Building Classification Models with scikit-learn

UNDERSTANDING CLASSIFICATION AS A MACHINE LEARNING PROBLEM

Janani Ravi
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Logistic regression for classification

Evaluating classification models

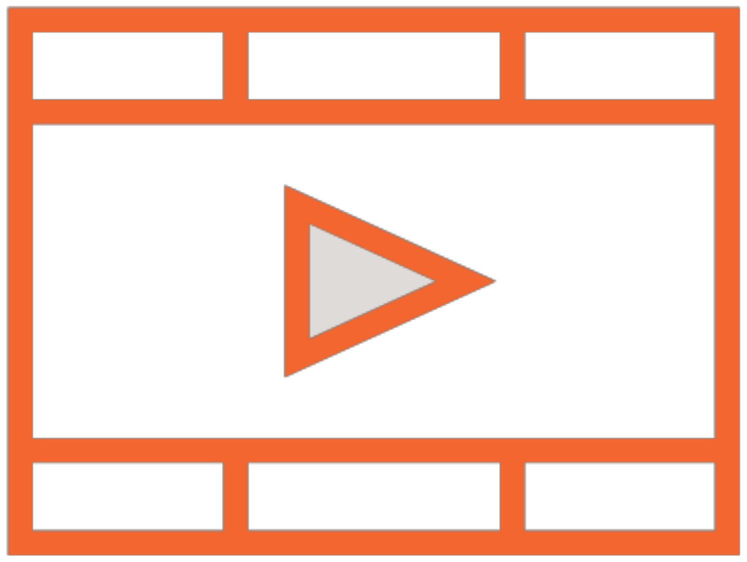
Accuracy, precision, and recall

ROC curves

Binary, multi-label, and multi-class classification

Prerequisites and Course Outline

Prerequisites



Basic Python programming

Basic understanding of the ML workflow

High school math

Prerequisite Courses

Building Your First scikit-learn Solution

Course Outline

Understanding the classification problem

Building a simple ML classifier

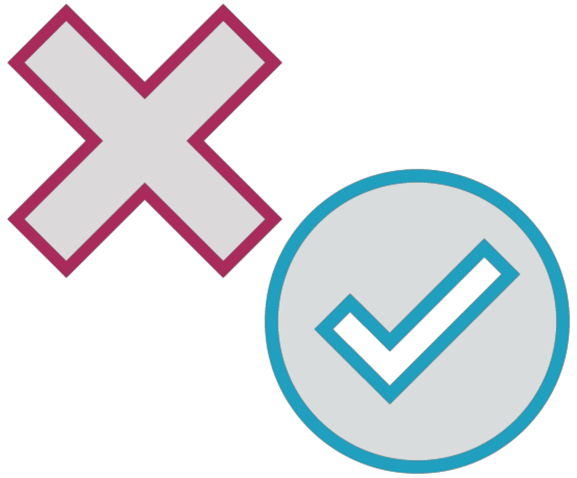
Choosing and implementing classification technique

Hyperparameter tuning for classification

Classifying images

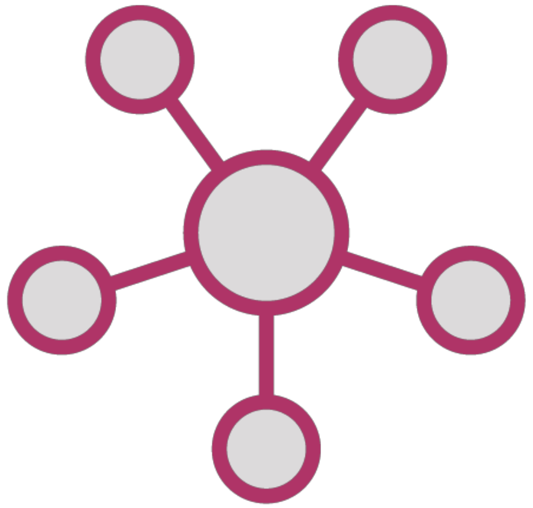
Classification and Classifiers

Types of Machine Learning Problems

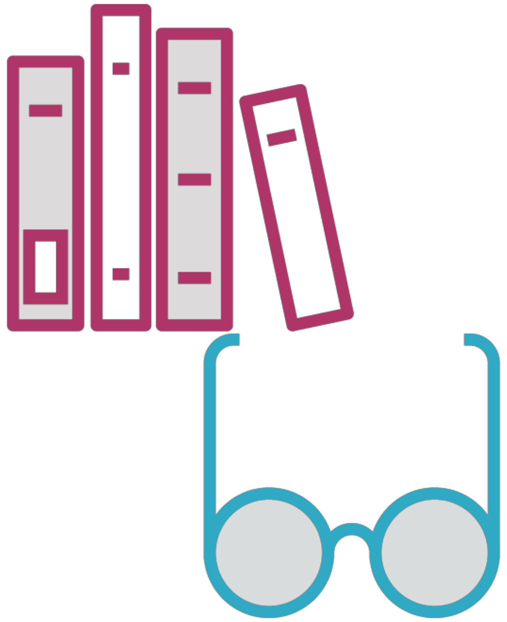


Classification

Regression

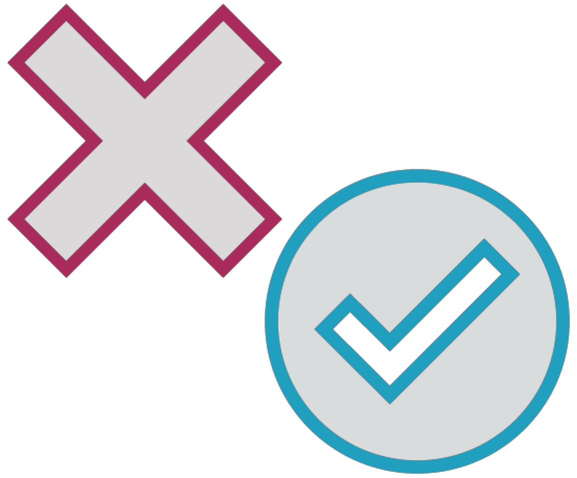


Clustering



Dimensionality reduction

Types of Machine Learning Problems



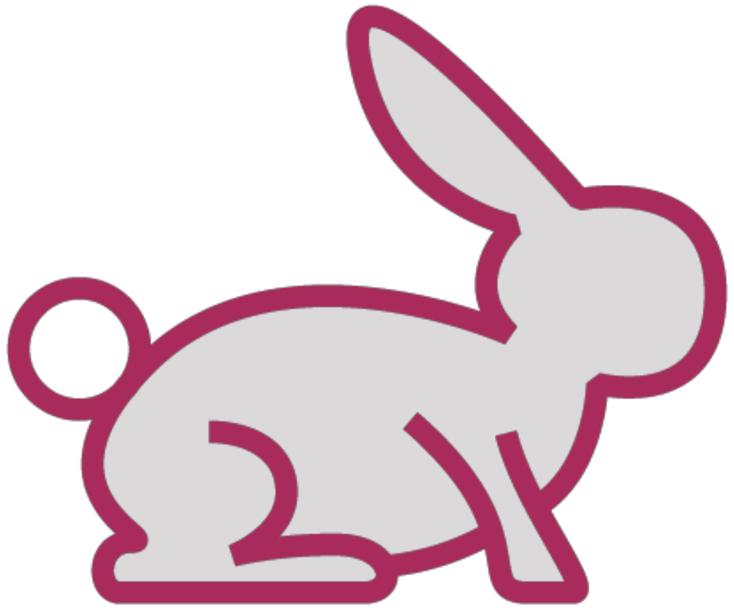
Classification

Regression

Clustering

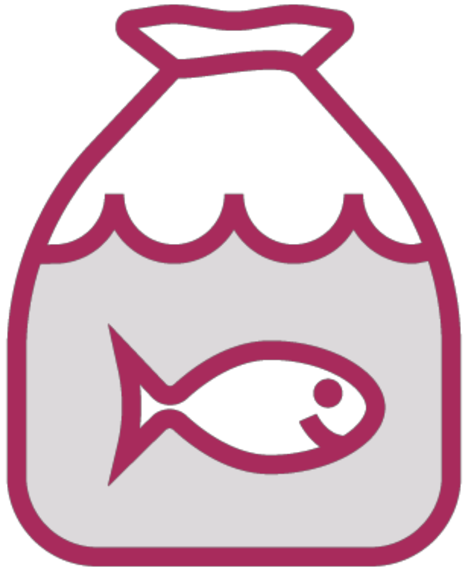
Dimensionality reduction

Whales: Fish or Mammals?



Mammals

Members of the infraorder *Cetacea*



Fish

Look like fish, swim like fish, move with fish

Whales: Fish or Mammals?

ML-based Classifier

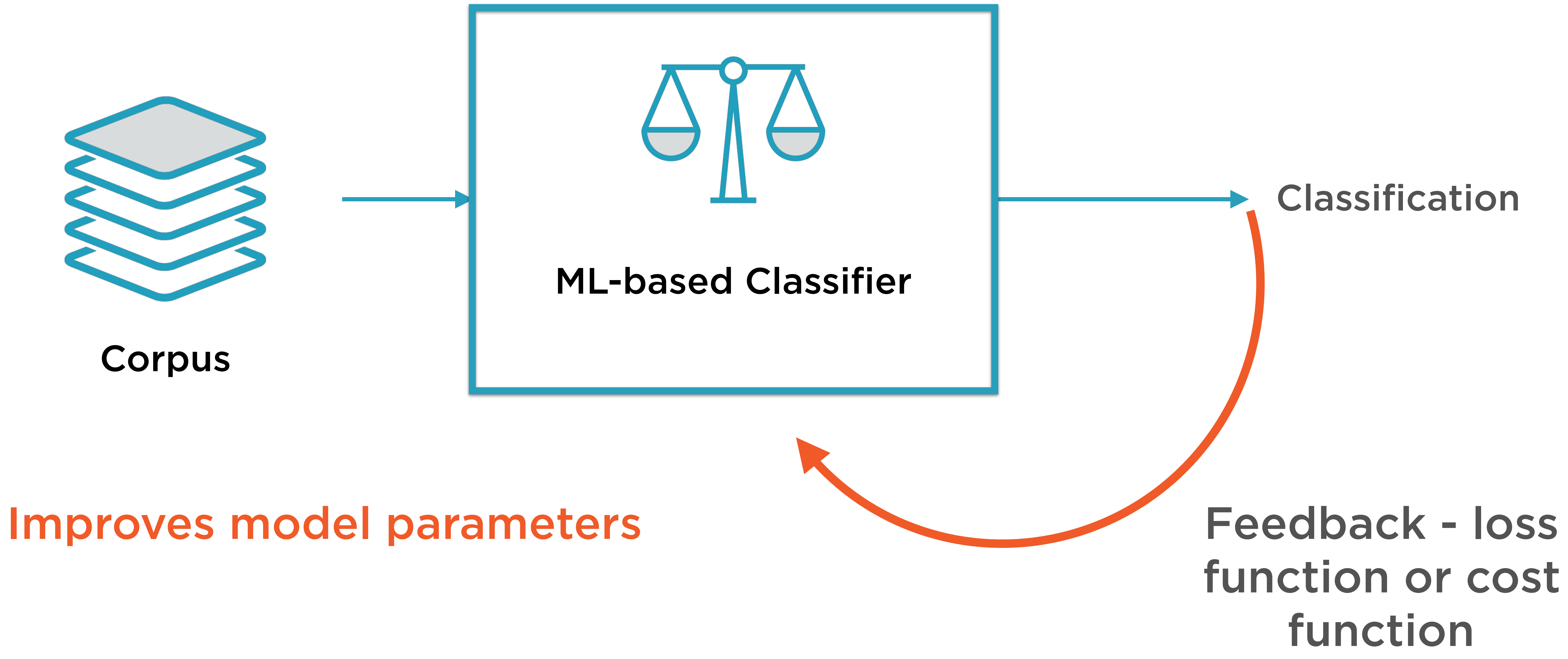
Training

Feed in a large corpus of data classified correctly

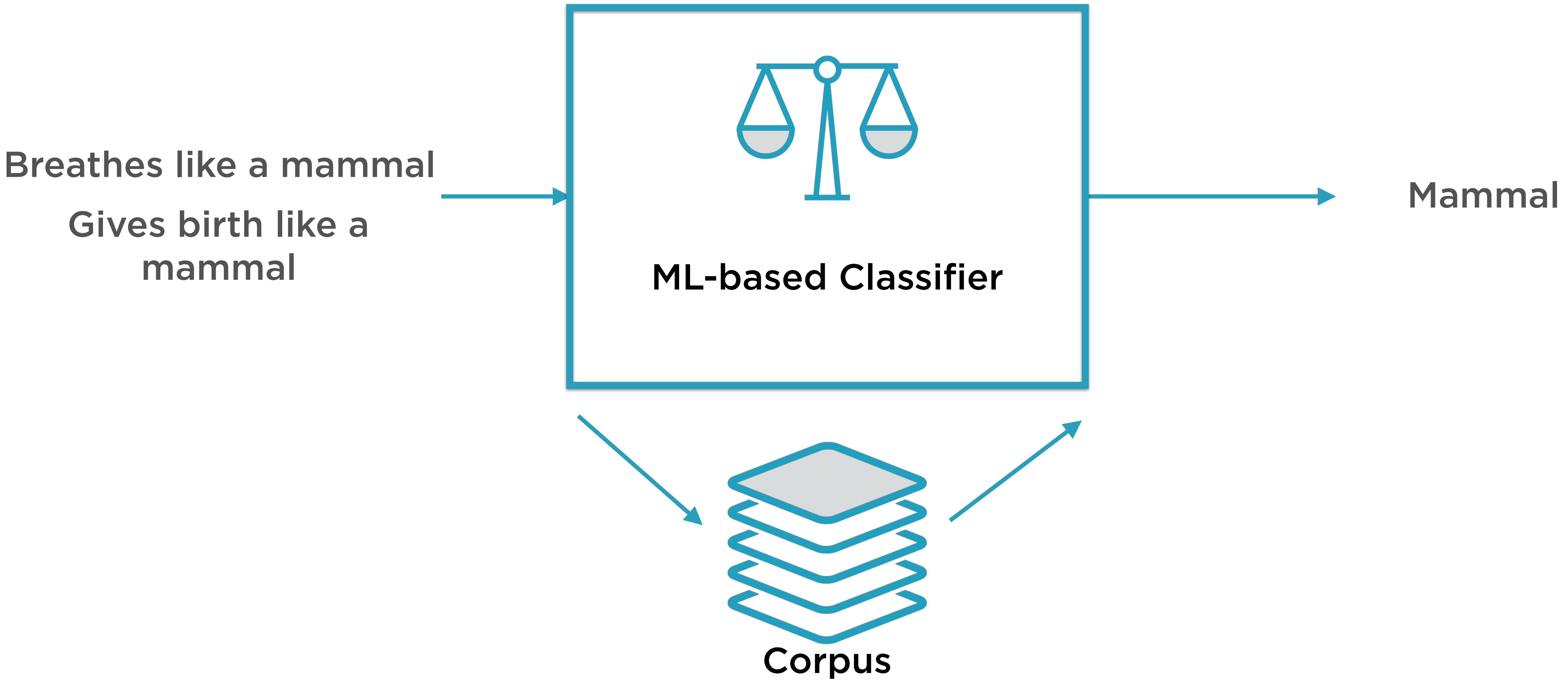
Prediction

Use it to classify new instances which it has not seen before

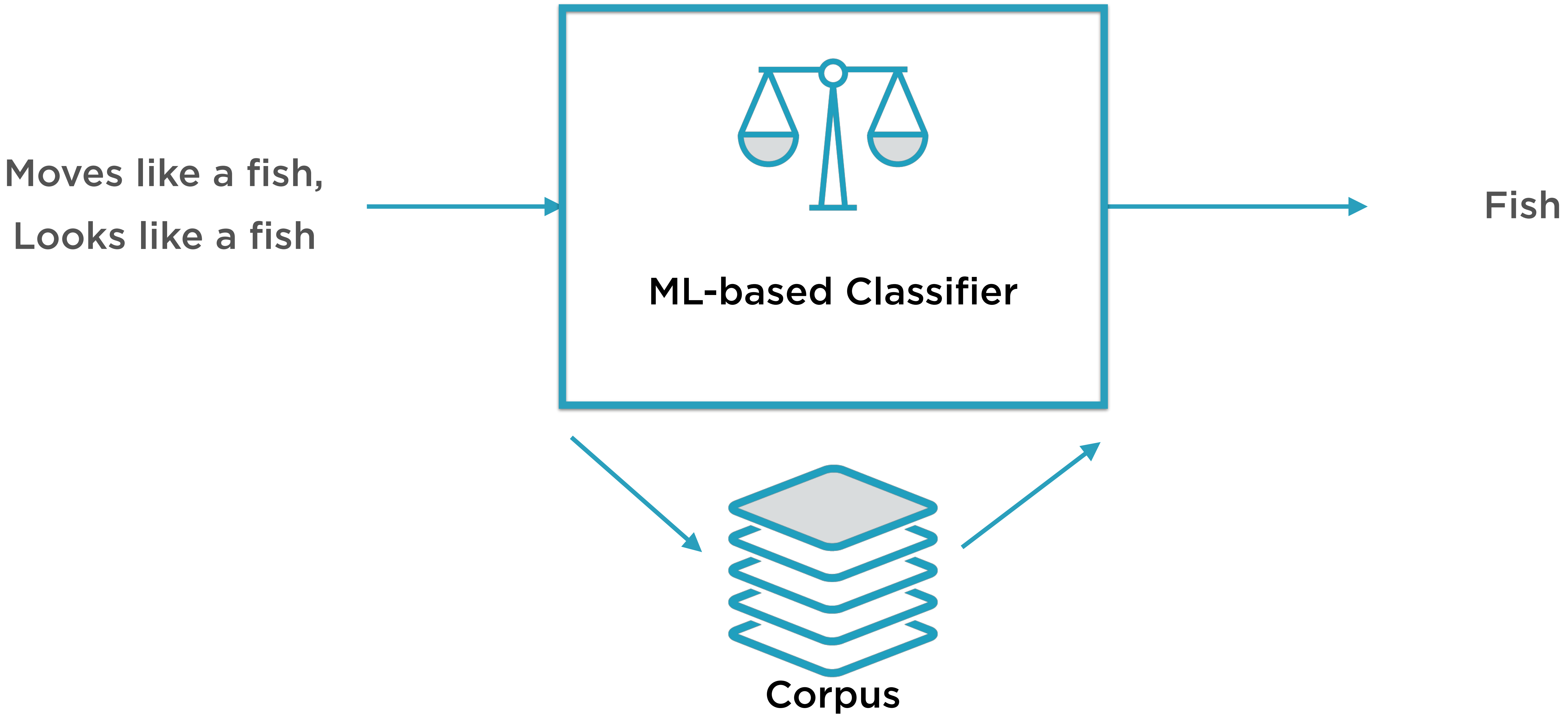
Training the ML-based Classifier



ML-based Binary Classifier



“Traditional” ML-based Binary Classifier



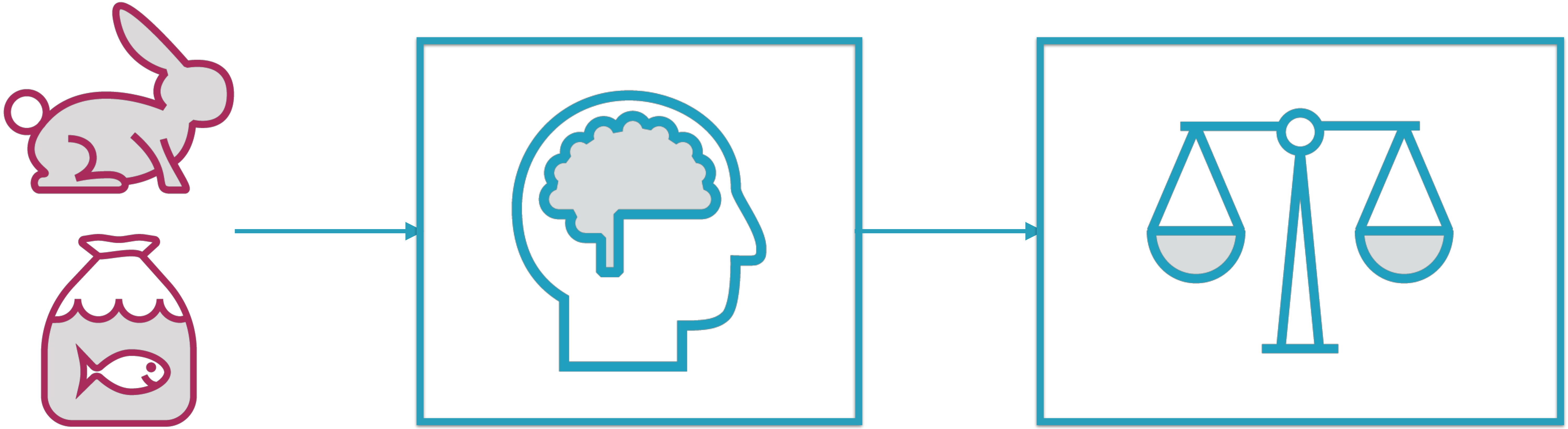
ML-based Binary Classifier

Corpus

Classification Algorithm

ML-based Classifier

ML-based Binary Classifier

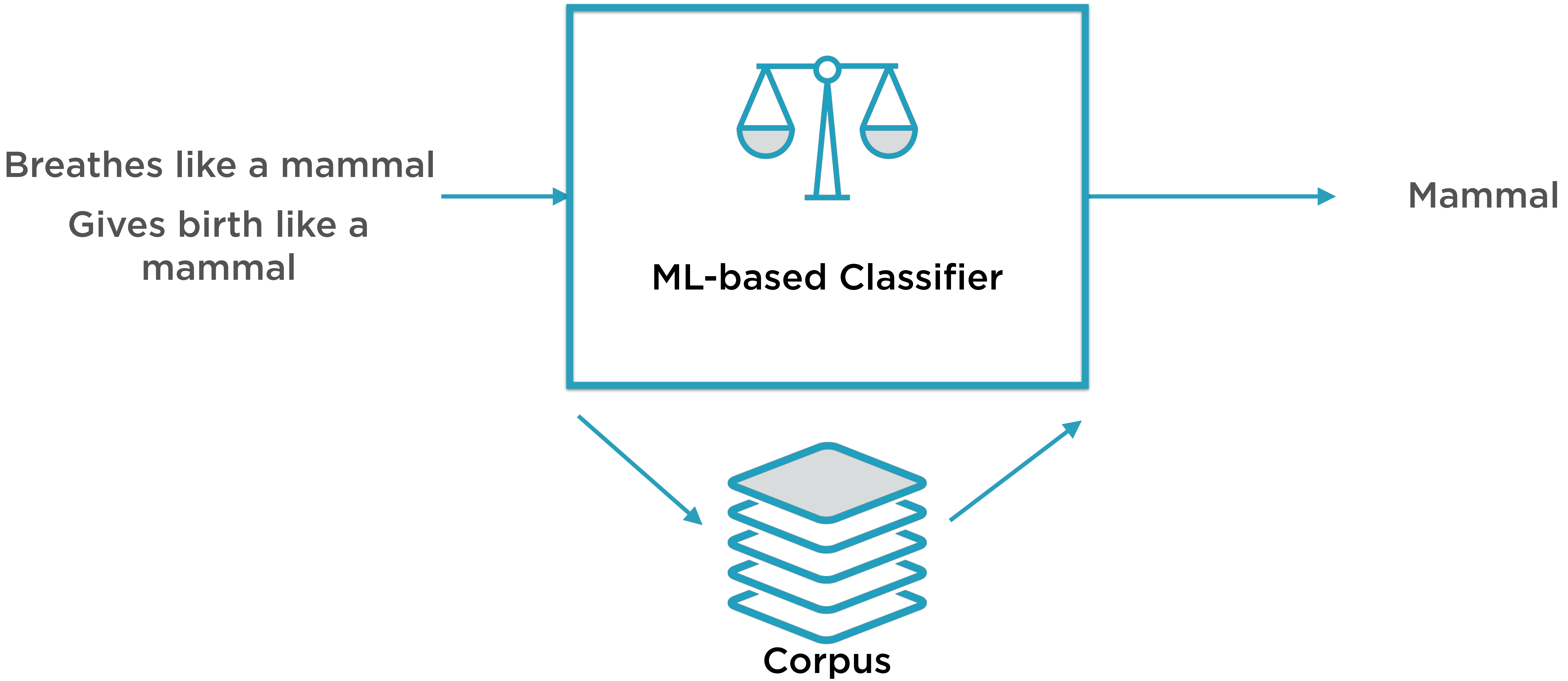


Corpus

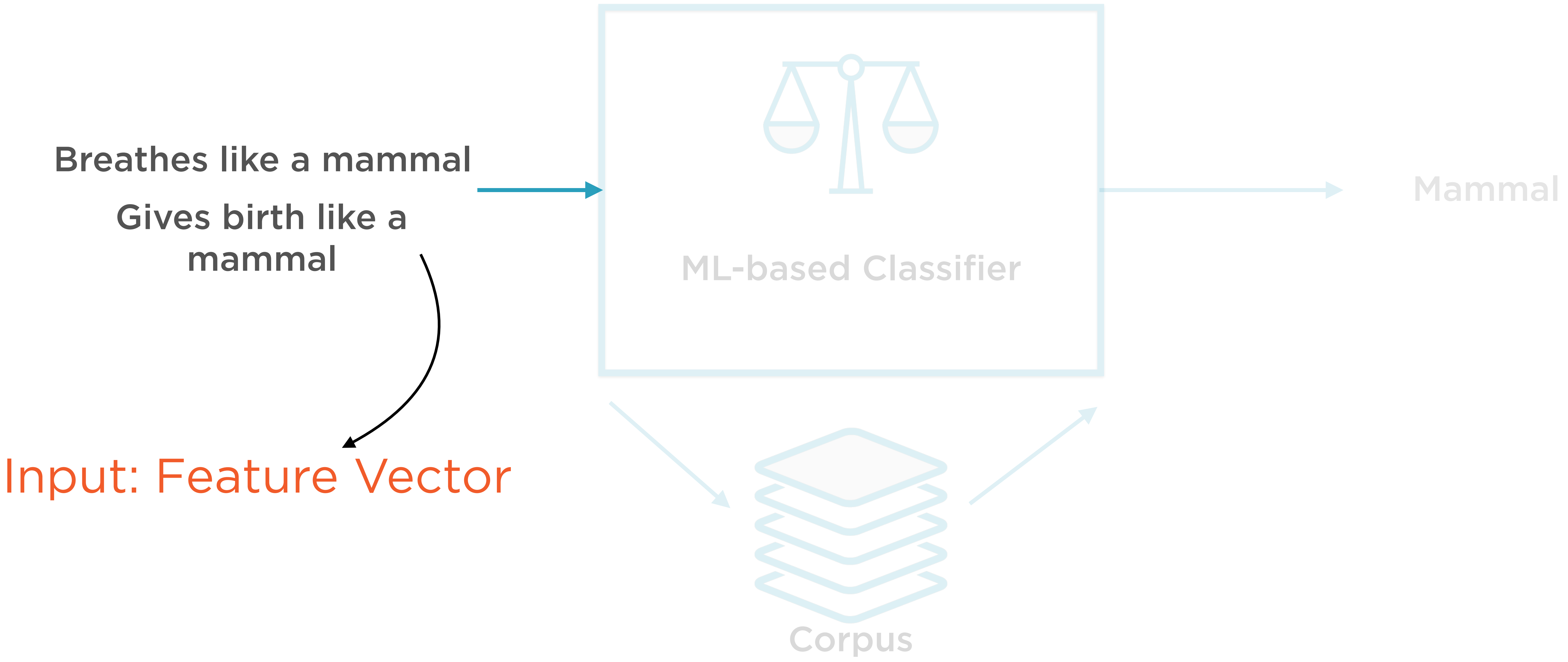
Naive Bayes, Support Vector Machines, Decision Trees

ML-based Classifier

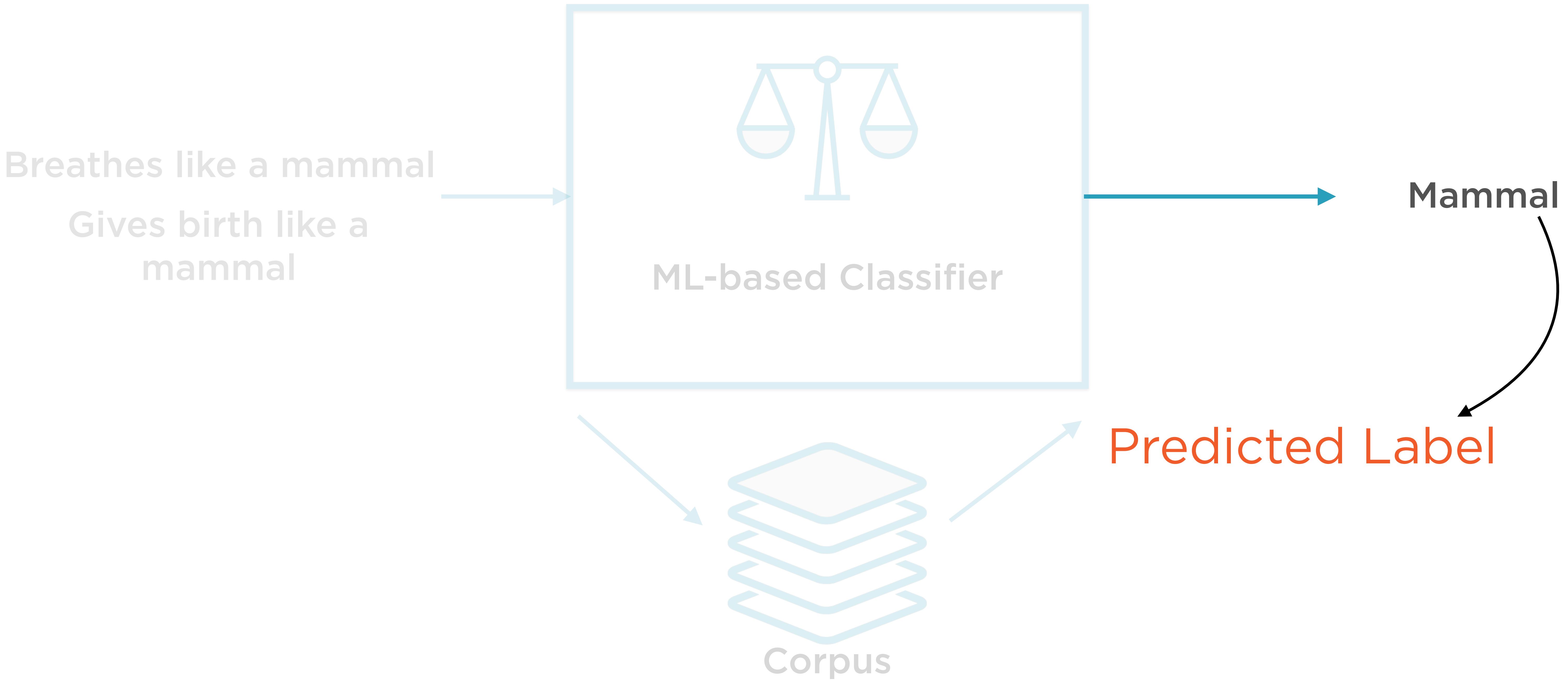
ML-based Binary Classifier



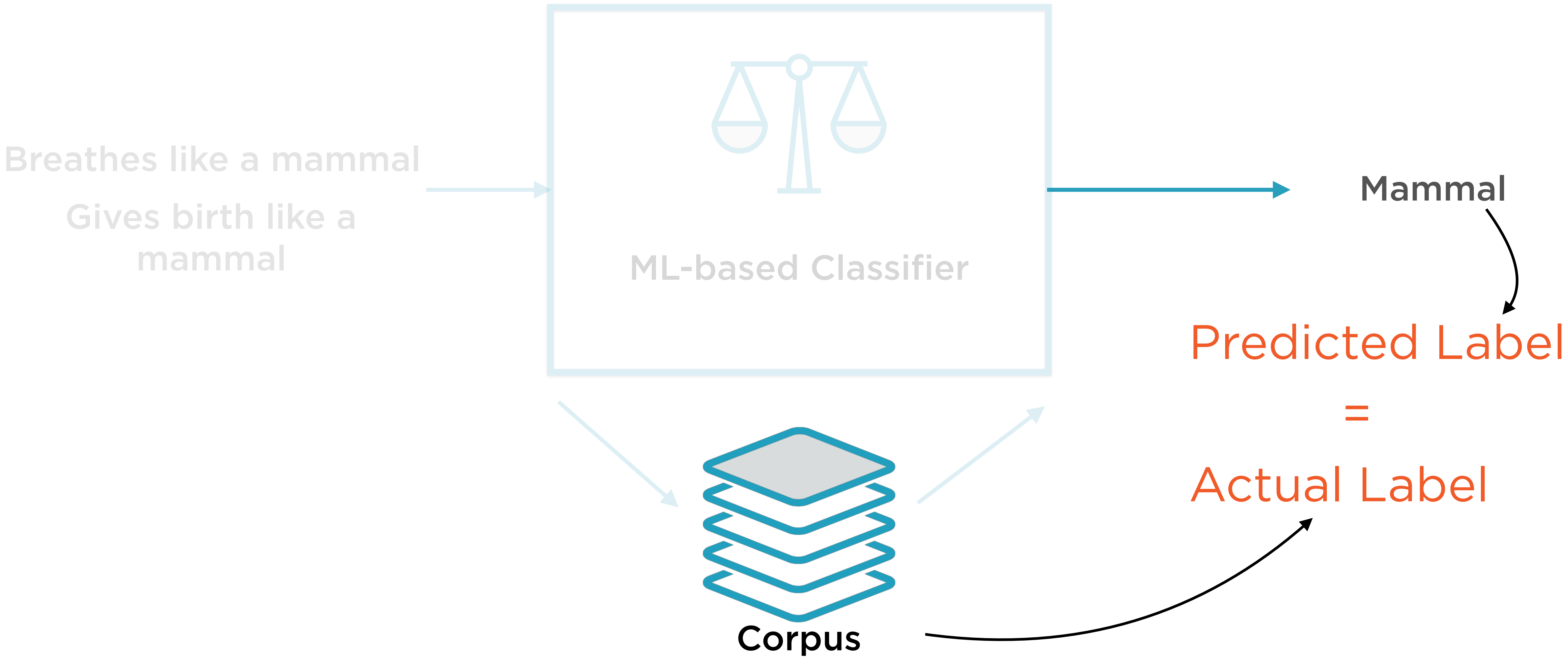
ML-based Binary Classifier



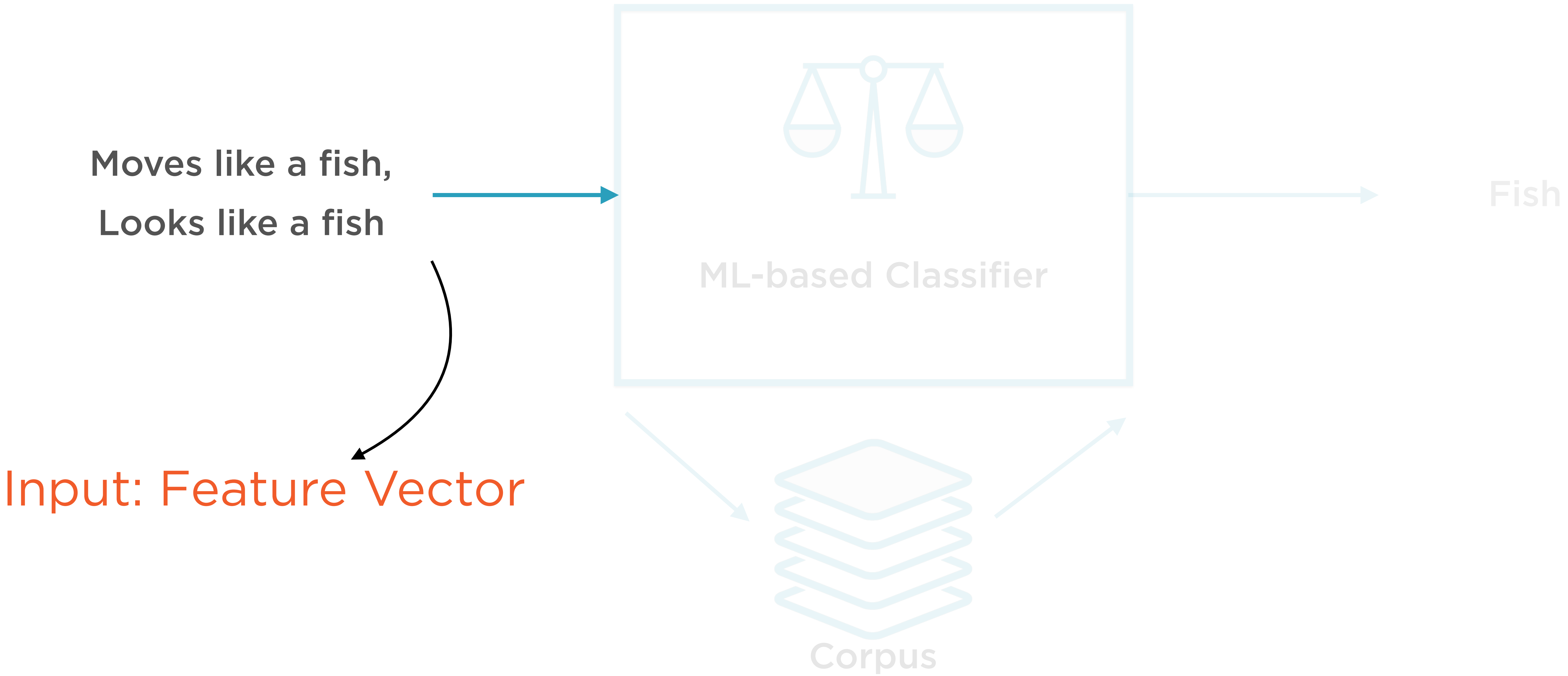
ML-based Binary Classifier



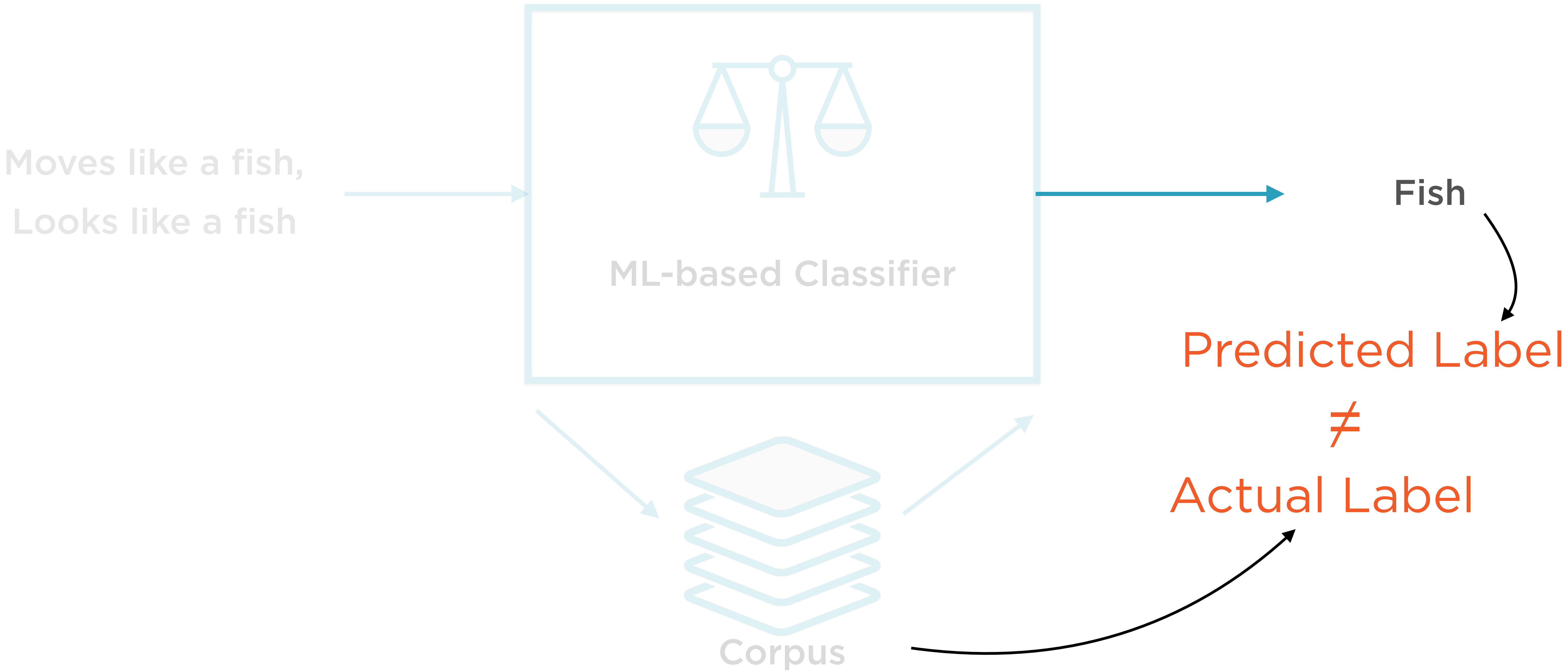
ML-based Binary Classifier



ML-based Binary Classifier



ML-based Binary Classifier



Logistic Regression: Intuition

Two Approaches to Deadlines

Start 5 minutes before deadline

Good luck with that

Start 1 year before deadline

Maybe overkill

Neither approach is optimal

Starting a Year in Advance

Probability of meeting the deadline

100%

Probability of getting other important work done

0%

Starting Five Minutes in Advance

Probability of meeting the deadline

0%

Probability of getting other important work done

100%

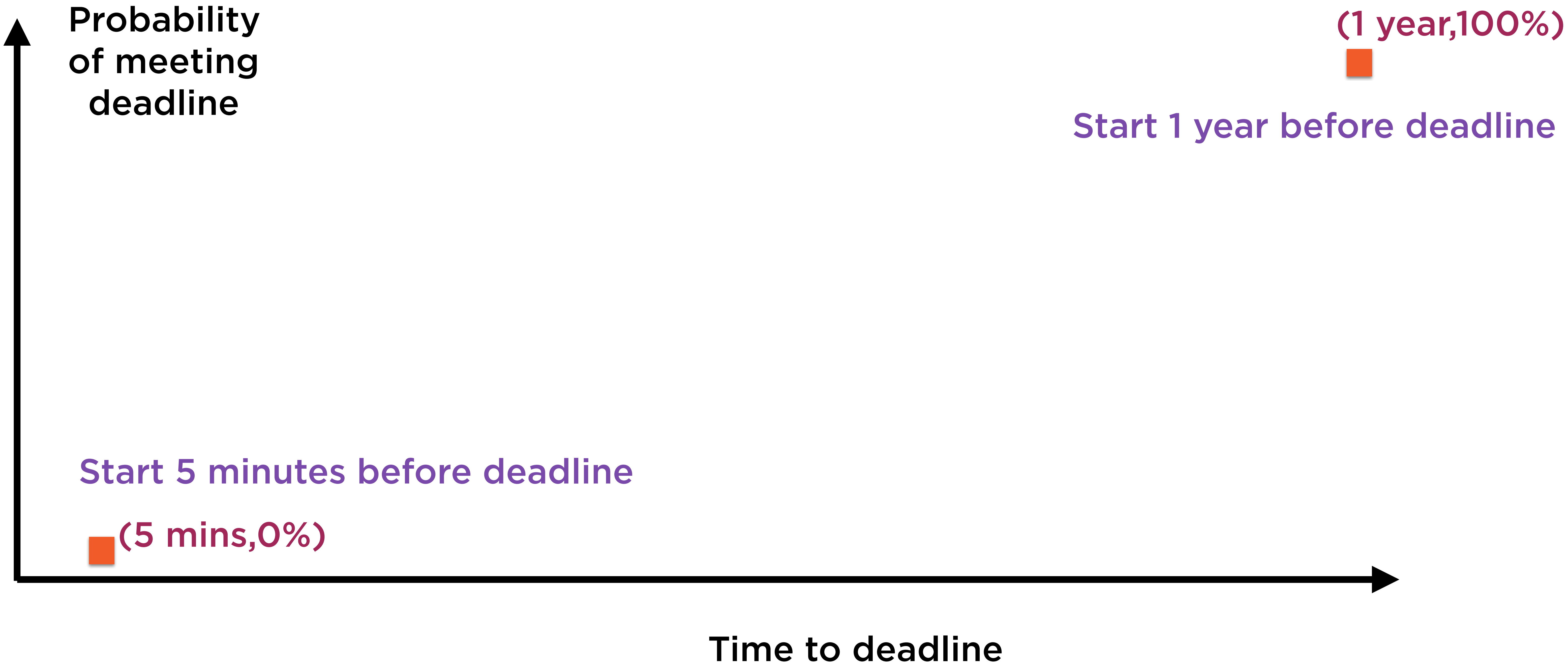
The Goldilocks Solution

Work fast	Work smart	Work hard
Start very late and hope for the best	Start as late as possible to be sure to make it	Start very early and do little else

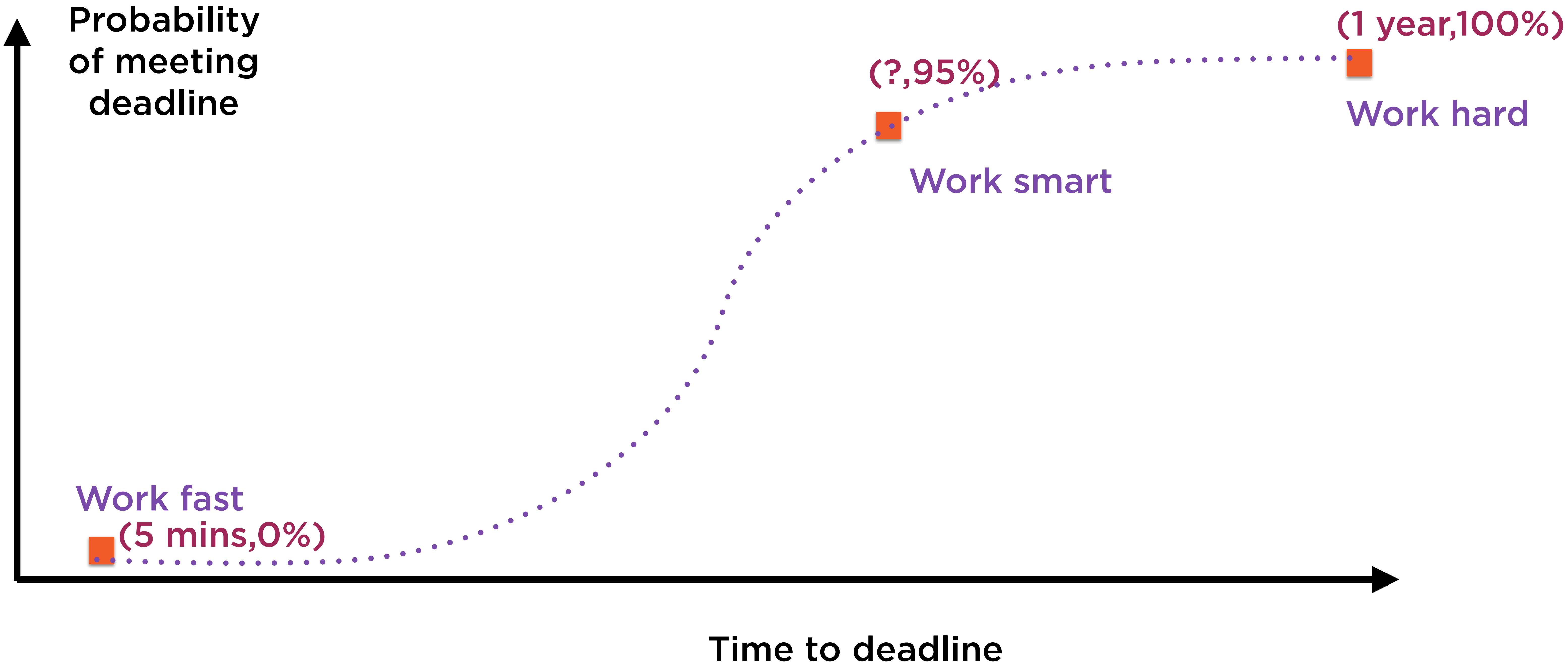
As usual, the middle path is best

Working Smart

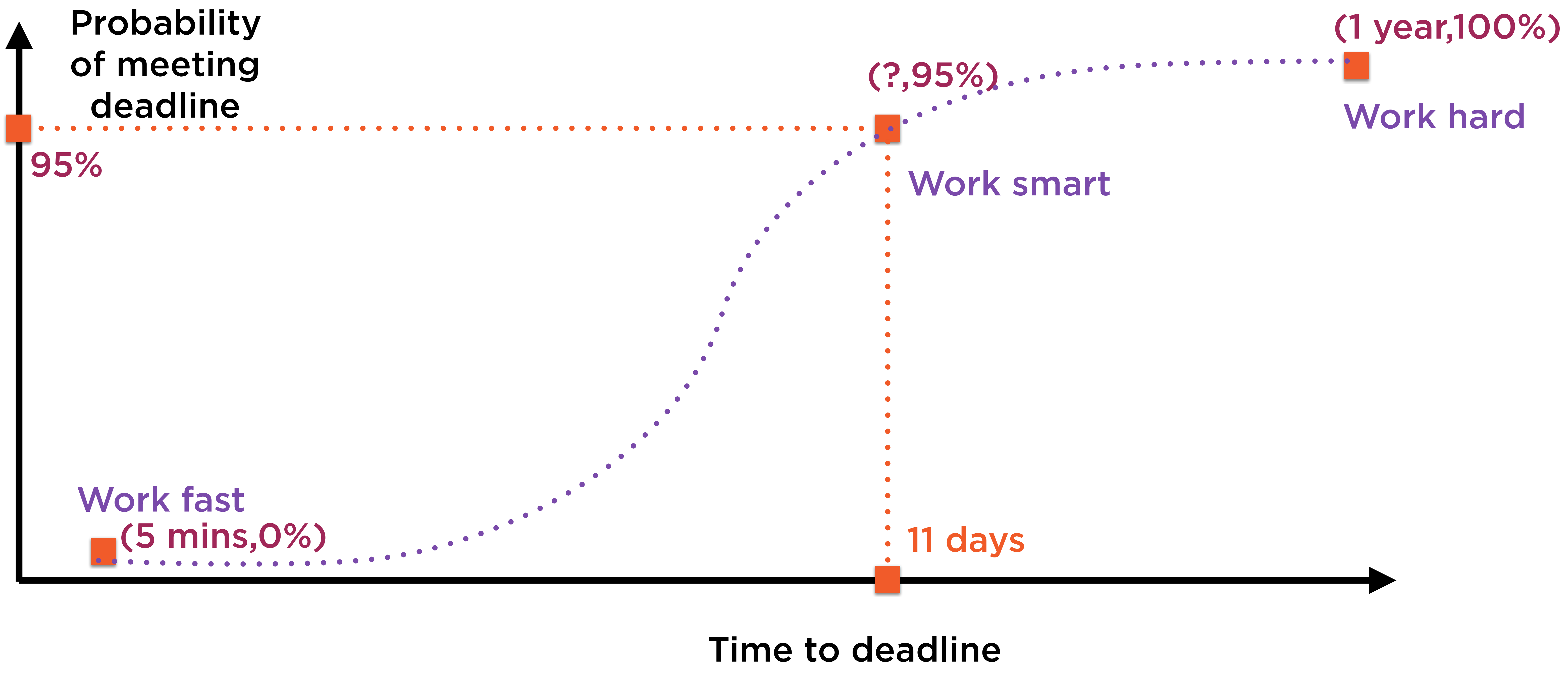
Working Hard, Fast, Smart



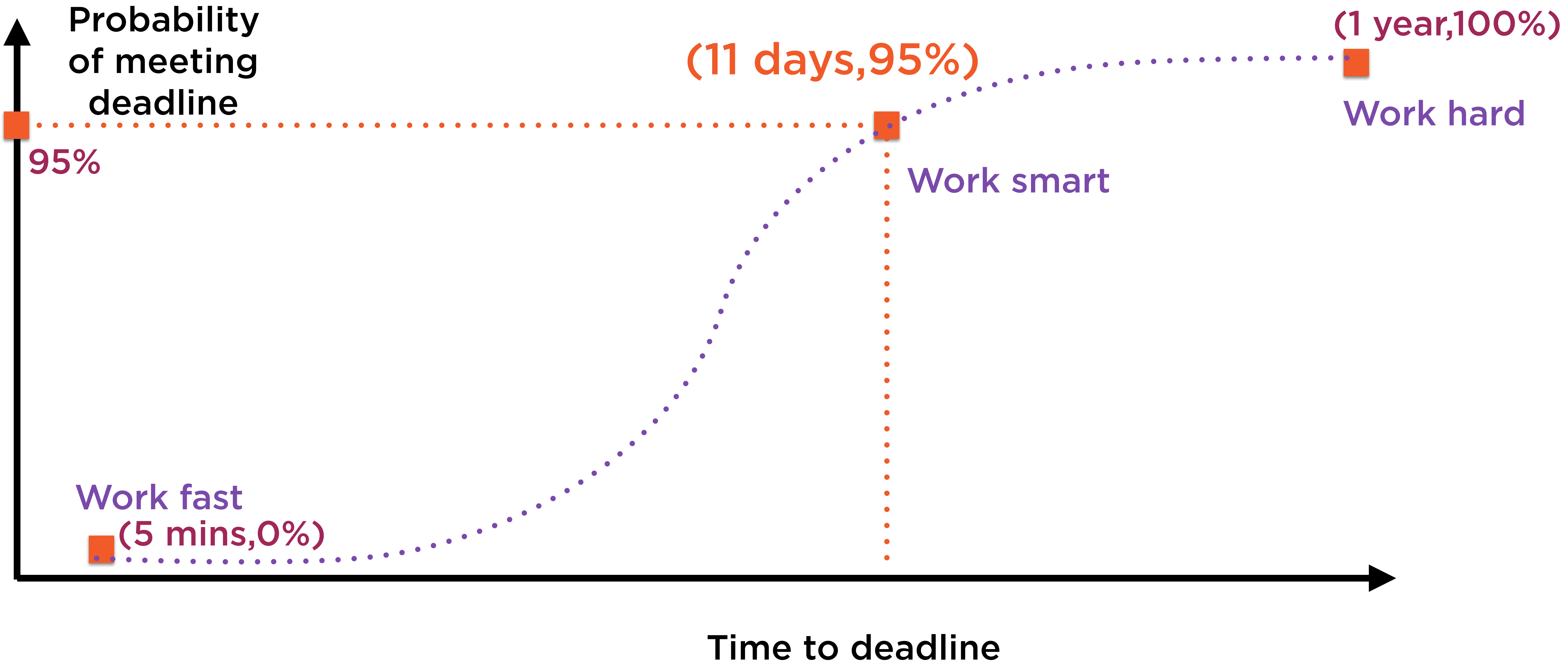
Working Hard, Fast, Smart



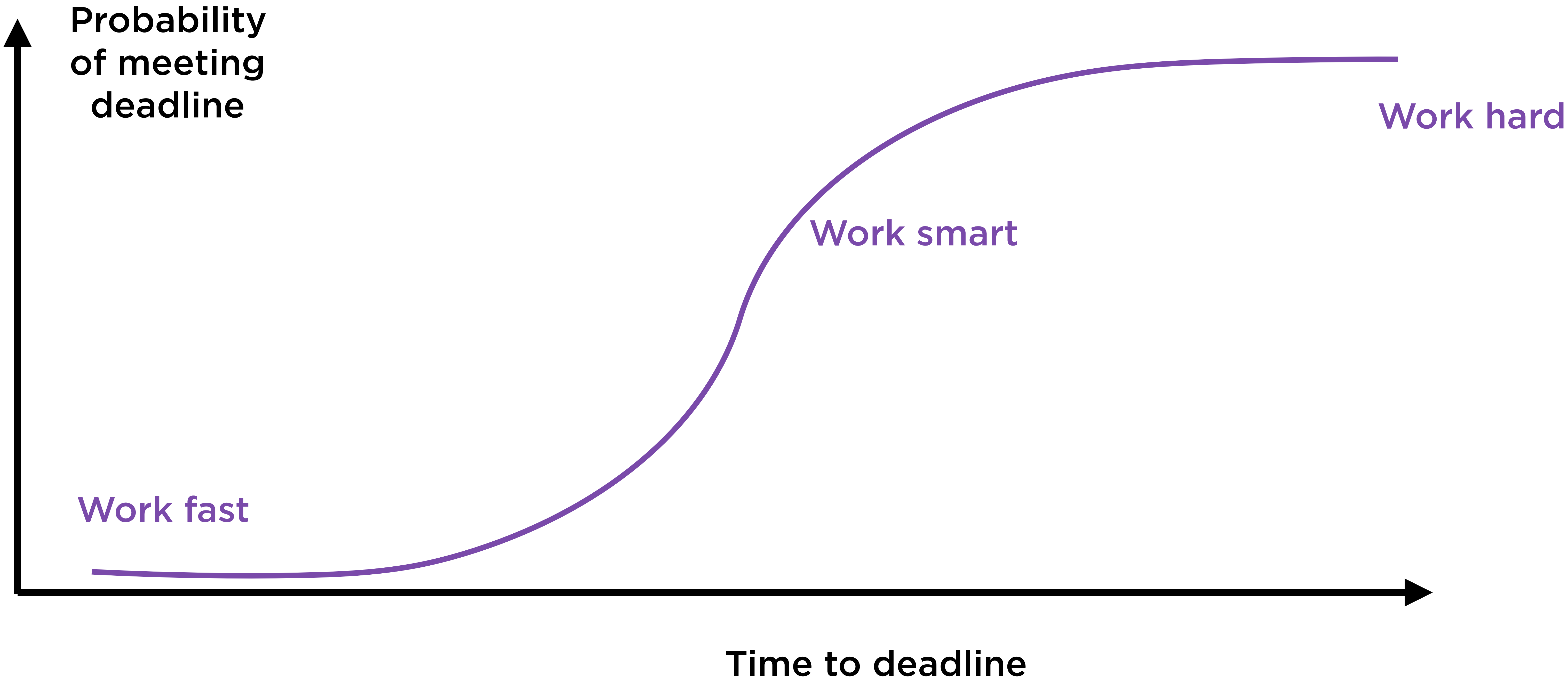
Working Hard, Fast, Smart



Working Hard, Fast, Smart

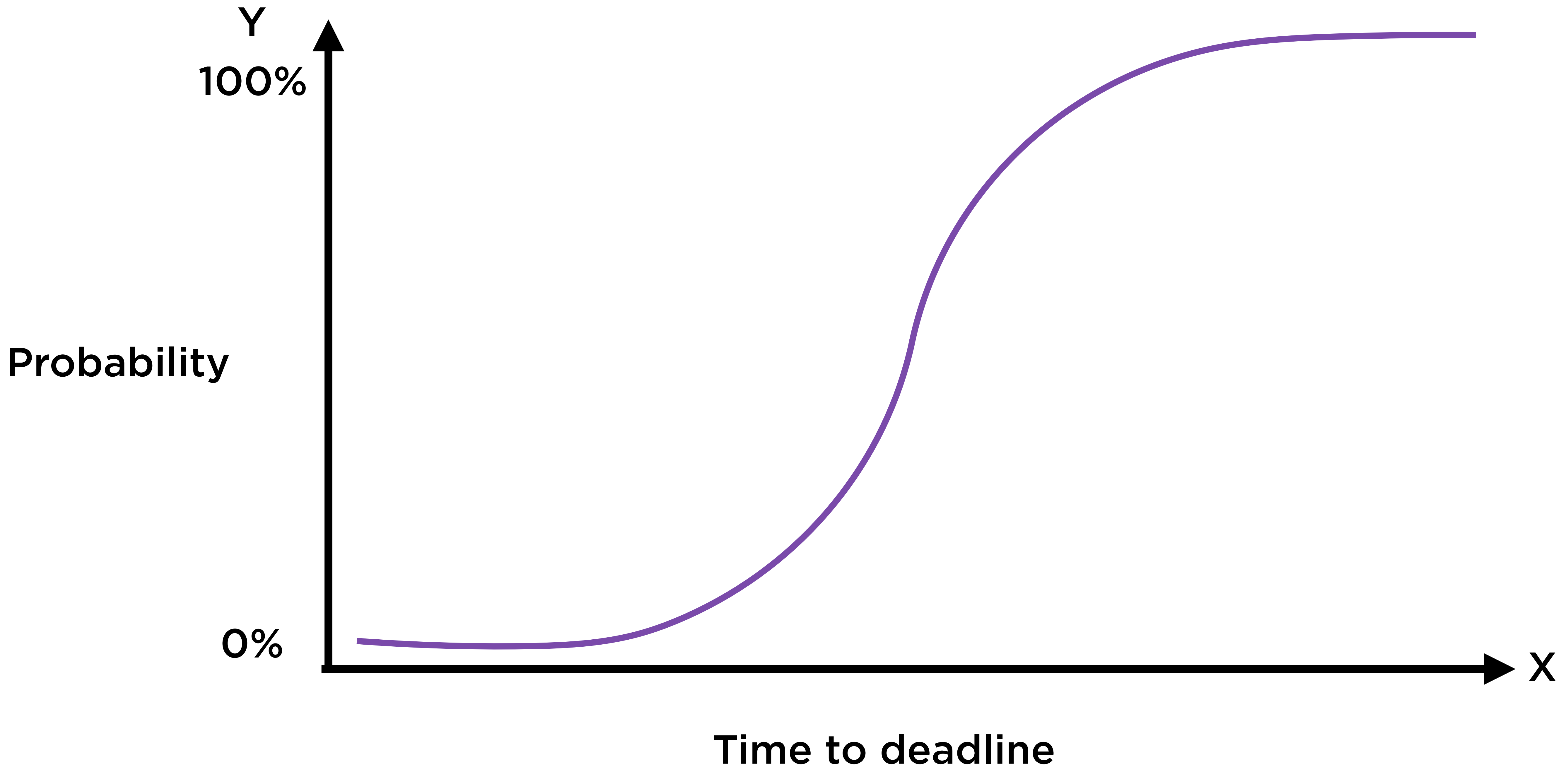


Working Hard, Fast, Smart

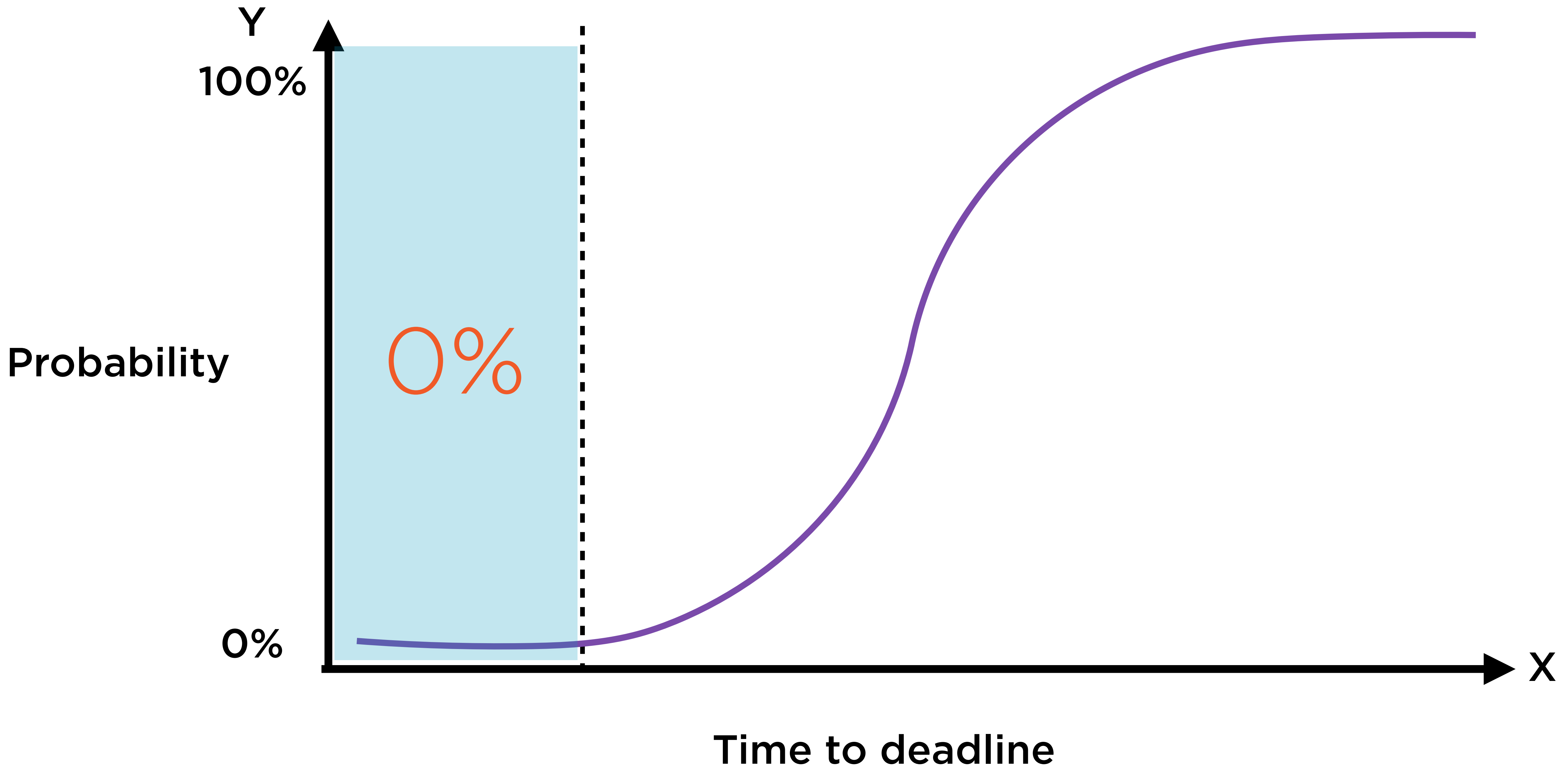


Logistic Regression helps find how probabilities are changed by actions

Working Smart with Logistic Regression

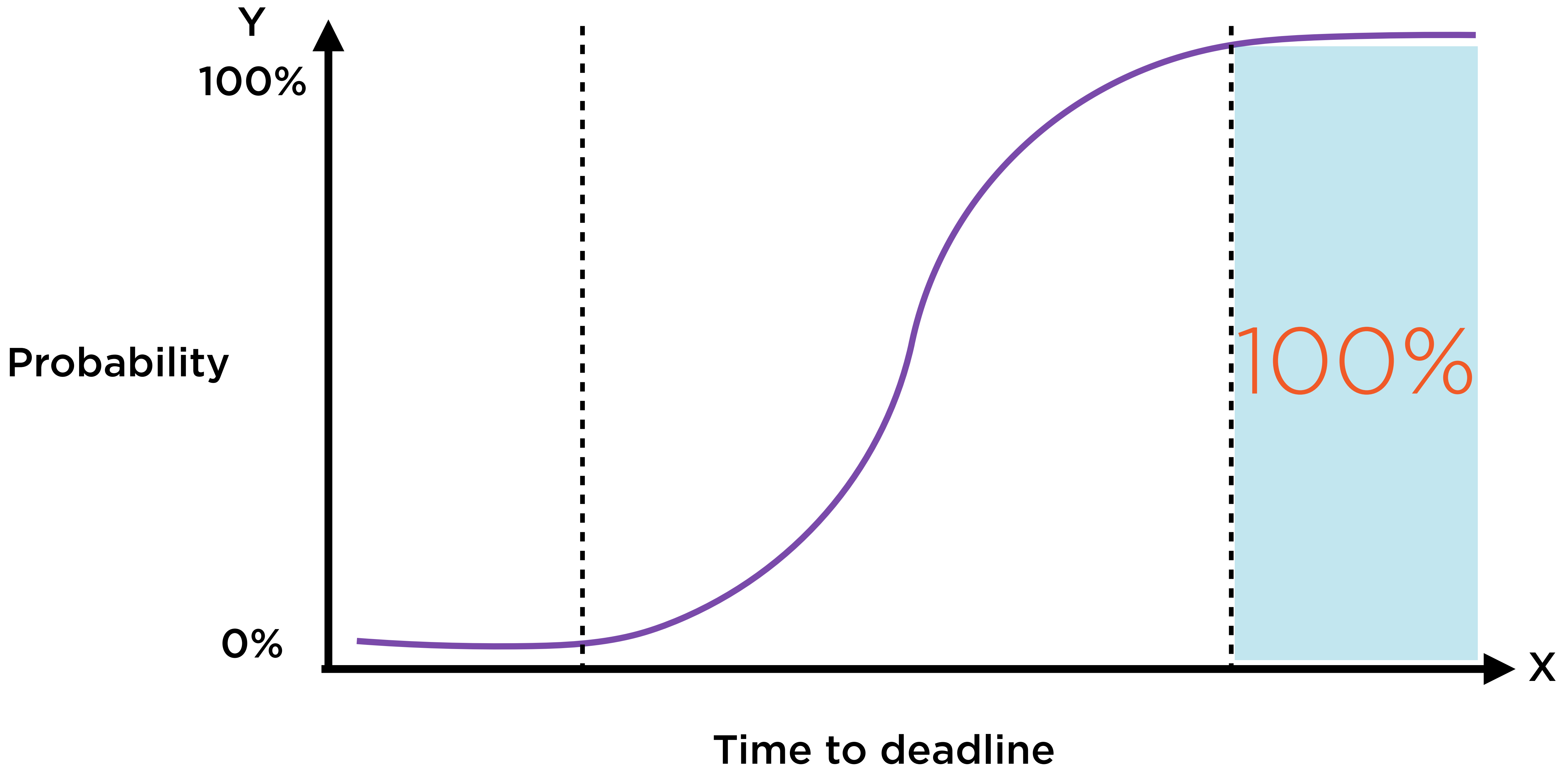


Working Smart with Logistic Regression



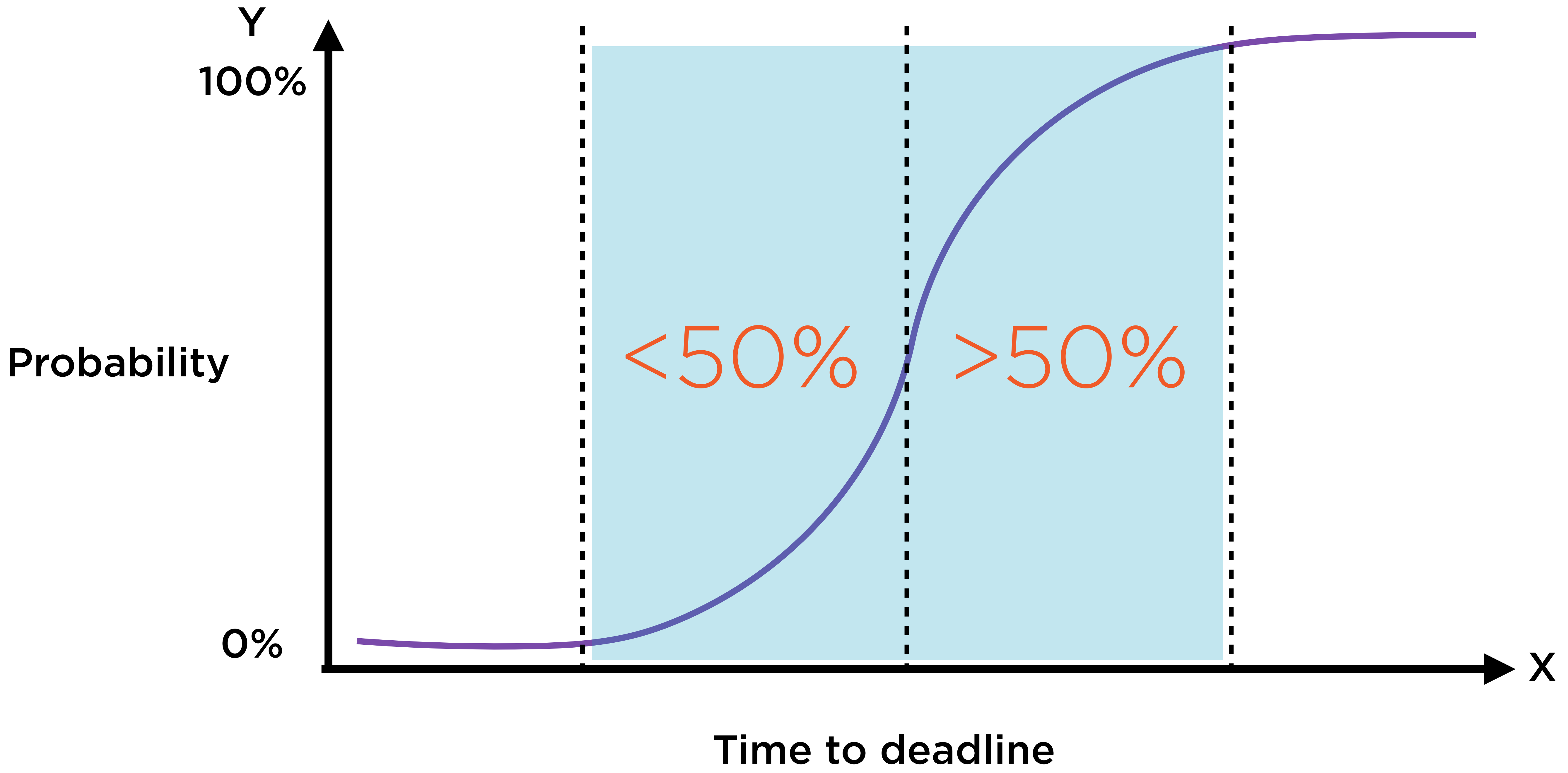
Start too late, and you'll definitely miss

Working Smart with Logistic Regression



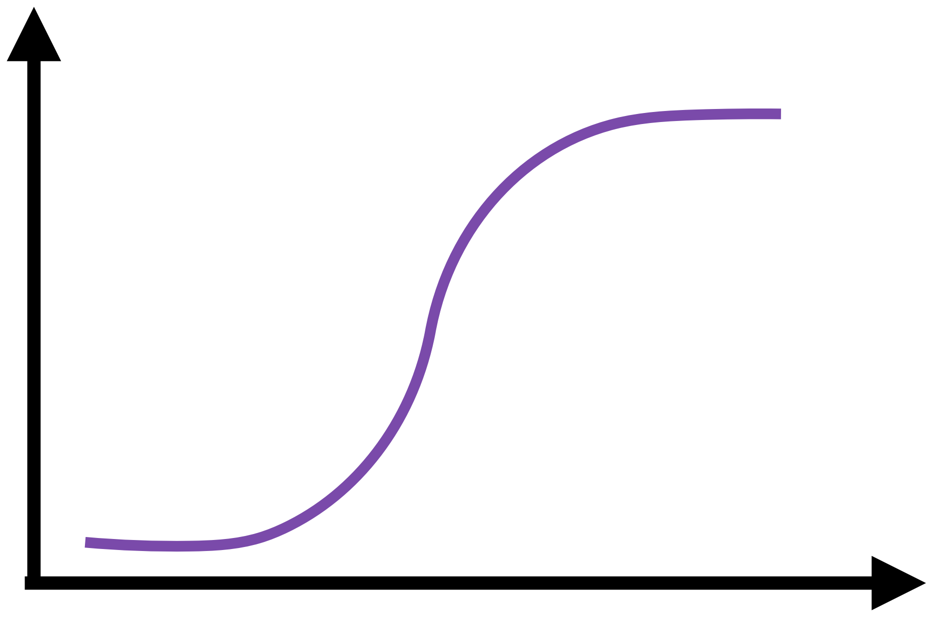
Start too early, and you'll definitely make it

Working Smart with Logistic Regression



Working smart is knowing when to start

Logistic Regression S-curves



y: hit or miss? (0 or 1?)

x: start time before deadline

p(y) : probability of y = 1

Logistic Regression S-curves

$$p(y_i) = \frac{1}{1 + e^{-(A+Bx_i)}}$$

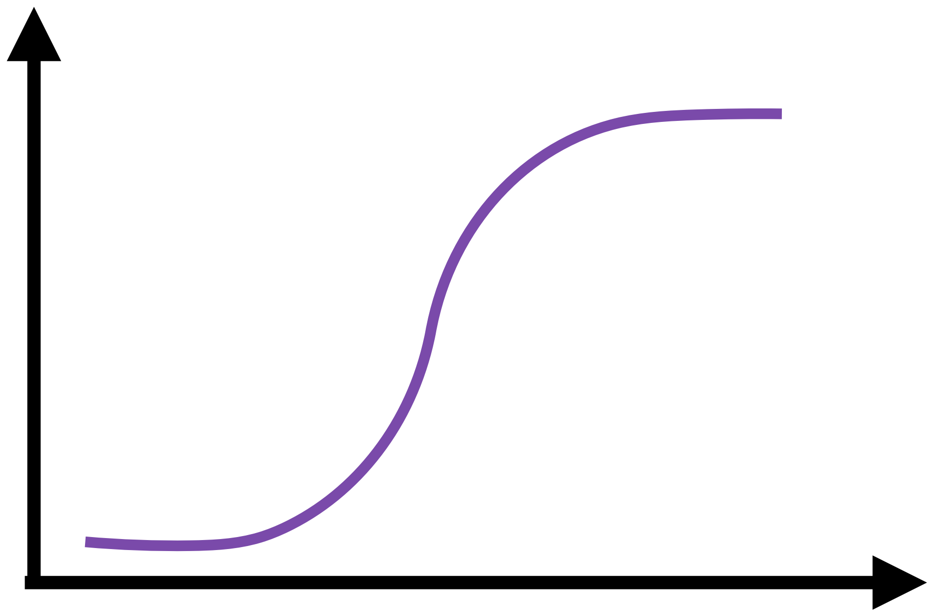
Logistic regression involves finding the "best fit" such curve

- A is the intercept

- B is the regression coefficient

(e is the constant 2.71828)

Logistic Regression S-curves

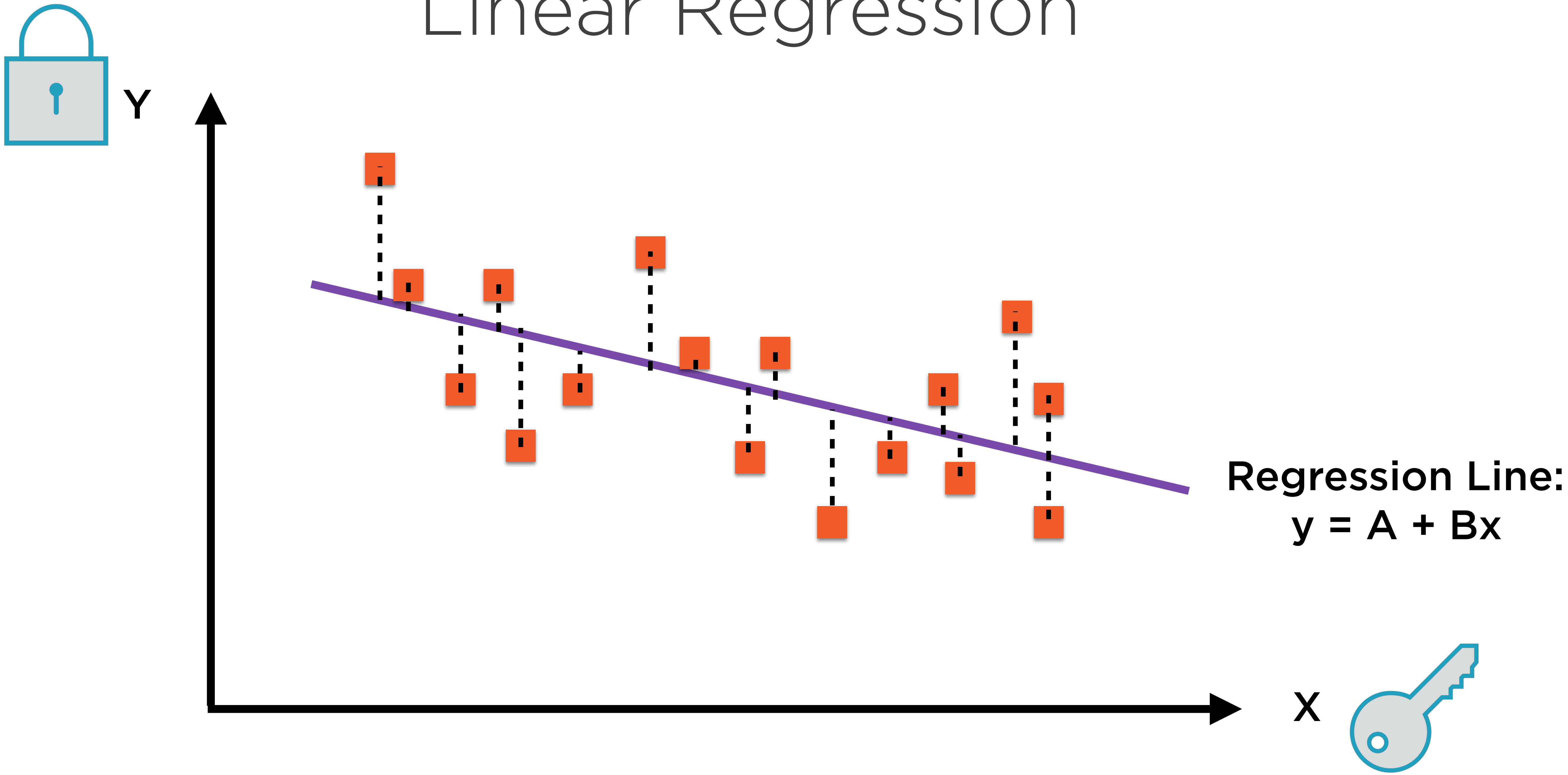


S-curves are widely studied, well understood

Logistic regression uses S-curve to estimate probabilities

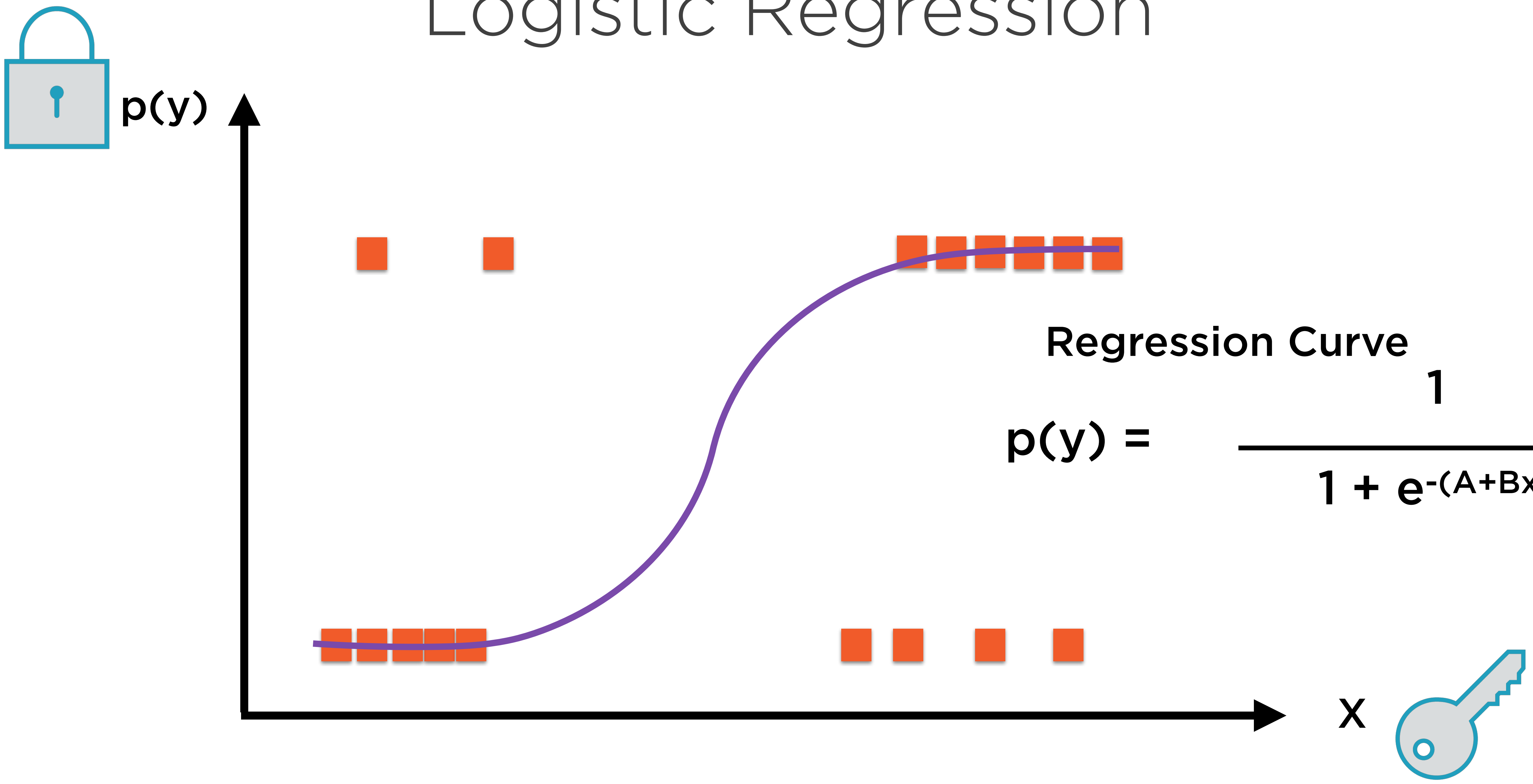
$$p(y) = \frac{1}{1 + e^{-(A+Bx)}}$$

Linear Regression



Finding the best fit line through these points

Logistic Regression



$$p(y) = \frac{1}{1 + e^{-(A+Bx)}}$$

Finding the best fit S-curve through these points

Logistic Regression

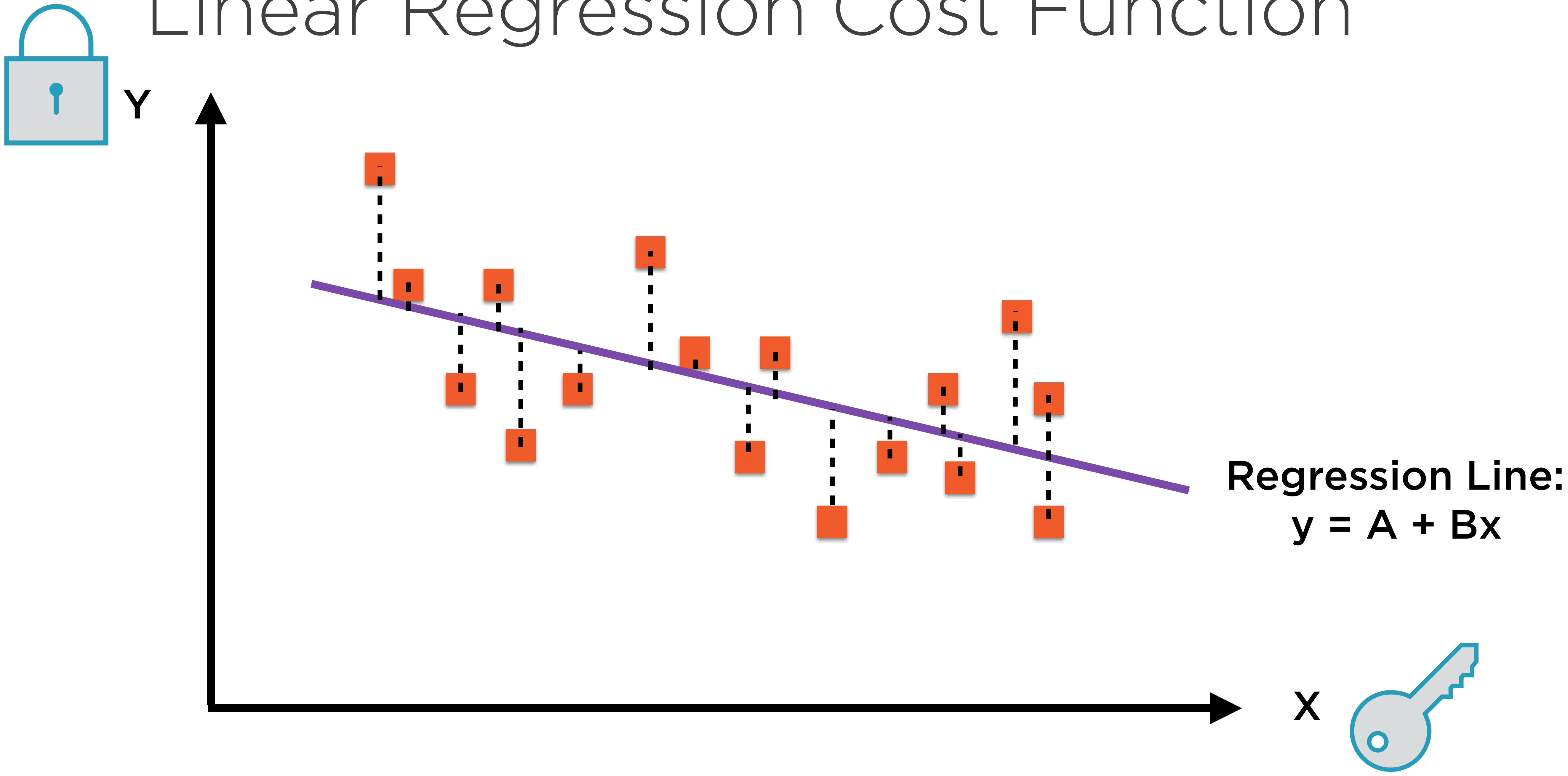
Regression Equation:

$$p(y_i) = \frac{1}{1 + e^{-(A+Bx_i)}}$$

Solve for A and B that “best fit” the data

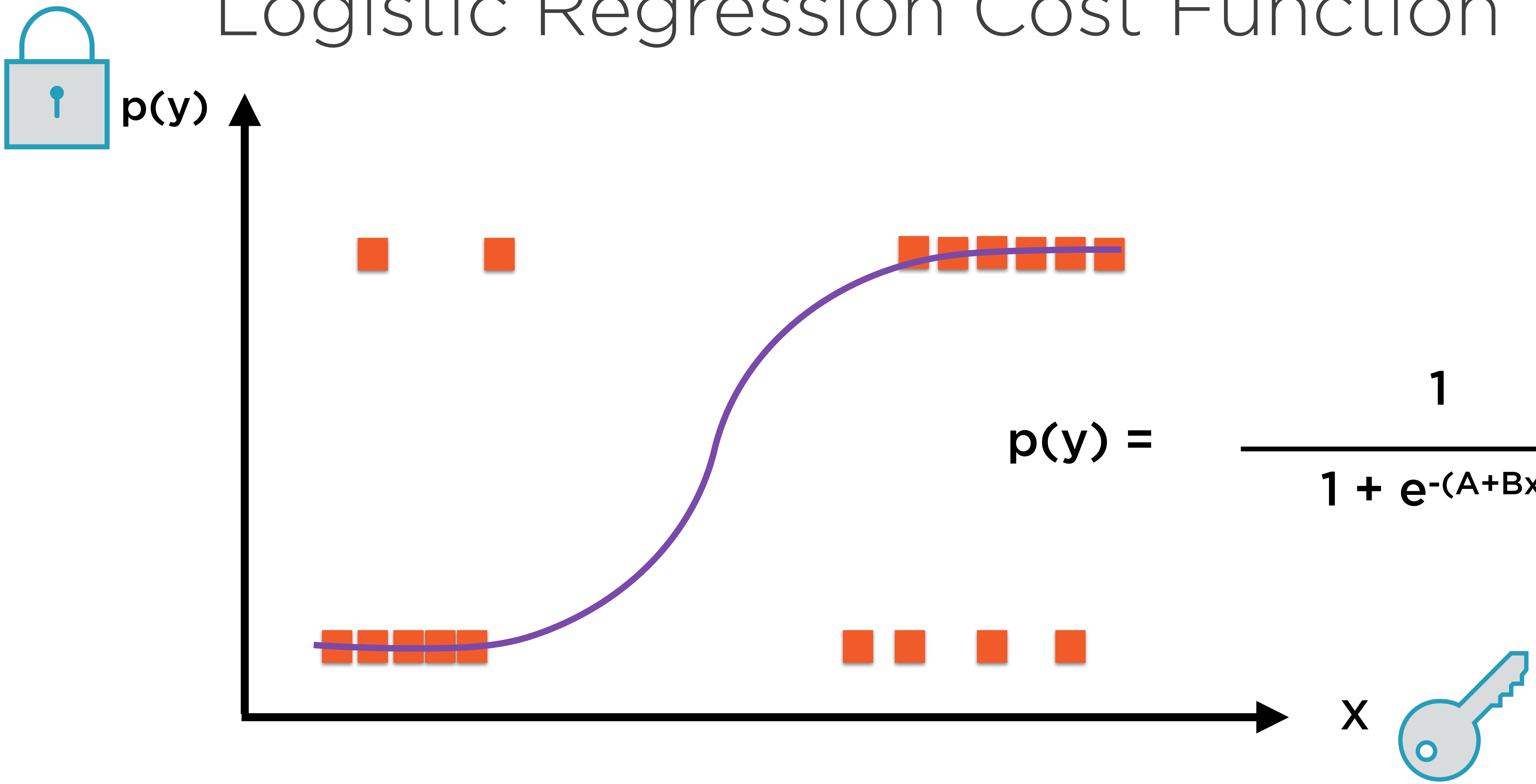
Cross Entropy: Loss Function

Linear Regression Cost Function



The mean square error measures how far the line is from the actual points

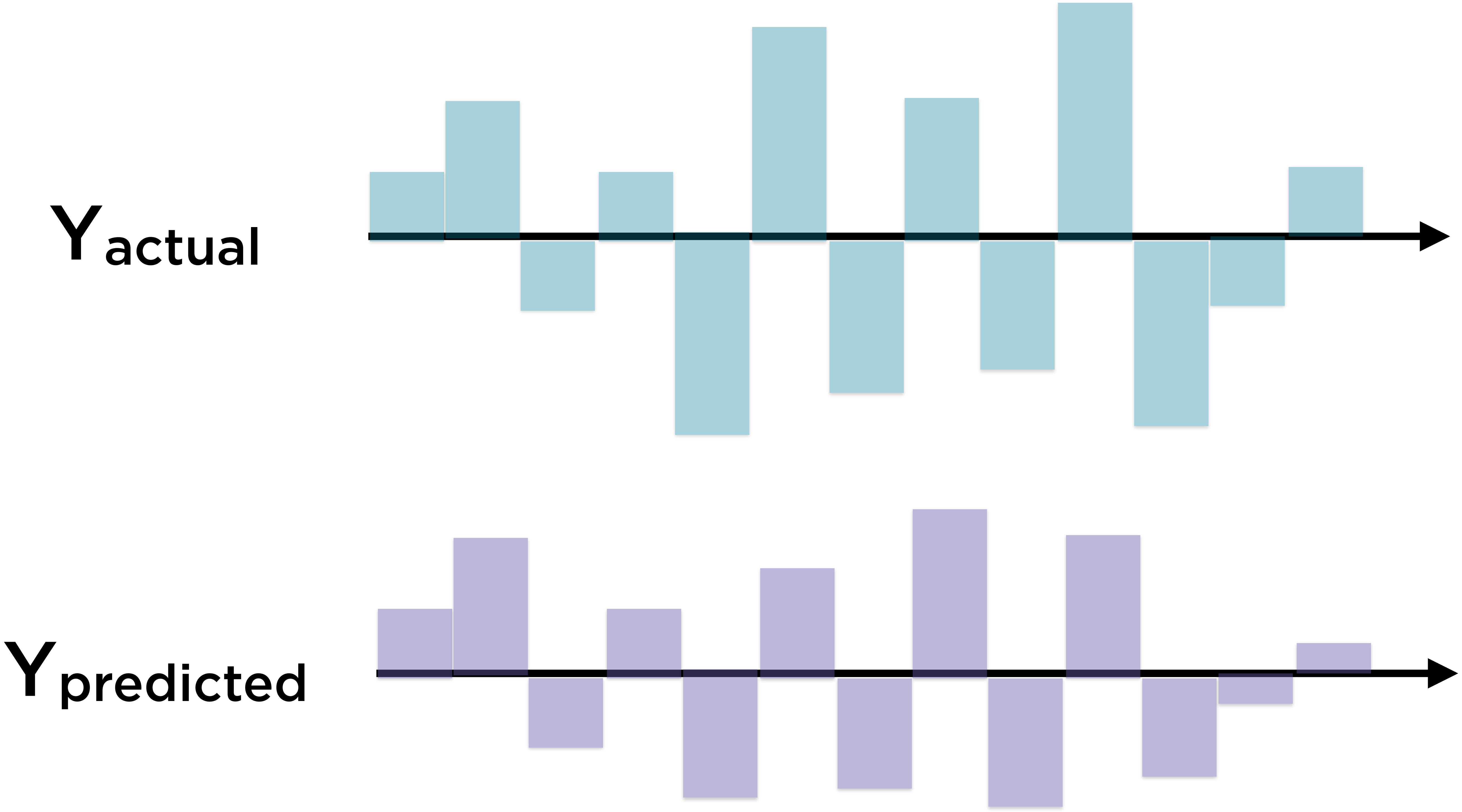
Logistic Regression Cost Function



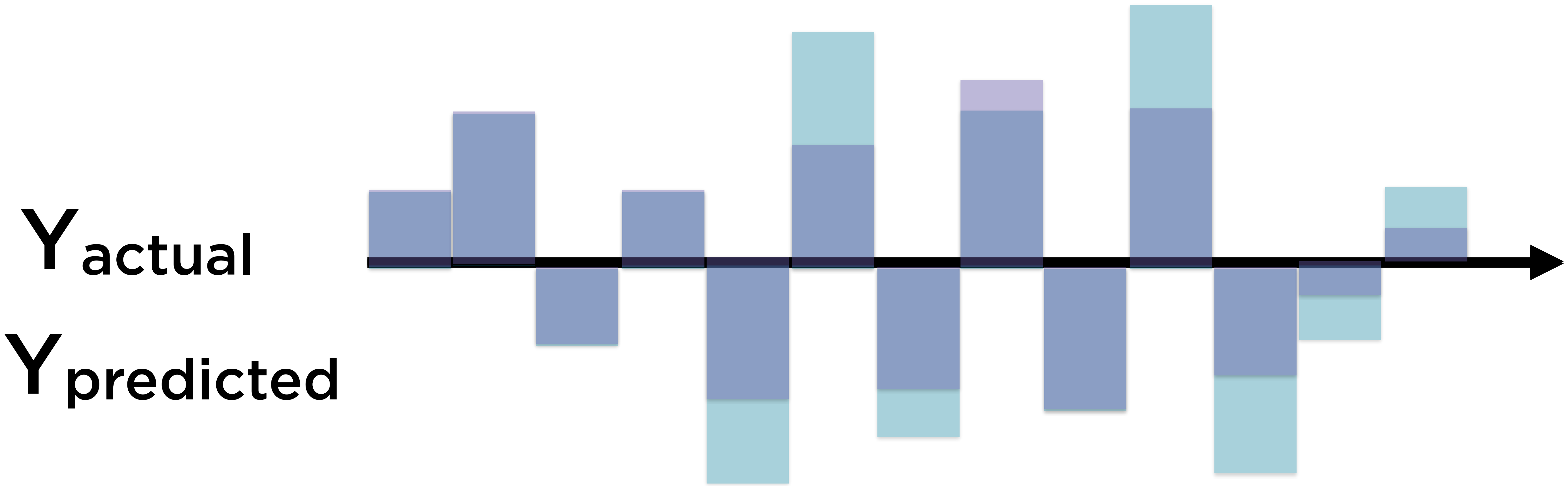
$$p(y) = \frac{1}{1 + e^{-(A+Bx)}}$$

Cross entropy measures how well the estimated probabilities match actual labels

Intuition: Low Cross Entropy

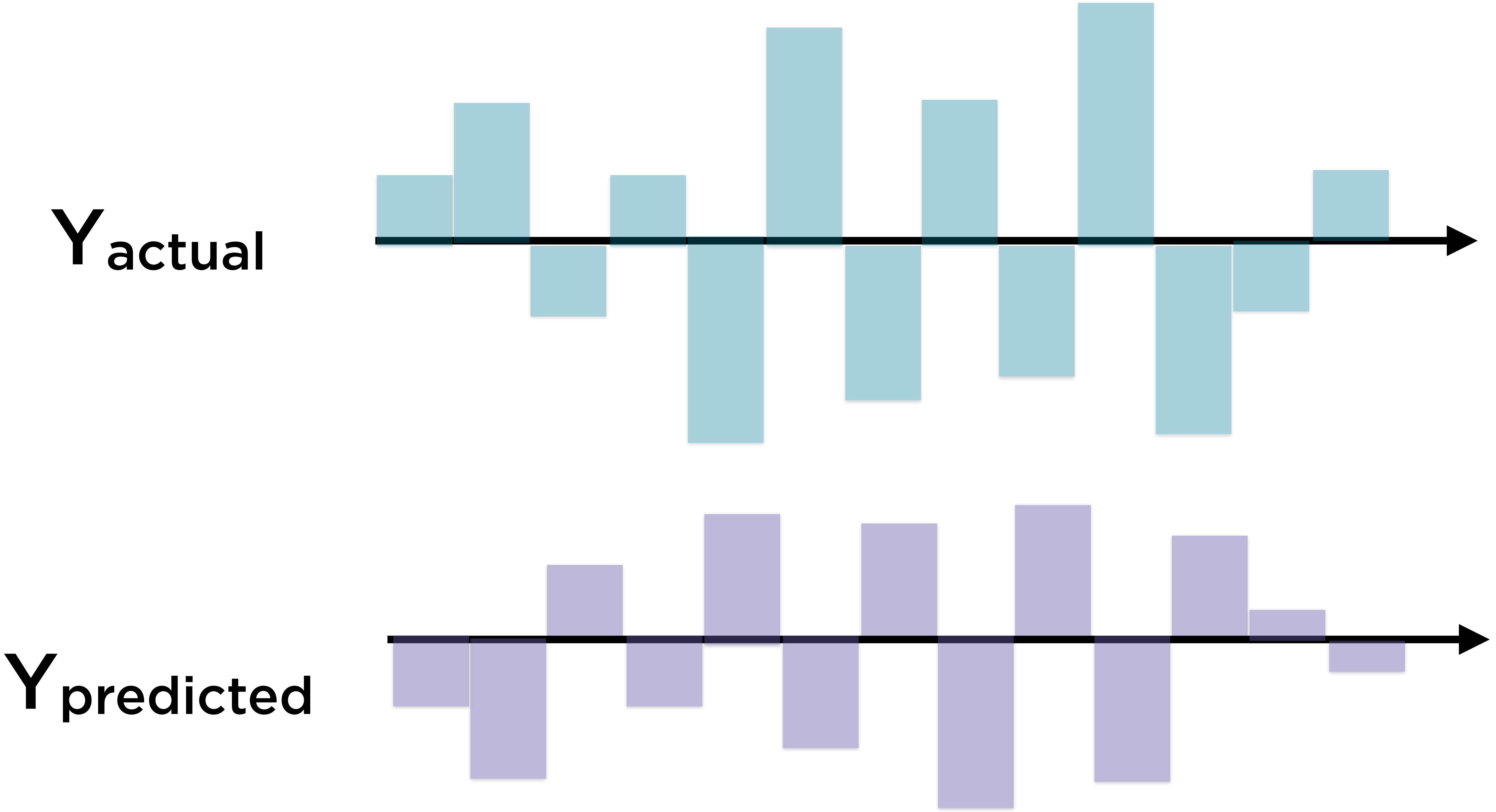


Intuition: Low Cross Entropy

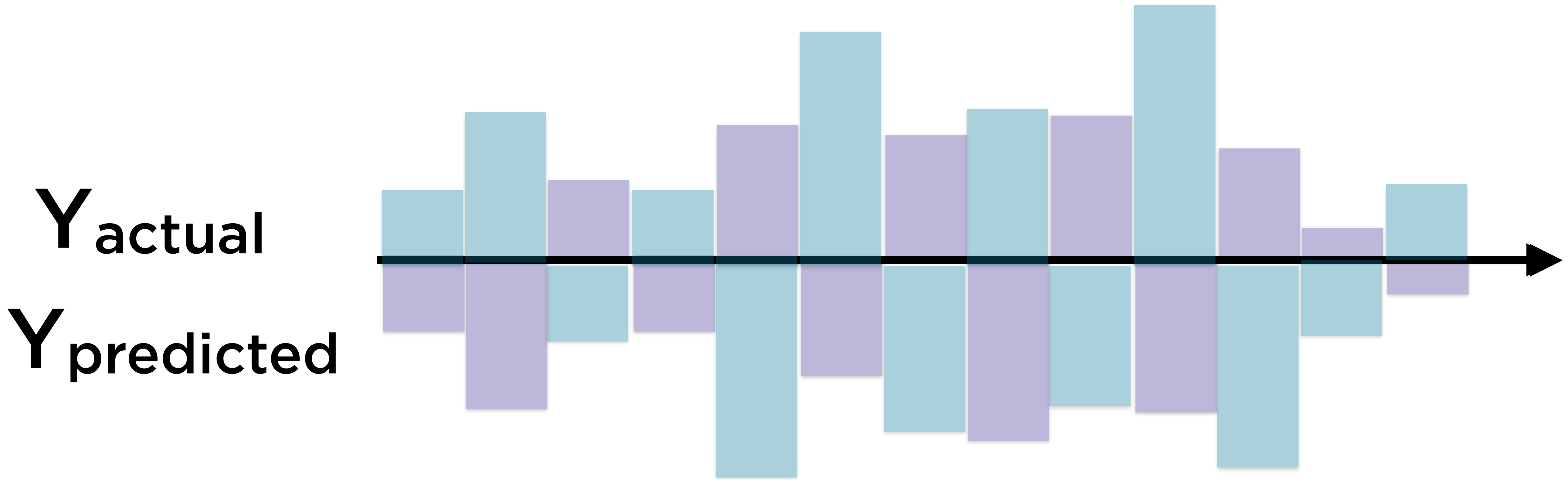


The labels of the two series are in-synch

Intuition: High Cross Entropy



Intuition: High Cross Entropy



The labels of the two series are out-of-synch

Accuracy, Precision, Recall

Accuracy

Compare predicted and actual labels

More matches = higher accuracy

High accuracy is good, but...

An algorithm might have high accuracy but still be a poor machine learning model

Its predictions are **useless**

All-is-well Binary Classifier

Here, accuracy for rare cancer may be 99.9999%, but...

Accuracy

Some labels maybe much more common/rare than others

Such a dataset is said to be skewed

Accuracy is a poor evaluation metric here

Confusion Matrix

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	**10 instances**	**4 instances**
No Cancer	**5 instances**	**1000 instances**

Confusion Matrix

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10	4
No Cancer	5	1000

True Positive

Actual Label	Predicted Labels: Cancer	Predicted Labels: No Cancer
Cancer	**10**	4
No Cancer	5	1000

Actual Label = Predicted Label

True Positive

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4
No Cancer	5	1000

Actual Label = Predicted Label

False Positive

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10	4
No Cancer	**5**	1000

Actual Label $\neq$ Predicted Label

False Positive

Predicted Labels

Actual Label	Cancer	No Cancer
Cancer	10	4
No Cancer	5 FP	1000

Actual Label $\neq$ Predicted Label

True Negative

Predicted Labels

Actual Label	Cancer	No Cancer
Cancer	10	4
No Cancer	5	1000

Actual Label = Predicted Label

True Negative

Actual Label	Predicted Labels: Cancer	Predicted Labels: No Cancer
Cancer	10	4
No Cancer	5	TN 1000

Actual Label = Predicted Label

False Negative

Predicted Labels

Actual Label	Cancer	No Cancer
Cancer	10	4
No Cancer	5	1000

Actual Label ≠ Predicted Label

False Negative

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10	4 FN
No Cancer	5	1000

Actual Label ≠ Predicted Label

Confusion Matrix

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

Accuracy

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

Accuracy

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

Actual Label = Predicted Label

Accuracy

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	TP 10	FN 4
No Cancer	FP 5	TN 1000

$$\text{Accuracy} = \frac{TP + TN}{\text{Num Instances}} = \frac{1010}{1019} = 99.12\%$$

Accuracy

Accuracy = 99.12%

Classifier gets it right 99.12% of the time

But...

Accuracy

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

People on chemotherapy, radiation when not required

Accuracy

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

Cancer not detected, no treatment prescribed

Accuracy is not a good metric to evaluate whether this model performs well

Precision

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

Precision

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

Precision = Accuracy when classifier flags cancer

Precision

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{10}{15} = 66.67\%$$

Precision

Precision = 66.67%

1 in 3 cancer diagnoses is incorrect

Recall

Predicted Labels

Actual Label	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

Recall

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

Recall = Accuracy when cancer actually present

Recall

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10 TP	4 FN
No Cancer	5 FP	1000 TN

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{10}{14} = 71.42\%$$

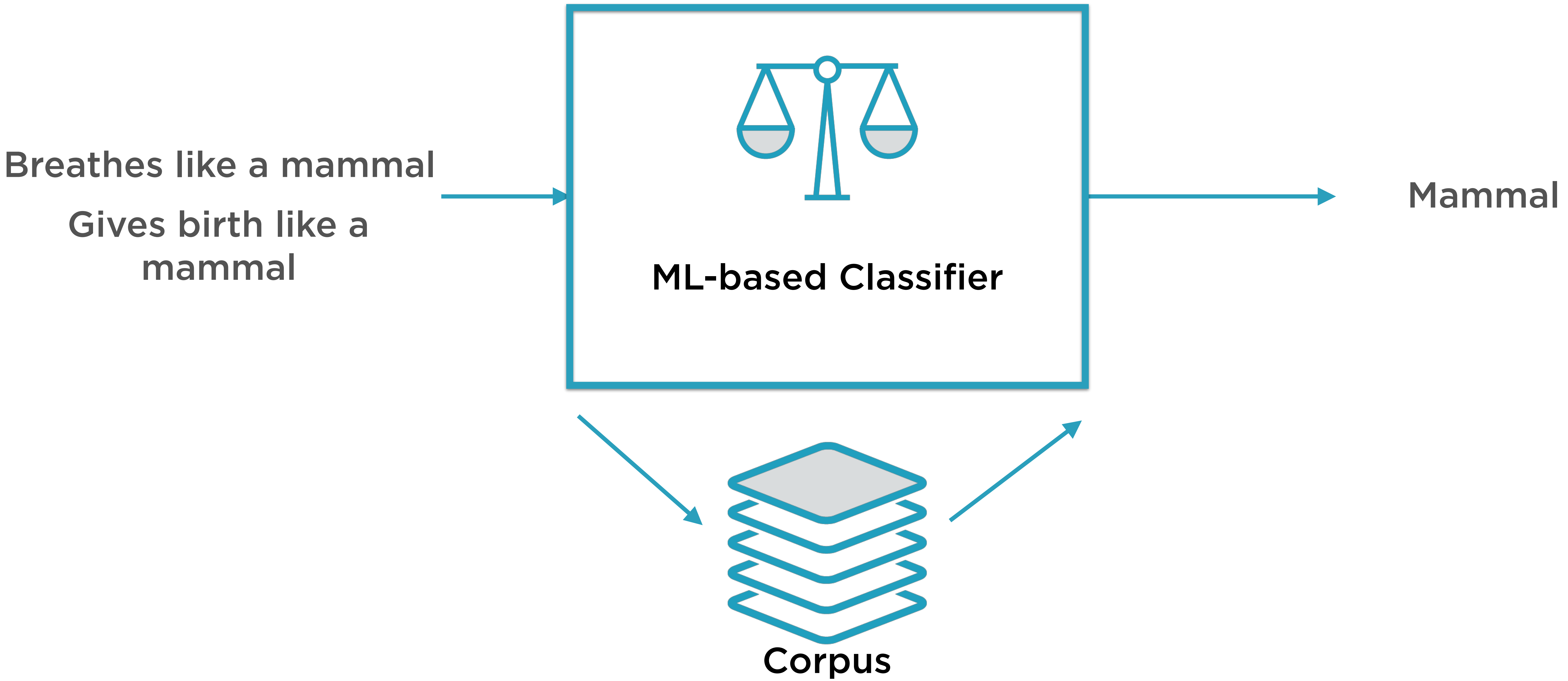
Recall

Recall = 71.42%

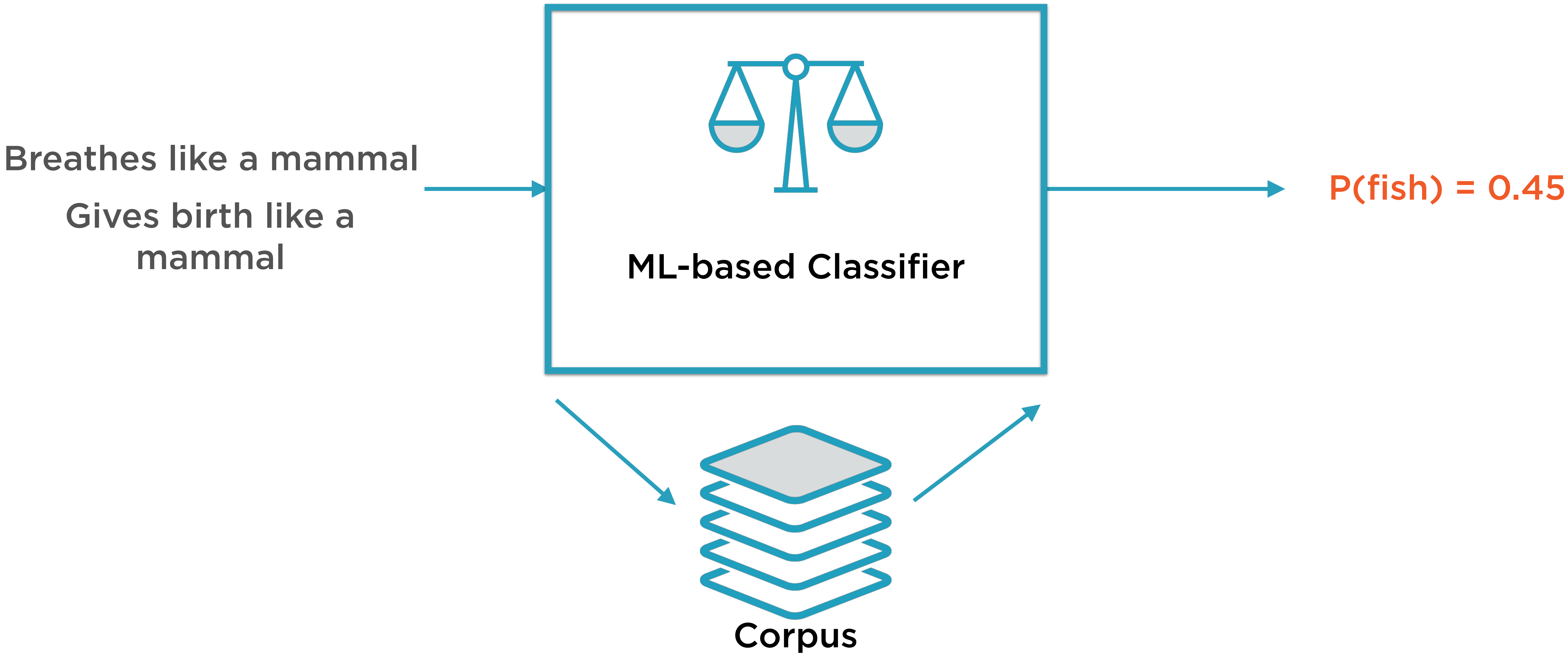
2 in 7 cancer cases missed

Evaluating Classifiers

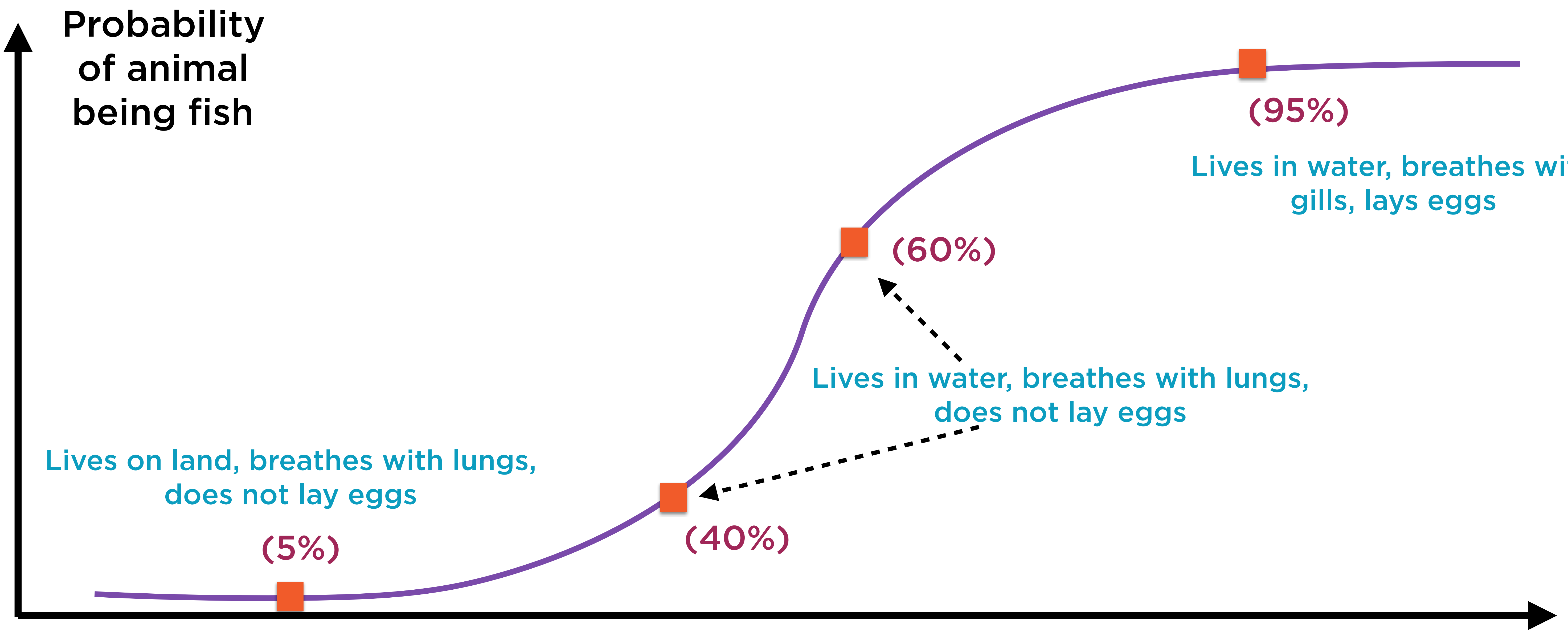
ML-based Binary Classifier



ML-based Binary Classifier

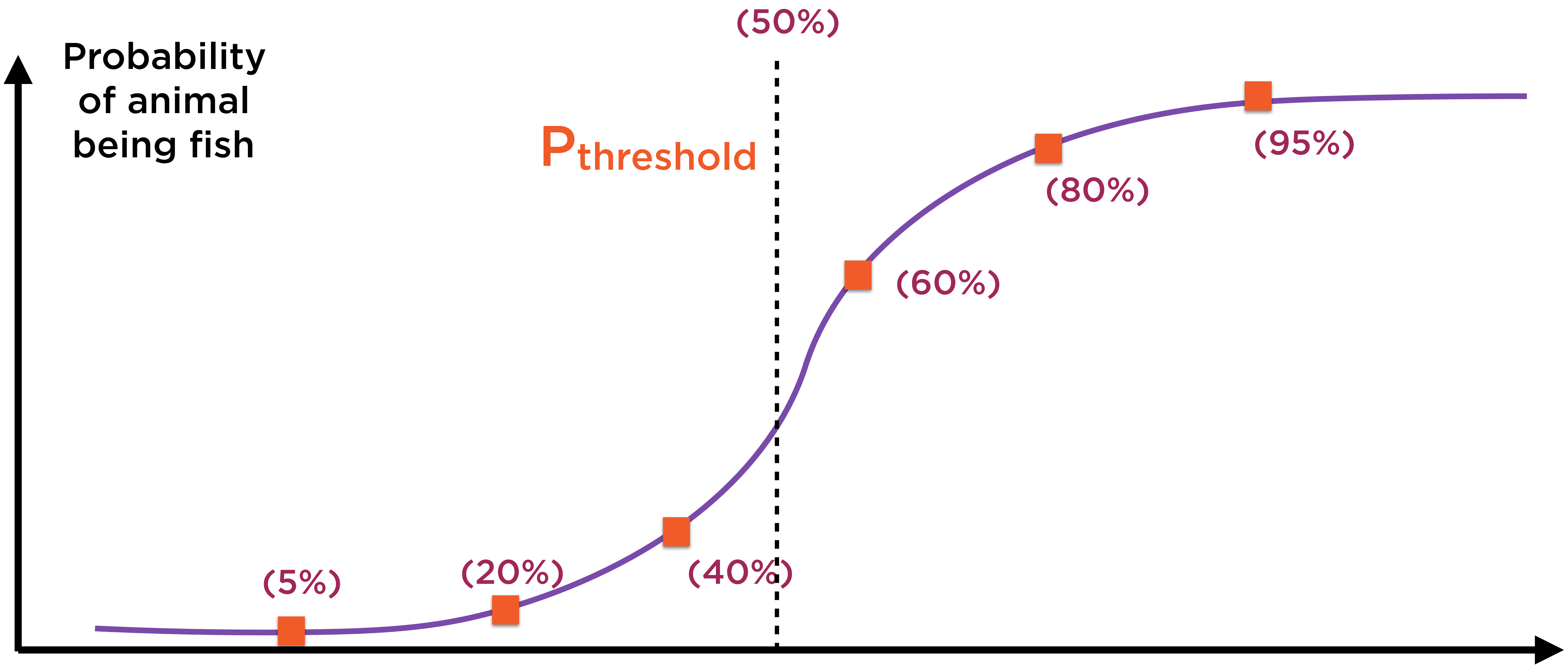


Applying Logistic Regression

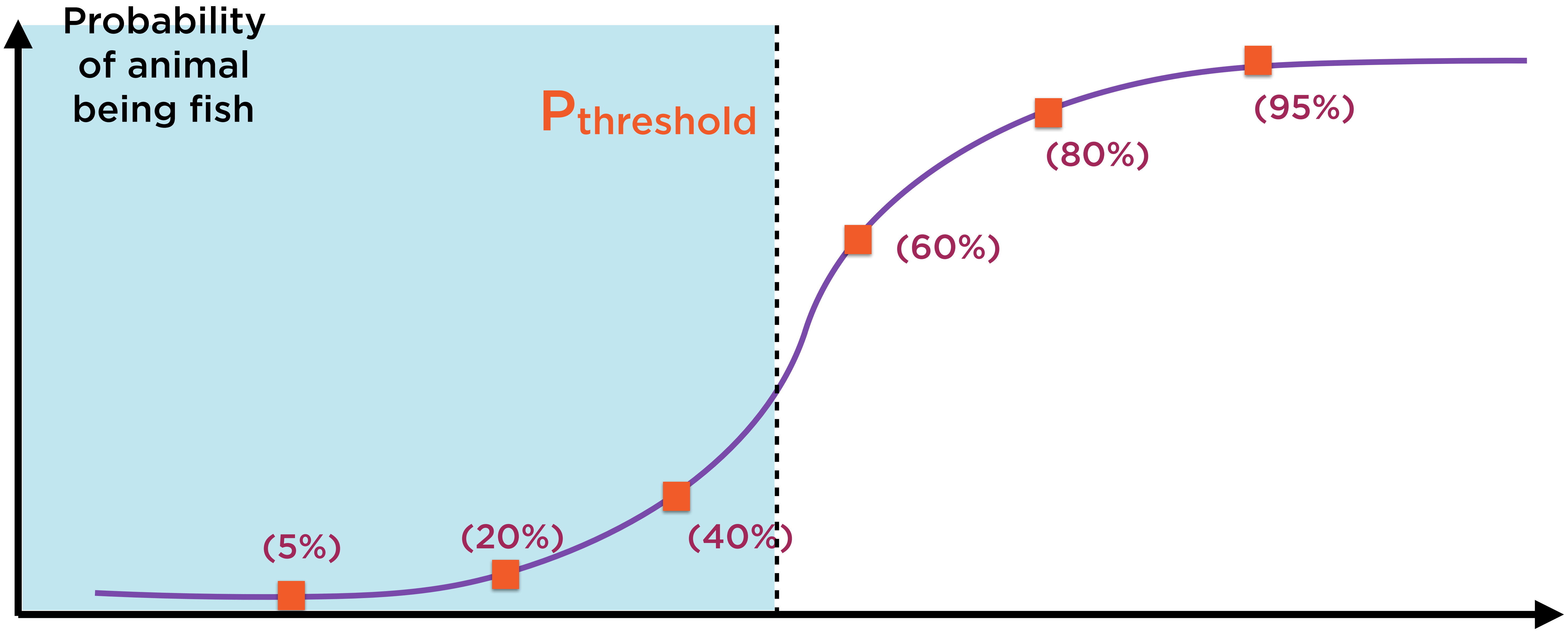


Whales: Fish or Mammals?

Choosing Decision Threshold

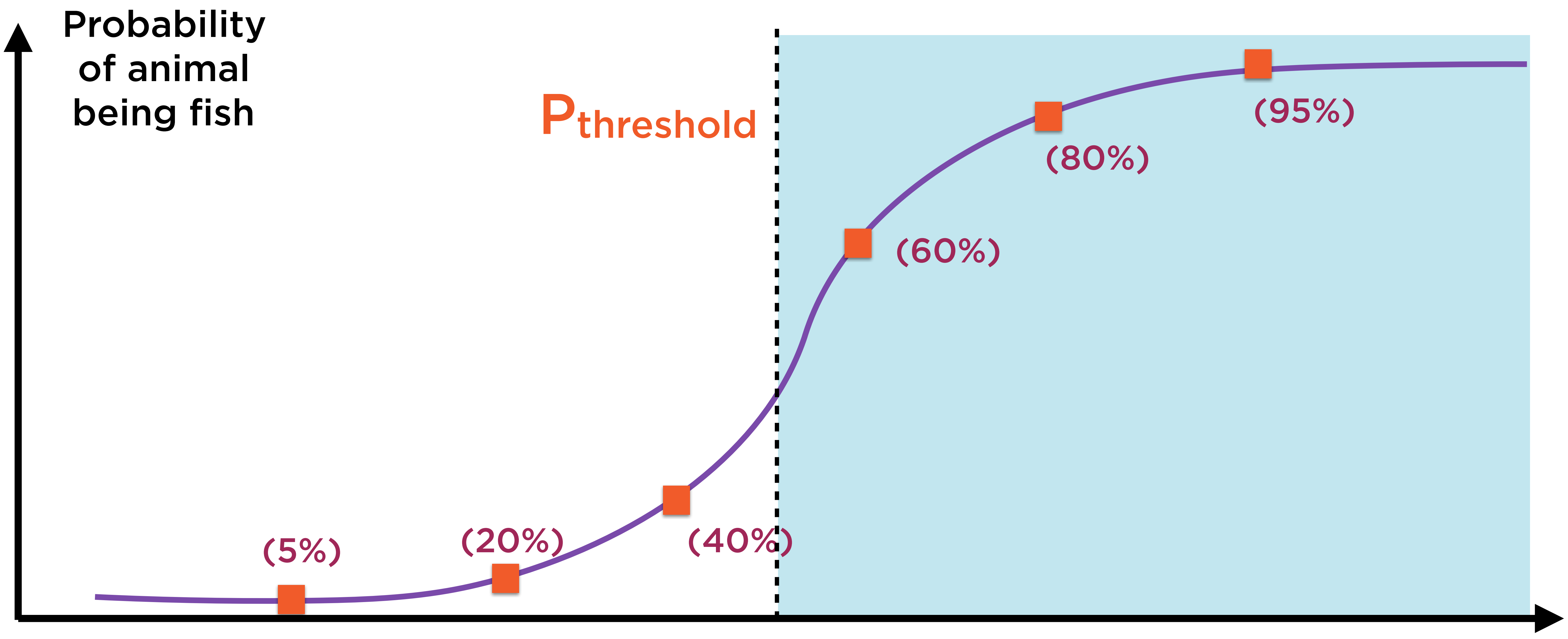


Choosing Decision Threshold



If probability < $P_{threshold}$, it's a mammal

Applying Logistic Regression



If probability > $P_{threshold}$, it's a fish

“Always Positive”

$P_{threshold} = 0$

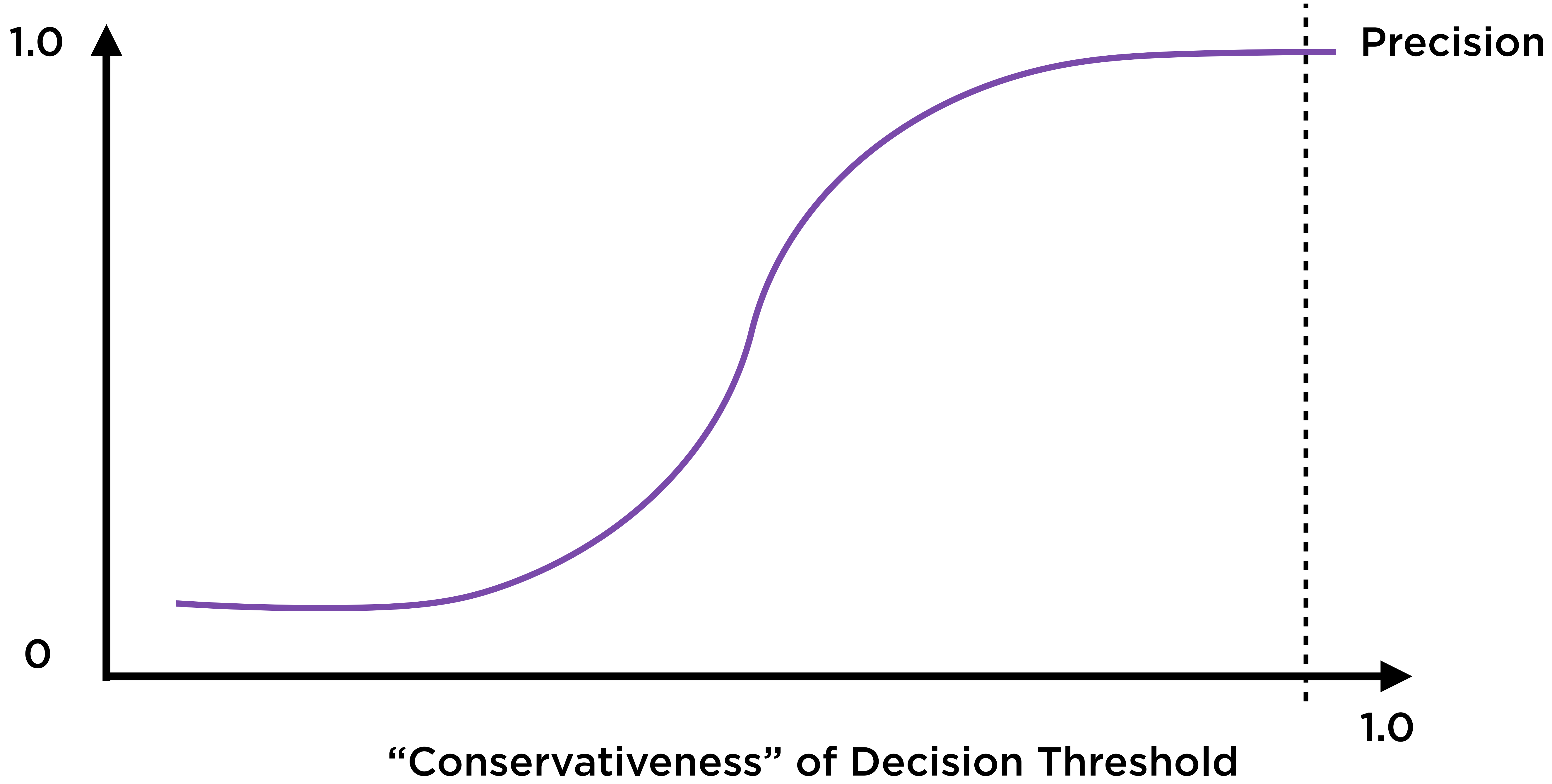
Actual \ Predicted	Cancer	No Cancer
Cancer	TP 14	FN 0
No Cancer	FP 1005	TN 0

Recall = 100%

Precision = 14/1019 = 13.7%

Classifier not conservative enough

Precision vs. “Conservativeness”



“Always Negative”

$P_{threshold} = 1$

Predicted

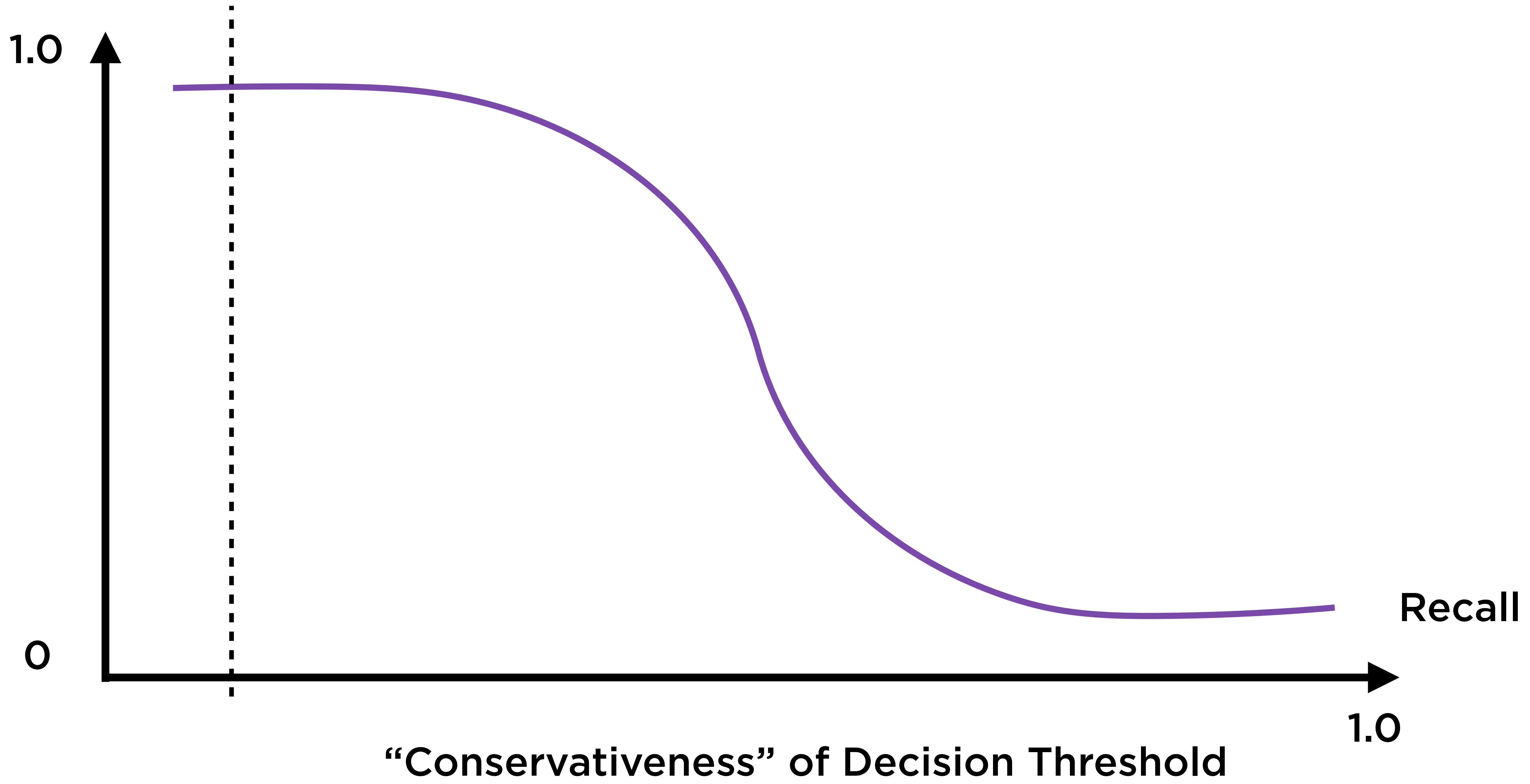
Actual	Cancer	No Cancer
Cancer	TP 0	FN 14
No Cancer	FP 0	TN 1005

Recall = 0%

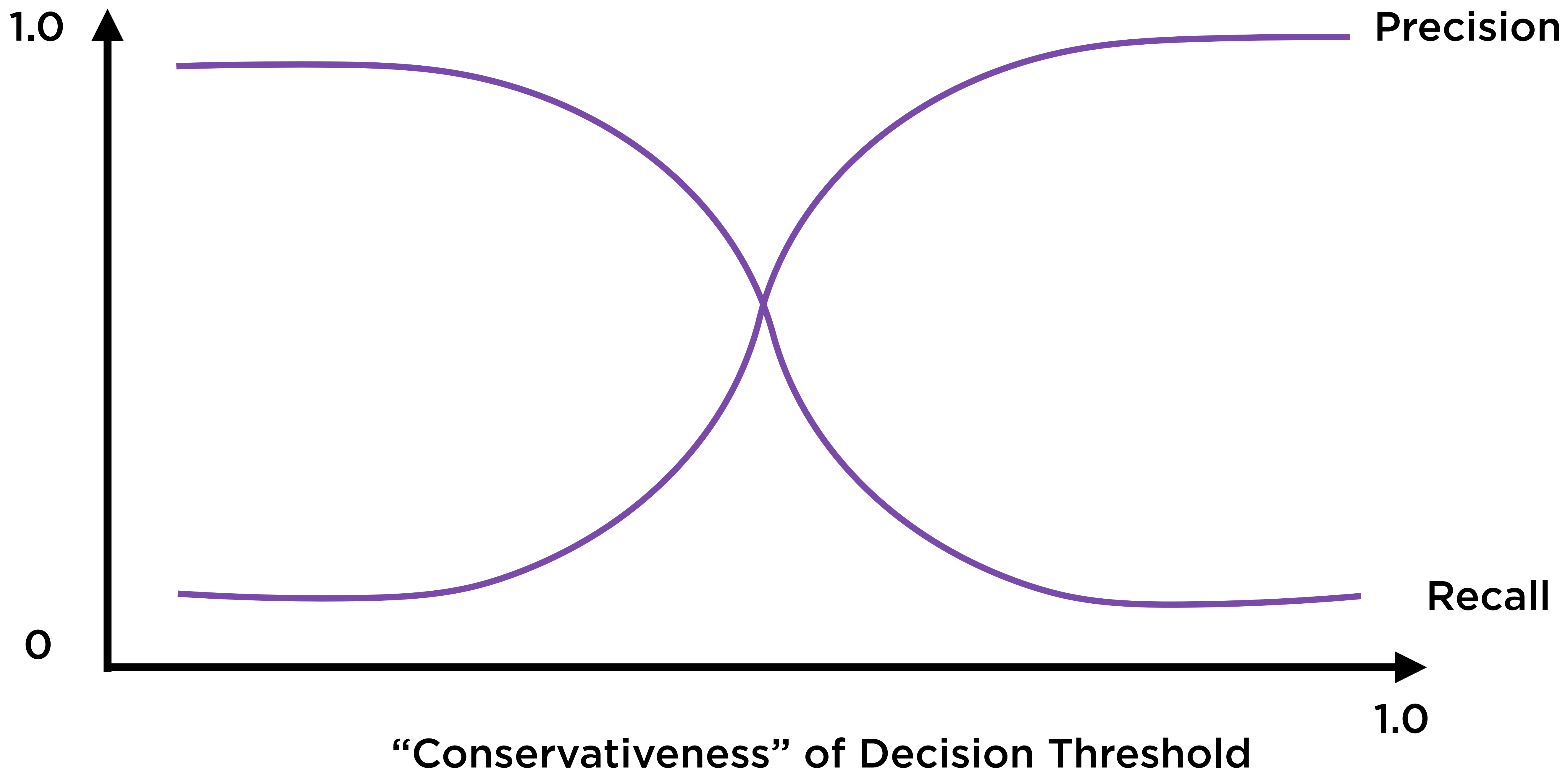
Precision = Infinite

Classifier too conservative

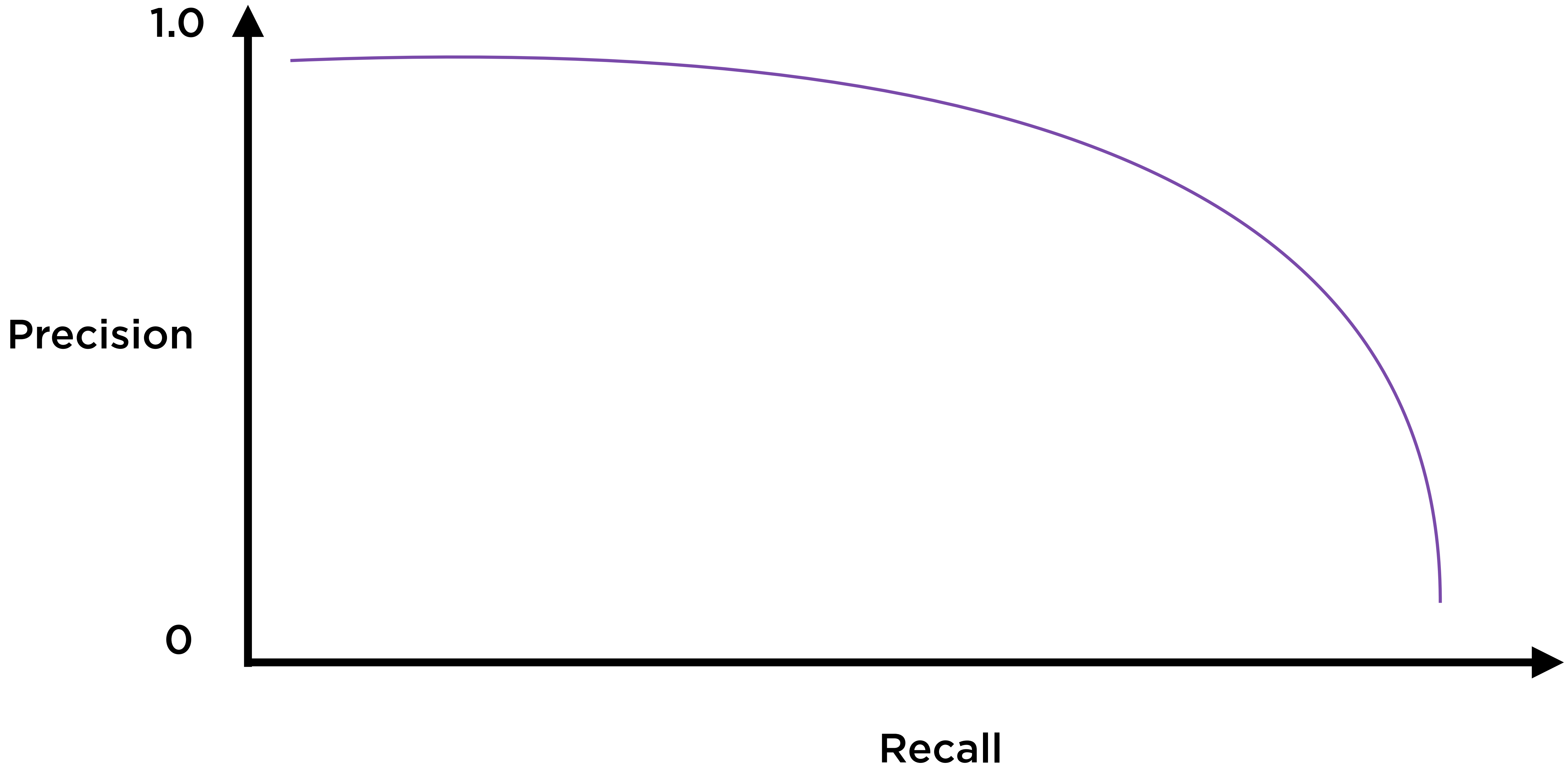
Recall vs. "Conservativeness"



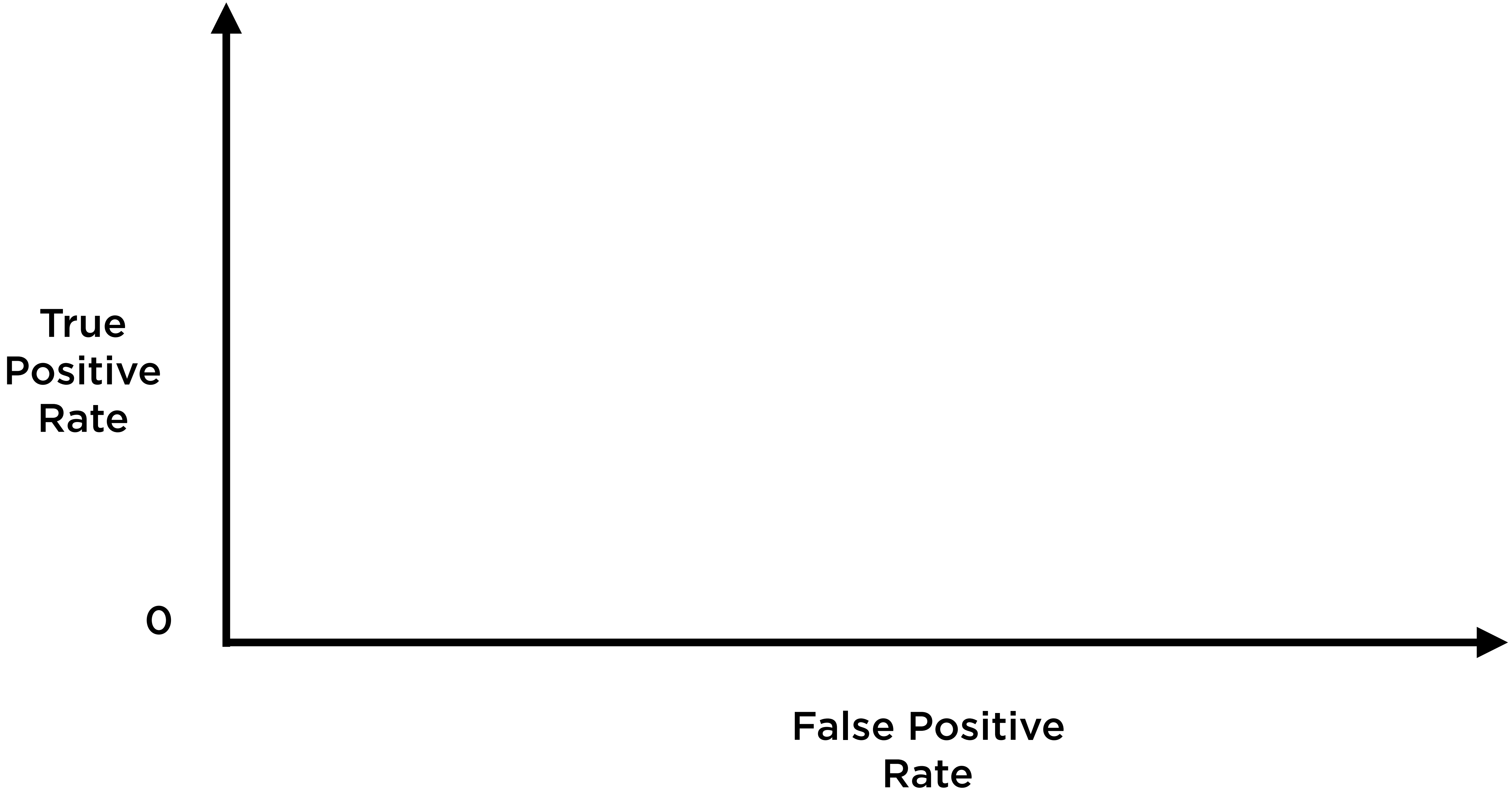
Precision-Recall Tradeoff



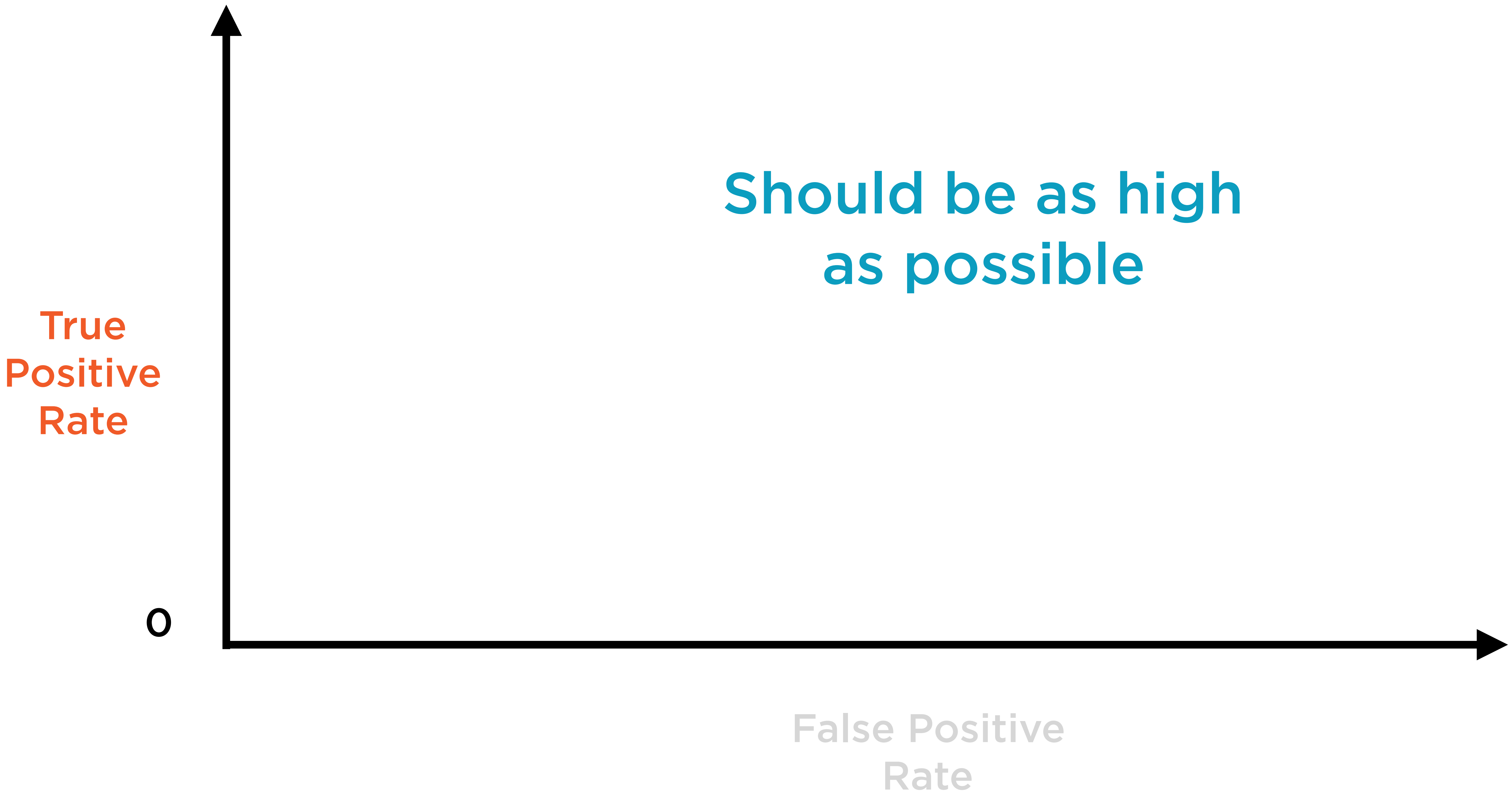
Precision-Recall Tradeoff



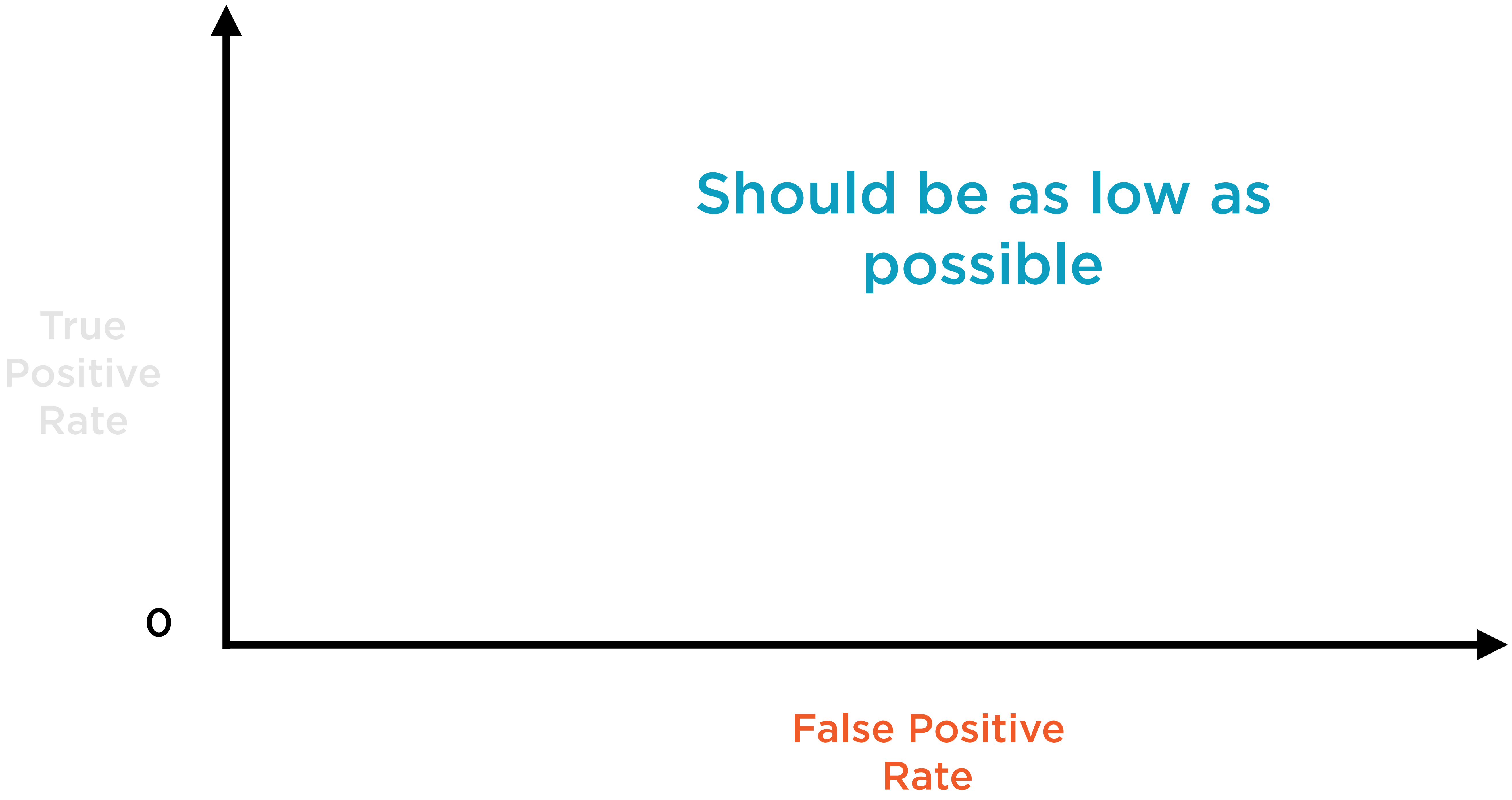
Choosing $P_{threshold}$



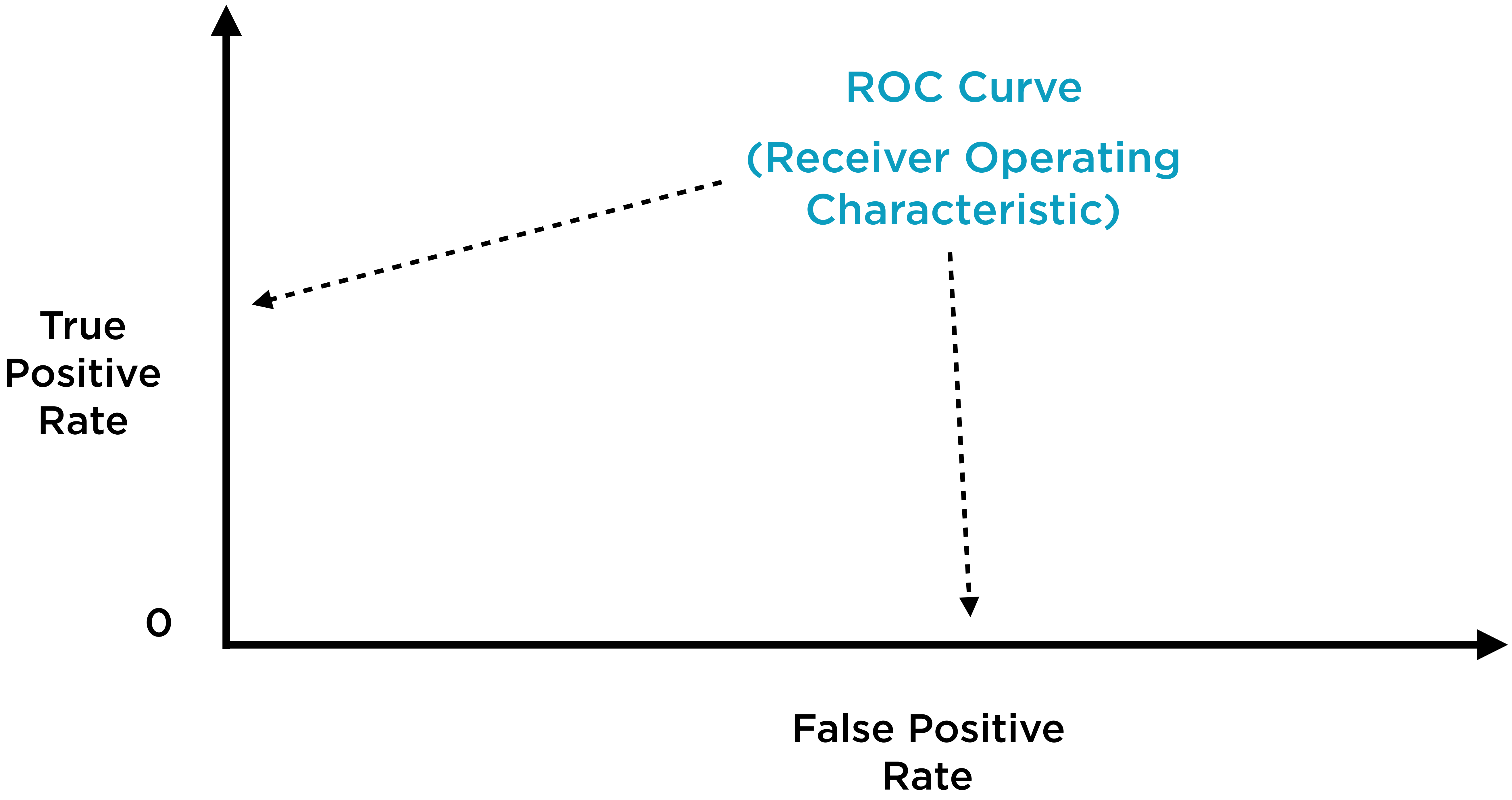
Choosing $P_{threshold}$



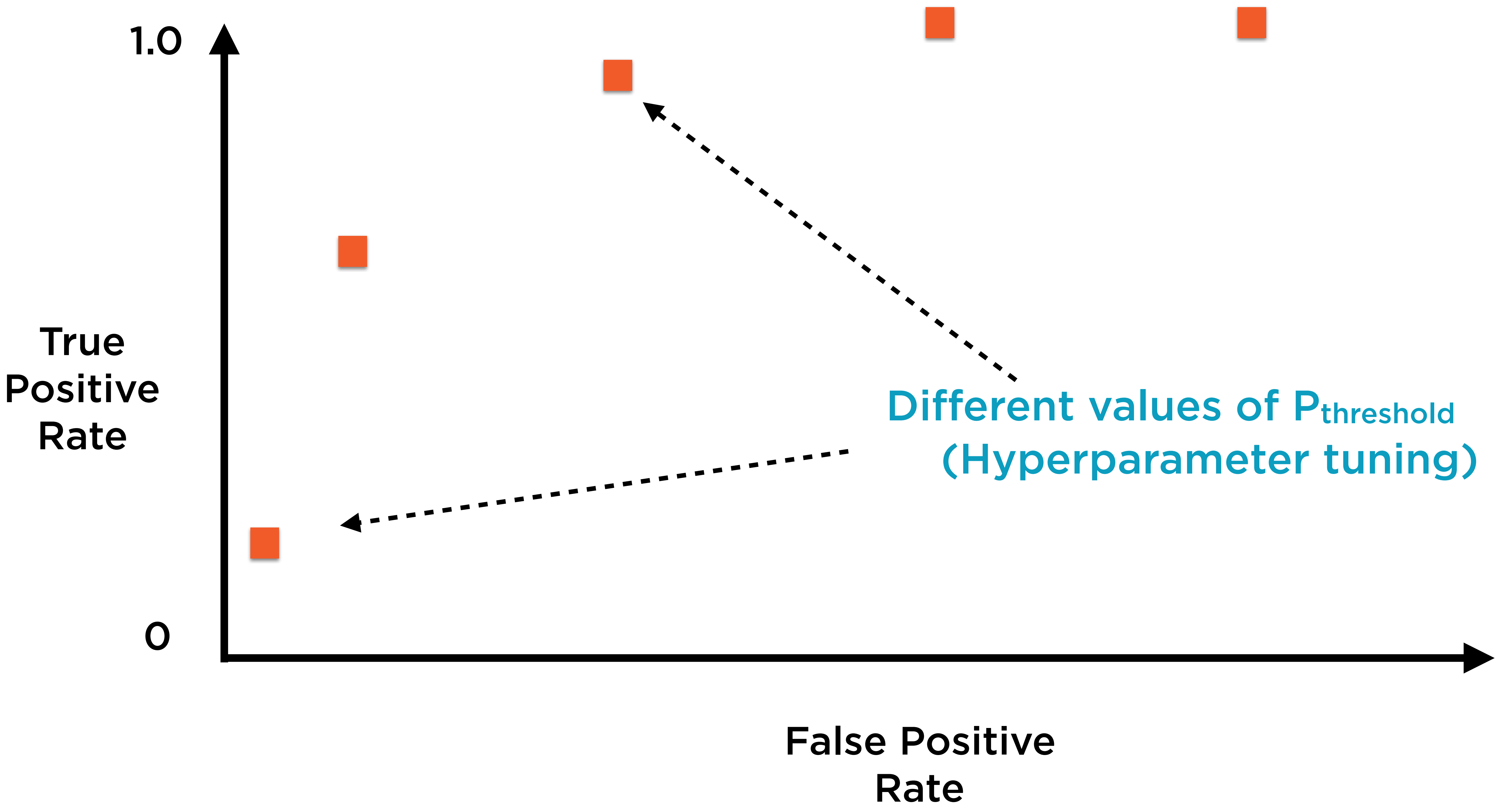
Choosing $P_{threshold}$



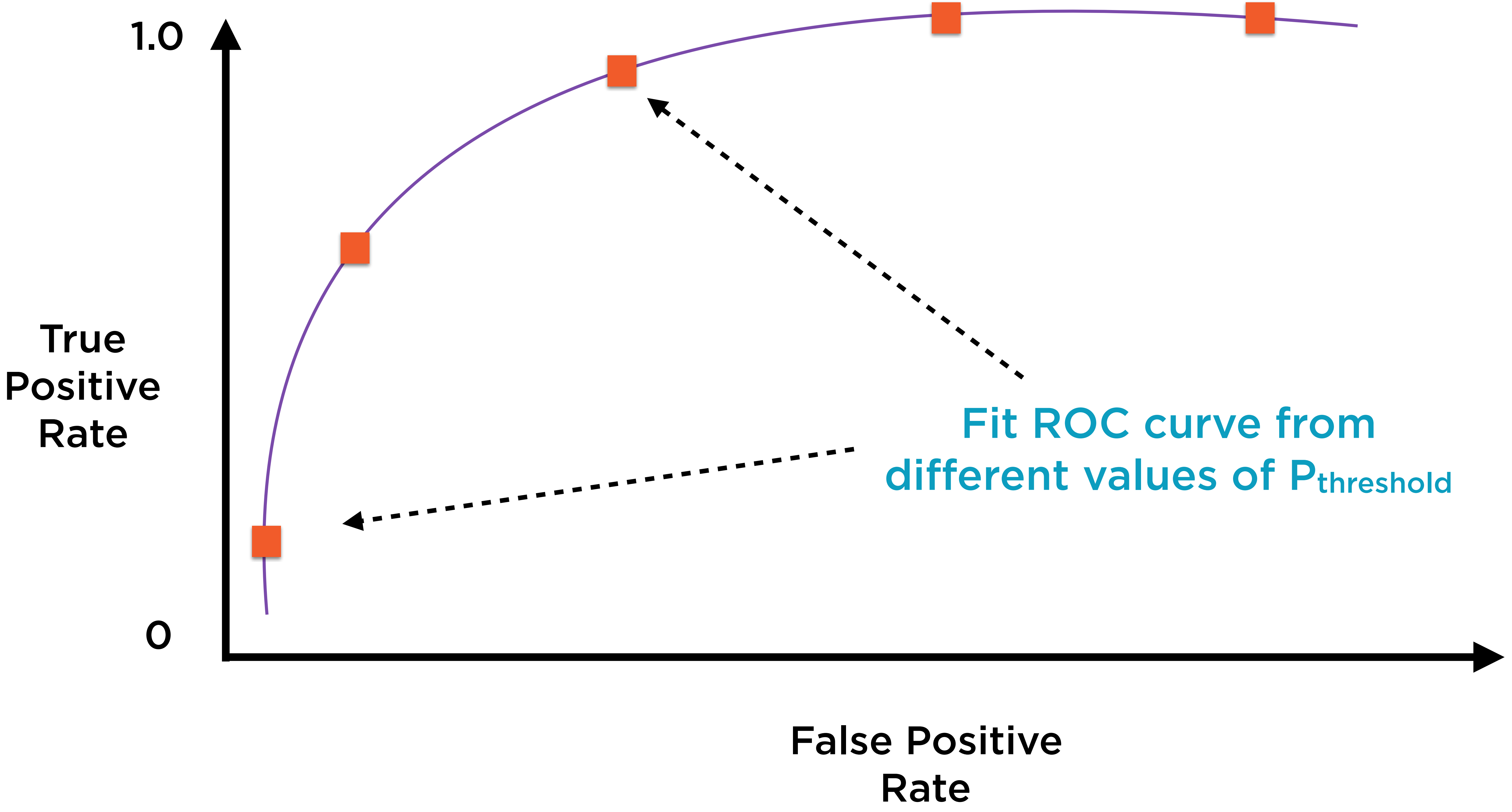
Choosing $P_{threshold}$



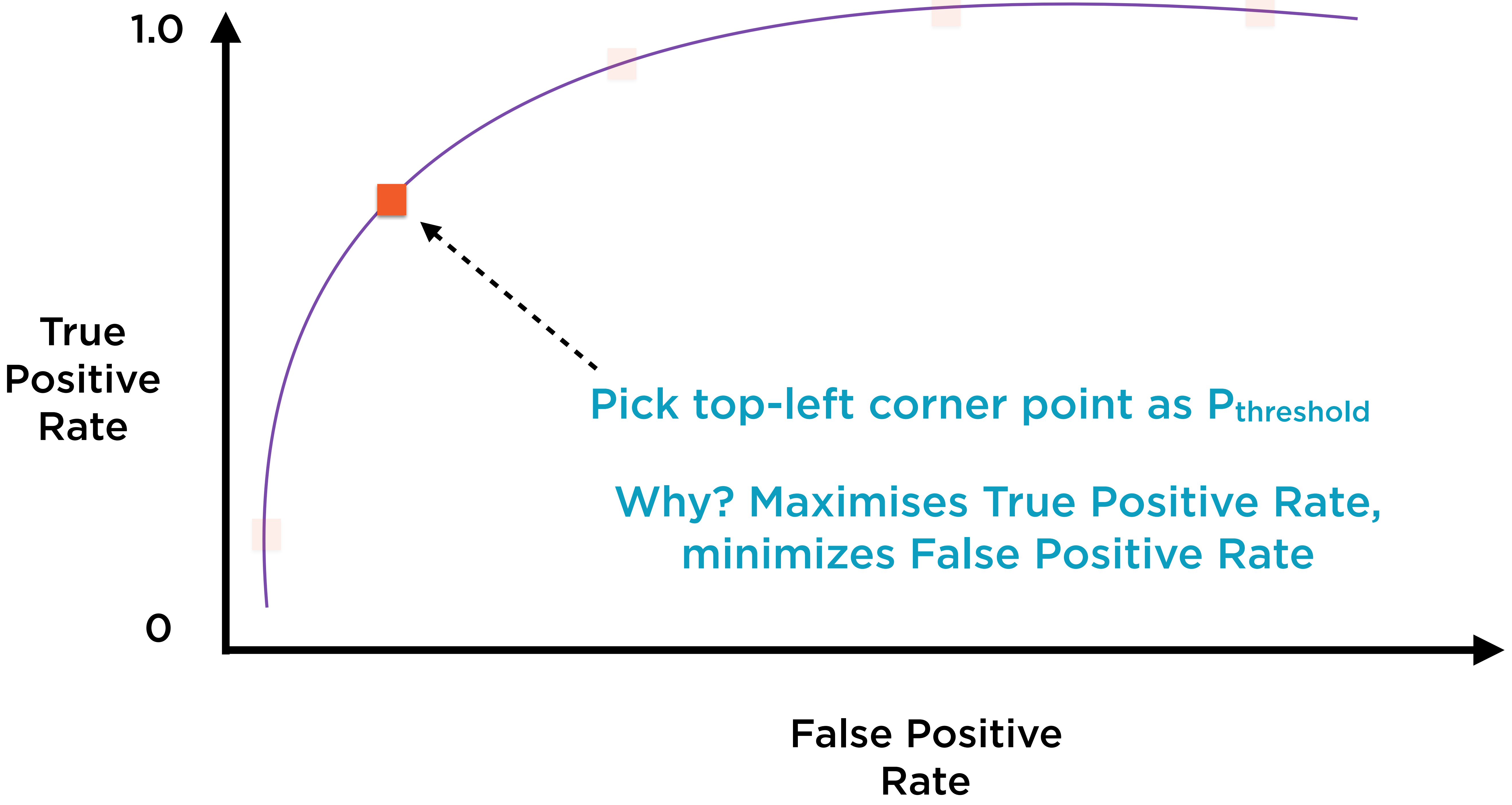
Choosing $P_{threshold}$



Choosing $P_{threshold}$

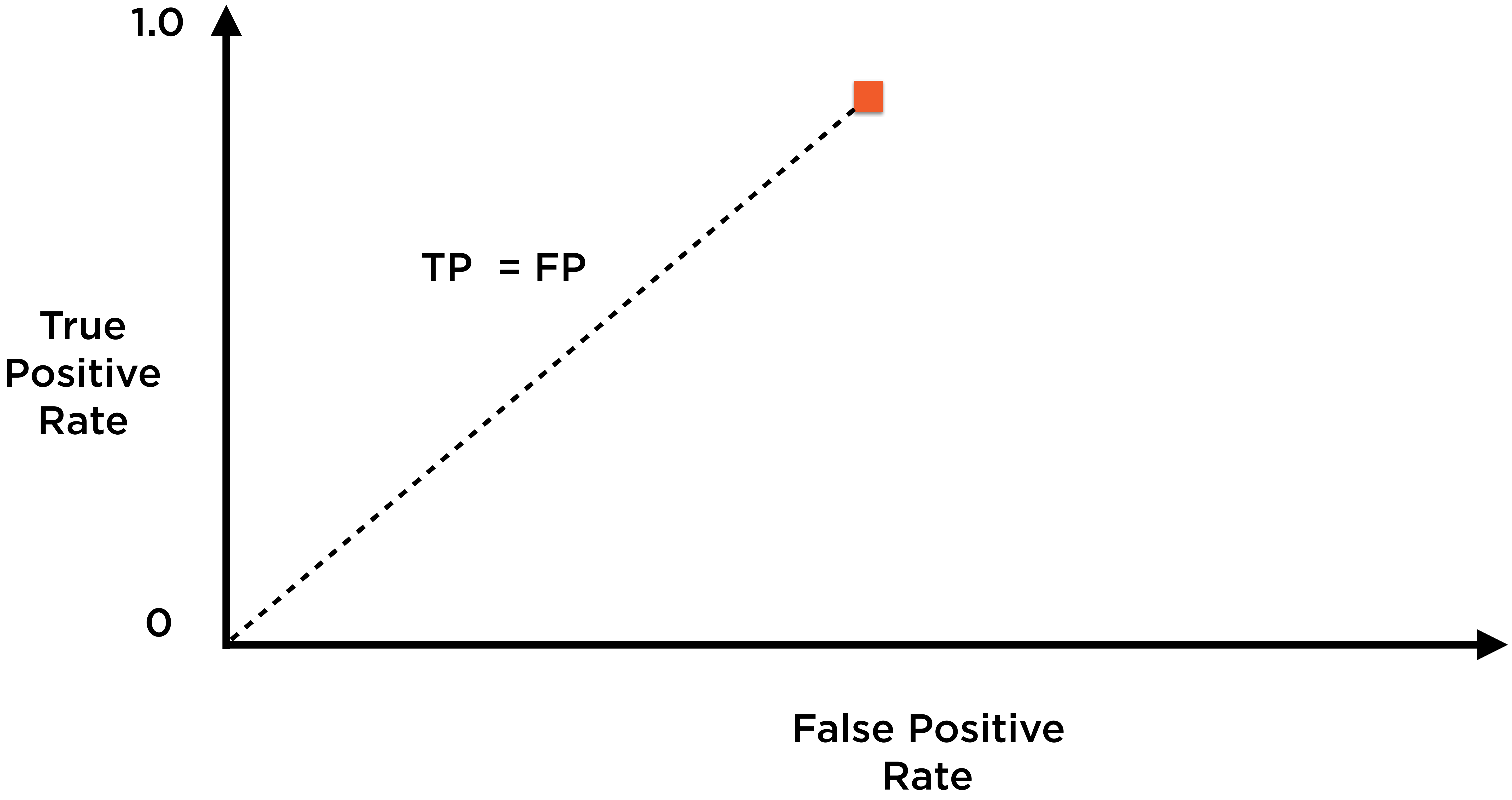


ROC Curve


1.0
0
True
Positive
Rate
Pick top-left corner point as $P_{threshold}$
Why? Maximises True Positive Rate,
minimizes False Positive Rate
False Positive
Rate

ROC of Perfect Classifier

ROC of Random Classifier



Types of Classification

Types of Classification Tasks

Binary

“Yes/No”, “True/False”, “Up/Down”

Output is binary categorical variable

Multiclass

Digit classification

Output variable takes 1 of N (>2) values

Multilabel

(“True”, “Female”), (“False”, “Female”)

Output is tuple of multiple binary variables (not disjoint)

Multioutput

(“Sunday”, “January”)

Multiclass + multilabel

Multilabel

Some algorithms are inherently multilabel

- Naive Bayes

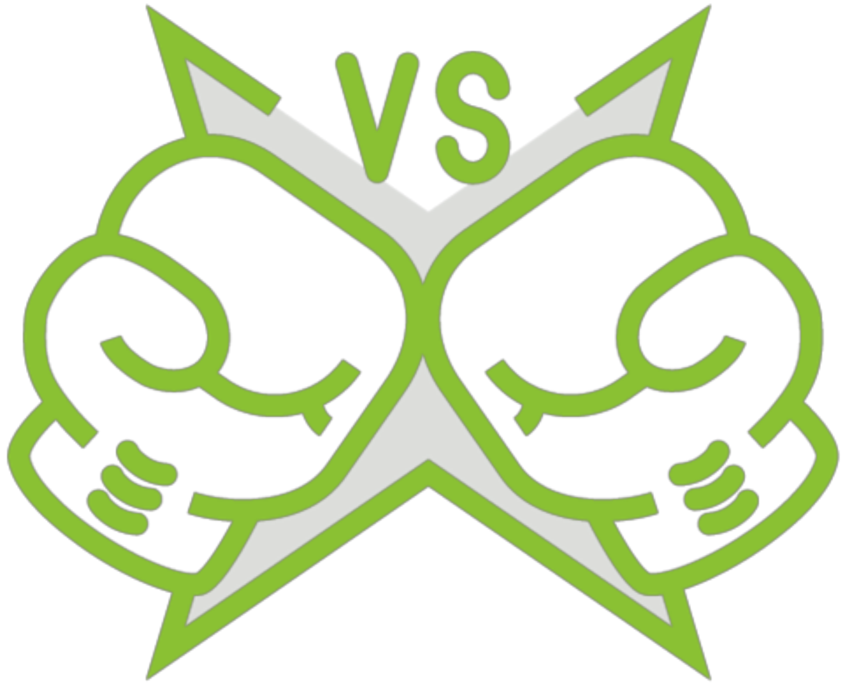
Multilabel

Many classification algorithms are inherently binary

- Logistic regression
- Support Vector Machines

Inherently binary classifiers can be generalised for multilabel classification

One vs. All



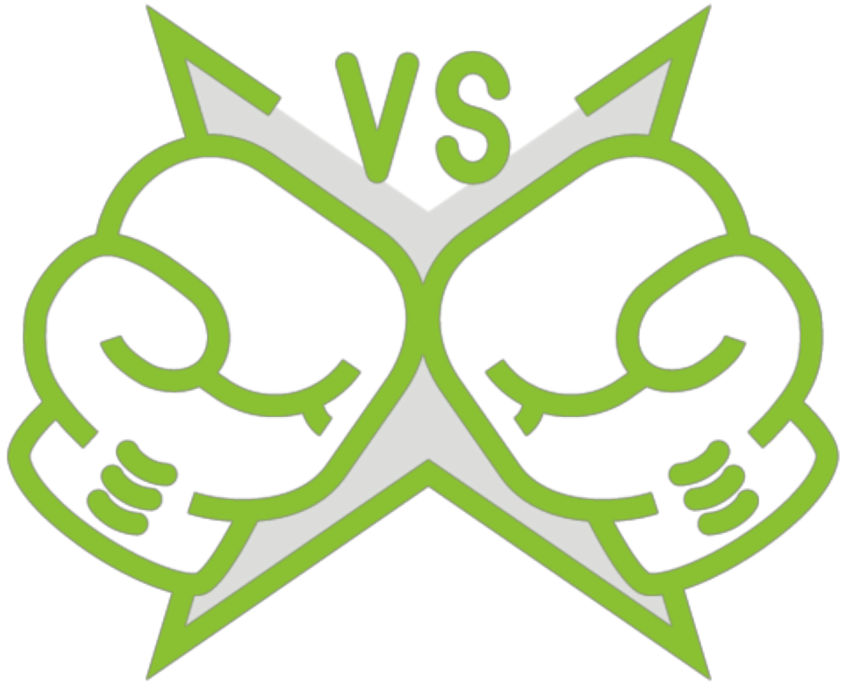
One-versus-all

Classifying digits 0-9

Train 10 binary classifiers

- 0-detector, 1-detector...
- Predicted label = output of detector with highest score

One vs. One



One-versus-one

Train 45 binary classifiers

- One detector for each pair of digits
- For N labels, need N(N-1)/2 classifiers
- Predicted label = output of digit that wins most duels

Summary

Logistic regression for classification

Evaluating classification models

Accuracy, precision, and recall

ROC curves

Binary, multi-label, and multi-class classification