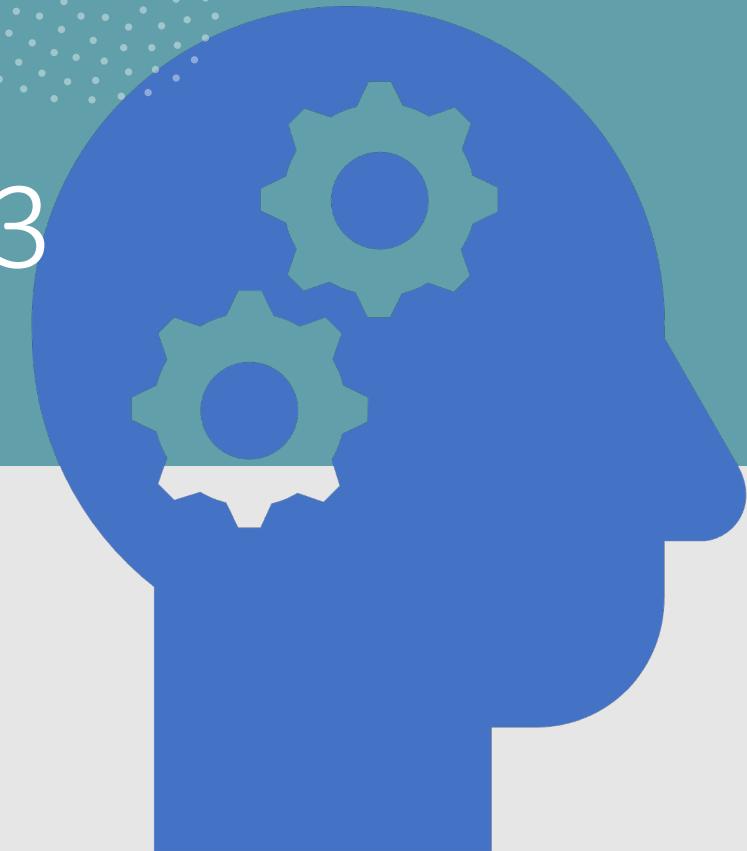


Machine Learning : 06048203

# Logistic Regression





# 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
  - Transforming Linear Regression to Logistic Regression
  - Mathematical Theory behind Logistic Regression
  - Simple Implementation of Logistic Regression for Classification Problem

# Logistic Regression

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

# Logistic Regression

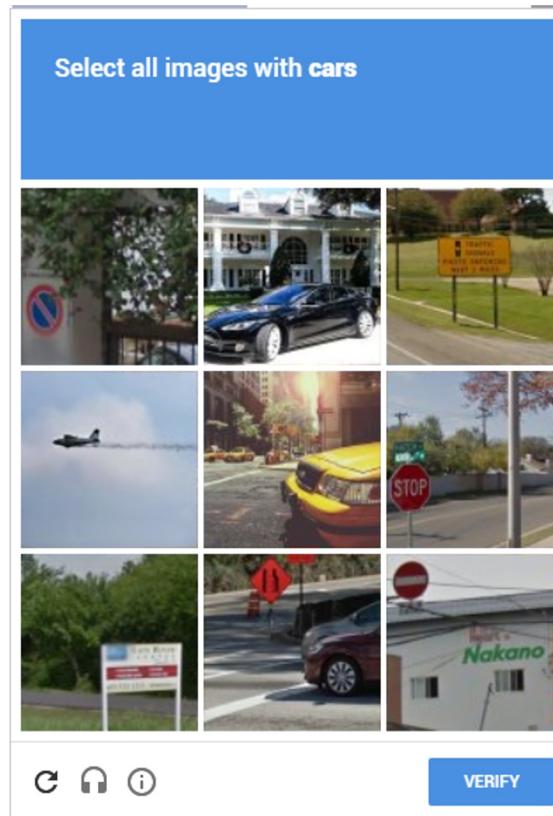
- Logistic Regression Section Overview
  - Multiclass Classification with Logistic Regression
  - Logistic Regression Project
  - Logistic Regression Project Solutions

# 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  $\hat{y}$  will be a category, meaning we won't be able to calculate a difference based on  $y - \hat{y}$ .
  - **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:

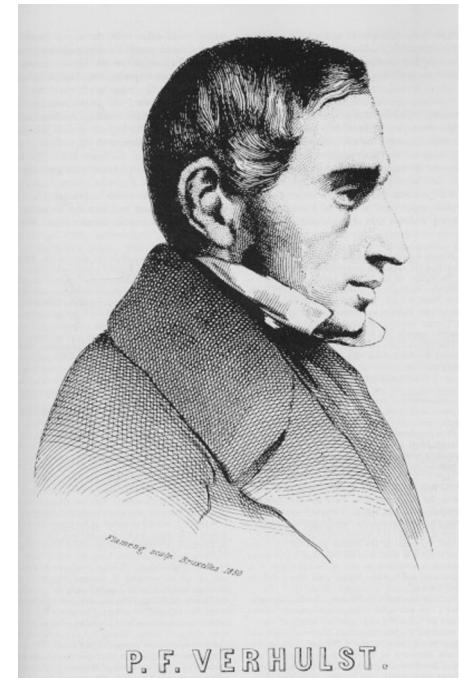
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

# Logistic Regression

- 1830-1850: Under guidance of Adolphe Quetelet, Pierre François Verhulst developed the logistic function:



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

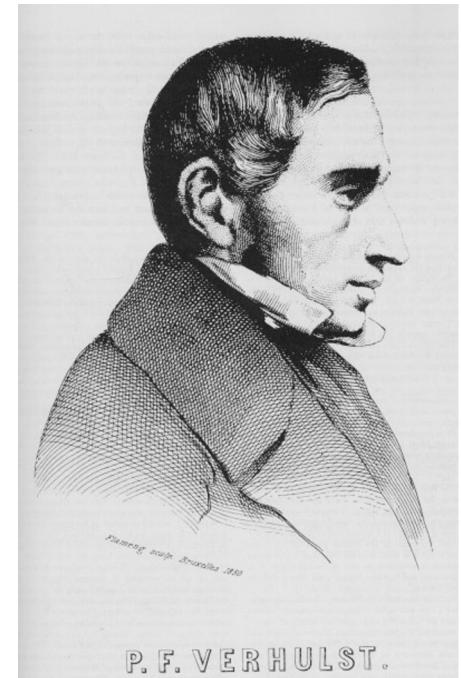


# Logistic Regression

- 1830-1850: Under guidance of Adolphe Quetelet, Pierre François Verhulst developed the logistic function:

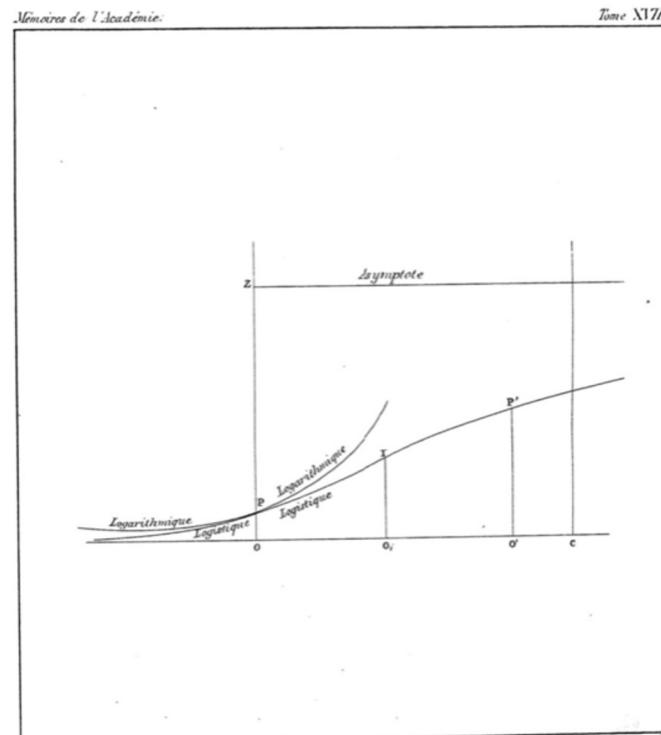
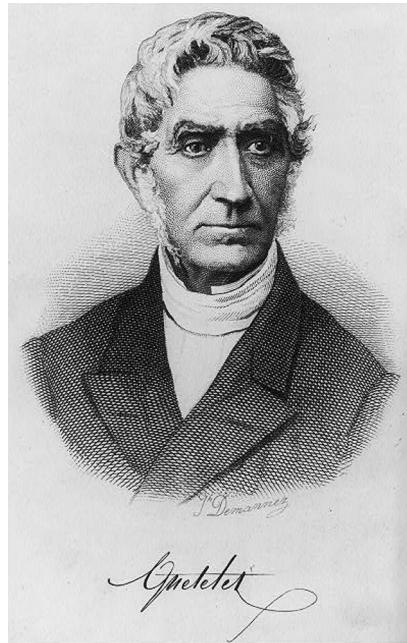


$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



# Logistic Regression

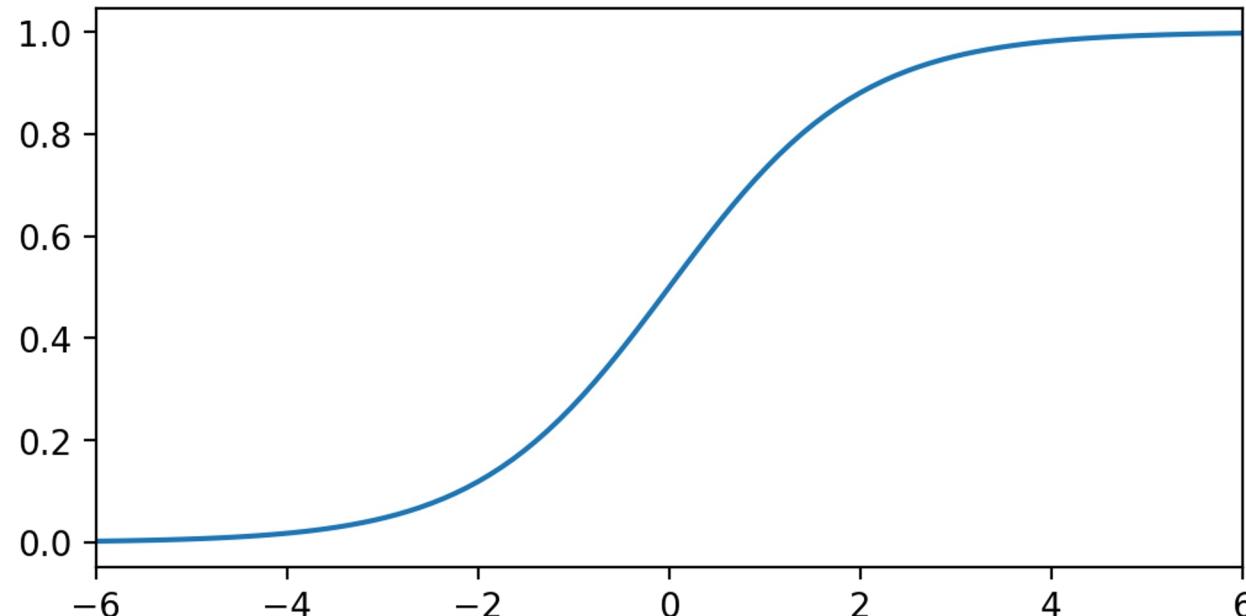
- 1830-1850: Developed for the purposes of modeling population growth.



P. F. VERHULST.

# Logistic Regression

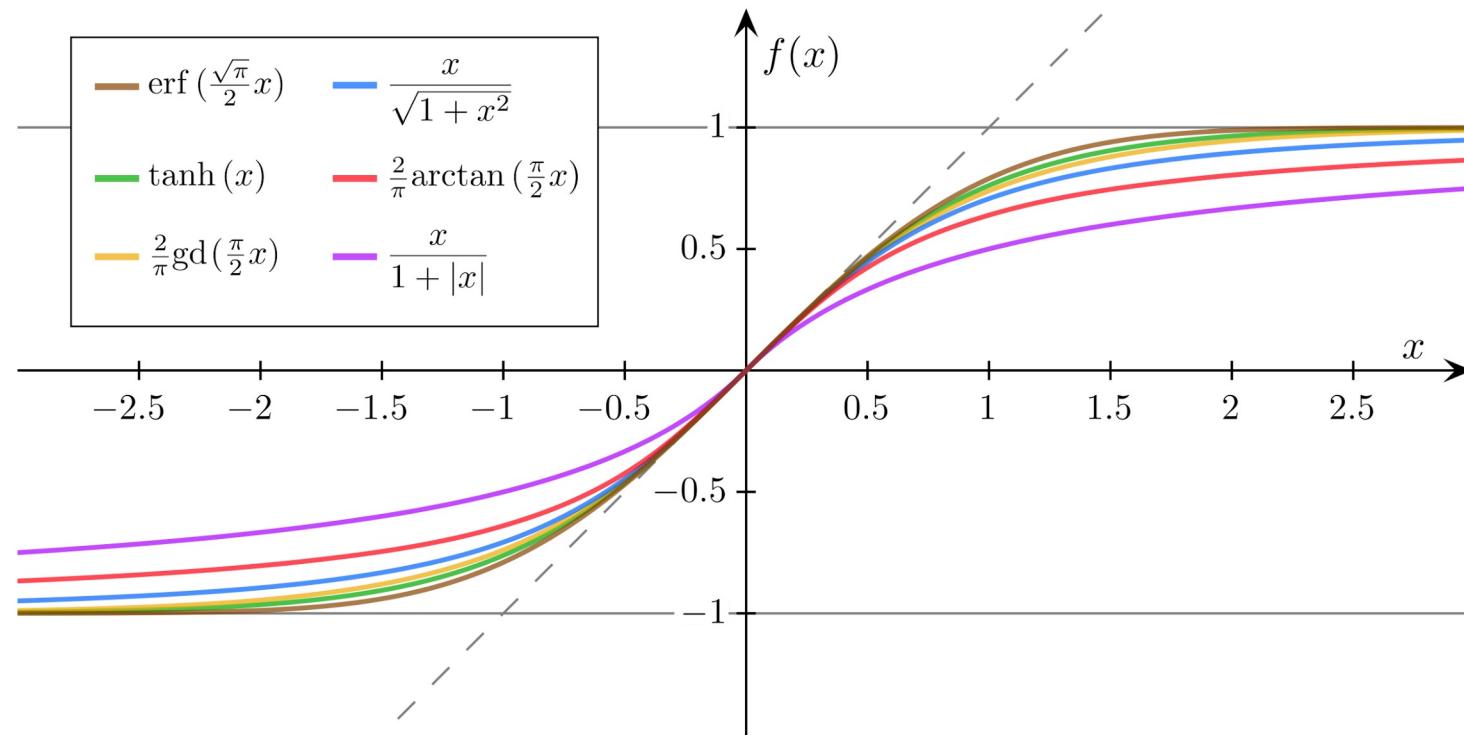
- Why the need for a logistic function versus a logarithmic function?



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

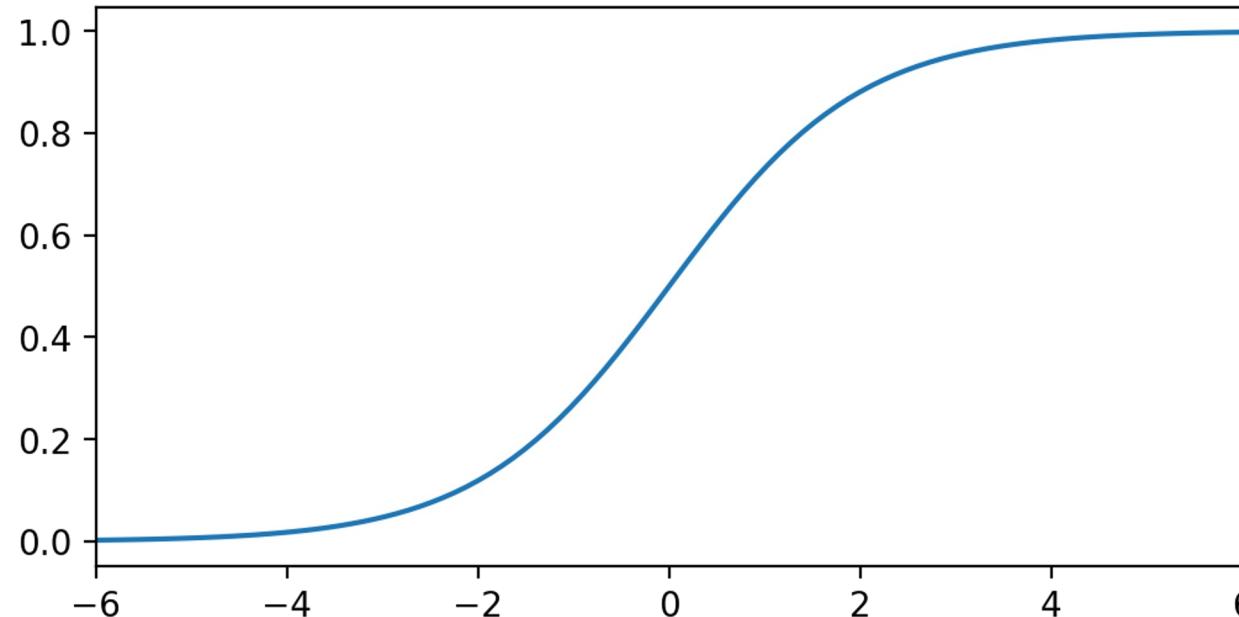
# Logistic Regression

- Note: There is a “family” of logistic functions.



# Logistic Regression

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



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

# **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)

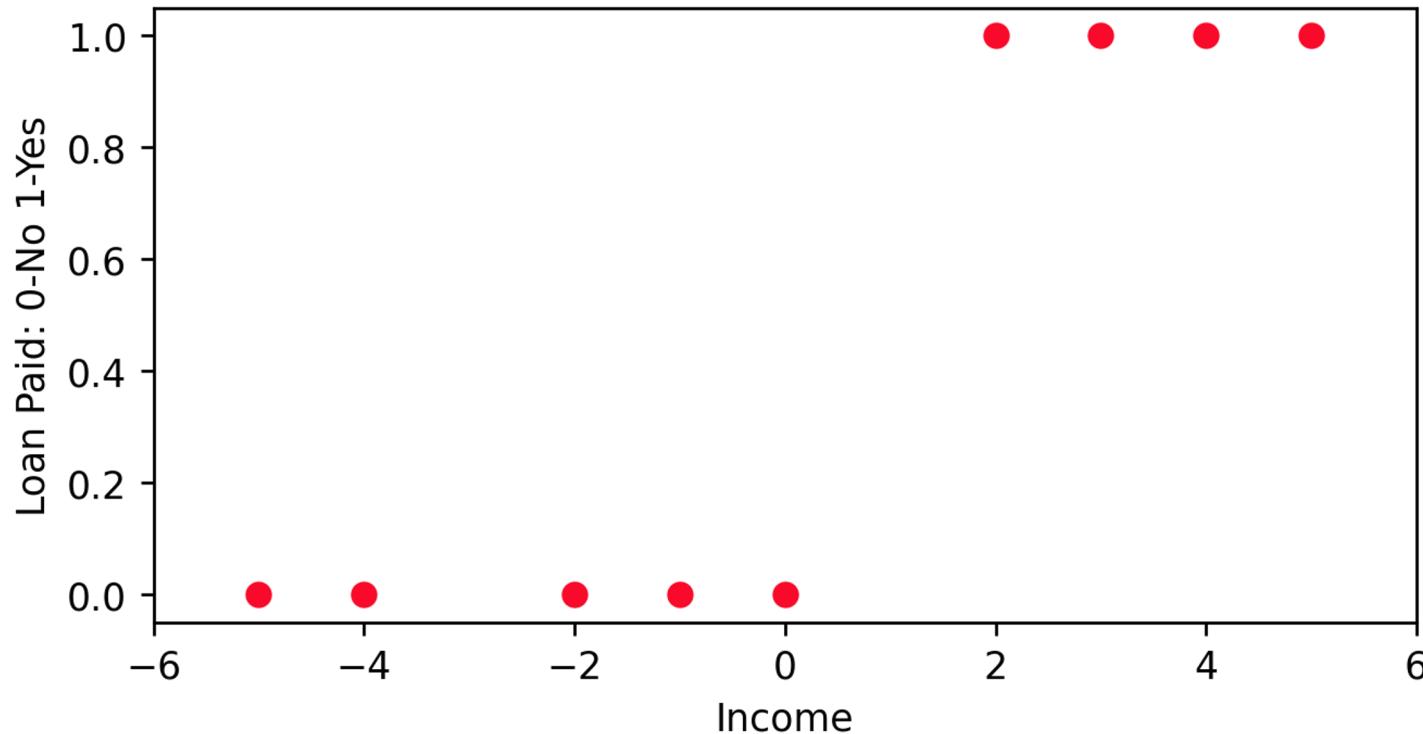
# Logistic Regression

- Our data set:

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

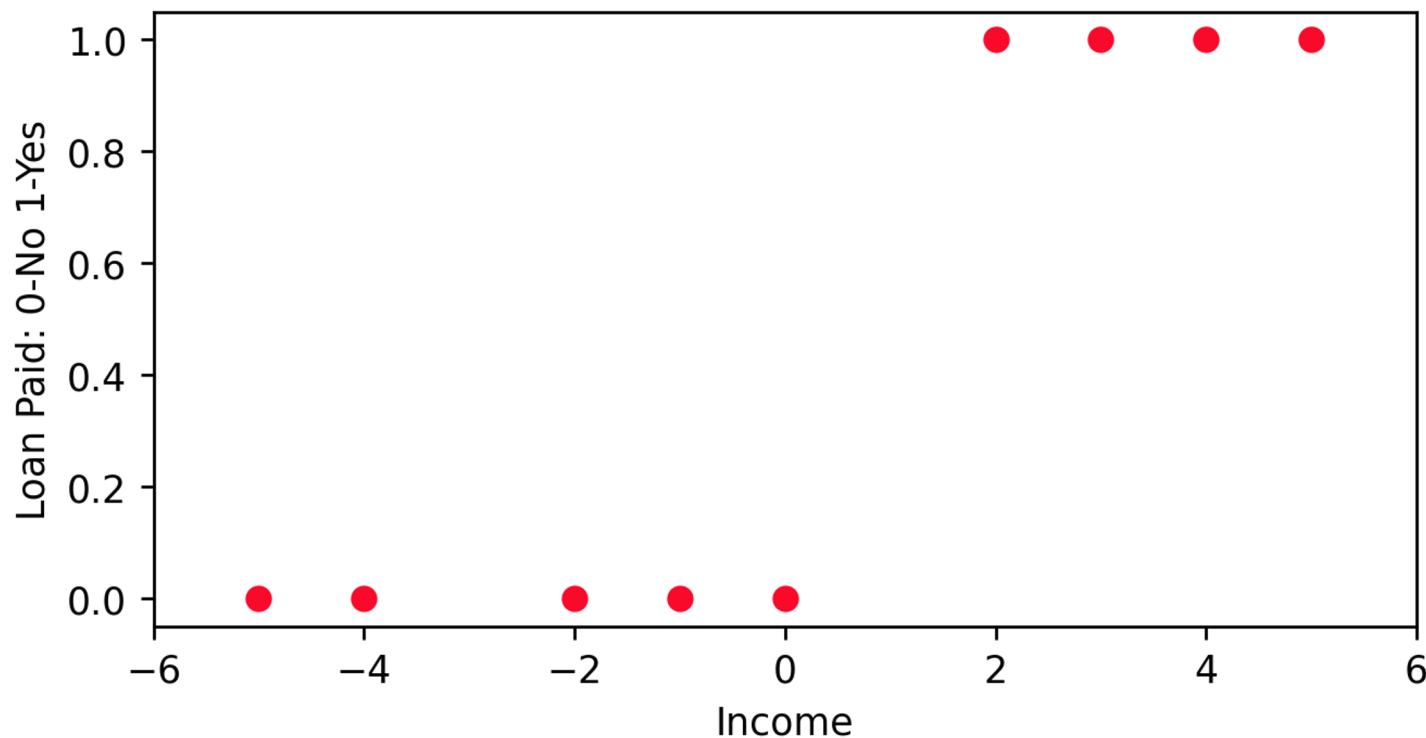
# Logistic Regression

- Let's begin by plotting income versus default:



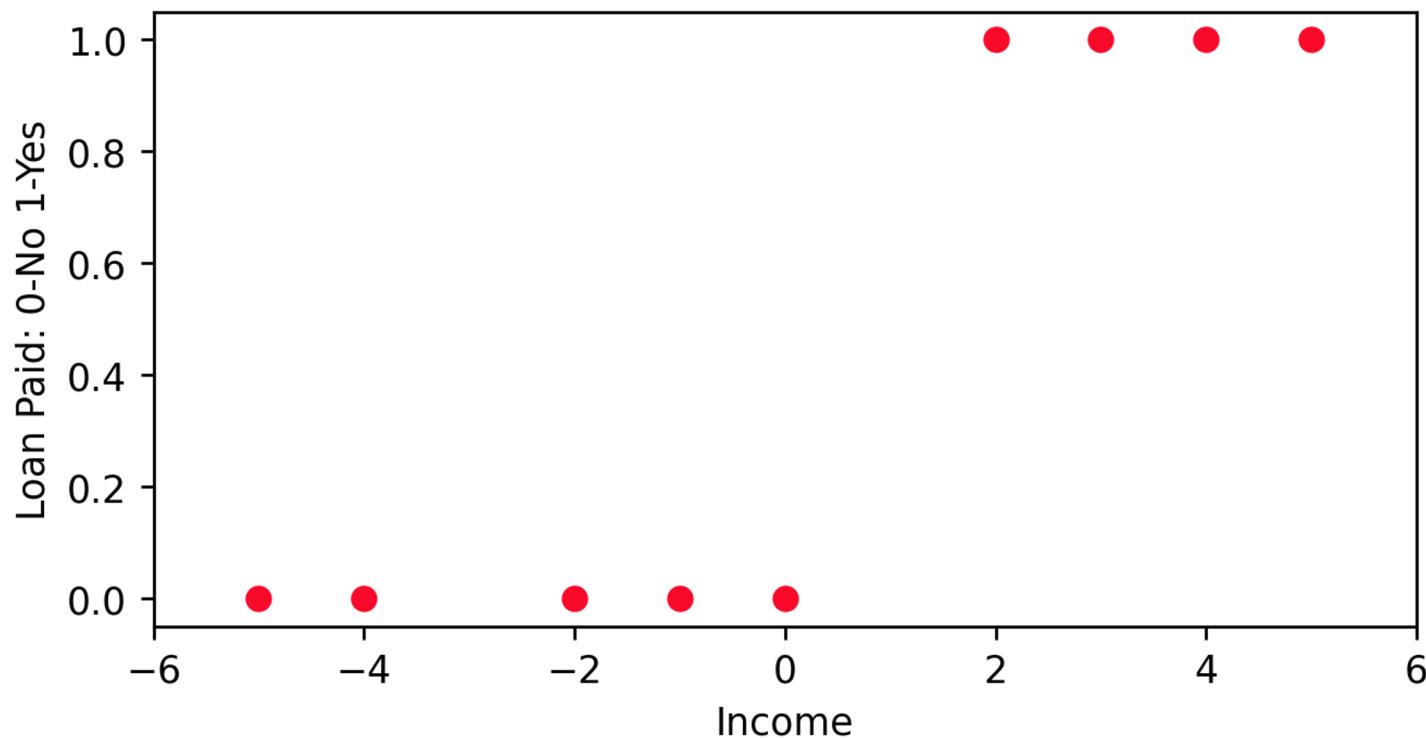
# Logistic Regression

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



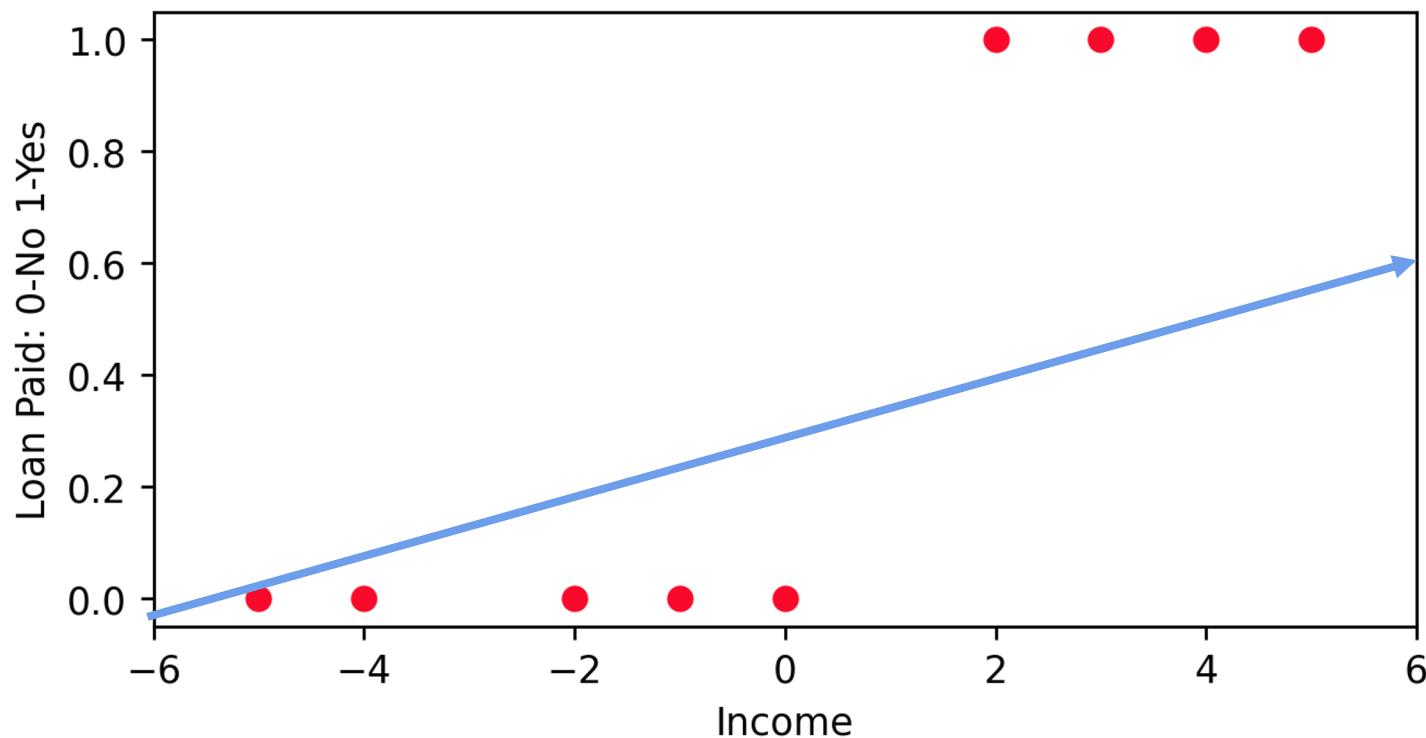
# Logistic Regression

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



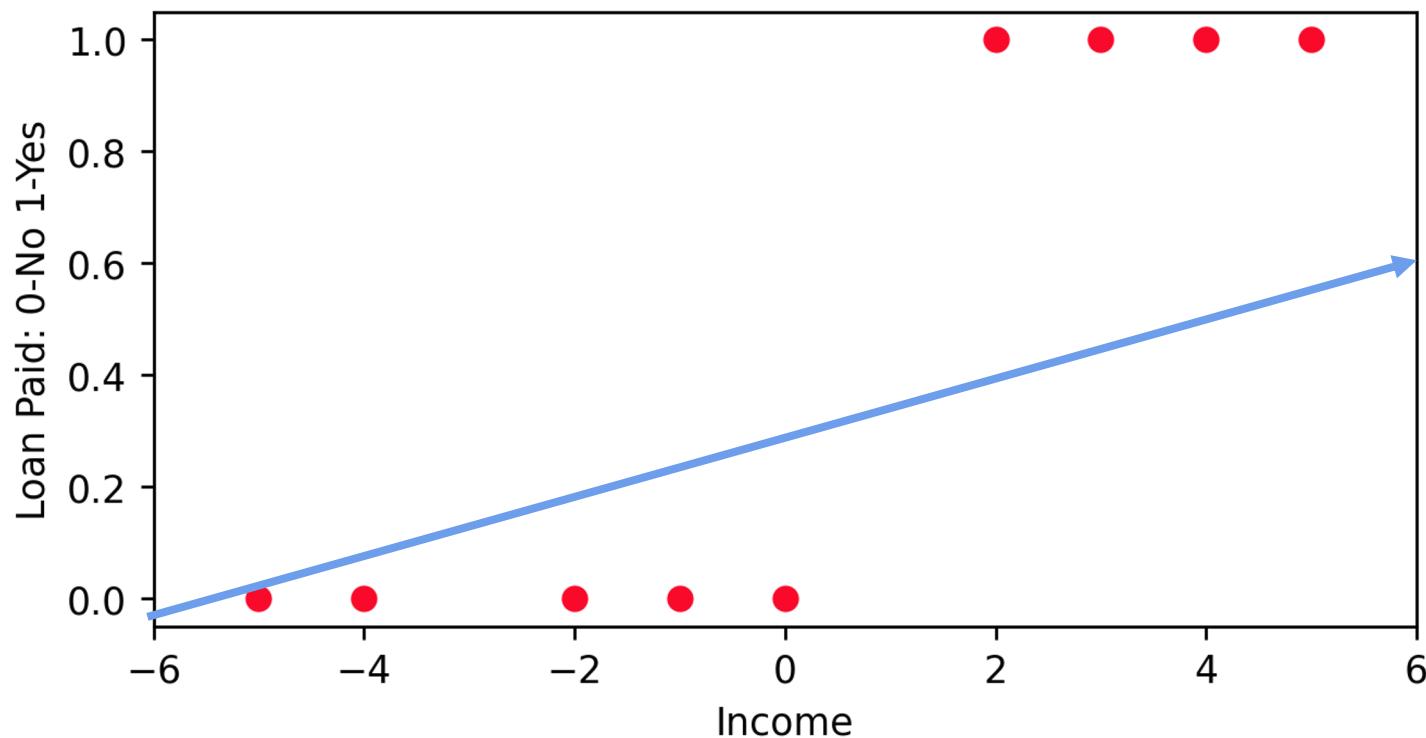
# Logistic Regression

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



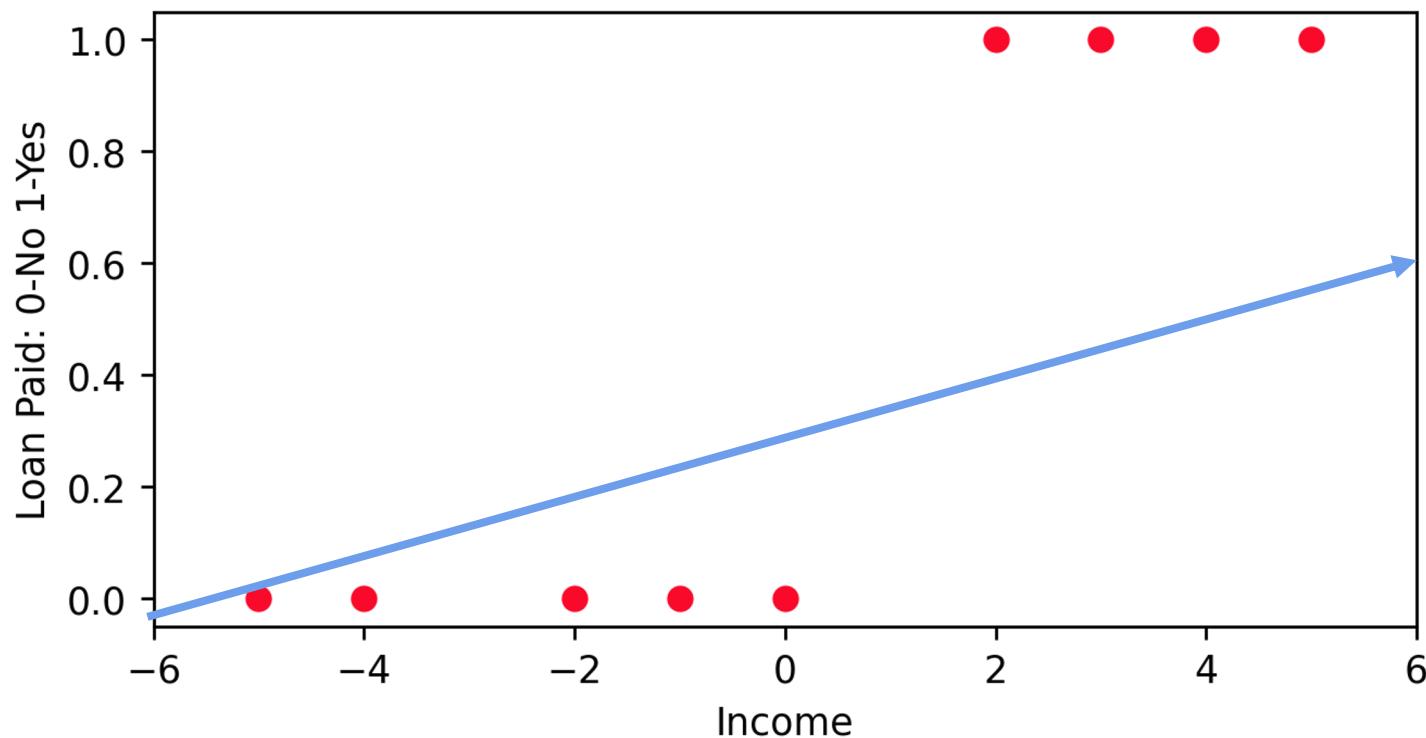
# Logistic Regression

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



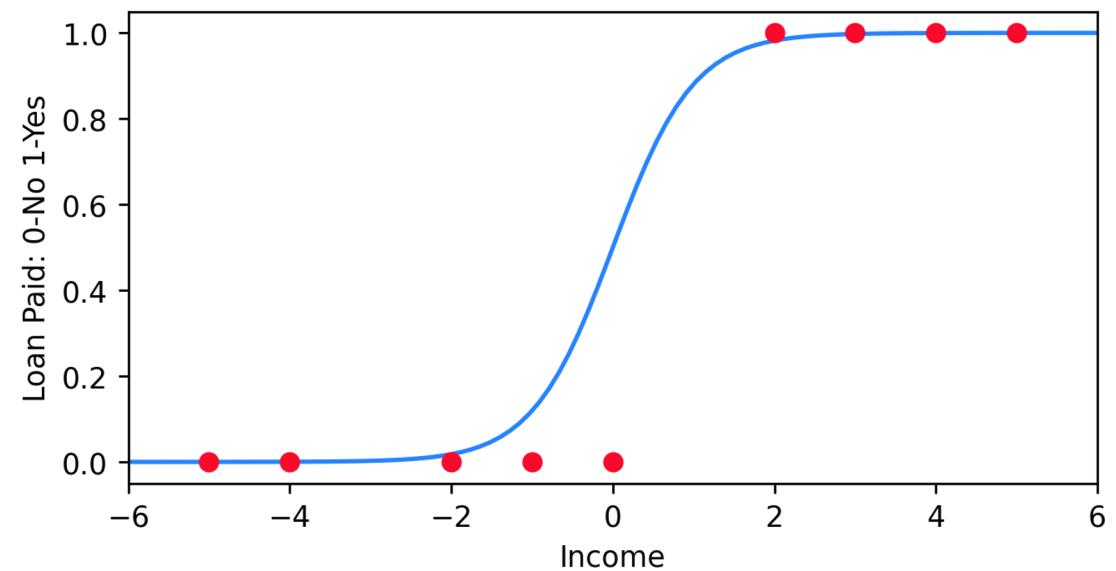
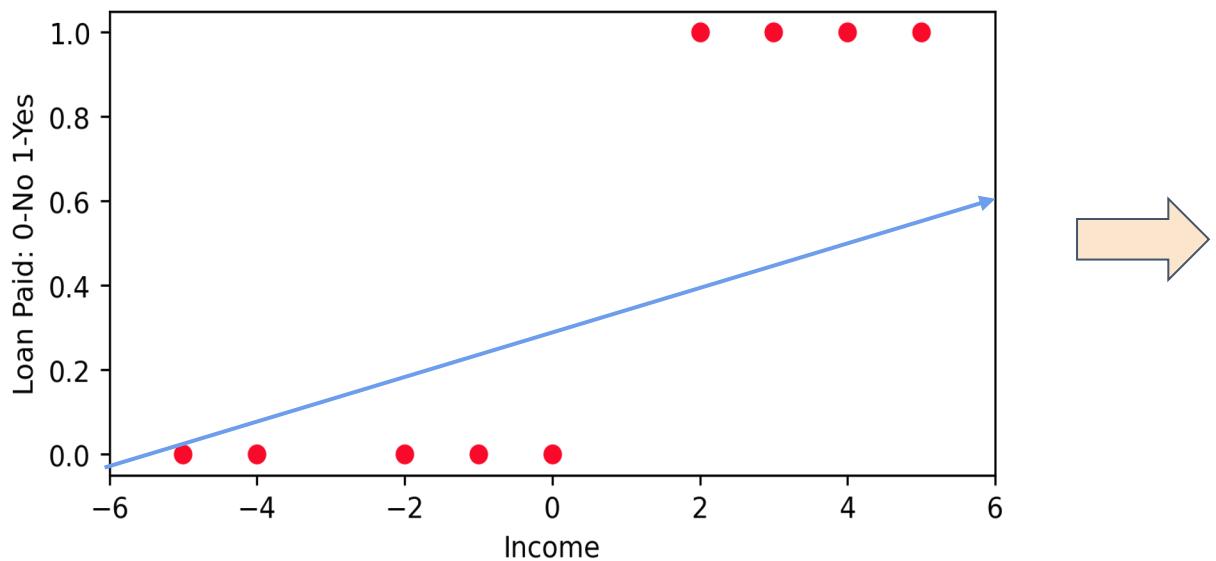
# Logistic Regression

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



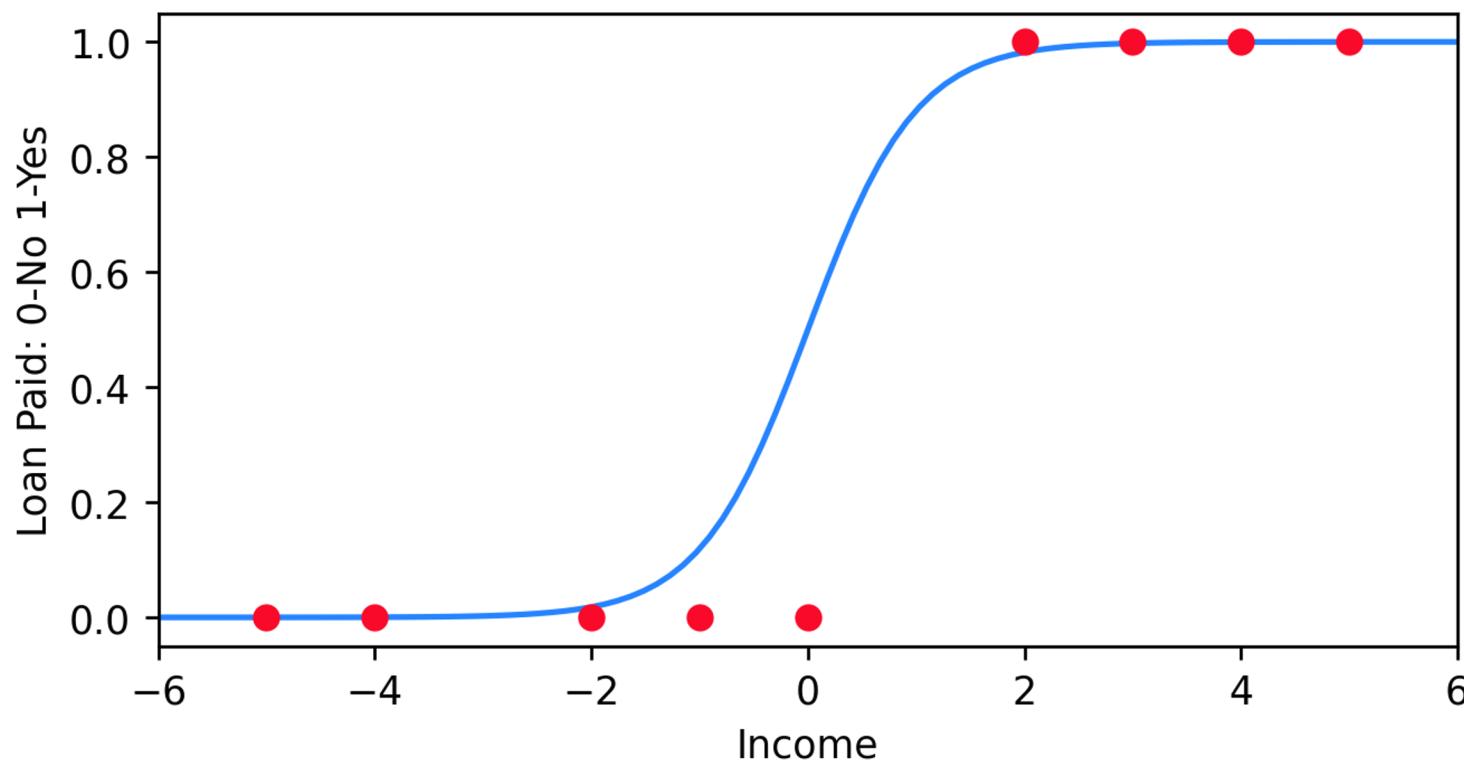
# Logistic Regression

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



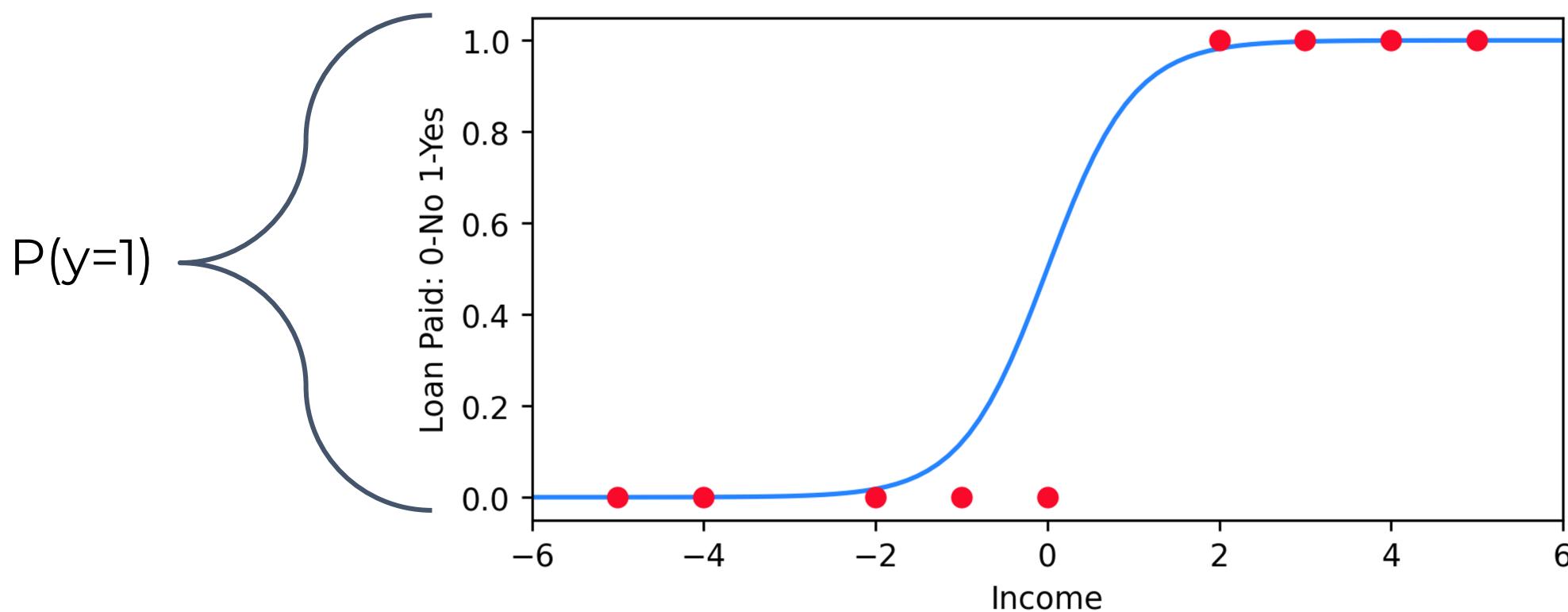
# Logistic Regression

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



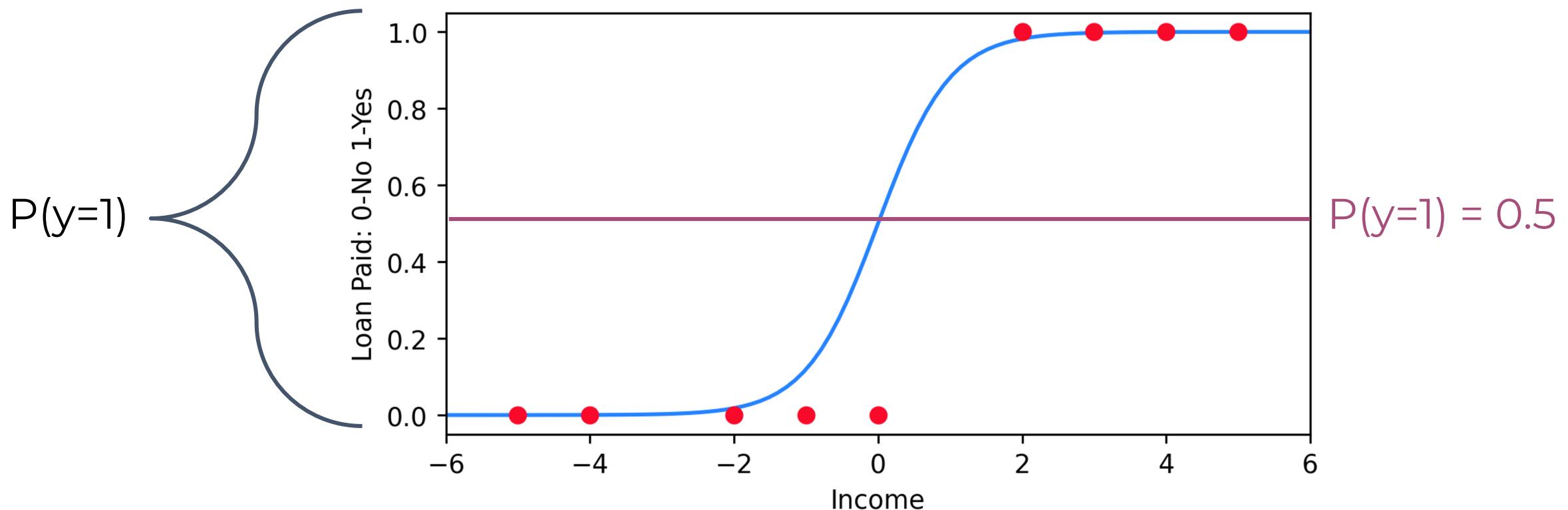
# Logistic Regression

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



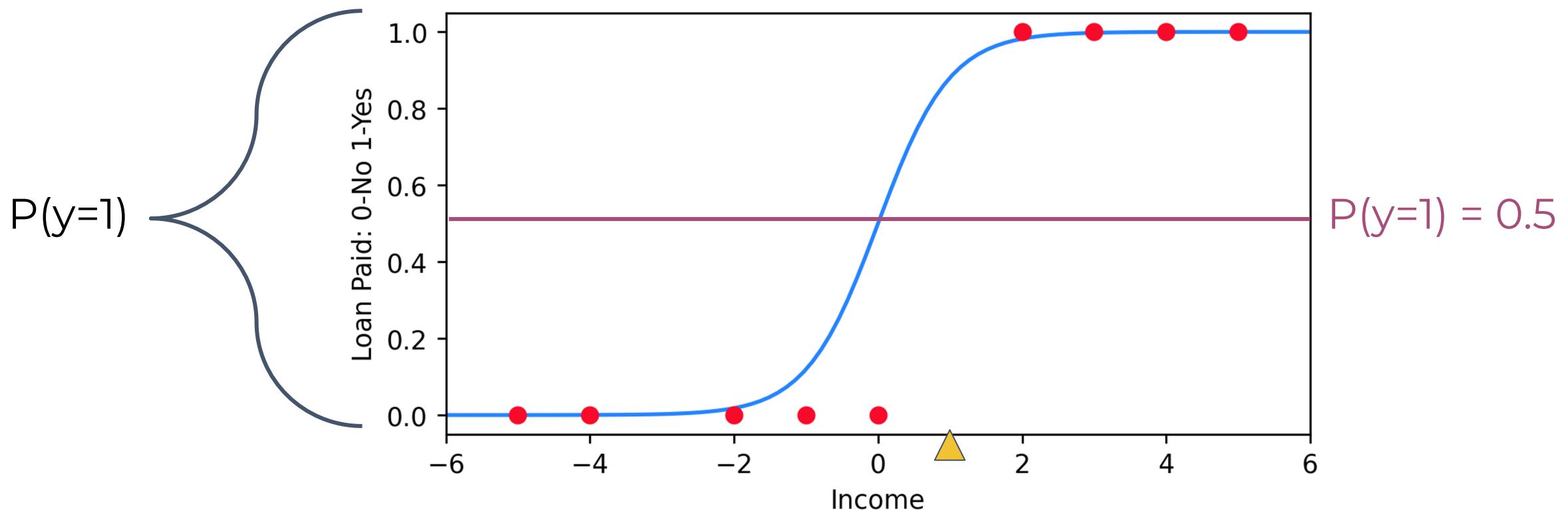
# Logistic Regression

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



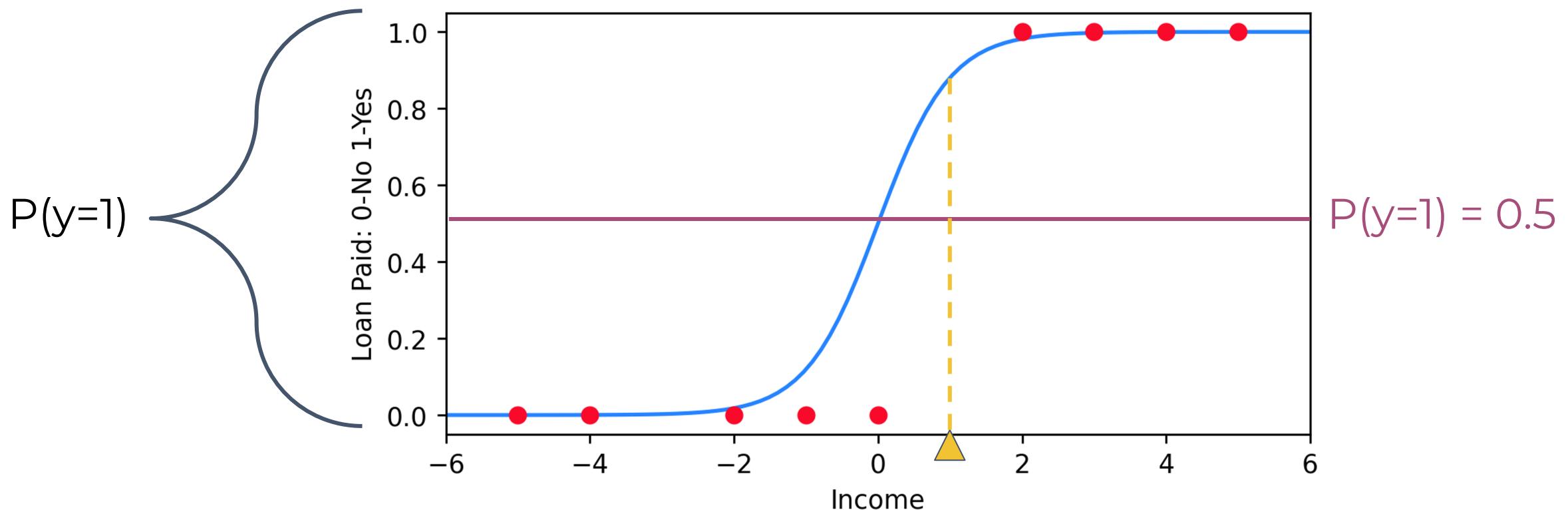
# Logistic Regression

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



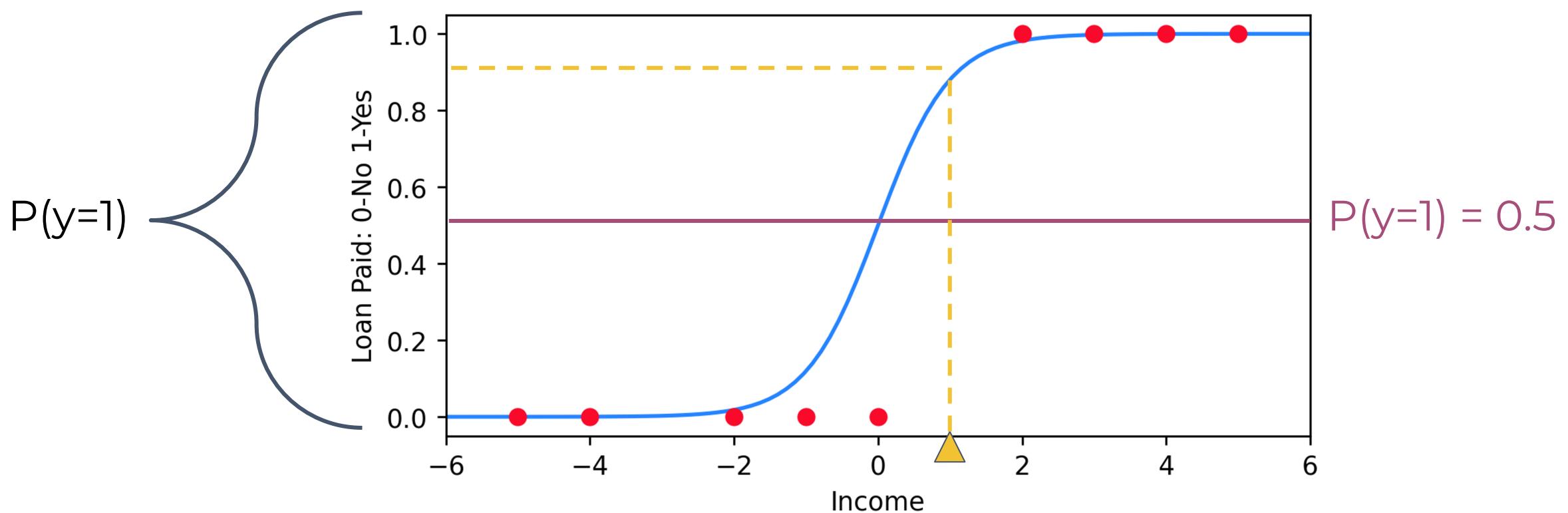
# Logistic Regression

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



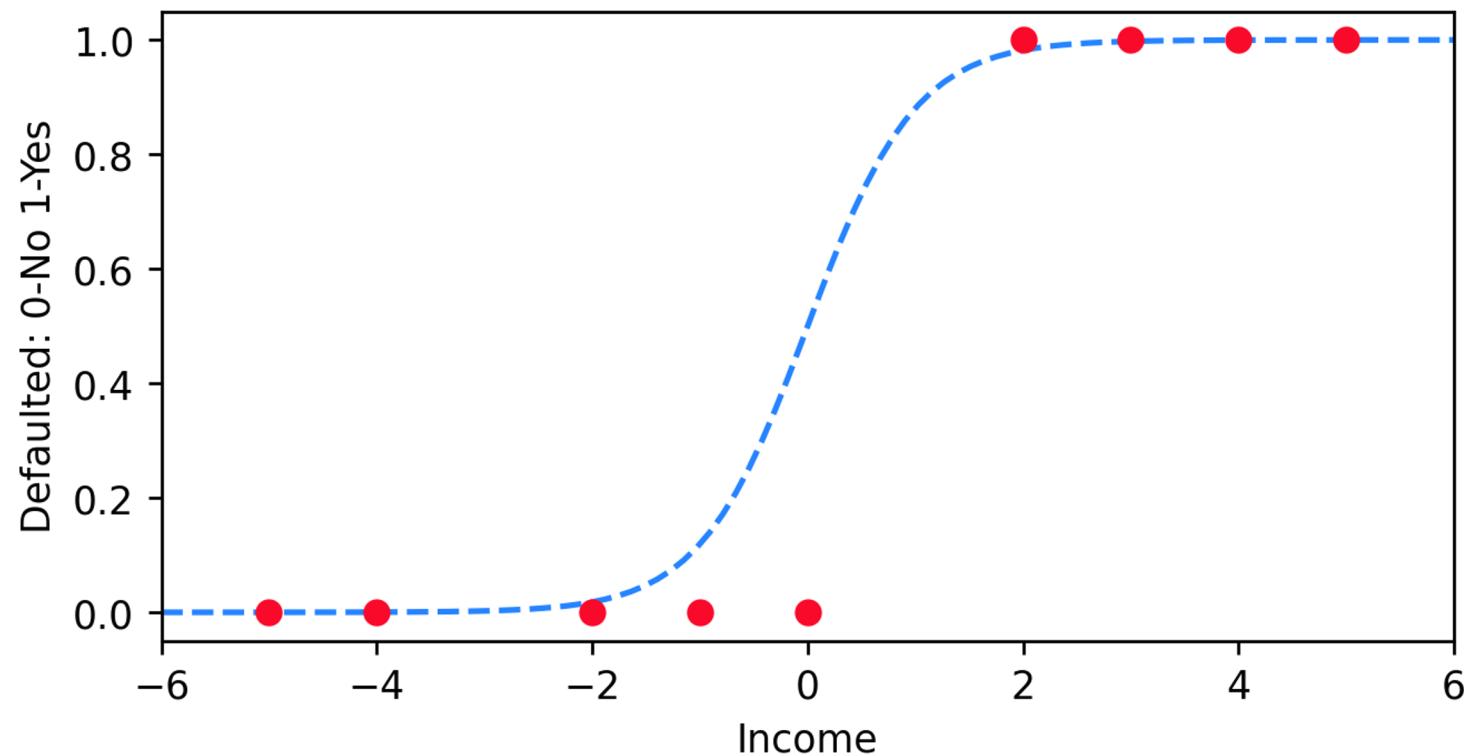
# Logistic Regression

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



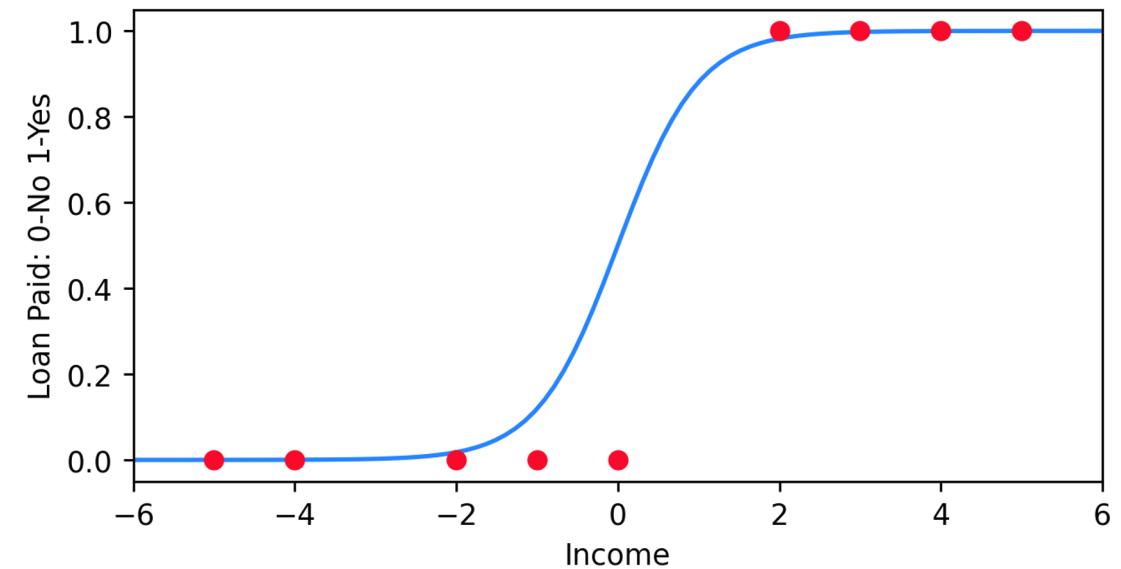
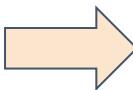
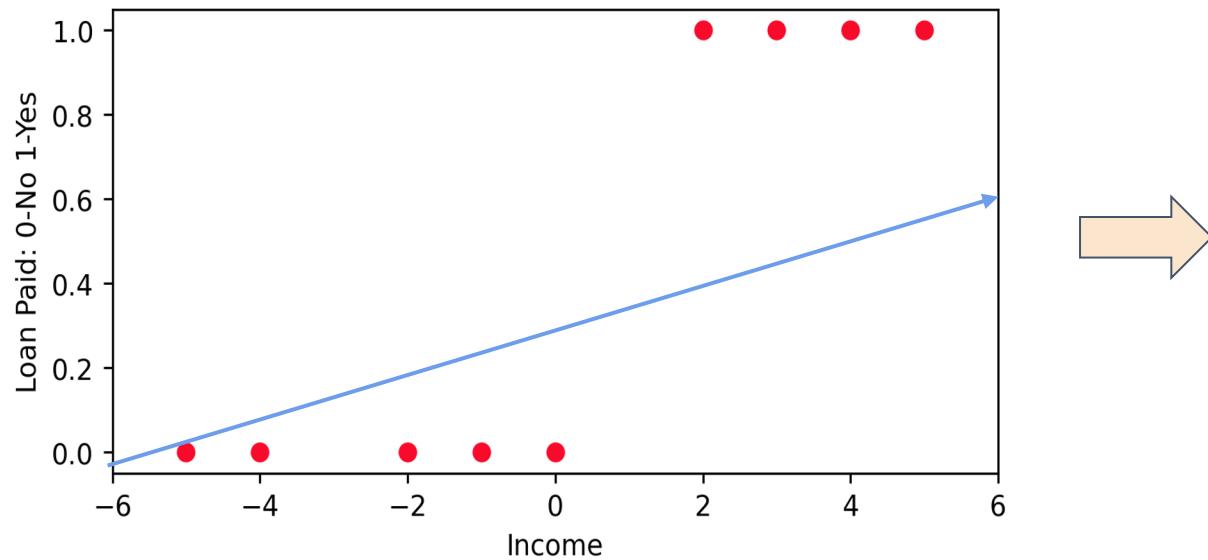
# Logistic Regression

- But how do we actually create this line?



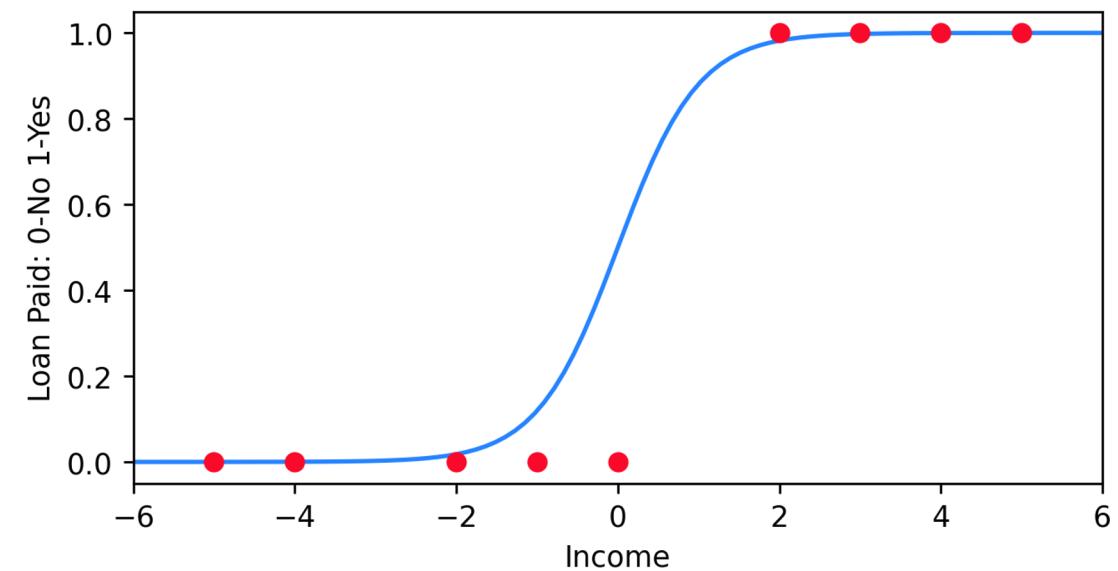
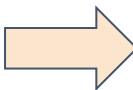
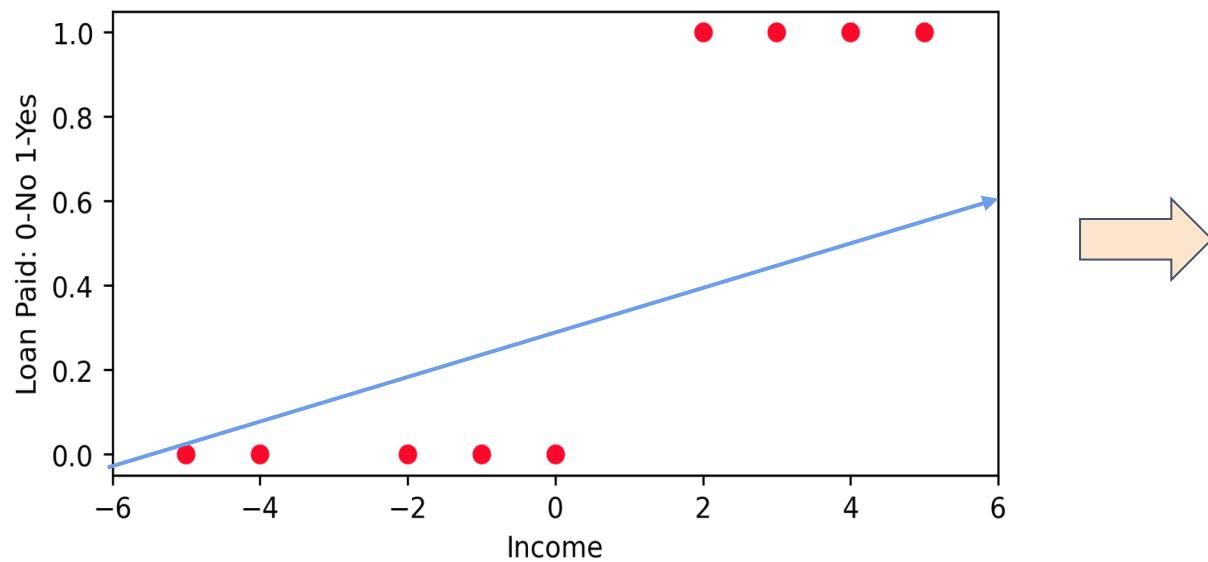
# Logistic Regression

- Fortunately, the mathematics of the conversion are quite simple!



# Logistic Regression

- In the next lecture we will go through the mathematical process of this conversion.

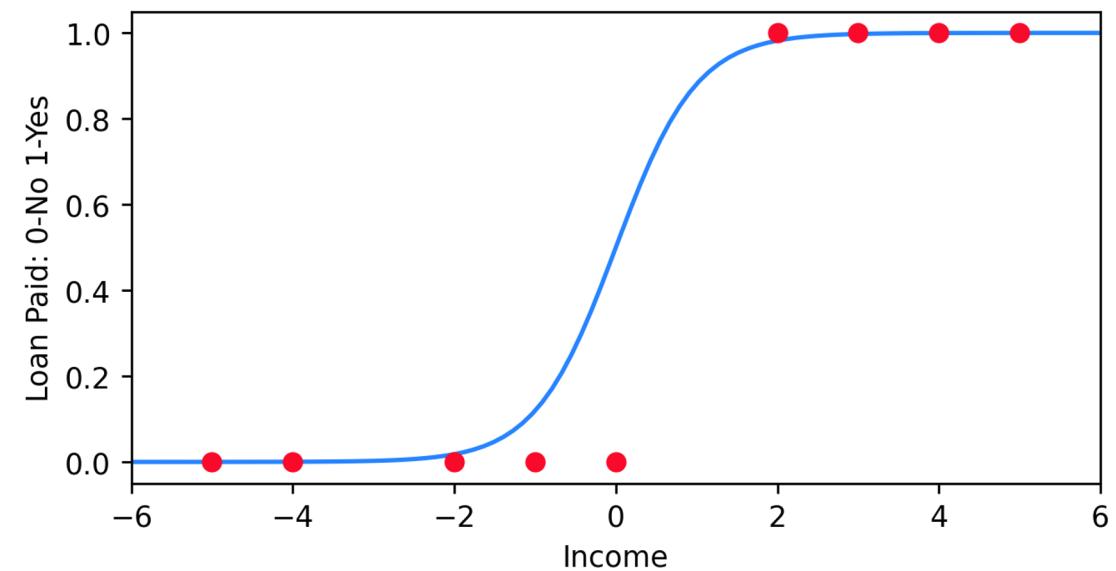
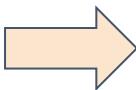
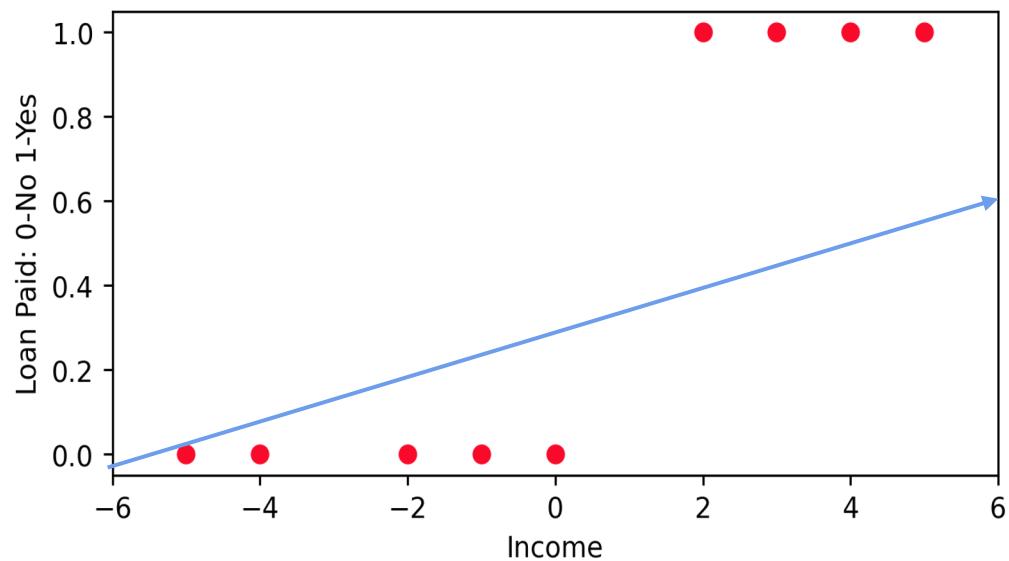


# **Logistic Regression Theory and Intuition**

Part Two: Linear to Logistic Math

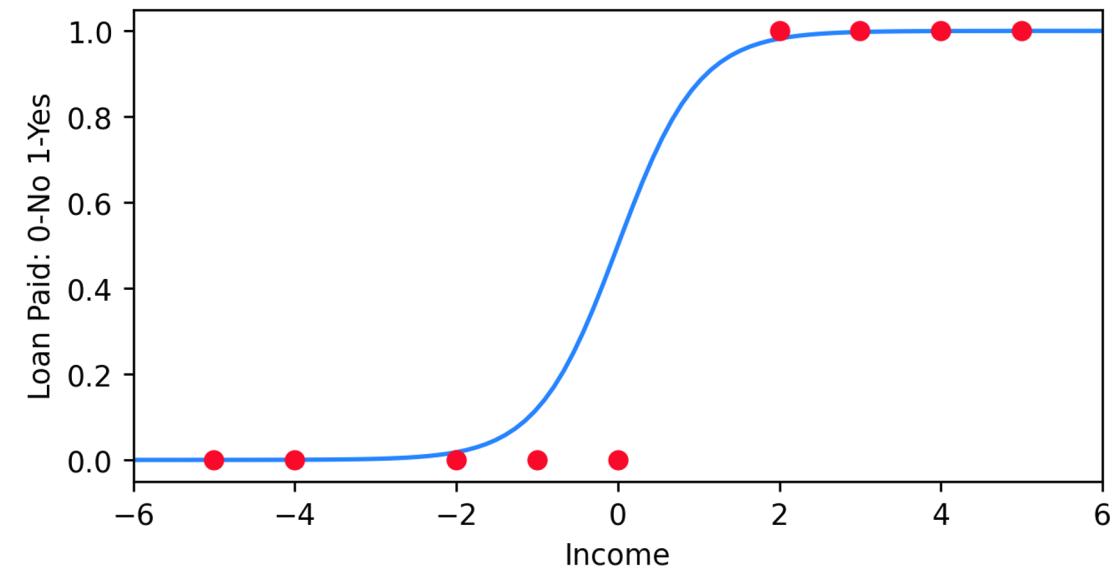
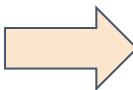
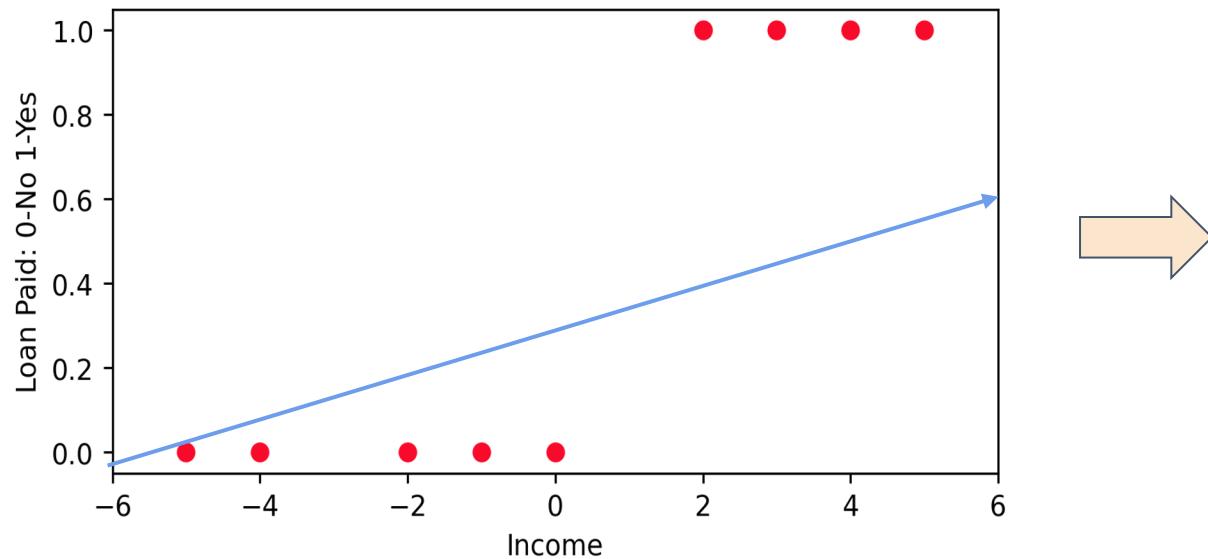
# Logistic Regression

- 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!

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

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

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

# 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

- How do we interpret the coefficients and their relation to  $\hat{y}$  ?

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

# **Logistic Regression Theory and Intuition**

**Part Three: Finding the Best Fit**

## Deriving the binary cross-entropy for logistic regression

Let us consider a predictor  $x$  and a binary (or Bernoulli) variable  $y$ .

Assuming there exist some relationship between  $x$  and  $y$ , an ideal model would predict

$$\mathcal{P}(y|\mathbf{x}) = \begin{cases} 1 & \text{if } y = 1 \\ 0 & \text{if } y = 0 \end{cases}$$

By using logistic regression, this unknown probability function is modeled as

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

## From the Bernoulli distribution to the binary cross-entropy

One way to assess how good of a job our model is doing is to compute the so-called *likelihood function*. Given  $m$  examples, this likelihood function is defined as

$$\mathcal{L}(\mathbf{w}) = \prod_{i=1}^m \hat{\mathcal{P}}(y_i | \mathbf{x}_i; \mathbf{w})$$

Ideally, we thus want to find the parameters  $\mathbf{w}$  that maximizes  $\mathcal{L}(\mathbf{w})$ . In practice, however, one usually does not work directly with this function but with its negative log for the sake of simplicity

$$-\log \mathcal{L}(\mathbf{w}) = -\sum_{i=1}^m \log \hat{\mathcal{P}}(y_i | \mathbf{x}_i; \mathbf{w})$$

logistic regression only models  $P(1|\mathbf{x}, \mathbf{w})$ ? Given that

$$\hat{\mathcal{P}}(0|\mathbf{x}; \mathbf{w}) = 1 - \hat{\mathcal{P}}(1|\mathbf{x}; \mathbf{w})$$

one can use a simple exponentiation trick to write

$$\hat{\mathcal{P}}(y|\mathbf{x}; \mathbf{w}) = \hat{\mathcal{P}}(1|\mathbf{x}; \mathbf{w})^y \times \hat{\mathcal{P}}(0|\mathbf{x}; \mathbf{w})^{1-y}$$

Inserting this expression into the negative log-likelihood function (and normalizing by the number of examples), we finally obtain the desired normalized binary cross-entropy

$$\begin{aligned} \mathcal{J}(\mathbf{w}) &= -\frac{1}{m} \sum_{i=1}^m y_i \log \hat{\mathcal{P}}(1|\mathbf{x}_i, \mathbf{w}) + (1 - y_i) \log (1 - \hat{\mathcal{P}}(0|\mathbf{x}_i, \mathbf{w})) \\ &= -\frac{1}{m} \sum_{i=1}^m y_i \log \sigma(\mathbf{w}^T \mathbf{x}_i) + (1 - y_i) \log (1 - \sigma(\mathbf{w}^T \mathbf{x}_i)) \end{aligned}$$

# Logistic Regression

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

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

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

# Logistic Regression

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

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

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

# **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 are 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
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

- What is accuracy?

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	4	2
	HEALTHY	1	93

- Accuracy:
  - How often is the model correct?

$$\text{Acc} = (\text{TP} + \text{TN}) / \text{Total}$$

# Classification Metrics

- Calculating accuracy:

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	4	2
	HEALTHY	1	93

$$(4+93)/100 = 97\% \text{ Accuracy}$$

- Accuracy:
  - How often is the model correct?

$$\text{Acc} = (\text{TP}+\text{TN})/\text{Total}$$

# Classification Metrics

- Is this a good value for accuracy?

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	4	2
	HEALTHY	1	93

$$(4+93)/100 = 97\% \text{ Accuracy}$$

- Accuracy:
  - How often is the model correct?

$$\text{Acc} = (\text{TP}+\text{TN})/\text{Total}$$

# Classification Metrics

- The accuracy paradox...

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	4	2
	HEALTHY	1	93

$(4+93)/100 = 97\% \text{ Accuracy}$

- Accuracy:
  - How often is the model correct?

$$\text{Acc} = (\text{TP}+\text{TN})/\text{Total}$$

# 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

$(0+95)/100 = 95\% \text{ Accuracy}$

- 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?

$(TP) / \text{Total Actual Positives}$

# 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

$$\text{Recall} = \frac{\text{TP}}{\text{Total Actual Positives}}$$

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

$\frac{\text{TP}}{\text{Total Actual Positives}}$

# Classification Metrics

- Let's begin with recall.

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	4	2
	HEALTHY	1	93

$$\text{Recall} = \frac{4}{5}$$

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

$\frac{\text{TP}}{\text{Total Actual Positives}}$

# 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?
- $$\frac{\text{TP}}{\text{Total Actual Positives}}$$

# Classification Metrics

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

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	0	0
	HEALTHY	5	95

Recall =  
 $(TP) / \text{Total Actual Positives}$

- Recall:
  - How many relevant cases are found?
$$(TP) / \text{Total Actual Positives}$$

# Classification Metrics

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

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	0	0
	HEALTHY	5	95

$$\text{Recall} = \frac{\text{TP}}{\text{Total Actual Positives}}$$
$$(0)/5 !$$

- Recall:
  - How many relevant cases are found?
$$\frac{\text{TP}}{\text{Total Actual Positives}}$$

# Classification Metrics

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

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	0	0
	HEALTHY	5	95

$$\text{Recall} = \frac{0}{5}$$

- Recall:
    - How many relevant cases are found?
- $\frac{\text{TP}}{\text{Total Actual Positives}}$

# 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 =  
 $(TP) / \text{Total Predicted Positives}$

- Precision:
    - When prediction is positive, how often is it correct?
- $(TP) / \text{Total Predicted Positives}$

# Classification Metrics

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

		ACTUAL	
		INFECTED	HEALTHY
PREDICTED	INFECTED	4	2
	HEALTHY	1	93

$$\text{Precision} = \frac{\text{TP}}{\text{Total Predicted Positives}}$$

- 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

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

- 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

$$\text{Precision} = \frac{(4)}{6}$$

- 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

$$\text{Precision} = 0.666$$

- 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.

---

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

# 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.

---

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

# Classification Metrics

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

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



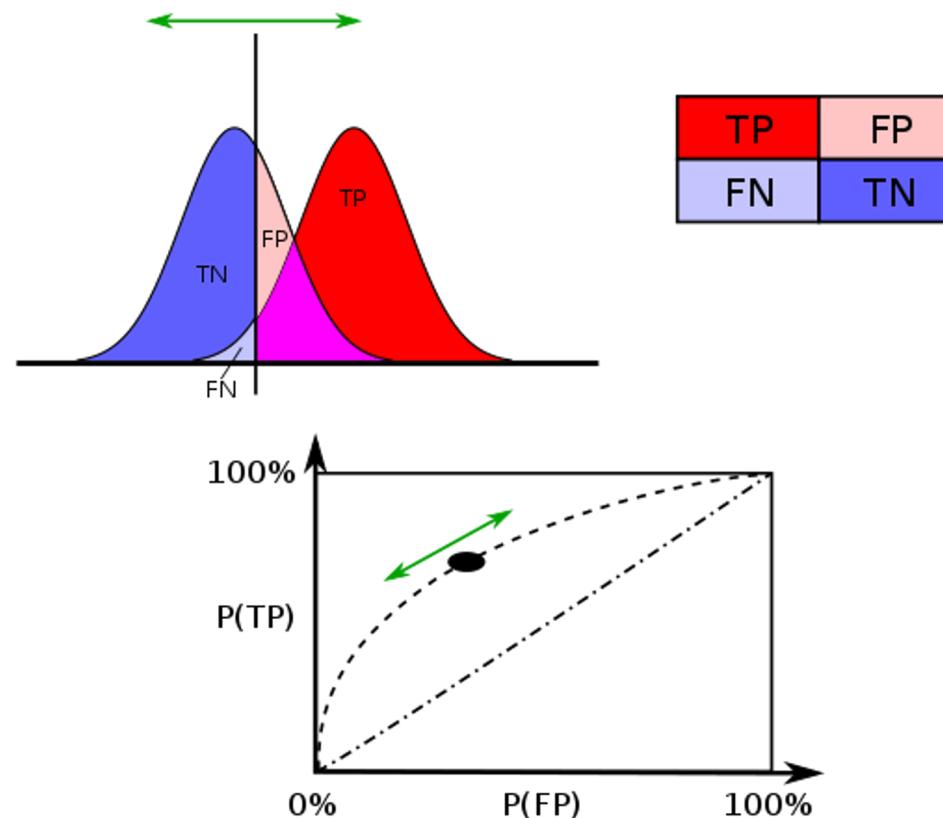
# Classification Metrics

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



# Classification Metrics

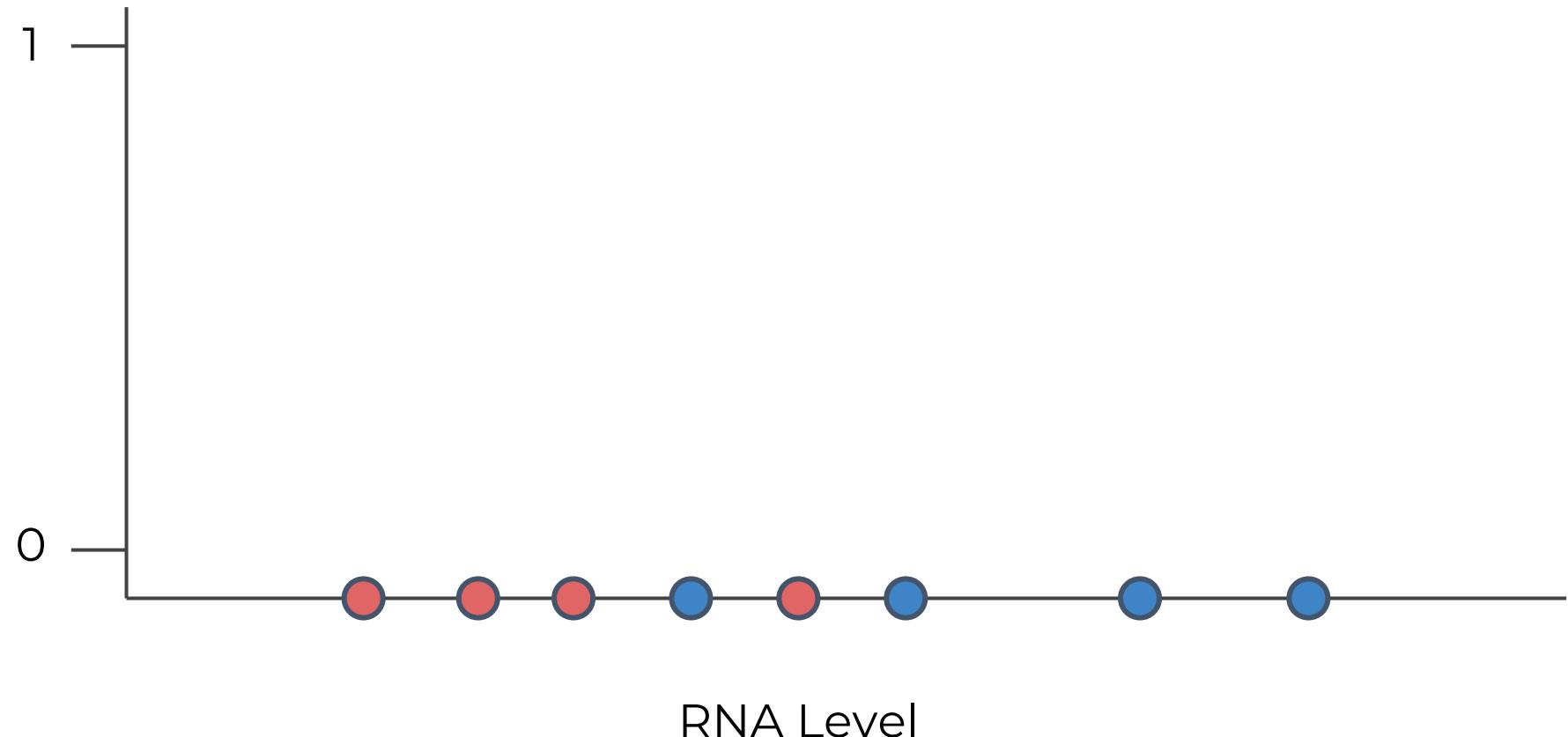
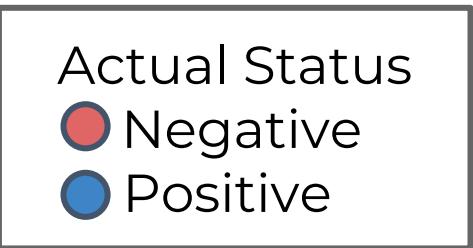
- There can be a trade-off between True Positives and False Positives.



# Classification Metrics

- Our previous infection test.

Infection Test  
0 - Negative  
1 - Positive

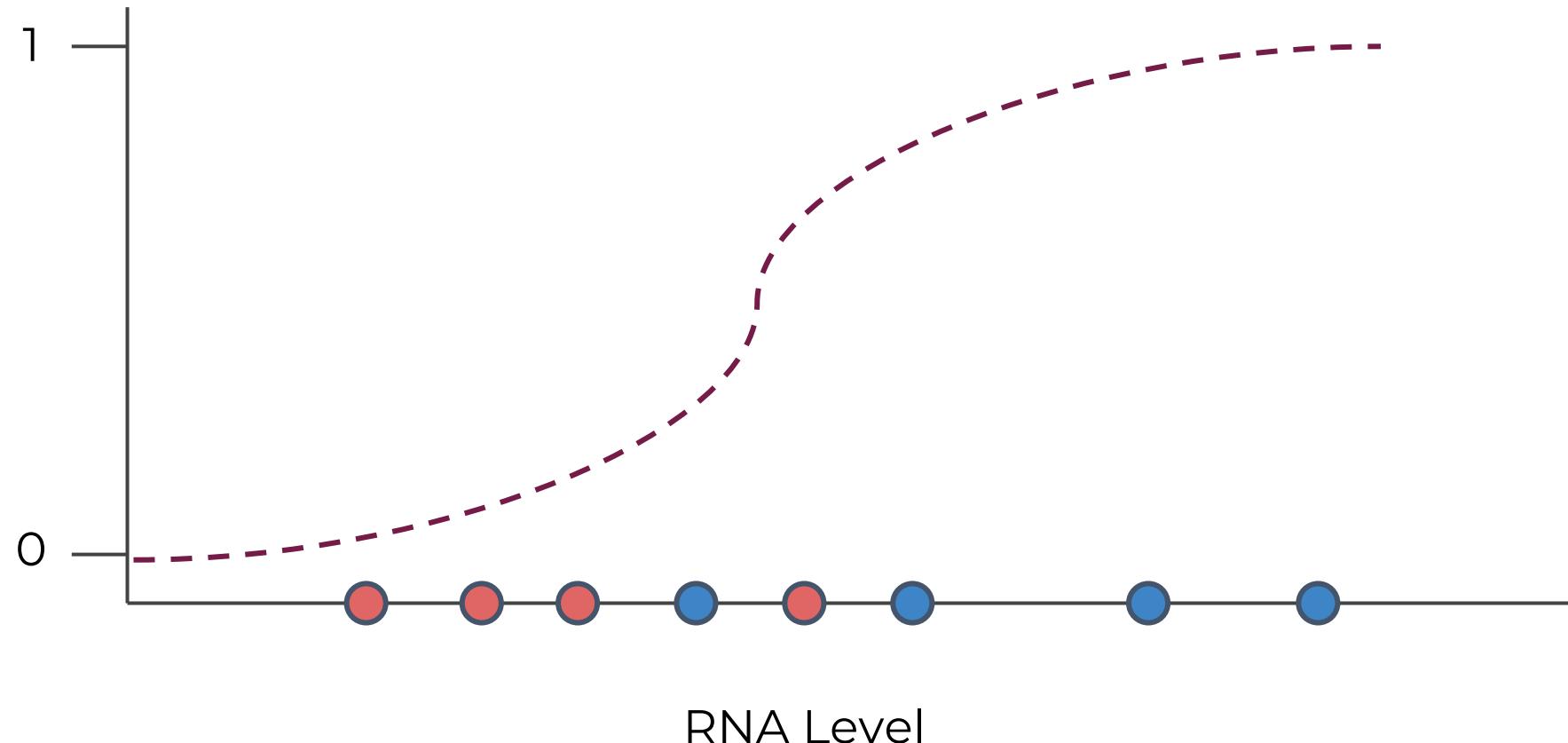


# Classification Metrics

- Fit logistic regression model.

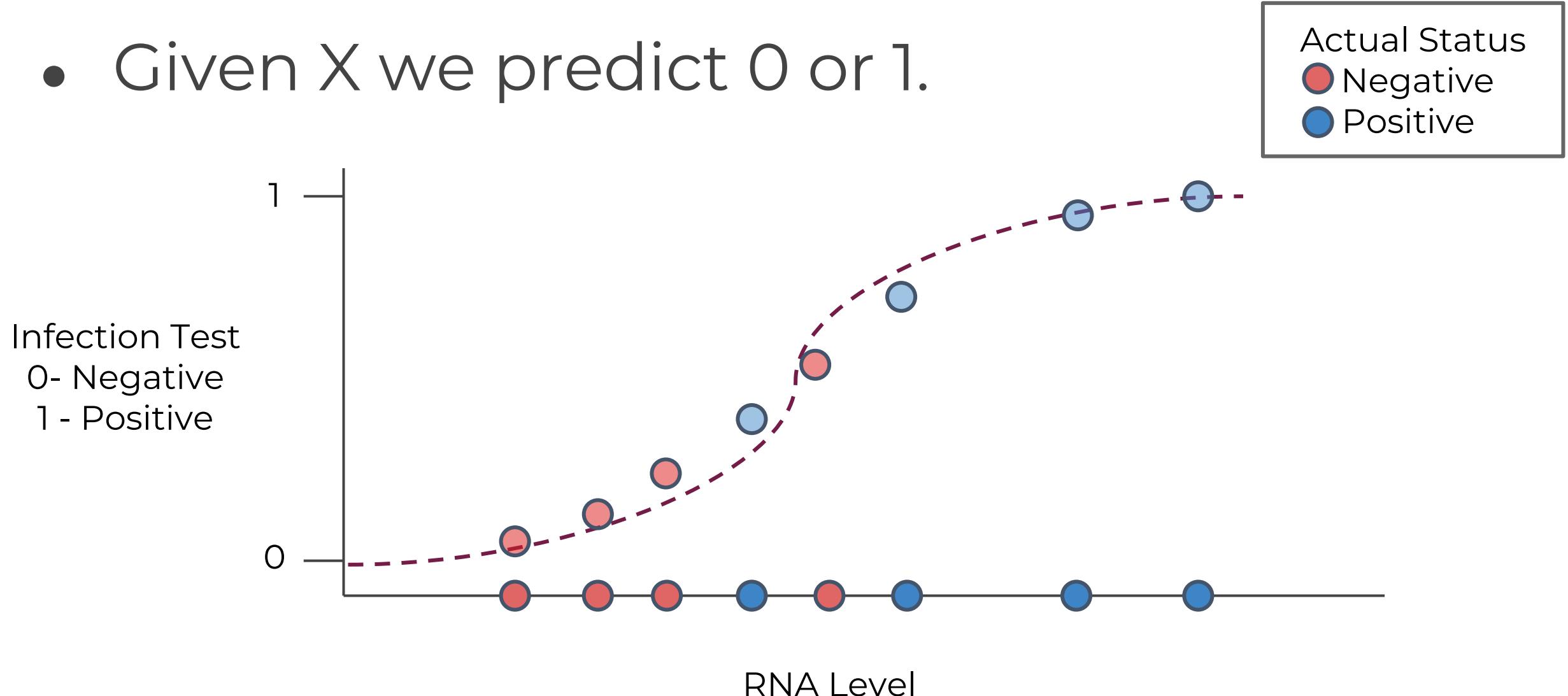
Infection Test  
0 - Negative  
1 - Positive

Actual Status  
● Negative  
● Positive



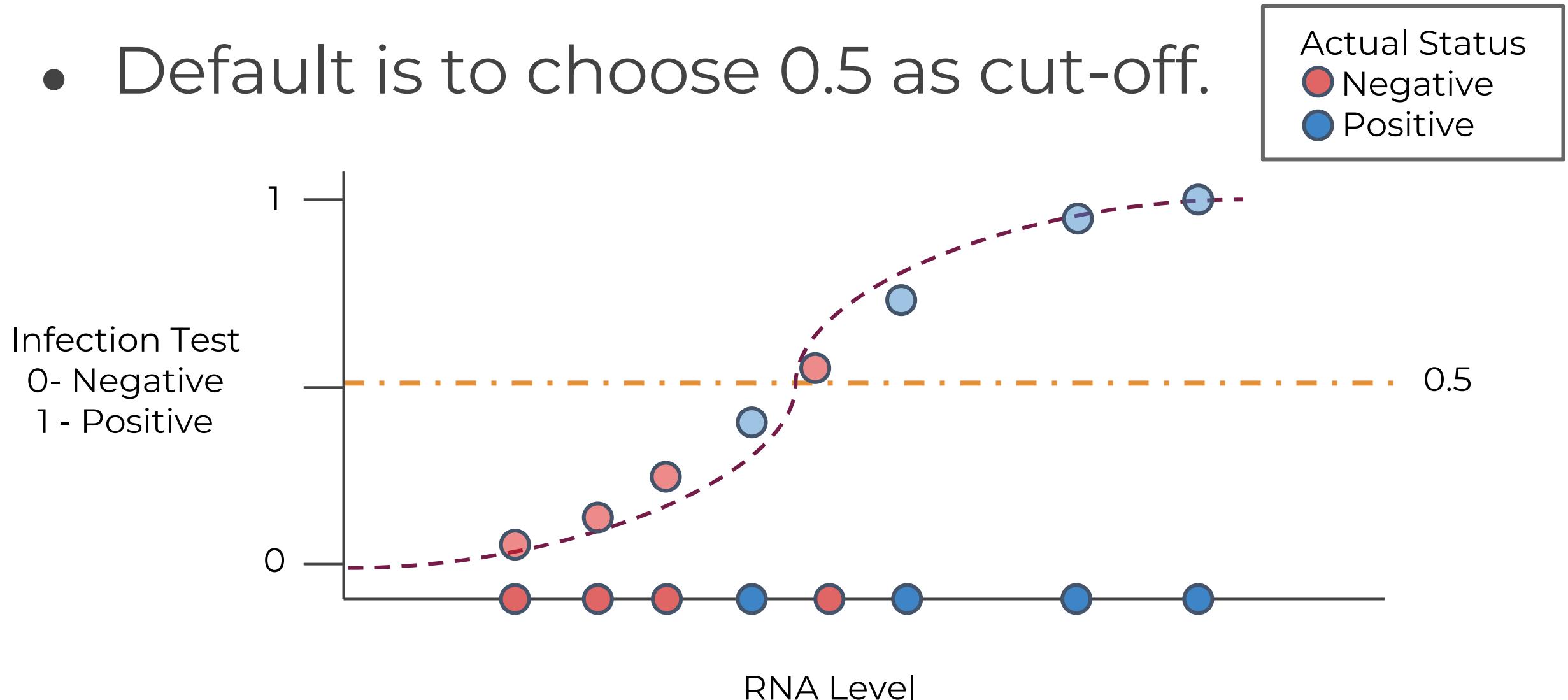
# Classification Metrics

- Given  $X$  we predict 0 or 1.



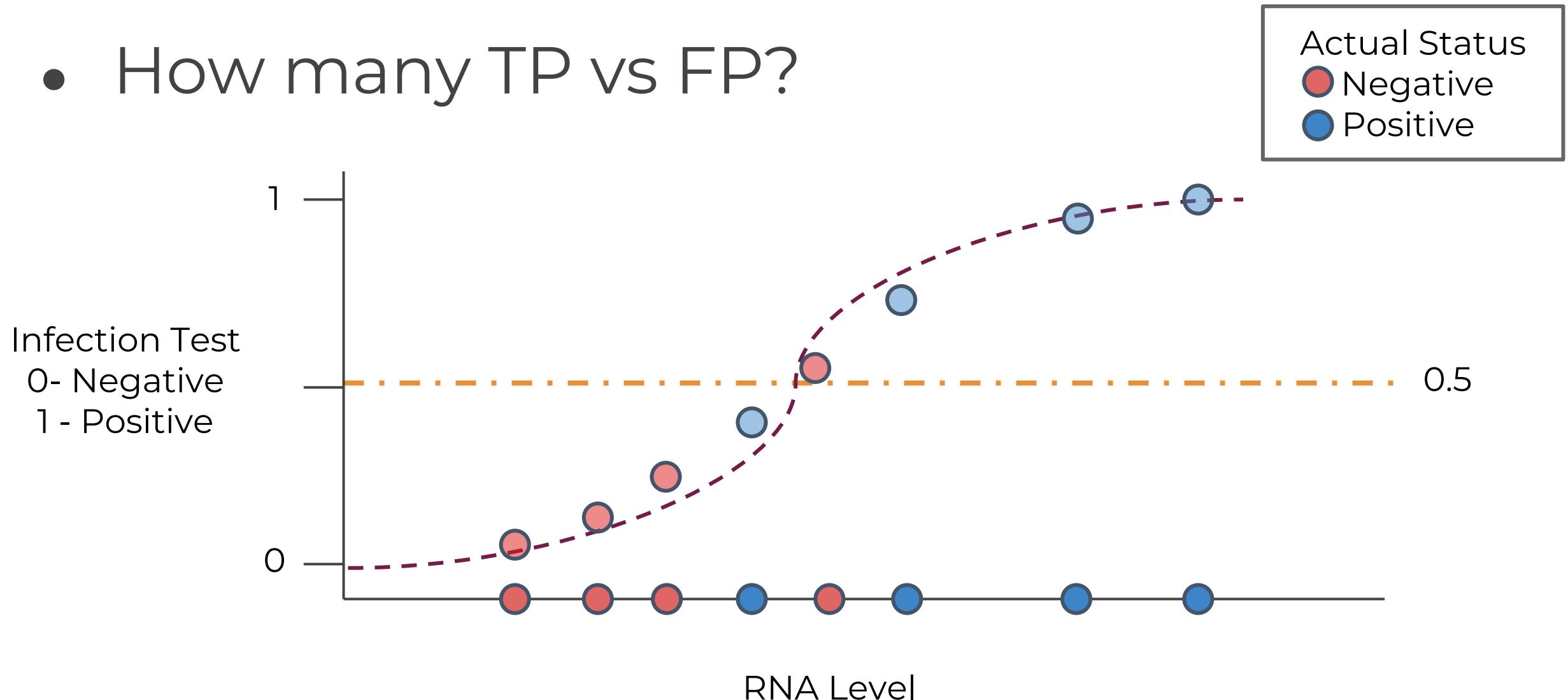
# Classification Metrics

- Default is to choose 0.5 as cut-off.



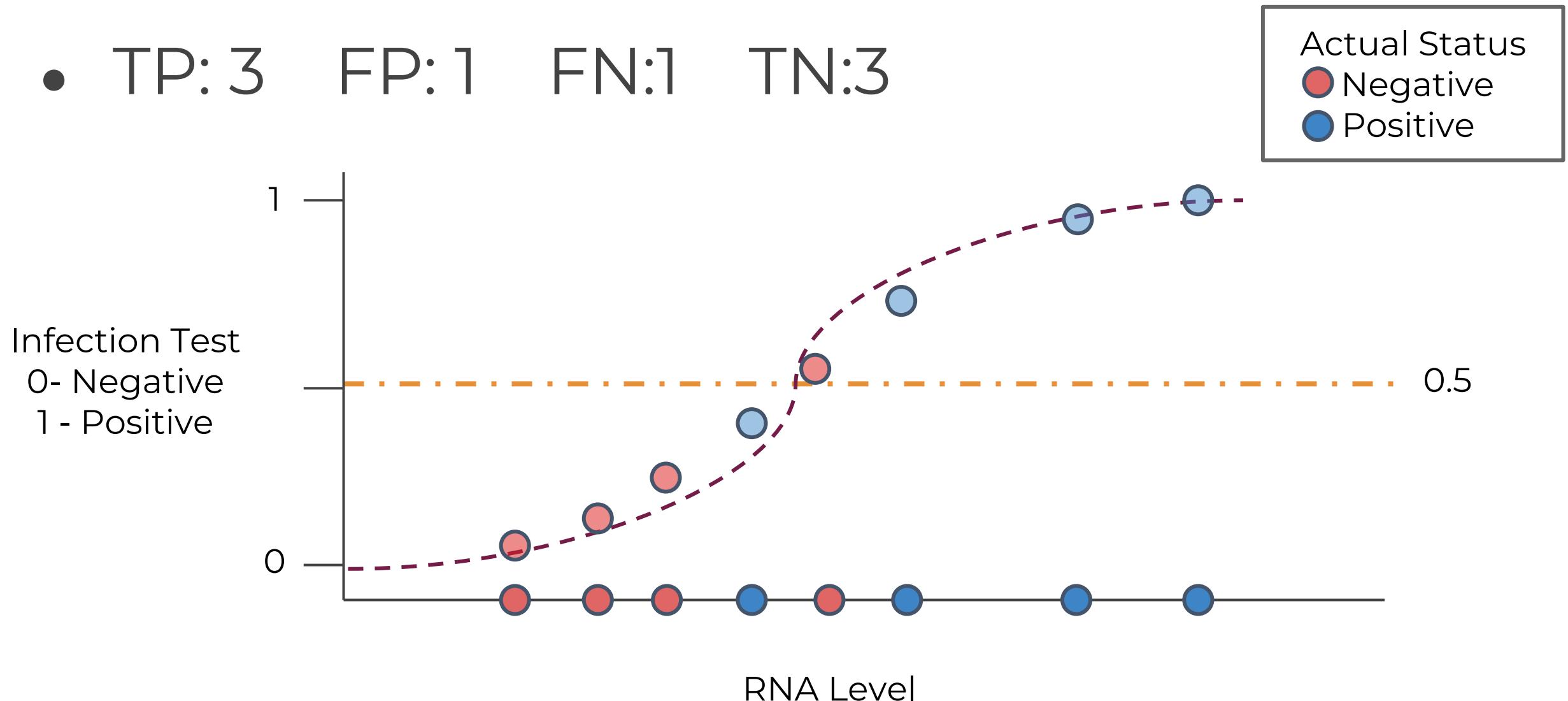
# Classification Metrics

- How many TP vs FP?



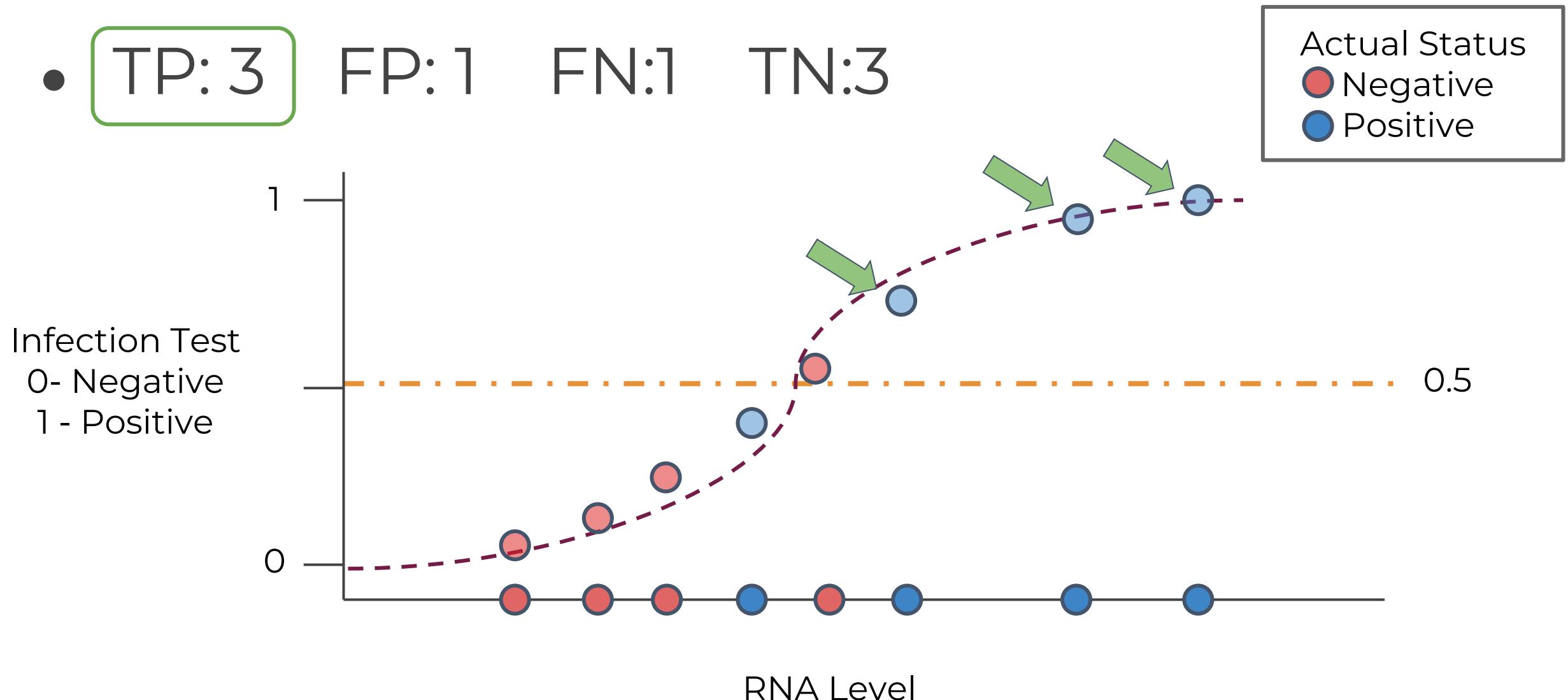
# Classification Metrics

- TP: 3   FP: 1   FN: 1   TN: 3



# Classification Metrics

- TP: 3   FP: 1   FN: 1   TN: 3



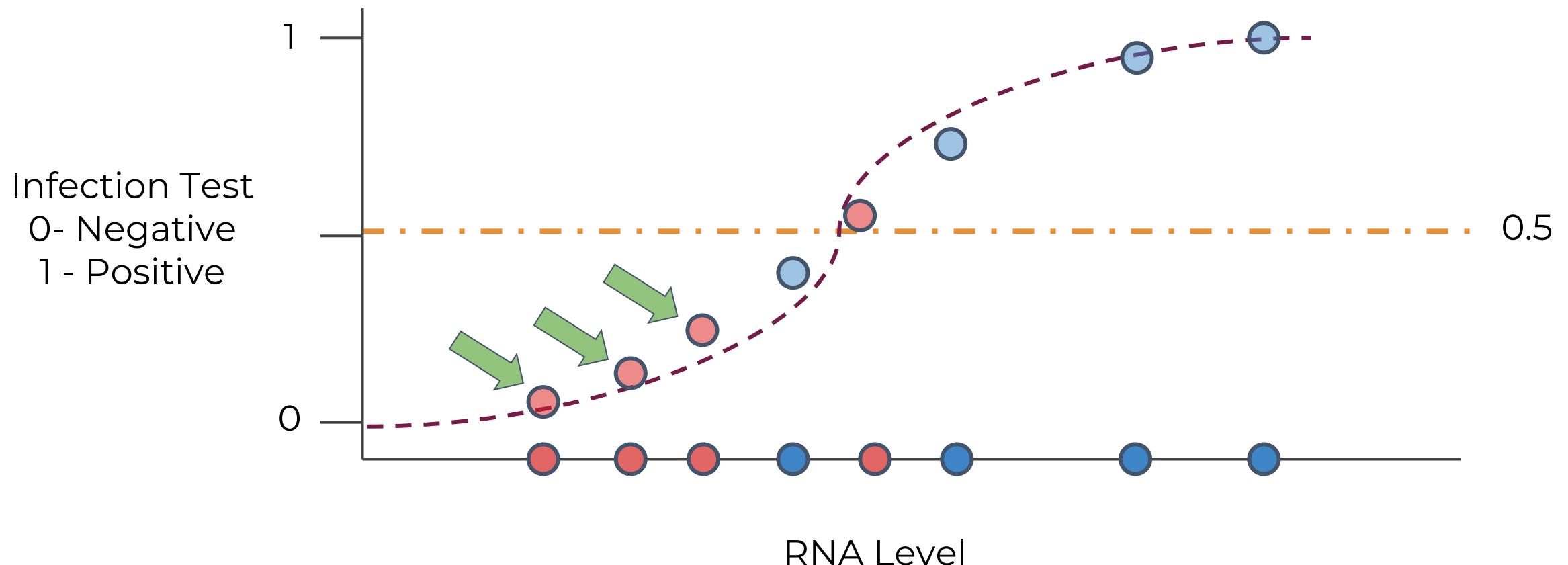
# Classification Metrics

- TP: 3   FP: 1   FN: 1

TN:3

Actual Status

- Negative
- Positive

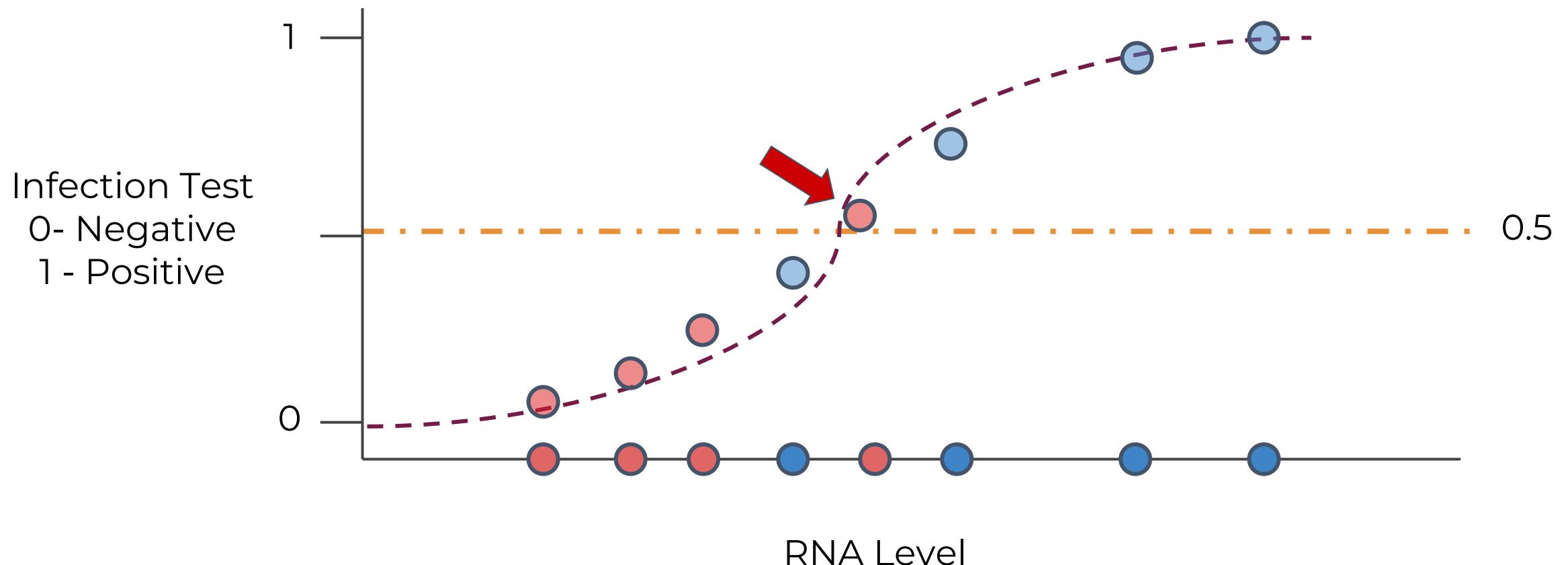


# Classification Metrics

- TP: 3    **FP: 1**    FN:1    TN:3

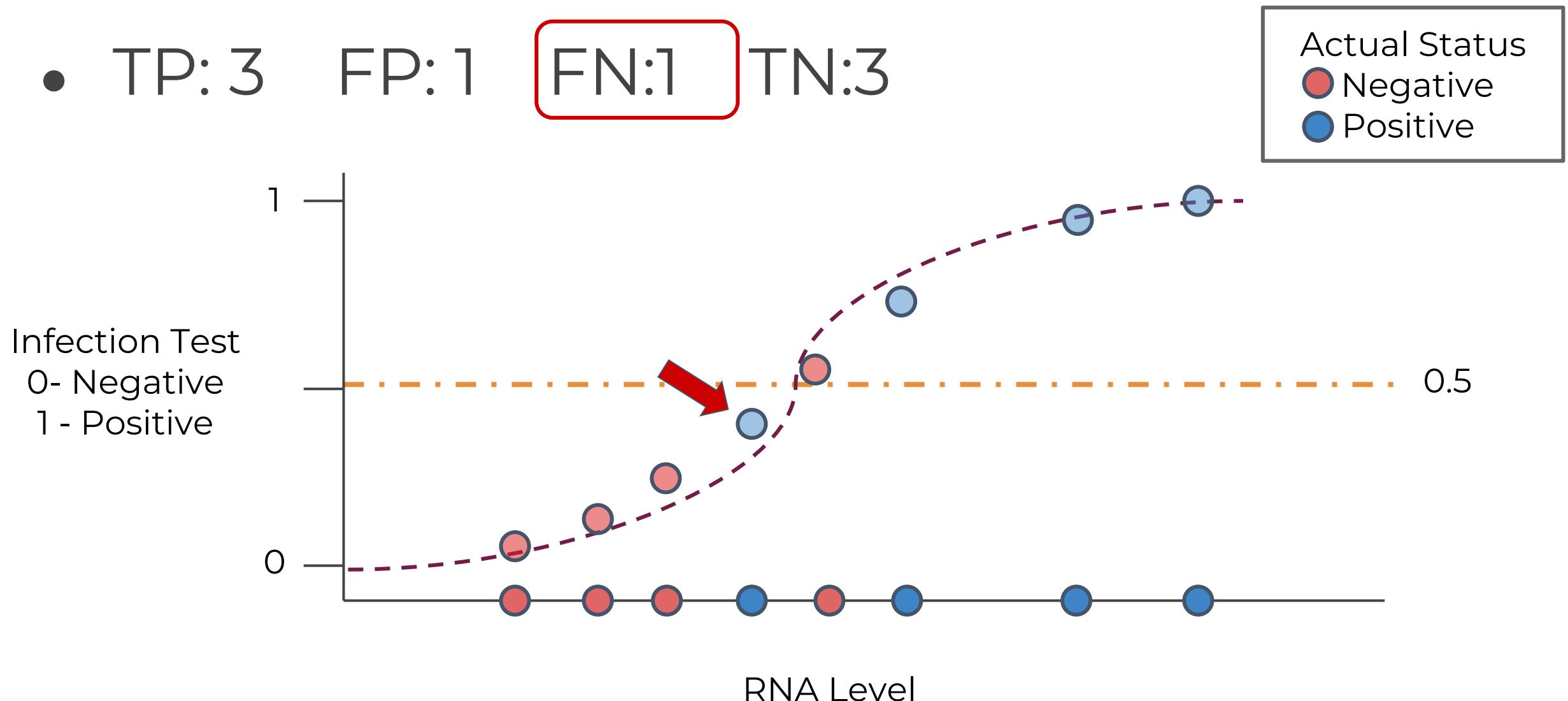
Actual Status

- Negative
- Positive



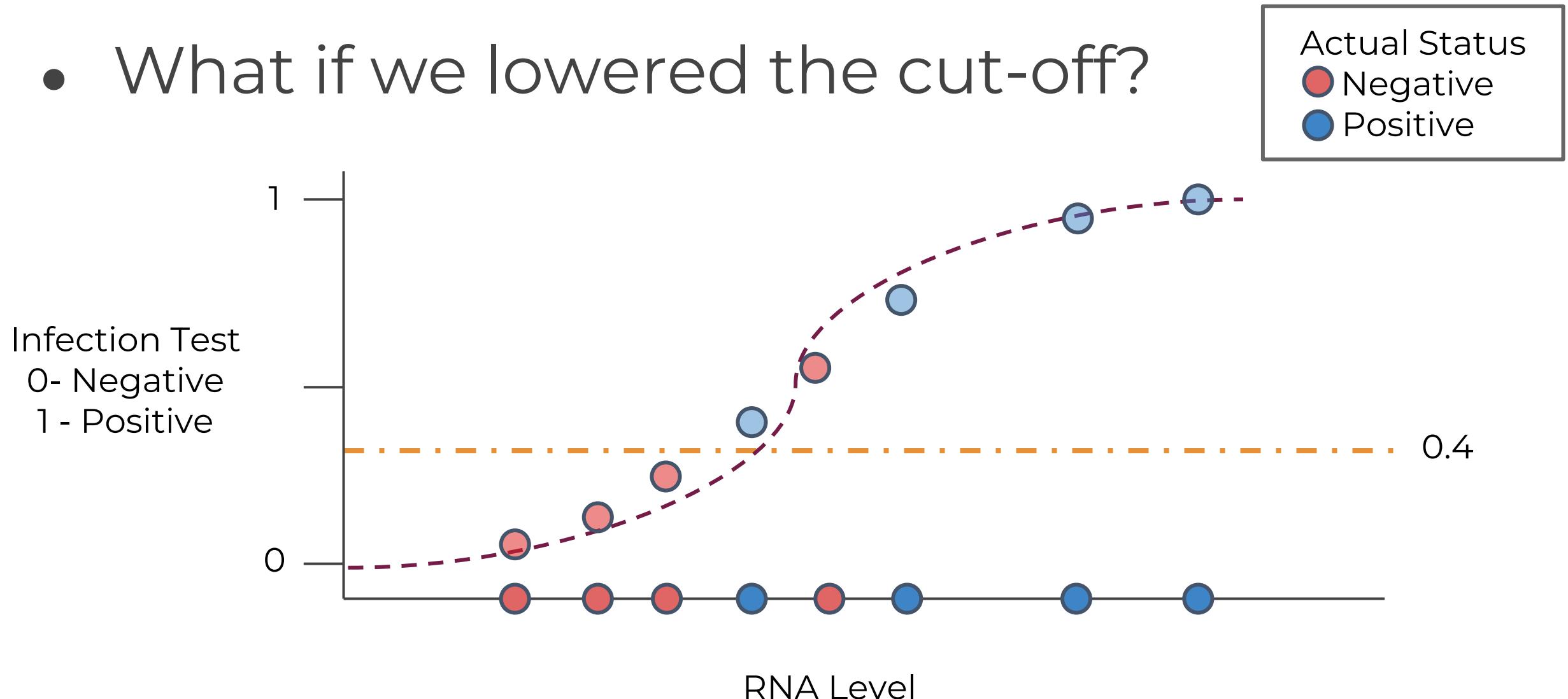
# Classification Metrics

- TP: 3   FP: 1   FN: 1   TN: 3



# Classification Metrics

- What if we lowered the cut-off?

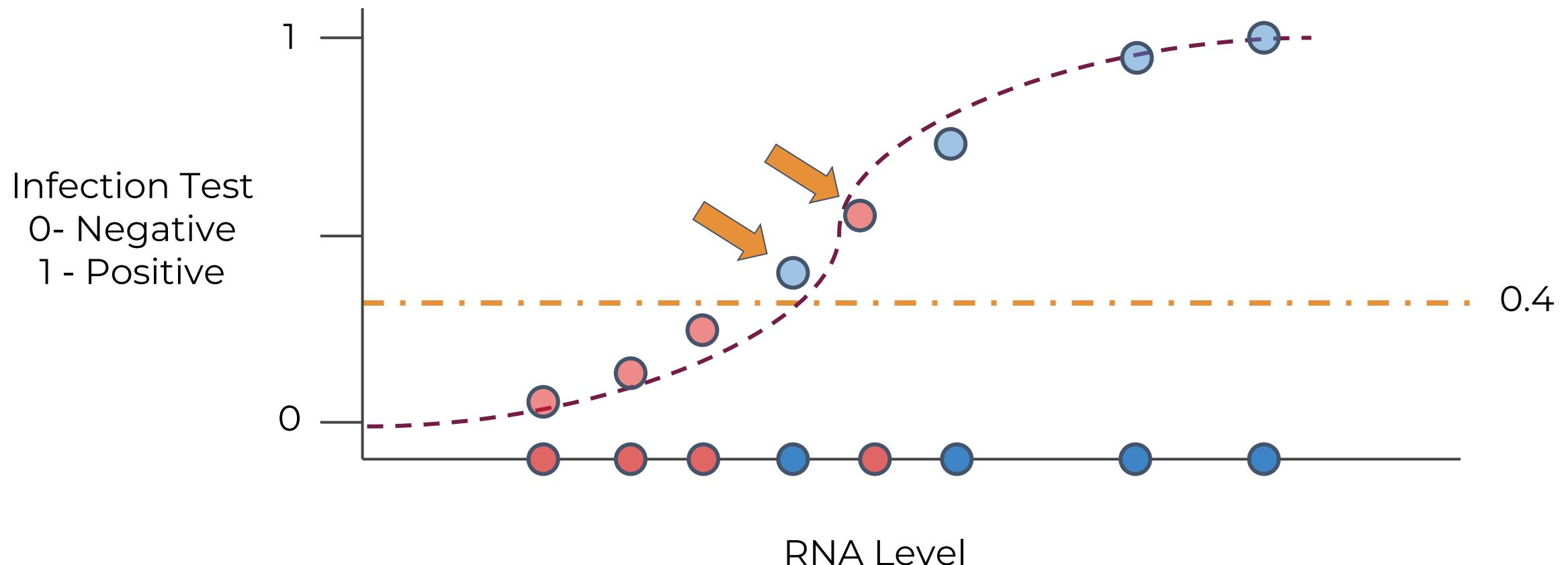


# Classification Metrics

- TP: 3    **FP: 2    FN:0**    TN:3

Actual Status

- Negative
- Positive



# Classification Metrics

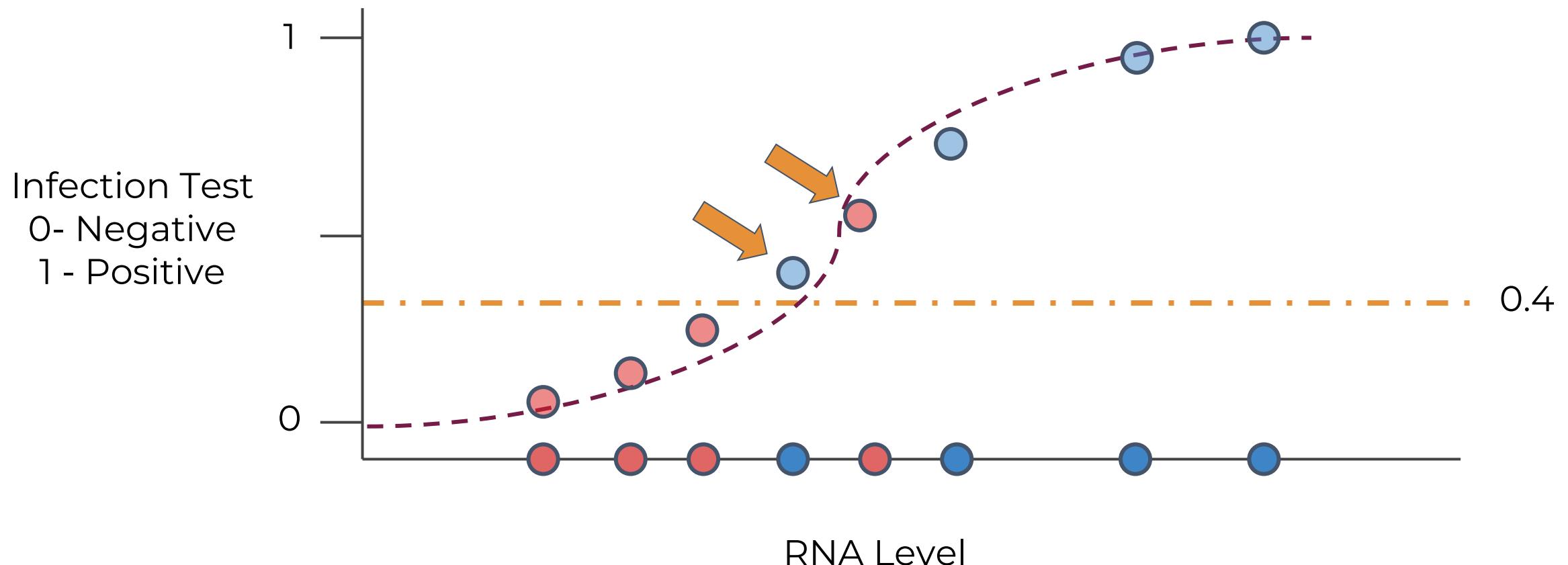
- In certain situations, we gladly accept more false positives to reduce false negatives.
- Imagine a dangerous virus test, we would much rather produce false positives and later do more stringent examination than accidentally release a false negative!

# Classification Metrics

- TP: 3    **FP: 2    FN:0**    TN:3

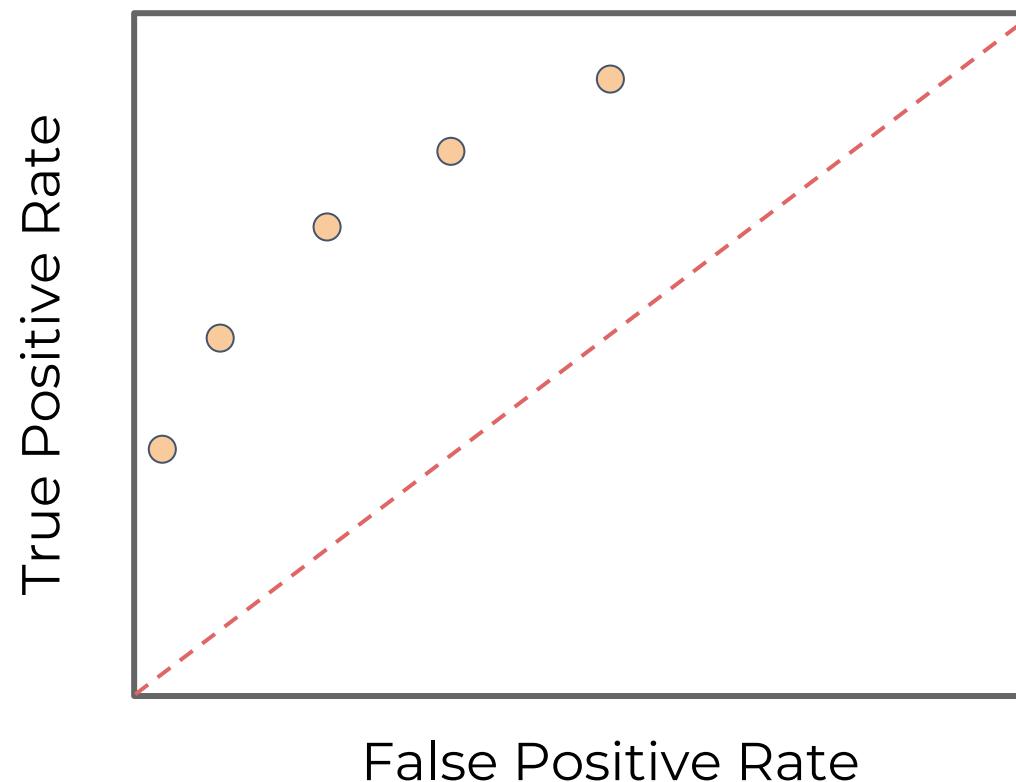
Actual Status

- Negative
- Positive



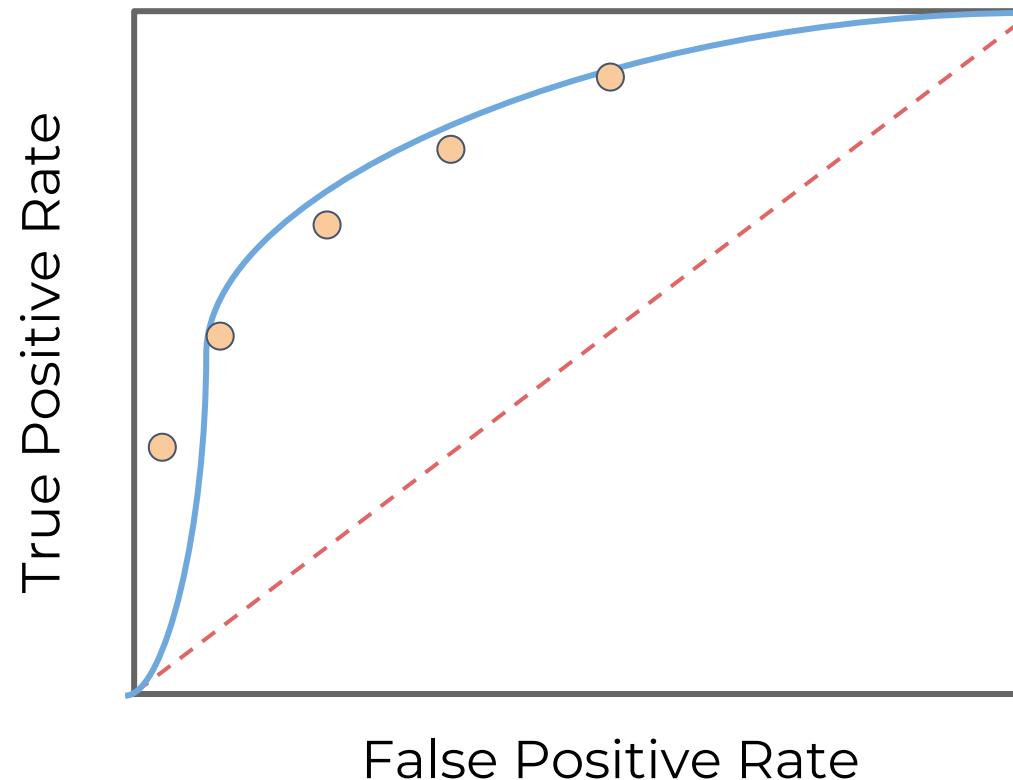
# Classification Metrics

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



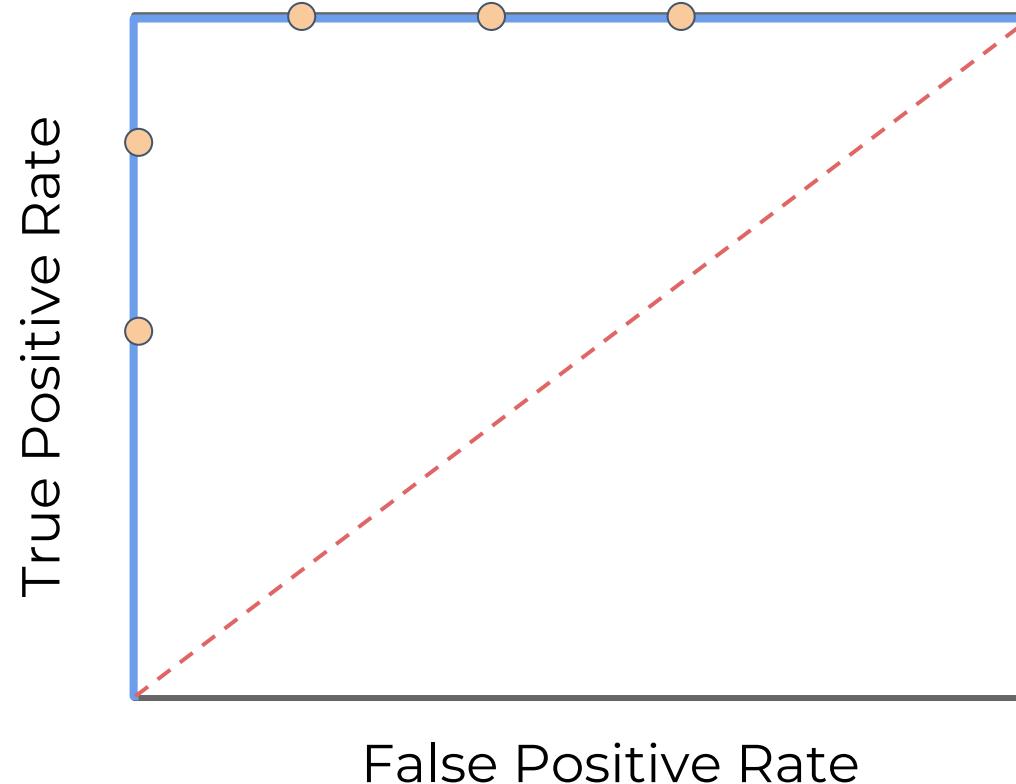
# Classification Metrics

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



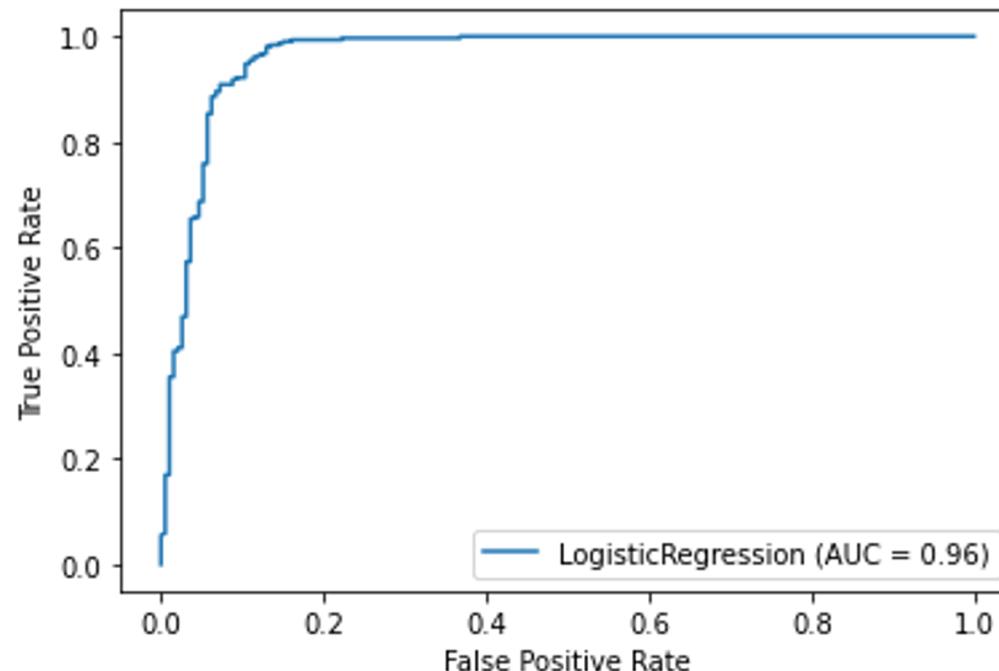
# Classification Metrics

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



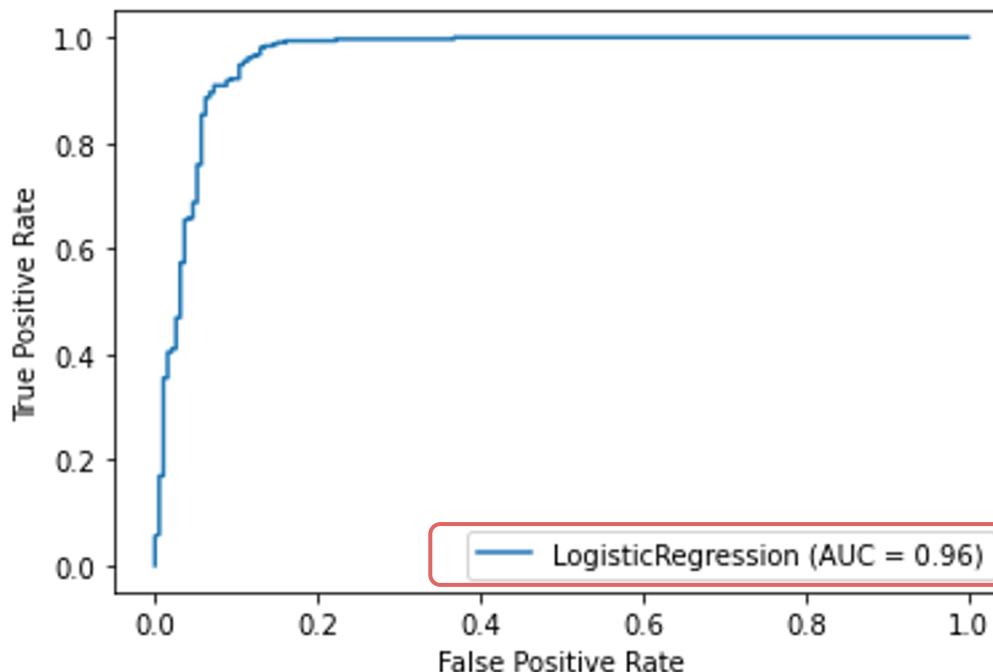
# Classification Metrics

- 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:

