1}{1+e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right)\right)$$

# Logistic Regression

- Just as with Linear Regression, gradient descent can solve this for us!

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m} y^j \log\left(\hat{y}^j\right) + (1 - y^j)\log\left(1 - \hat{y}^j\right)$$

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m}\left(y^j \log\left(\frac{1}{1 + e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right) + (1 - y^j)\log\left(1 - \frac{1}{1 + e^{-\sum_{i=0}^{n}\beta_i x_i^j}}\right)\right)$$

# Logistic Regression

- Don’t worry about fully understanding this gradient descent.
- In practice we never have to implement it ourselves.
- Main takeaway should be the relationship between log odds and probability.

# Logistic Regression

- Now that we have an intuition of what happens “behind the scenes”, let’s explore Logistic Regression with Python!

# Classification Performance Metrics

Part One: Confusion Matrix Basics

# Classification Metrics

- You’ve probably heard of terms such as “false positive” or “false negative”. As well as metrics like “accuracy”.
- But what do these terms actually mean mathematically?

# Classification Metrics

- Imagine we’ve developed a test or model to detect presence of a virus infection in a person based on some biological feature.
- We could treat this as a Logistic Regression, predicting:
  - 0 - Not Infected (Tests Negative)
  - 1 - Infected (Tests Positive)

# Classification Metrics

- It is unlikely our model will perform perfectly. This means there 4 possible outcomes:
    - Infected person tests positive.
    - Healthy person tests negative.

# Classification Metrics

- It is unlikely our model will perform perfectly. This means there 4 possible outcomes:
    - Infected person tests positive.
    - Healthy person tests negative.
        - *Note, these are the outcomes we want! But it is unlikely our test is perfect...*

# Classification Metrics

- It is unlikely our model will perform perfectly. This means there 4 possible outcomes:
    - Infected person tests positive.
    - Healthy person tests negative.
    - Infected person tests negative.
    - Healthy person tests positive.

# Classification Metrics

- Based off these 4 possibilities, there are many error metrics we can calculate.
- First, let's start by visualizing these four possibilities as a matrix.

# Classification Metrics

- Confusion Matrix

| | ACTUAL | |
|---|---|---|
| | INFECTED | HEALTHY |
| | | |
| | | |

# Classification Metrics

- Confusion Matrix

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | | |
| | HEALTHY | | |

# Classification Metrics

- Confusion Matrix

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | TRUE POSITIVE | |
| | HEALTHY | | |

# Classification Metrics

- Confusion Matrix

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | TRUE POSITIVE | |
| | HEALTHY | | TRUE NEGATIVE |

# Classification Metrics

- Confusion Matrix

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | TRUE POSITIVE | FALSE POSITIVE |
| | HEALTHY | | TRUE NEGATIVE |

# Classification Metrics

- Confusion Matrix

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | TRUE POSITIVE | FALSE POSITIVE |
| | HEALTHY | FALSE NEGATIVE | TRUE NEGATIVE |

# Classification Metrics

- Imagine a test group of 100 people:

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | | |
| | HEALTHY | | |

# Classification Metrics

- 5 are infected. 95 are healthy.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | | |
| | HEALTHY | | |

# Classification Metrics

- We tested all of them with these results:

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

# Classification Metrics

- What is accuracy?

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

# Classification Metrics

- What is accuracy?

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

- Accuracy:
  - How often is the model correct?

Acc = (TP+TN)/Total

# Classification Metrics

- Calculating accuracy:

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

(4+93)/100 = 97% Accuracy

- Accuracy:
  - How often is the model correct?

Acc = (TP+TN)/Total

# Classification Metrics

- Is this a good value for accuracy?

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

(4+93)/100 = 97% Accuracy

- Accuracy:
  - How often is the model correct?

Acc = (TP+TN)/Total

# Classification Metrics

- The accuracy paradox...

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

(4+93)/100 = 97% Accuracy

- Accuracy:
  - How often is the model correct?

Acc = (TP+TN)/Total

# Classification Metrics

- Imagine we **always** report back “healthy”

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

# Classification Metrics

- Imagine we **always** report back "healthy"

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 0 | 0 |
| | HEALTHY | 5 | 95 |

# Classification Metrics

- Imagine we **always** report back "healthy"

ACTUAL

| | | INFECTED | HEALTHY |
|---|---|---|---|
| PREDICTED | INFECTED | 0 | 0 |
| | HEALTHY | 5 | 95 |

(0+95)/100 = 95% Accuracy

- Accuracy:
  - How often is the model correct?

95% accuracy for a model that always returns "healthy"!

# Classification Metrics

- You may be thinking, "*The numbers here are arbitrary, we just happen to get good accuracy in this made up case. Real world data would reflect poor accuracy if a model always returned the same result*".

# Classification Metrics

- This is the accuracy paradox!
  - Any classifier dealing with **imbalanced** classes has to confront the issue of the accuracy paradox.
  - **Imbalanced** classes will always result in a distorted accuracy reflecting better performance than what is truly warranted.

# Classification Metrics

- **Imbalanced** classes are often found in real world data sets.
  - Medical conditions can affect small portions of the population.
  - Fraud is not common (e.g. Real vs. Fraud credit card usage).

# Classification Metrics

- If a class is only a small percentage (**n%**), then a classifier that always predicts the majority class will always have an accuracy of (1-n).
- In our previous example we saw infected were only 5% of the data.
- Allowing the accuracy to be 95%.

# Classification Metrics

- This means we shouldn’t solely rely on accuracy as a metric!
- This is where precision,recall, and f1-score will come in.
- Let’s explore these other metrics in the next lecture.

# Classification Performance Metrics

Part Two: Precision and Recall

# Classification Metrics

- We already know how to calculate accuracy and its associated paradox.
- Let’s explore three more metrics that can help give a clearer picture of performance:
  - Recall (a.k.a. sensitivity)
  - Precision
  - F1-Score

# Classification Metrics

- Let's begin with recall.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

- Recall:
  - When it actually is a positive case, how often is it correct?

(TP)/Total Actual Positives

# Classification Metrics

- Let’s begin with recall.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Recall =
(TP)/Total Actual Positives

- Recall:
  - When it actually is a positive case, how often is it correct?

(TP)/Total Actual Positives

# Classification Metrics

- Let's begin with recall.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Recall =
(TP)/5

- Recall:
  - When it actually is a positive case, how often is it correct?

(TP)/Total Actual Positives

# Classification Metrics

- Let's begin with recall.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Recall =
(4)/5

- Recall:
  - When it actually is a positive case, how often is it correct?

(TP)/Total Actual Positives

# Classification Metrics

- Let's begin with recall.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Recall = 0.8

- Recall:
  - How many relevant cases are found?

(TP)/Total Actual Positives

# Classification Metrics

- What’s the recall if we always classify as “healthy”?

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 0 | 0 |
| | HEALTHY | 5 | 95 |

Recall =
(TP)/Total Actual Positives

- Recall:
  - How many relevant cases are found?

(TP)/Total Actual Positives

# Classification Metrics

- What's the recall if we always classify as "healthy"?

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 0 | 0 |
| | HEALTHY | 5 | 95 |

Recall =
(0)/5 !

- Recall:
  - How many relevant cases are found?

(TP)/Total Actual Positives

# Classification Metrics

- A recall of 0 alerts you the model isn't catching cases!

ACTUAL

| | | INFECTED | HEALTHY |
|---|---|---|---|
| PREDICTED | INFECTED | 0 | 0 |
| | HEALTHY | 5 | 95 |

Recall =
(0)/5 !

- Recall:
  - How many relevant cases are found?

(TP)/Total Actual Positives

# Classification Metrics

- Now let's explore **precision**.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Precision =
(TP)/Total Predicted Positives

- Precision:
  - When prediction is positive, how often is it correct?

(TP)/Total Predicted Positives

# Classification Metrics

- Now let’s explore **precision**.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Precision =
(TP)/Total Predicted Positives

- Precision:
  - When prediction is positive, how often is it correct?

(TP)/Total Predicted Positives

# Classification Metrics

- Now let's explore **precision**.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Precision =
(TP)/6

- Precision:
  - When prediction is positive, how often is it correct?

(TP)/Total Predicted Positives

# Classification Metrics

- Now let's explore **precision**.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Precision =
(TP)/6

- Precision:
  - When prediction is positive, how often is it correct?

(TP)/Total Predicted Positives

# Classification Metrics

- Now let's explore **precision**.

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Precision = (4)/6

- Precision:
  - When prediction is positive, how often is it correct?

(TP)/Total Predicted Positives

# Classification Metrics

- Now let's explore **precision**.

ACTUAL

| | | INFECTED | HEALTHY |
|---|---|---|---|
| PREDICTED | INFECTED | 4 | 2 |
| | HEALTHY | 1 | 93 |

Precision = 0.666

- Precision:
  - When prediction is positive, how often is it correct?

(TP)/Total Predicted Positives

# Classification Metrics

- What's the **precision** if we always classify as "healthy"?

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 0 | 0 |
| | HEALTHY | 5 | 95 |

Precision =
(TP)/Total Predicted Positives

- Precision:
  - When prediction is positive, how often is it correct?

(TP)/Total Predicted Positives

# Classification Metrics

- What's the **precision** if we always classify as "healthy"?

| | | ACTUAL | |
|---|---|---|---|
| | | INFECTED | HEALTHY |
| PREDICTED | INFECTED | 0 | 0 |
| | HEALTHY | 5 | 95 |

Precision = 0/0

- Precision:
  - When prediction is positive, how often is it correct?

(TP)/Total Predicted Positives

# Classification Metrics

- Recall and Precision can help illuminate our performance specifically in regards to the relevant or positive case.
- Depending on the model, there is typically a trade-off between precision and recall, which we will explore later on with the ROC curve.

# Classification Metrics

- Since precision and recall are related to each other through the numerator (TP), we often also report the F1-Score, which is the harmonic mean of precision and recall.

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

# Classification Metrics

- The harmonic mean (instead of the normal mean) allows the entire harmonic mean to go to zero if **either** precision or recall ends up being zero.

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

# Classification Metrics

- As a final note on the confusion matrix, there are **many** more metrics available:

| | | | | | | |
|---|---|---|---|---|---|---|
| | | **True condition** | | | | |
| | Total population | Condition positive | Condition negative | Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$ | Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$ | |
| **Predicted condition** | Predicted condition positive | **True positive** | **False positive**, Type I error | Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$ | False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$ | |
| | Predicted condition negative | **False negative**, Type II error | **True negative** | False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$ | Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$ | |
| | | True positive rate (TPR), Recall, Sensitivity, probability of detection, Power = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$ | False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$ | Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$ | Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$ | $F_1$ score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$ |
| | | False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$ | Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$ | Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$ | | |

# Classification Metrics

- Finally, let's explore a way to visualize the relationships between metrics such as precision and recall with curves.

# Classification Performance Metrics

Part Three: ROC Curves

# Classification Metrics

- During World War 2, Radar technology was developed to help detect incoming enemy aircraft.

# Classification Metrics

- The technology was so new, the US Army wanted to develop a methodology to evaluate radar operator performance.

# Classification Metrics

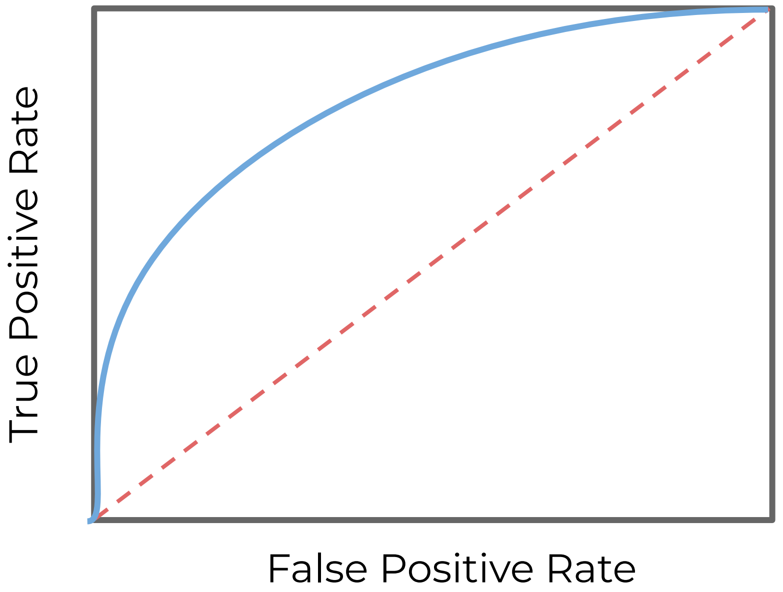
- They developed the Receiver Operator Characteristic curve.

# Classification Metrics

- They developed the Receiver Operator Characteristic curve.

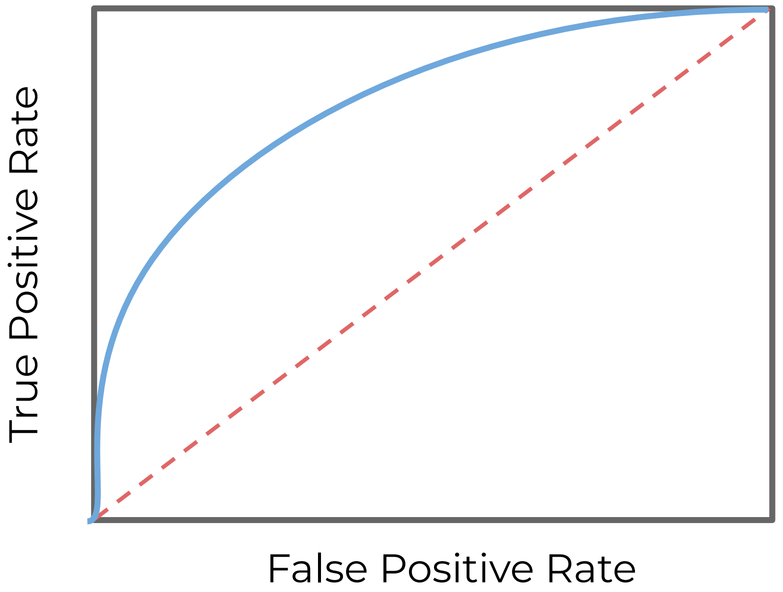
# Classification Metrics

- They developed the Receiver Operator Characteristic curve.



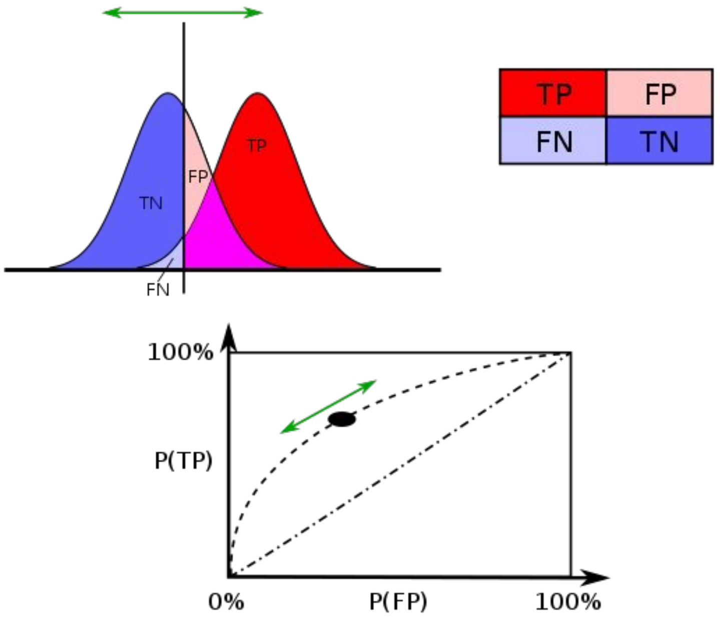
# Classification Metrics

- There can be a trade-off between True Positives and False Positives.



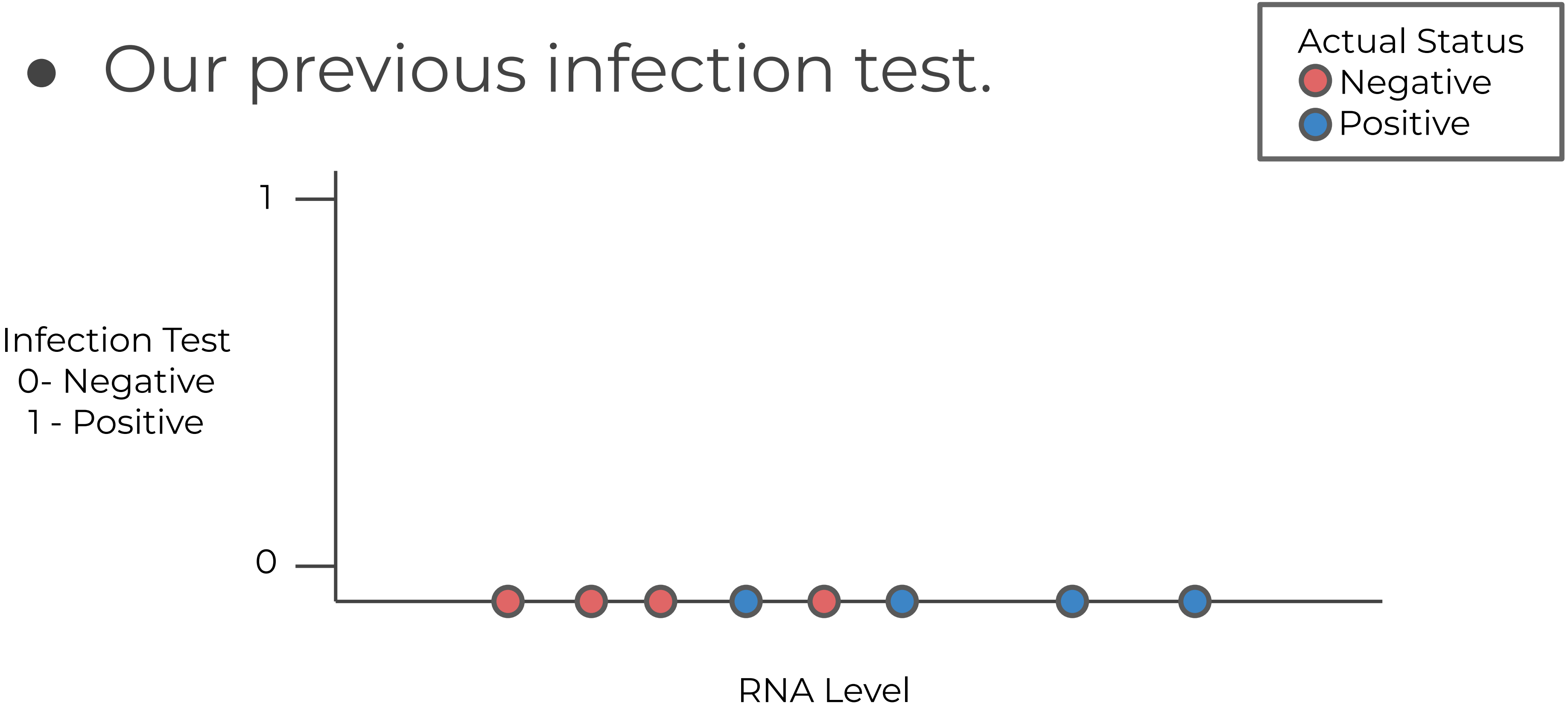
# Classification Metrics

- There can be a trade-off between True Positives and False Positives.



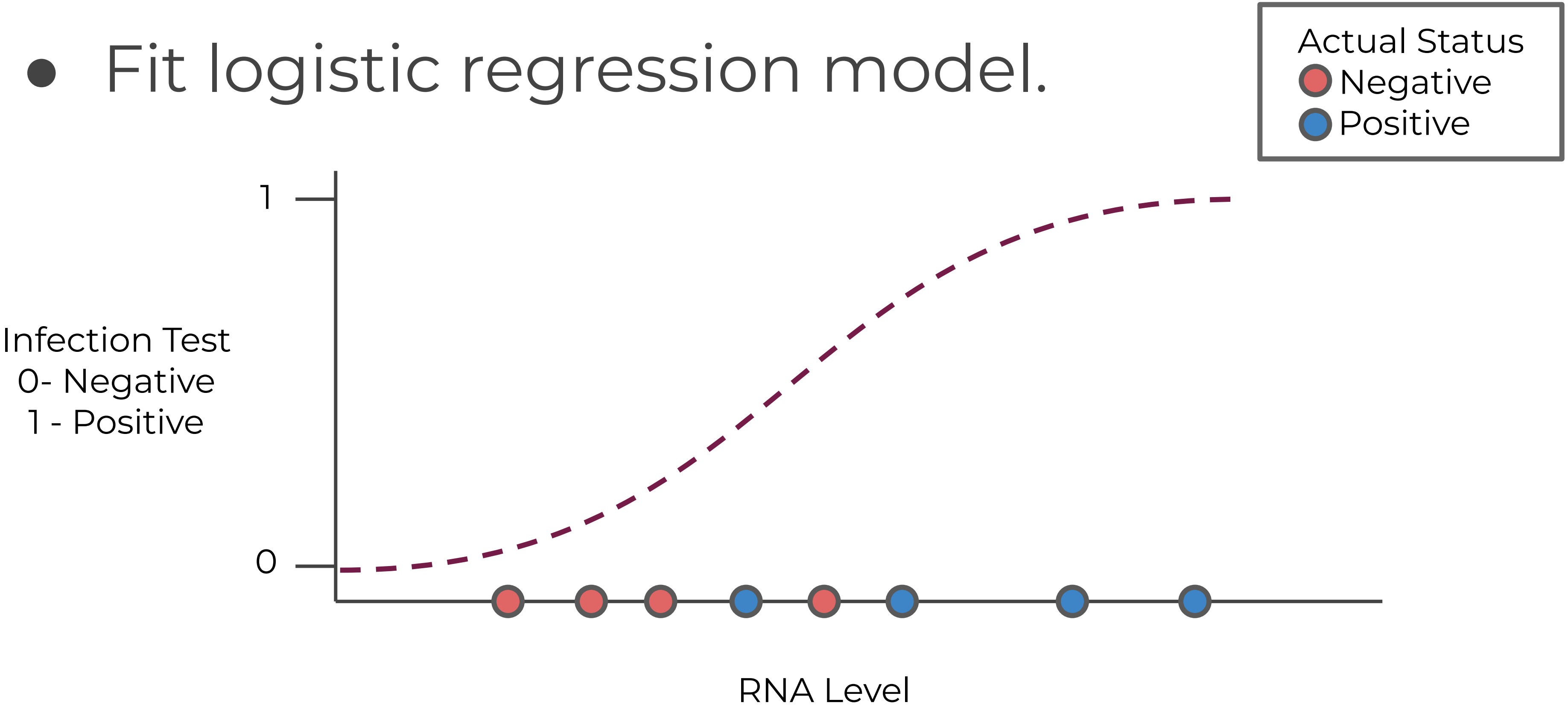
# Classification Metrics

- Our previous infection test.



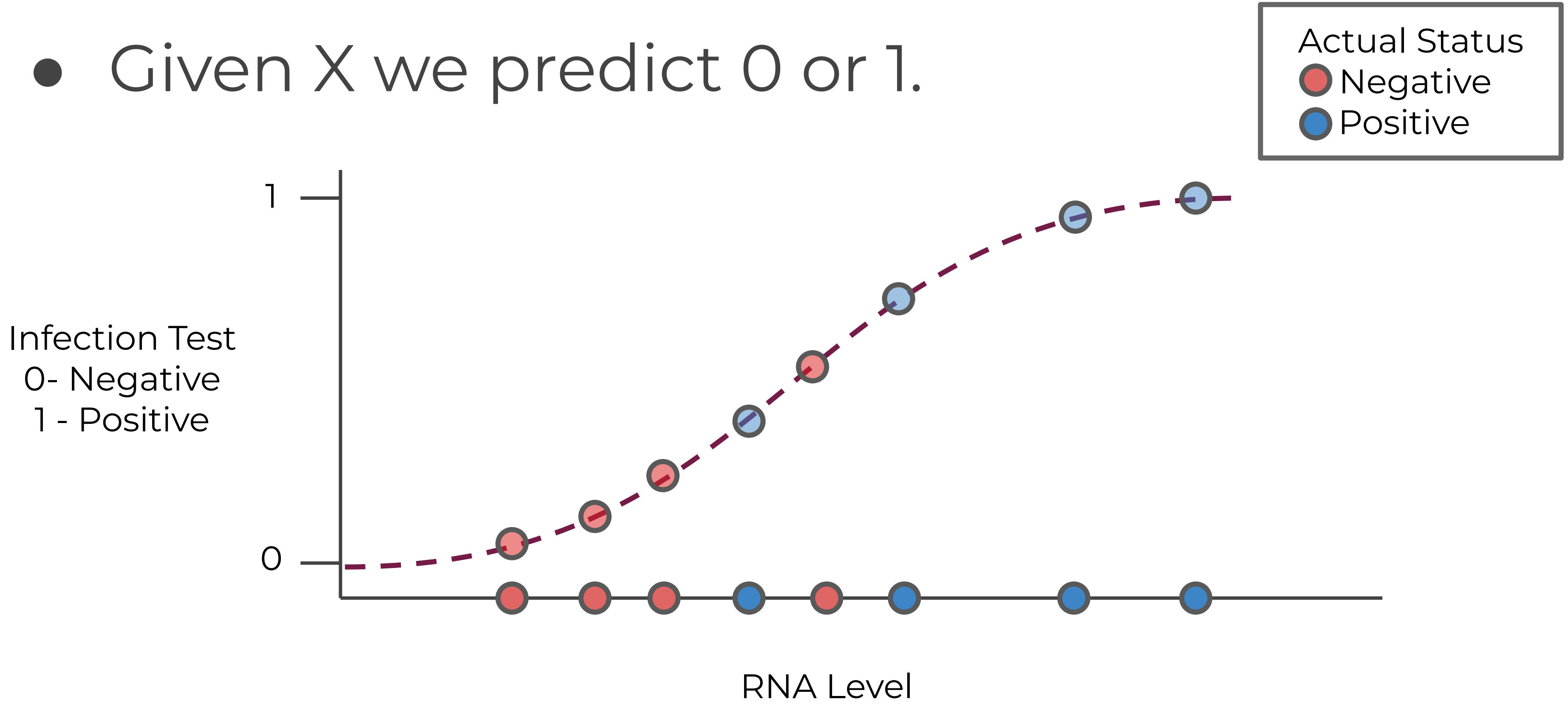
# Classification Metrics

- Fit logistic regression model.



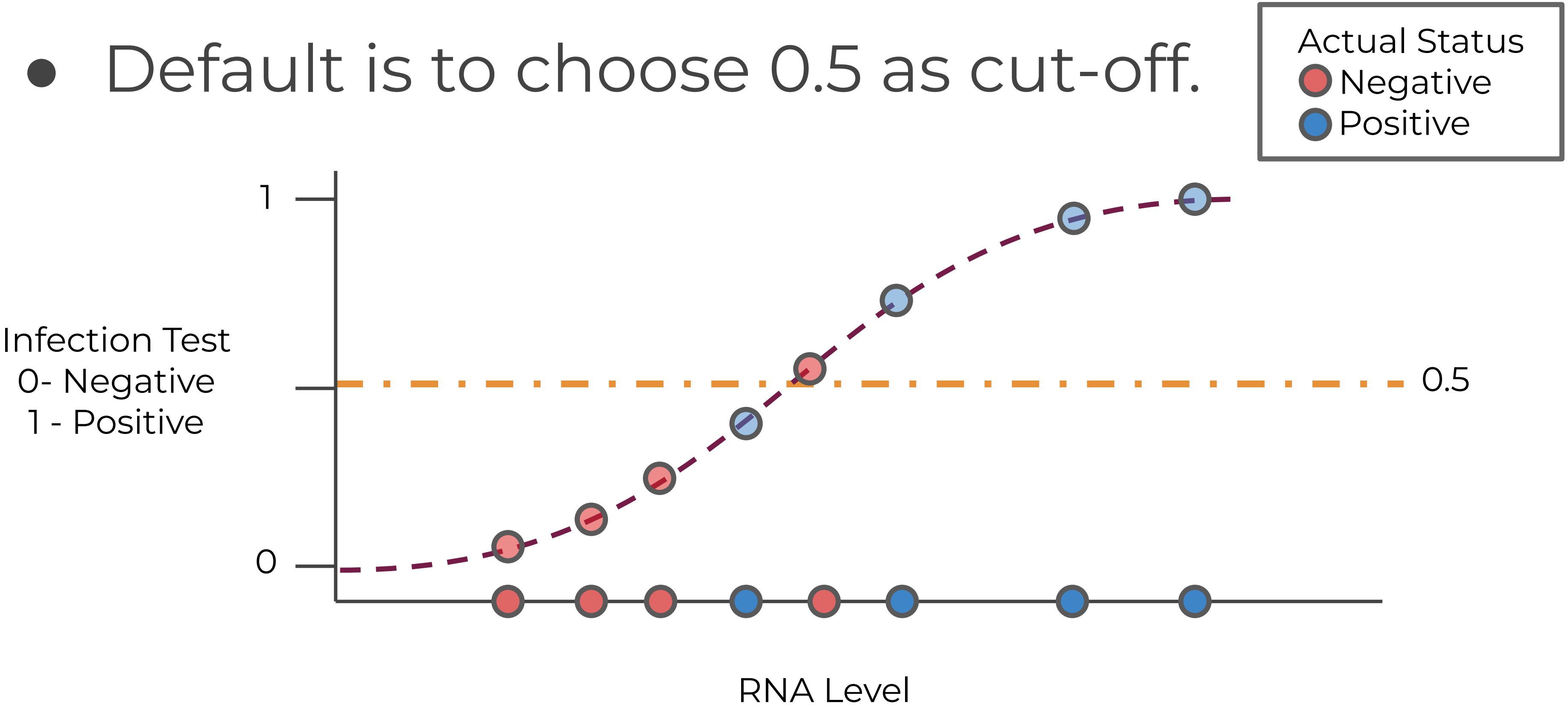
# Classification Metrics

- Given X we predict 0 or 1.



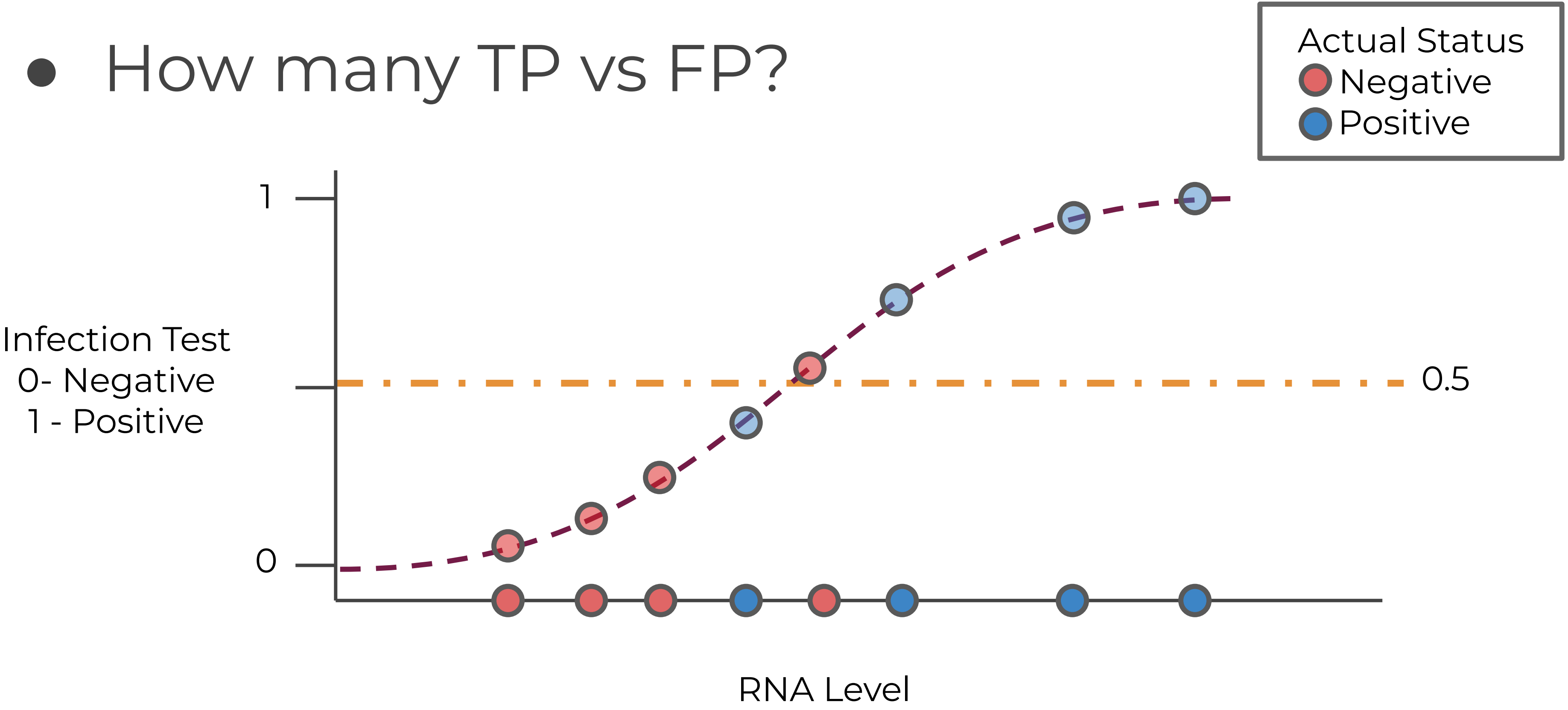
# Classification Metrics

- Default is to choose 0.5 as cut-off.



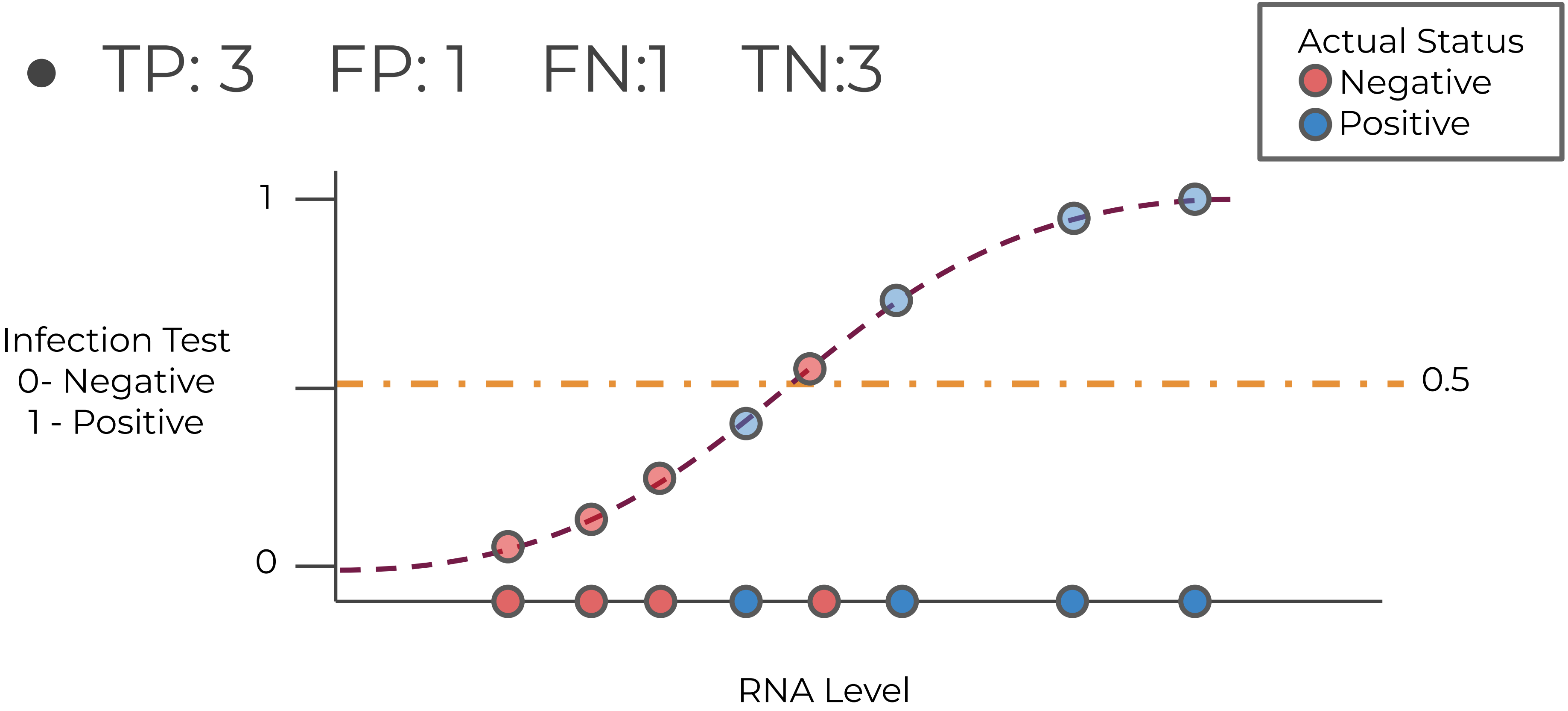
# Classification Metrics

- How many TP vs FP?



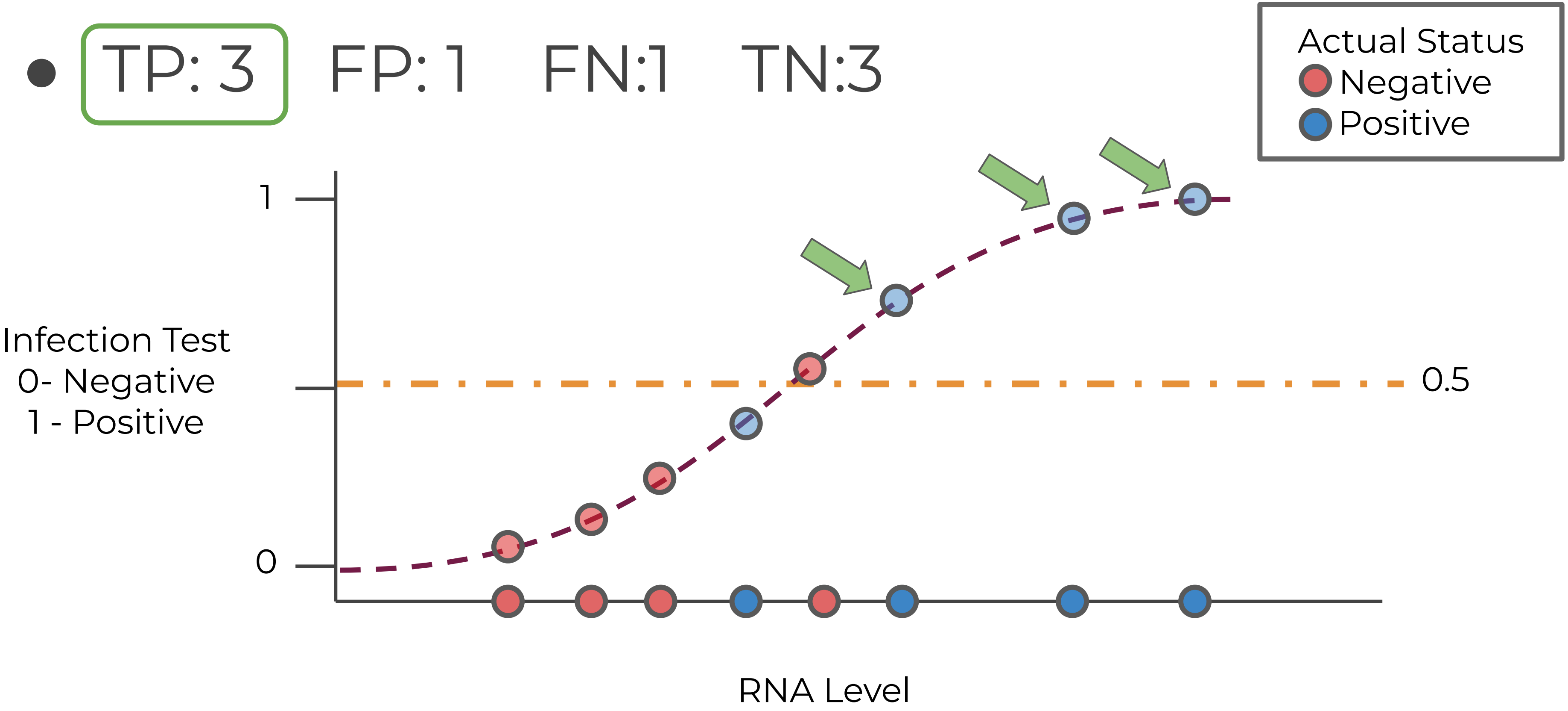
# Classification Metrics

- TP: 3 FP: 1 FN:1 TN:3



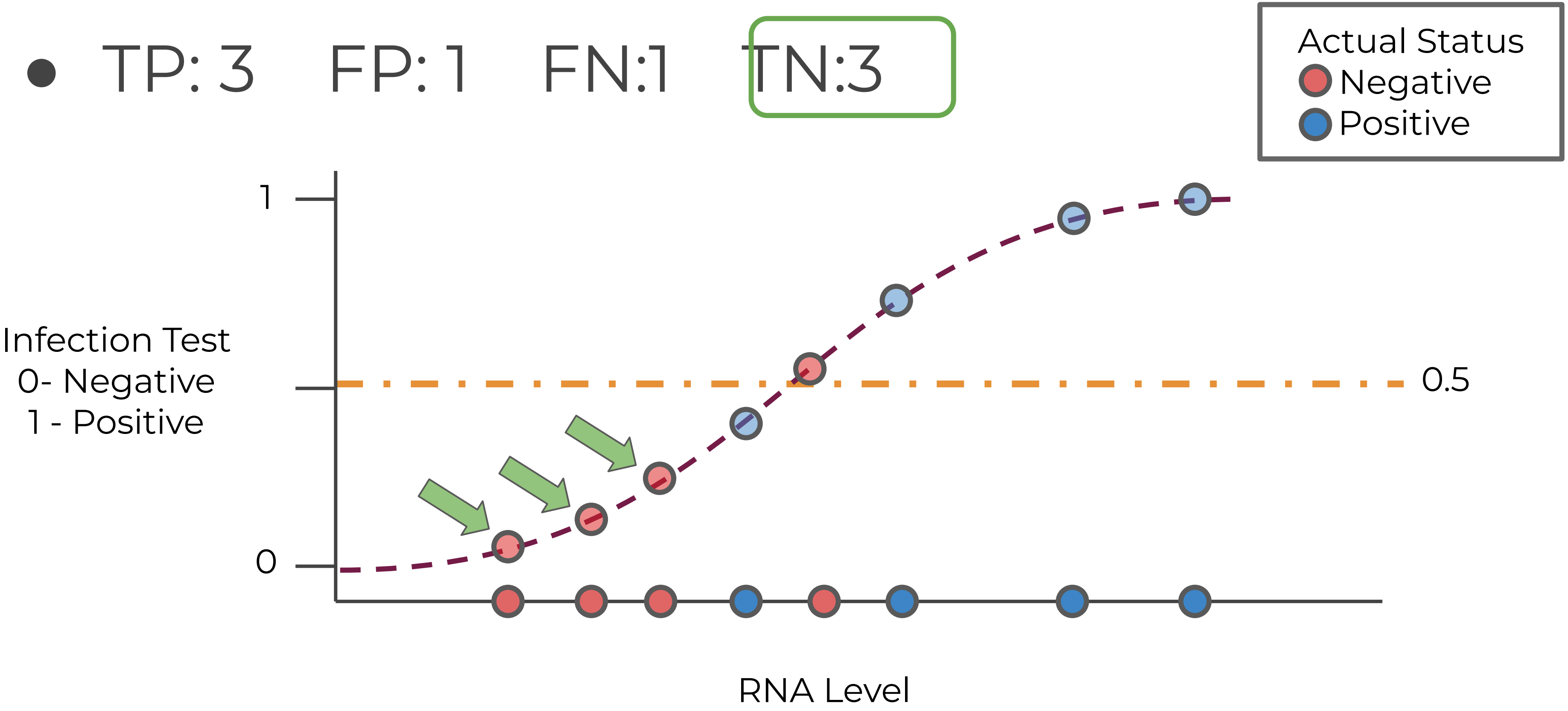
# Classification Metrics

- TP: 3 FP: 1 FN:1 TN:3



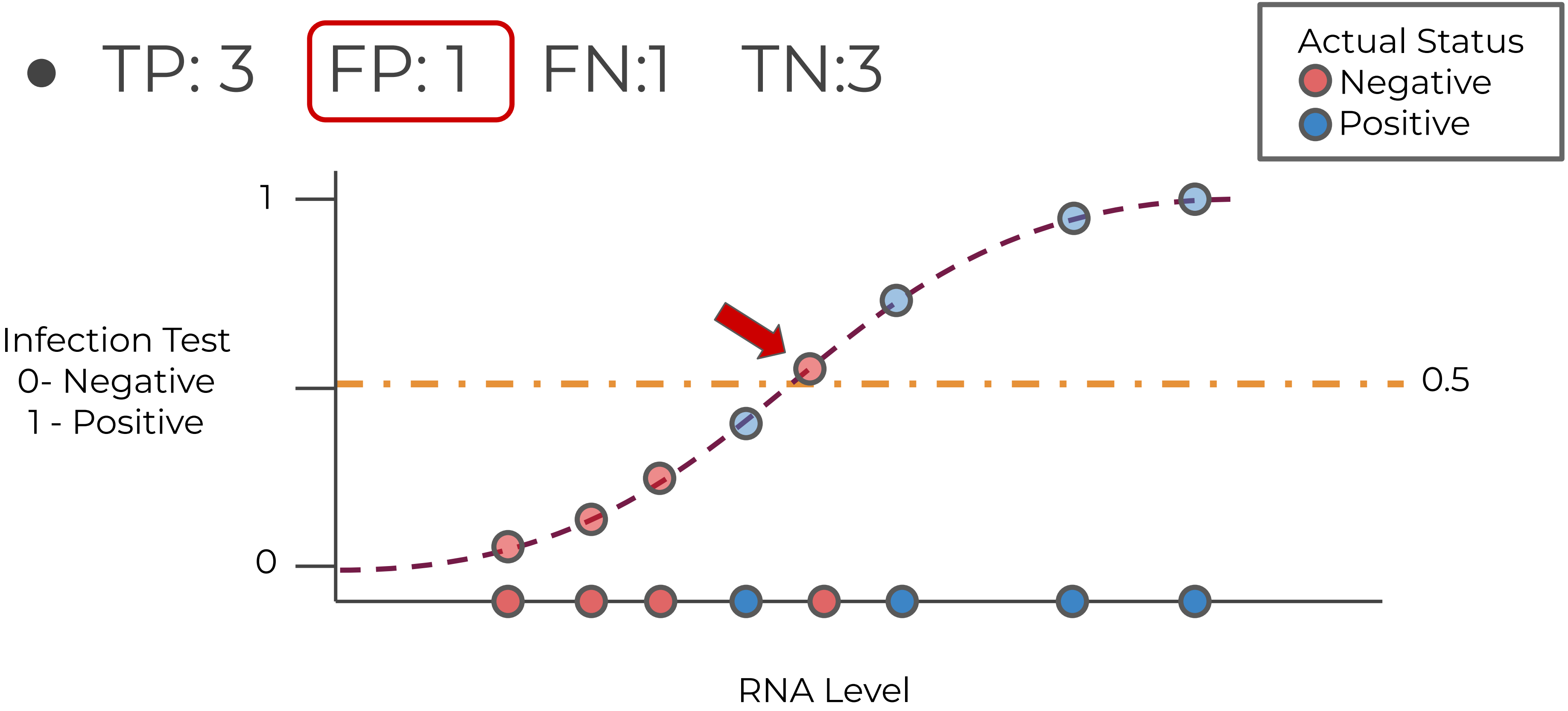
# Classification Metrics

- TP: 3 FP: 1 FN:1 TN:3



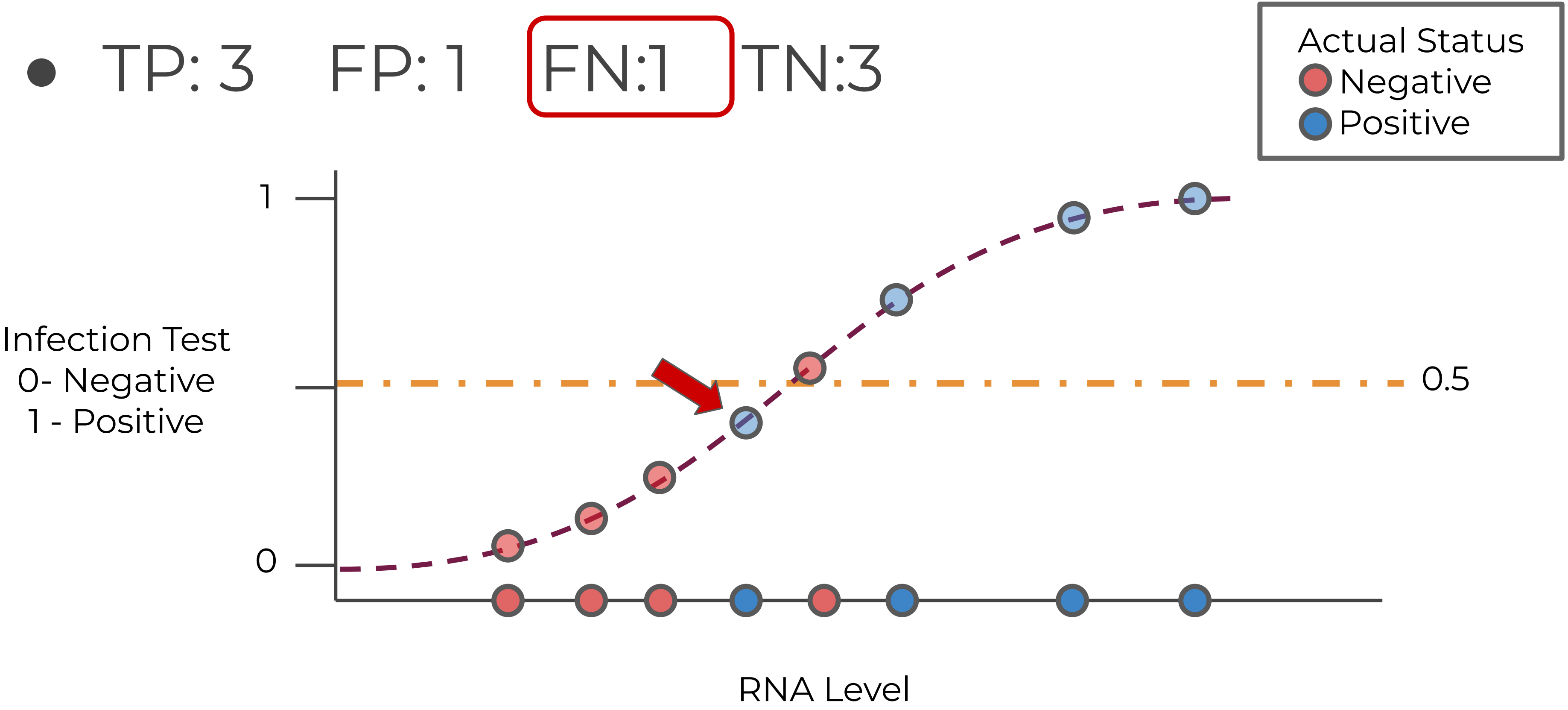
# Classification Metrics

- TP: 3 $\boxed{\text{FP: 1}}$ FN:1 TN:3



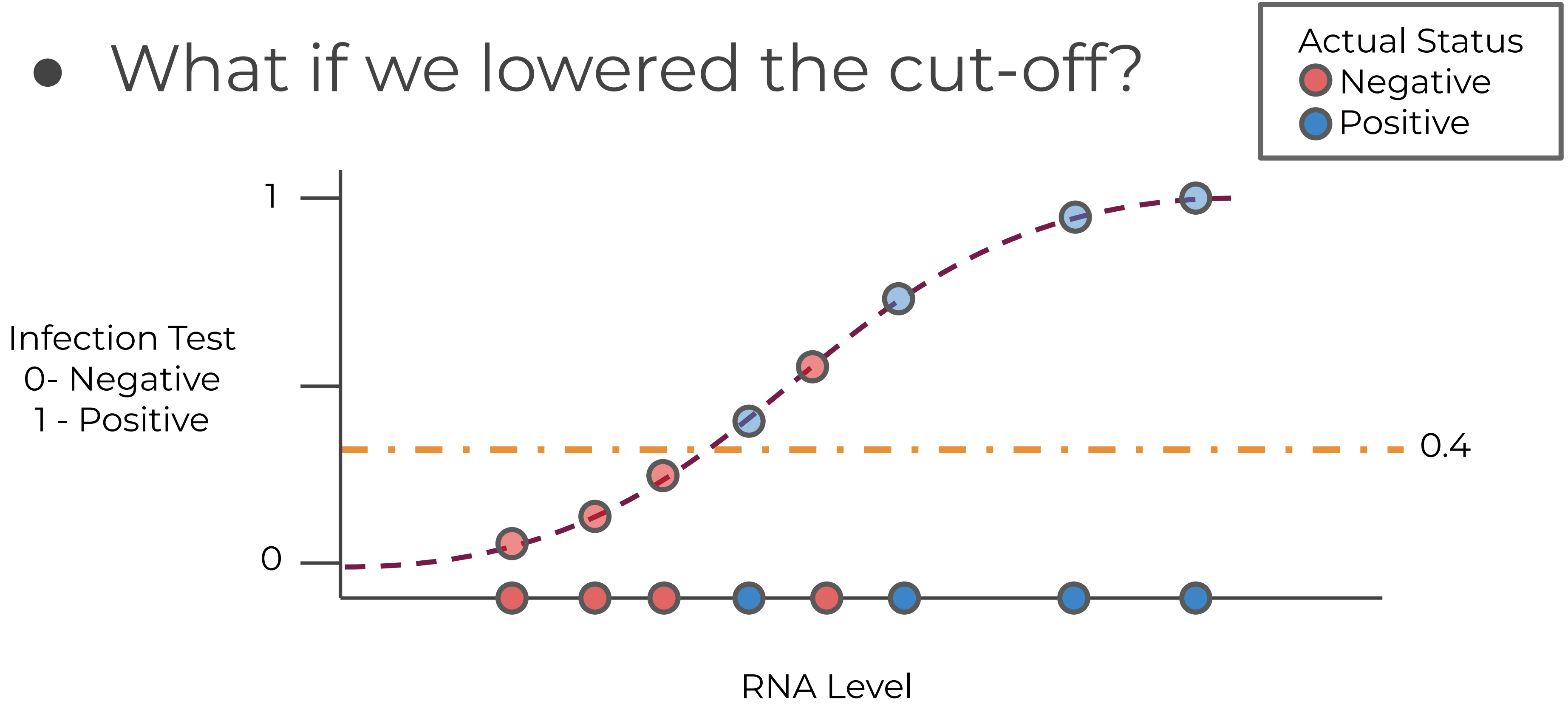
# Classification Metrics

- TP: 3 FP: 1 $\boxed{\text{FN:1}}$ TN:3



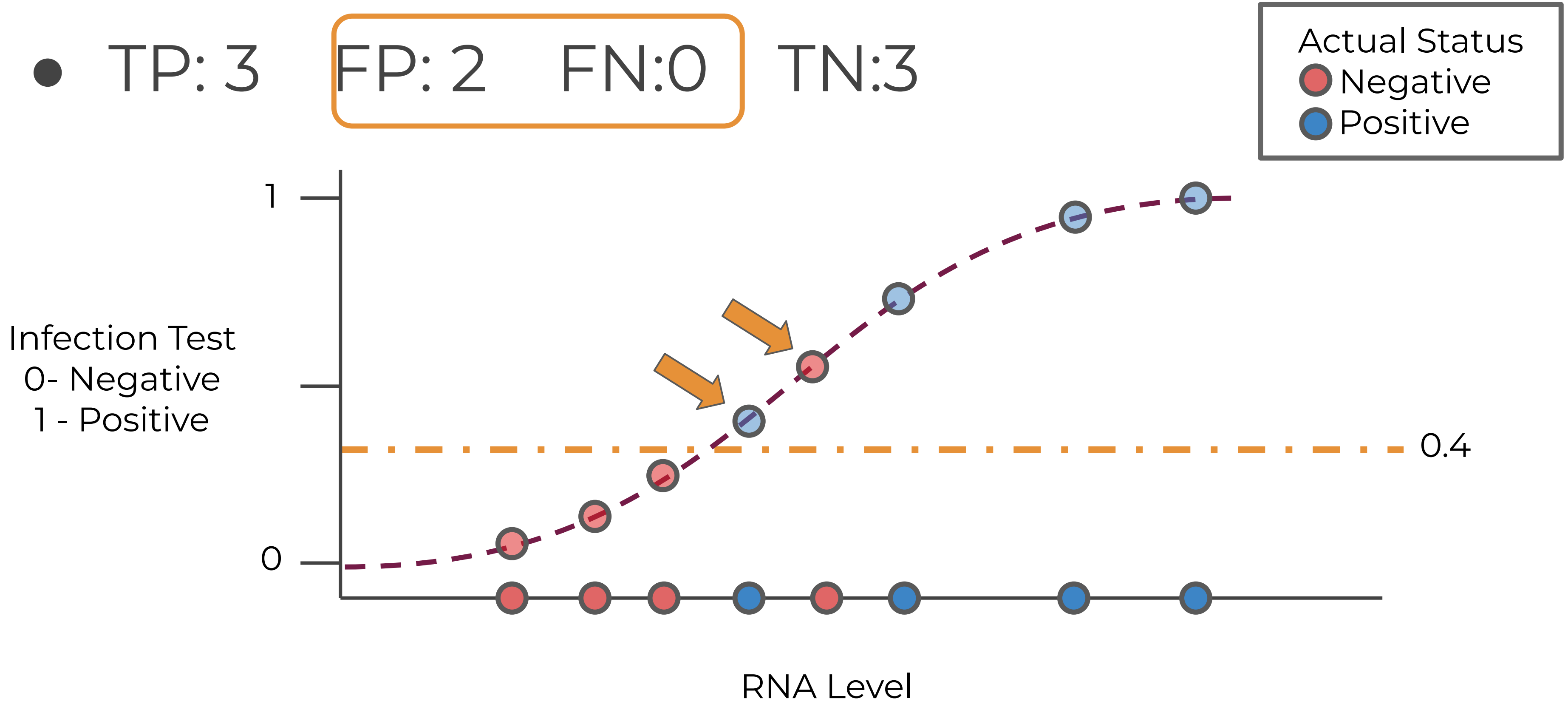
# Classification Metrics

- What if we lowered the cut-off?



# Classification Metrics

- TP: 3 FP: 2 FN:0 TN:3

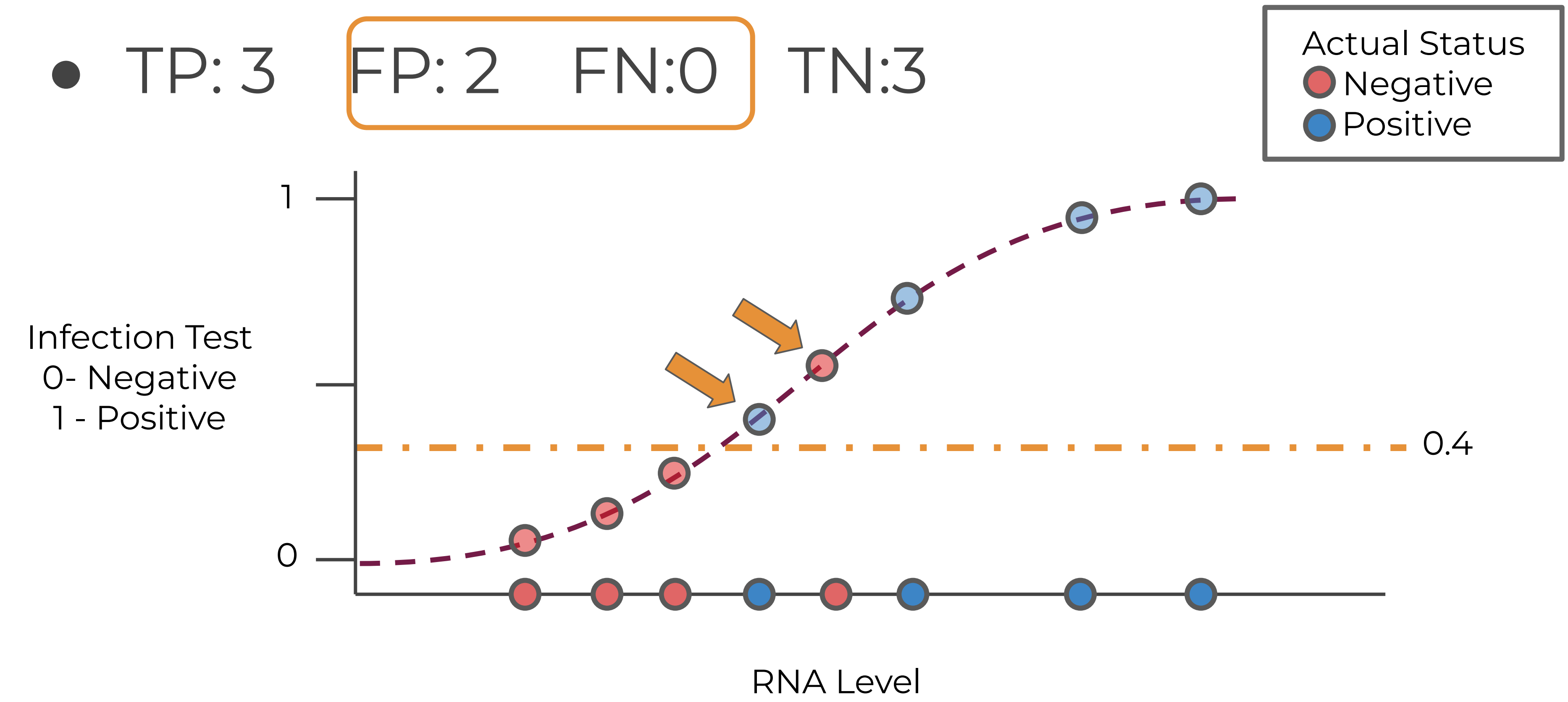


# Classification Metrics

- In certain situations, we gladly accept more false positives to reduce false negatives.
- Imagine a dangerous virus test, we would much rather produce false positives and later do more stringent examination than accidentally release a false negative!

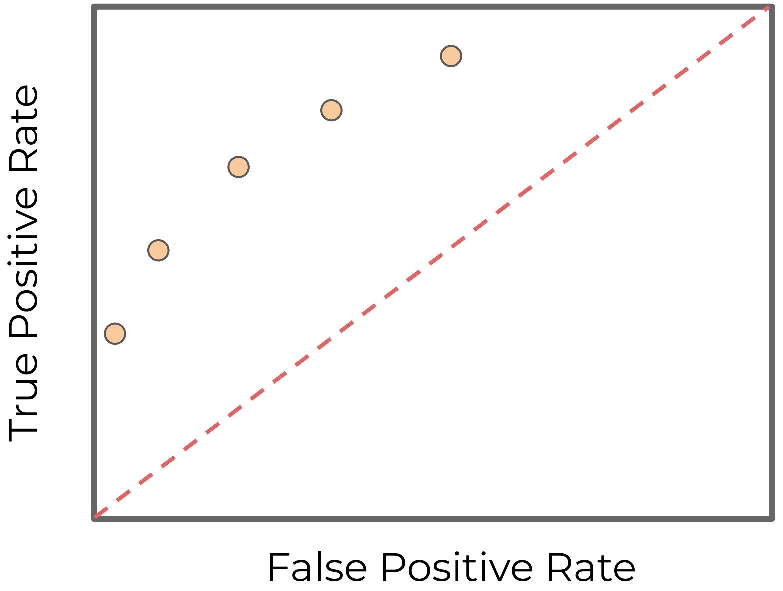
# Classification Metrics

- TP: 3 FP: 2 FN:0 TN:3



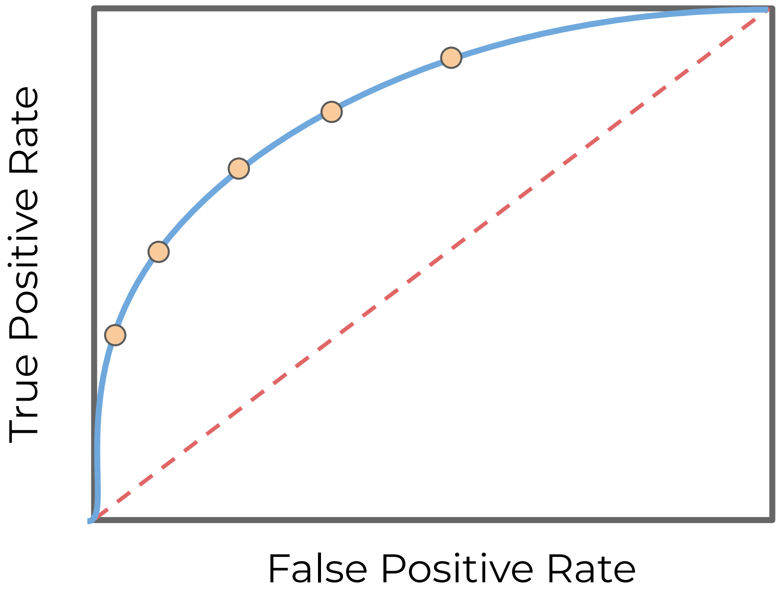
# Classification Metrics

- Chart the True vs. False positives for various cut-offs for the ROC curve.



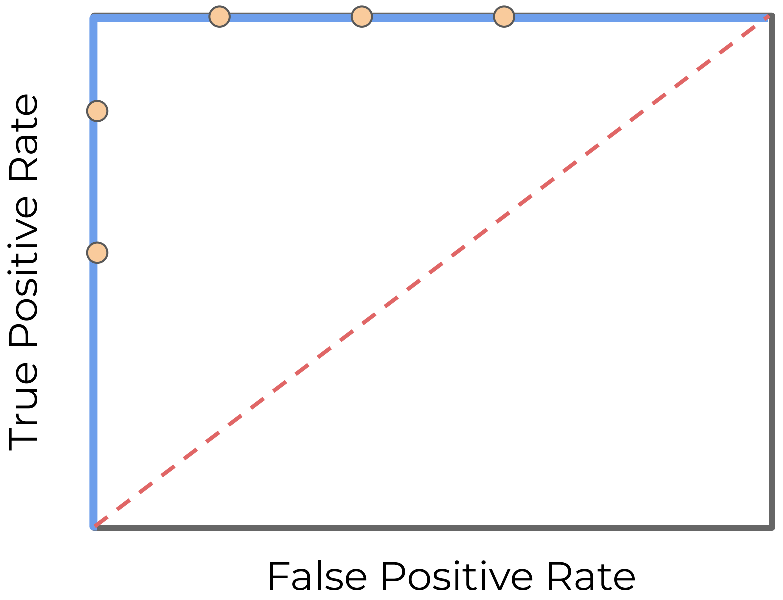
# Classification Metrics

- By changing the cut-off limit, we can adjust our True vs. False Positives!



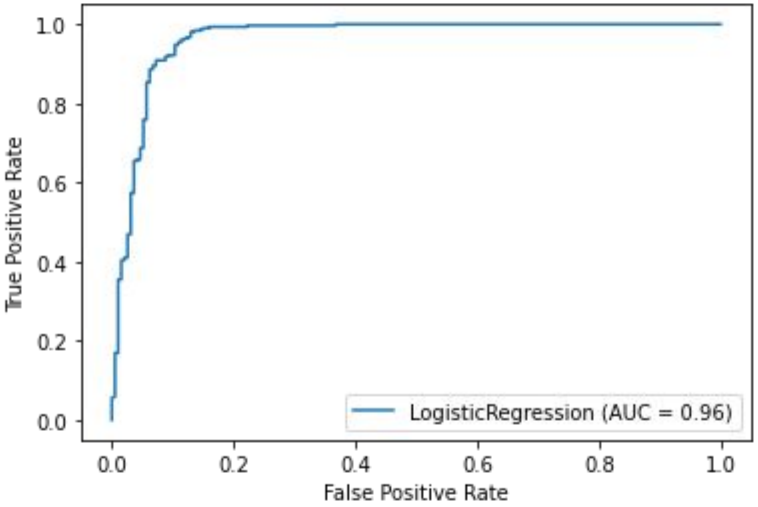
# Classification Metrics

- A perfect model would have a zero FPR.
- Random guessing is the red line.



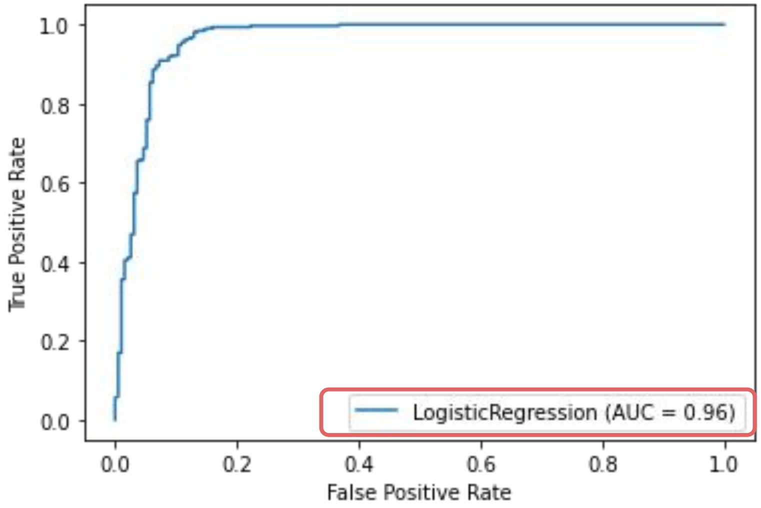
# Classification Metrics

- Realistically with smaller data sets the ROC curves are not as smooth.



# Classification Metrics

- AUC - Area Under the Curve , allows us to compare ROCs for different models.



# Classification Metrics

- Can also create precision vs. recall curves: