# Implementing Predictive Models for Categorical Data

**Janani Ravi**
CO-FOUNDER, LOONYCORN

www.loonycorn.com

# Overview

Classification to predict categorical variables

Intuition behind logistic regression

Evaluating classifiers using accuracy, precision, and recall

Building a classification model using logistic regression

Selecting relevant features to build the classifier using statistical techniques

# Types of Data

## Categorical

Male/Female, Month of year

## Numeric (Continuous)

Weight in lbs, Temperature in °F

Use regression to predict
numeric (continuous) y-variables

Use classification to predict
categorical (discrete) y-variables

# Numeric (Continuous) vs. Categorical Data

| Numeric (Continuous) | Categorical |
| --- | --- |
| E.g. height or weight of individuals | E.g. day of week, month of year, gender, letter grade |
| Can take any value | Finite set of permissible values |
| Predicted using regression models | Predicted using classification models |
| Always can be sorted on magnitude | Categories may or may not be sortable |

# 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

Start very late and hope for the best

## Work smart

Start as late as possible to be sure to make it

## Work hard

Start very early and do little else

**As usual, the middle path is best**

# Working Smart

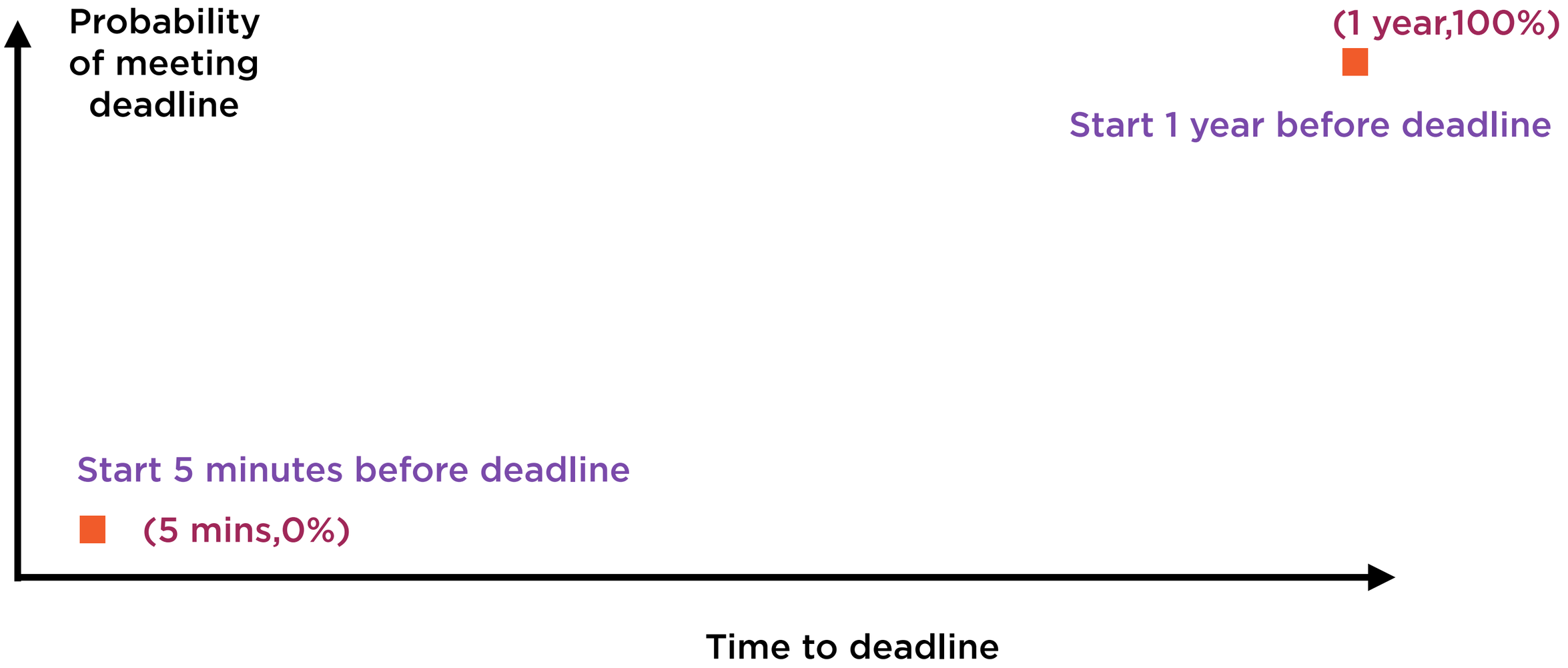
**Probability of meeting the deadline**

95%

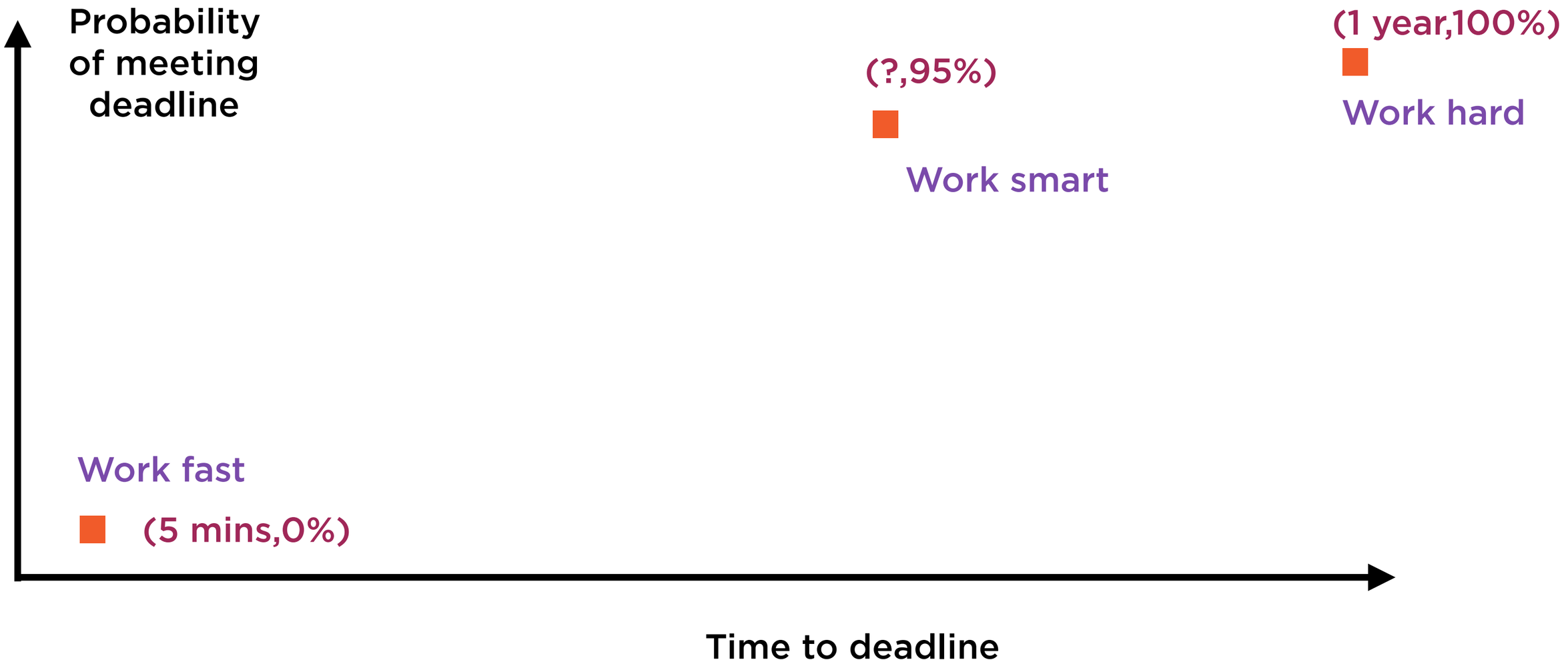
**Probability of getting other important work done**

95%

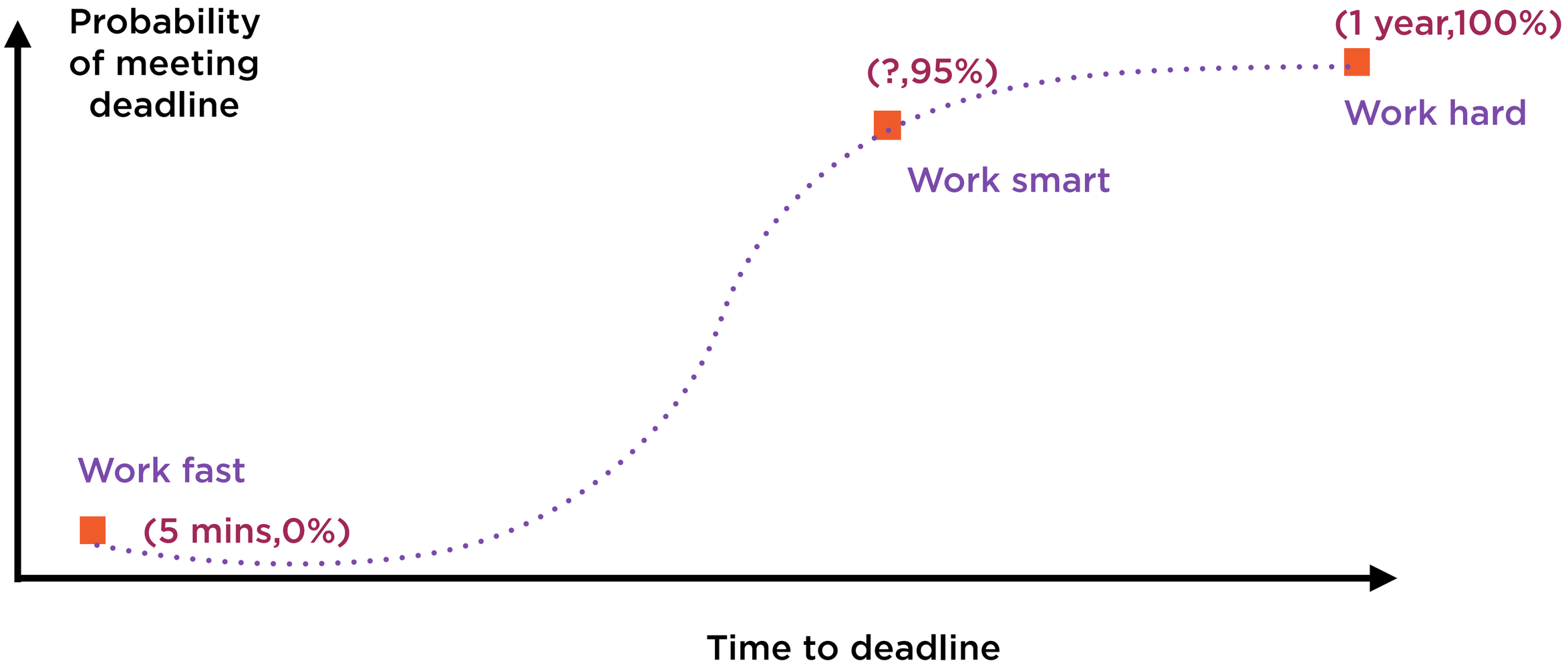
# Working Hard, Fast, Smart



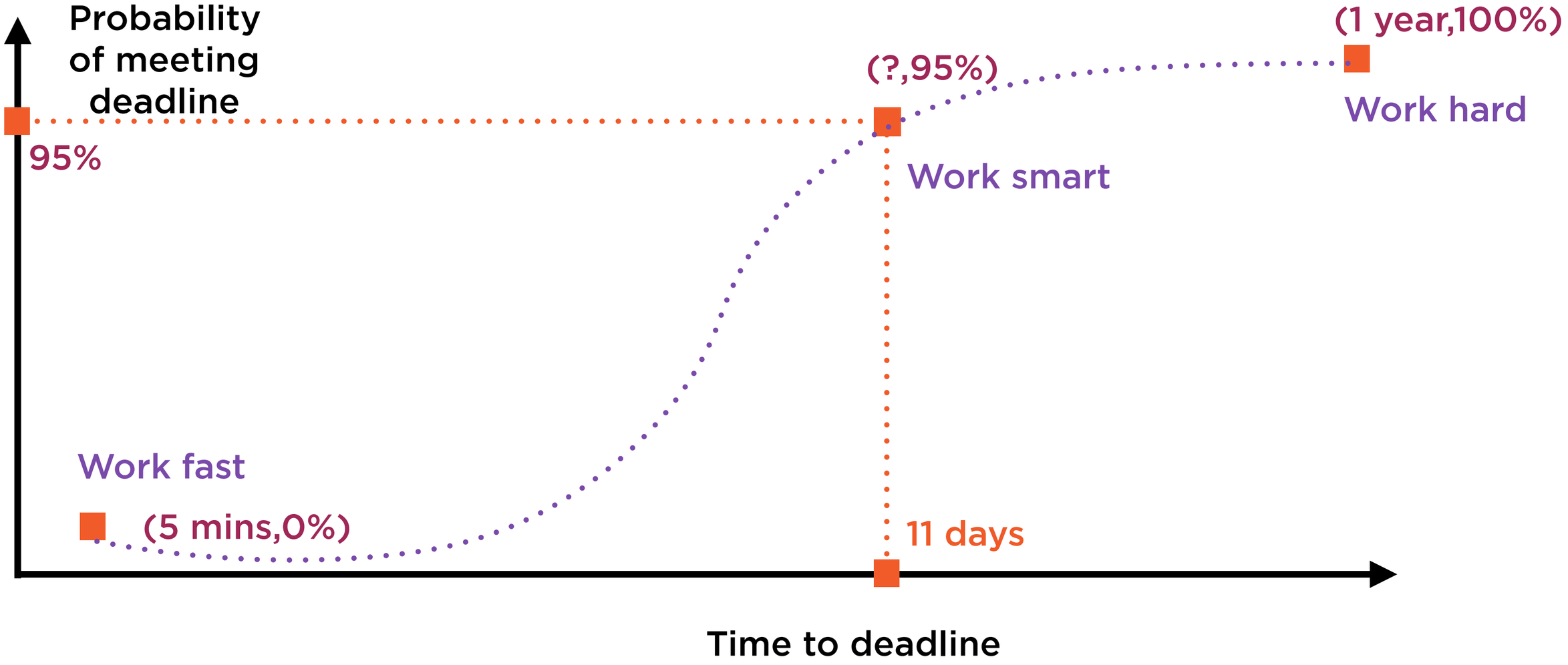
# Working Hard, Fast, Smart



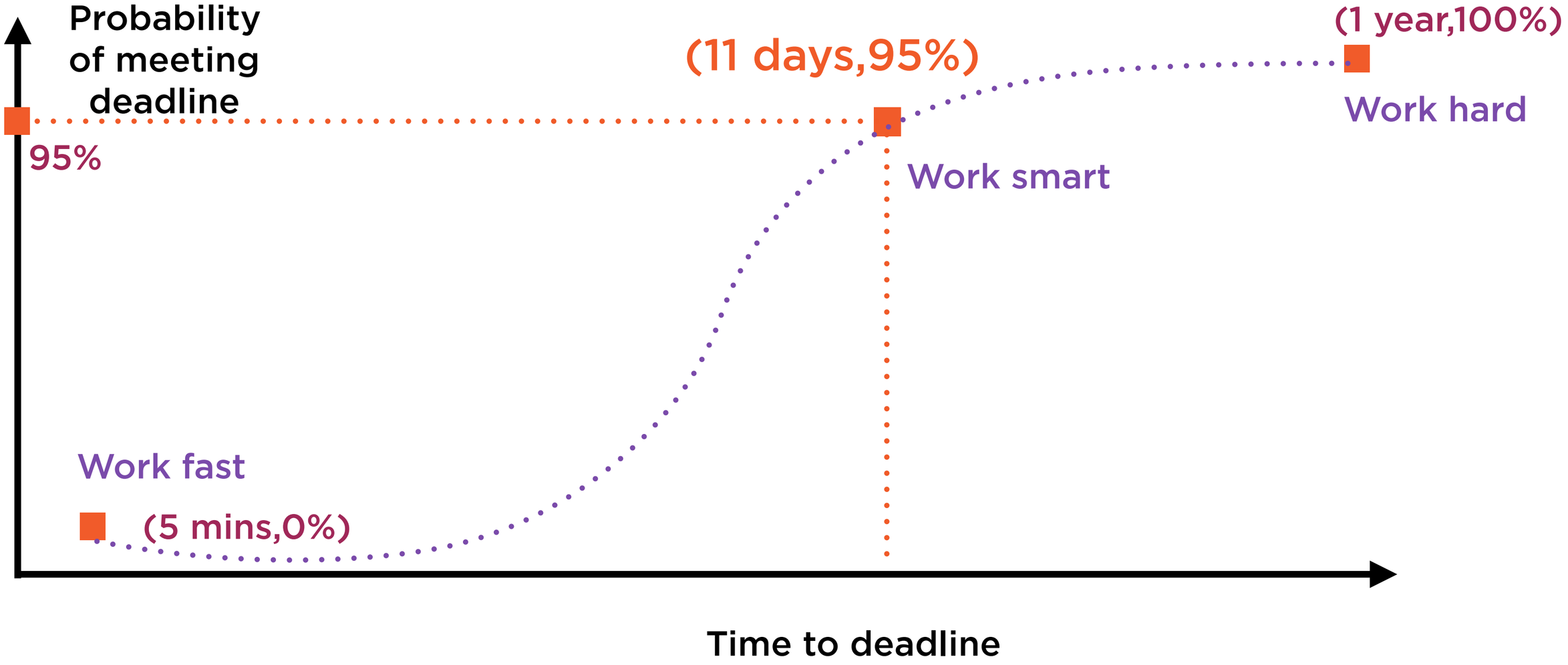
# Working Hard, Fast, Smart



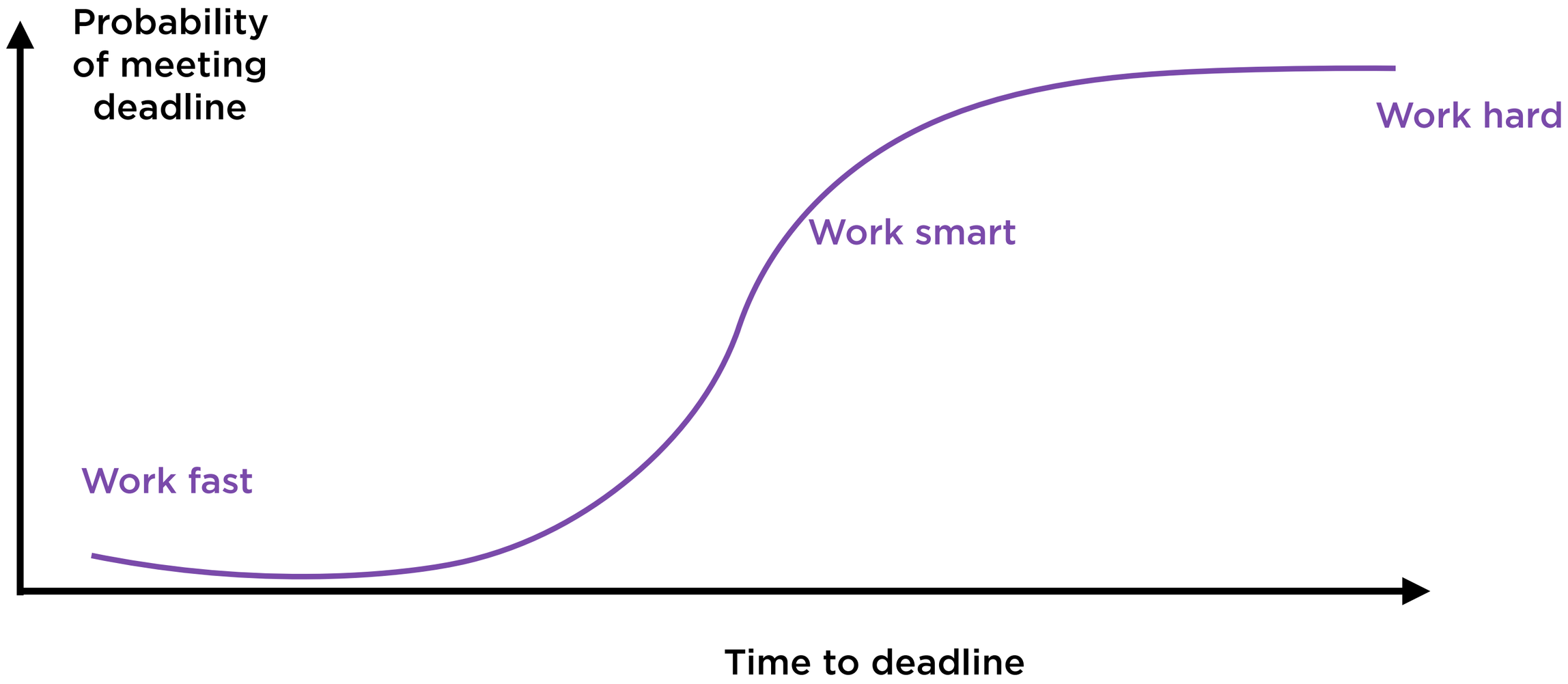
# Working Hard, Fast, Smart



# Working Hard, Fast, Smart

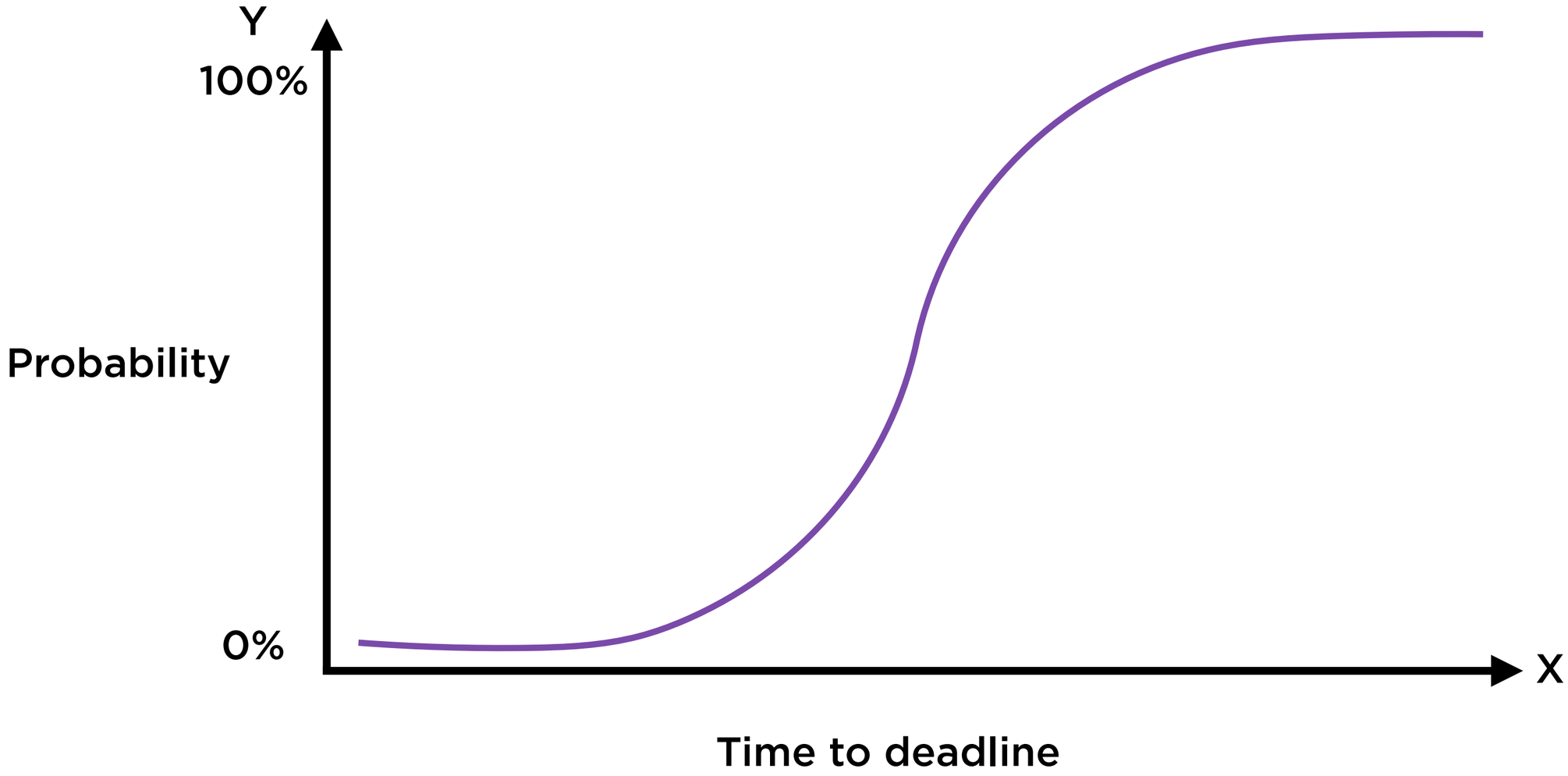


# Working Hard, Fast, Smart

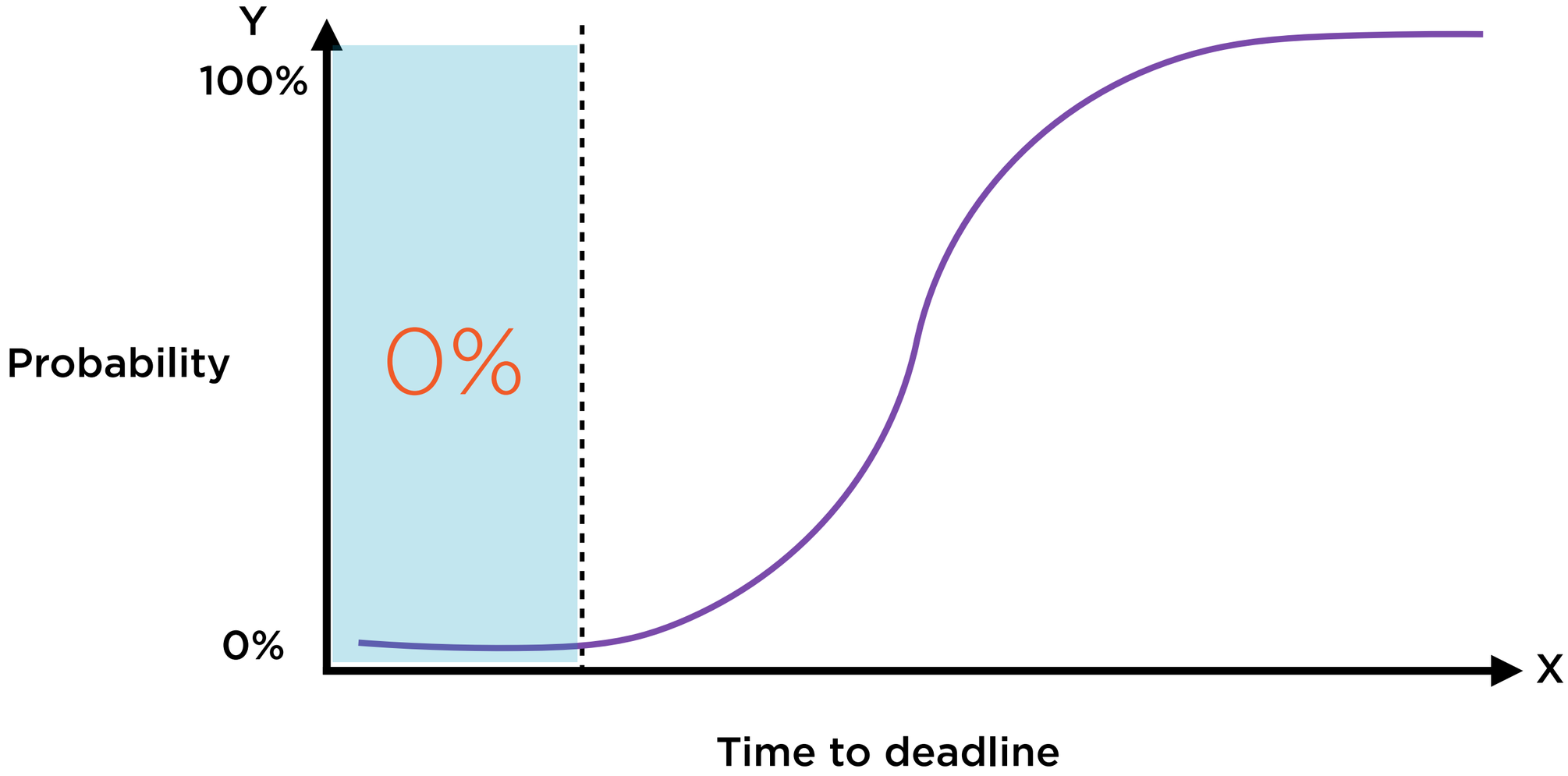


Logistic Regression helps find how probabilities are changed by actions

# Working Smart with Logistic Regression

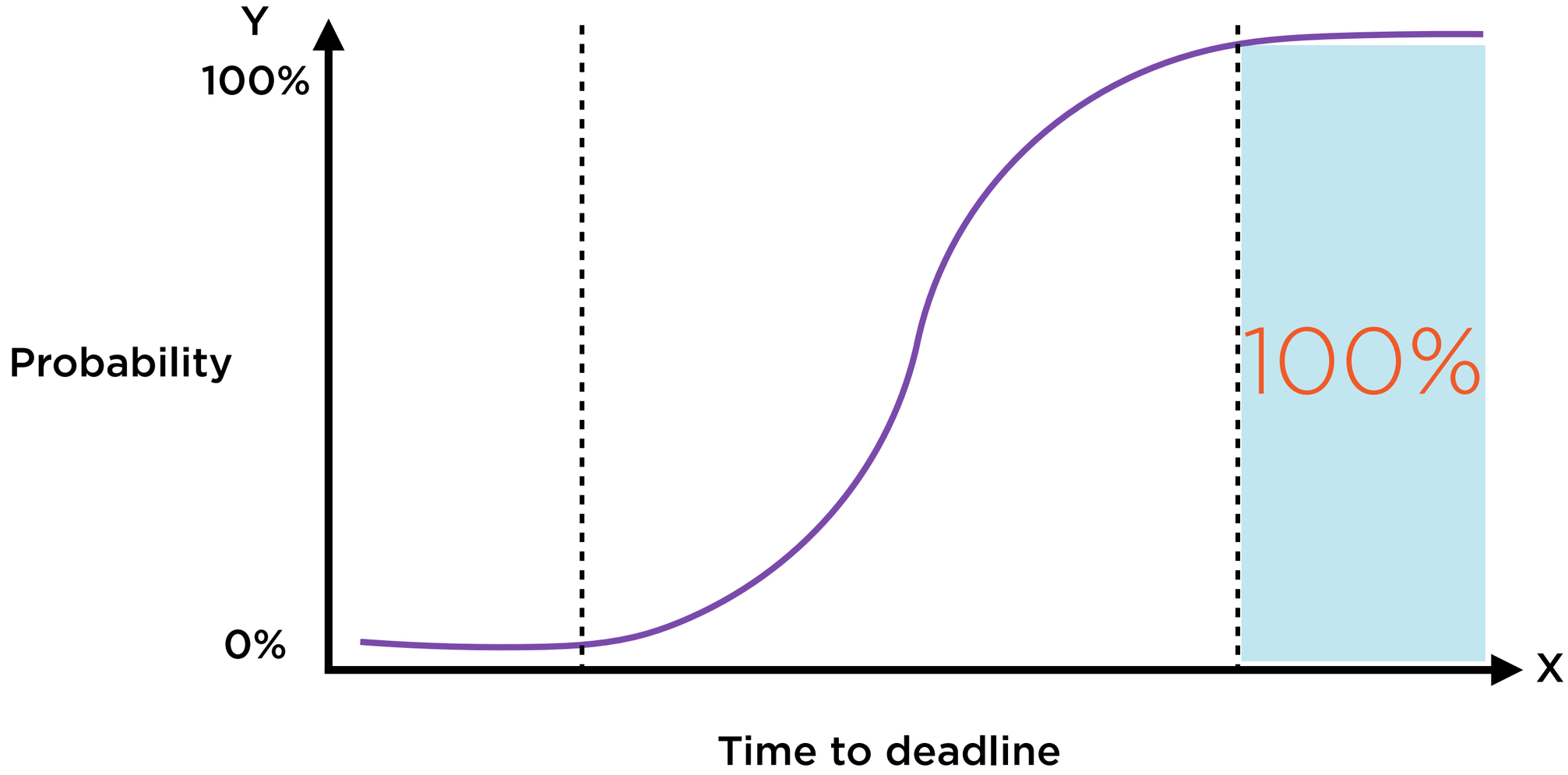


# Working Smart with Logistic Regression



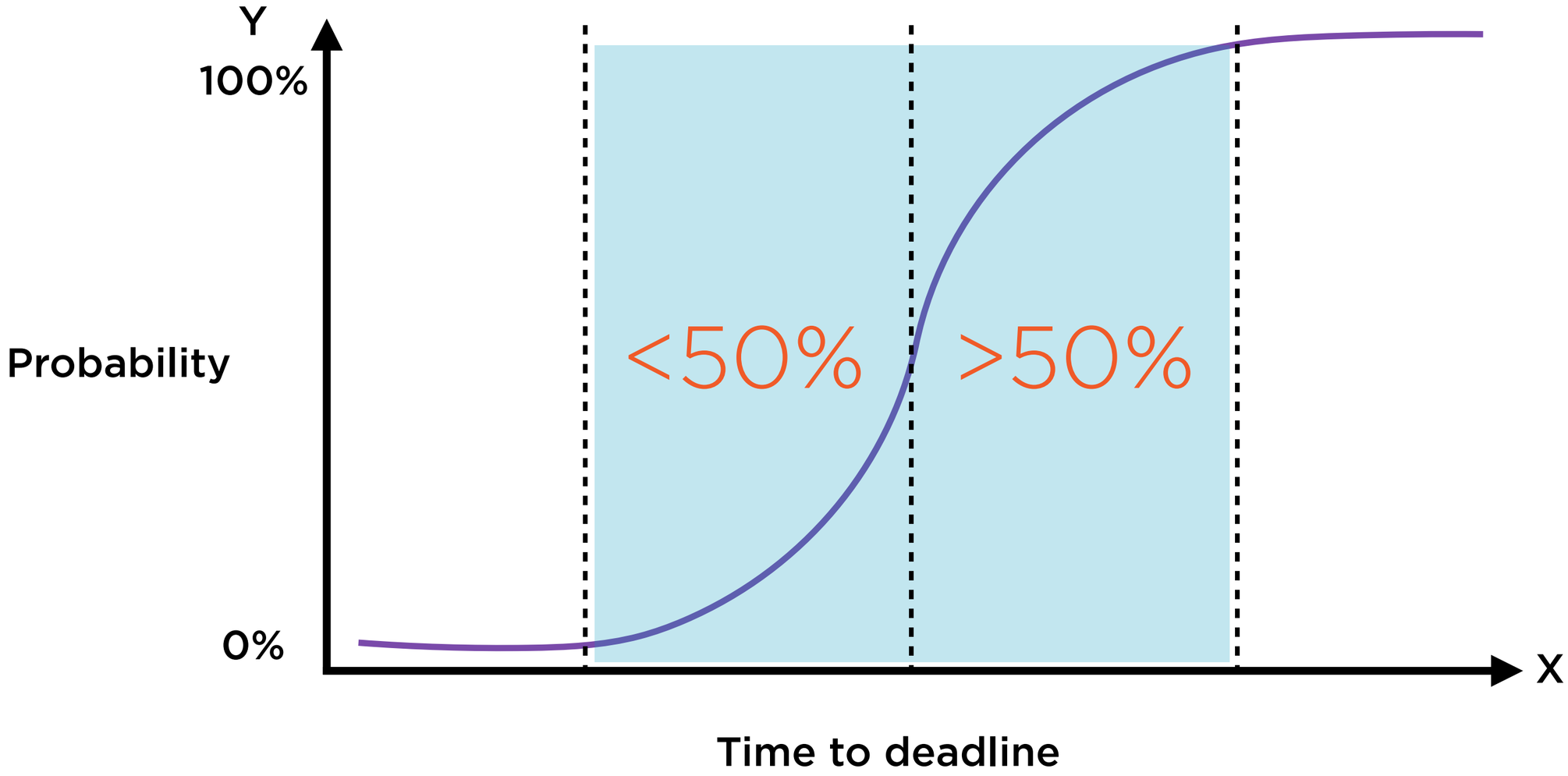

**Start too late, and you'll definitely miss**

# Working Smart with Logistic Regression

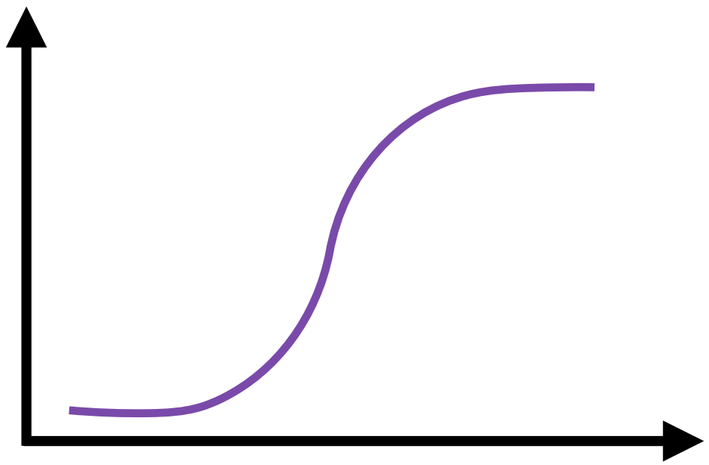



**Start too early, and you'll definitely make it**

# Working Smart with Logistic Regression




**Working smart is knowing when to start**



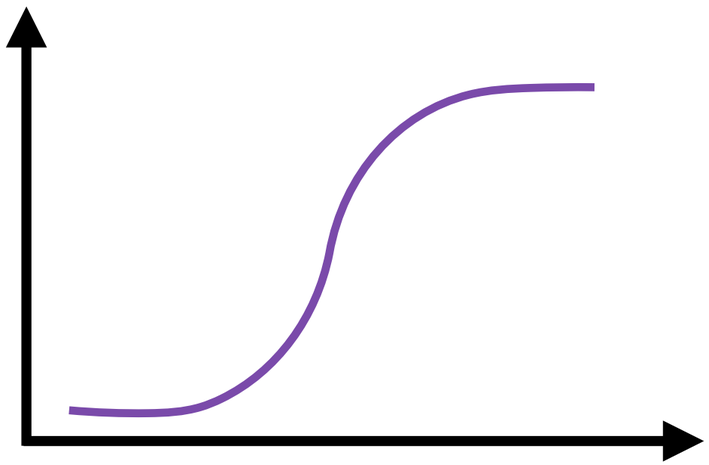
**Y-axis: probability of meeting deadline**

**X-axis: time to deadline**

**Meeting or missing deadline is binary**

**Probability curve flattens at ends**

- floor of 0
- ceiling of 1



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

x: start time before deadline

p(y) : probability of y = 1

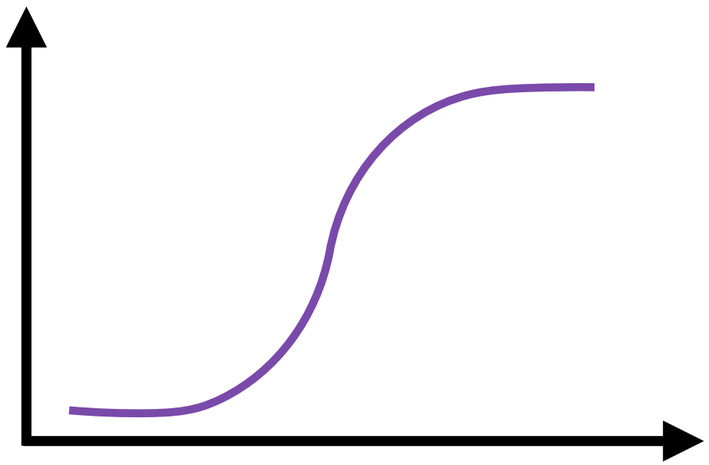
$$p(y_i) = \frac{1}{1 + e^{-(A+Bx_i)}}$$

**Logistic regression involves finding the "best fit" such curve**

- A is the intercept

- B is the regression coefficient

***(e is the constant 2.71828)***

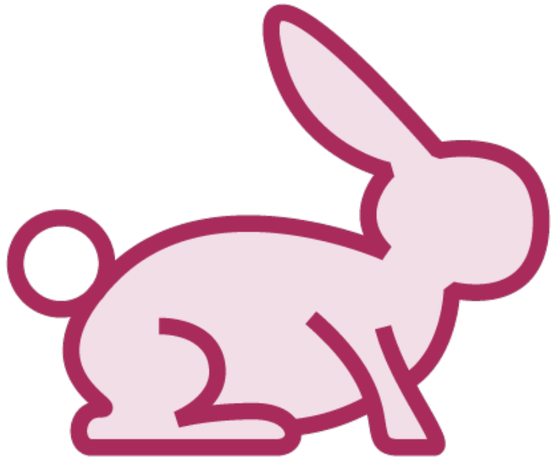


S-curves are widely studied, well understood

Logistic regression uses S-curve to estimate probabilities

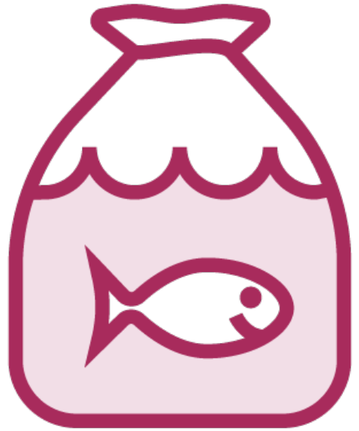
$$p(y) = \frac{1}{1 + e^{-(A+Bx)}}$$

# Whales: Fish or Mammals



**Mammal**

**Member of the infraorder *Cetacea***



**Fish**

**Looks like a fish, swims like a fish, moves like a fish**

# ML-based Binary Classifier

**Corpus**

**Classification Algorithm**

**ML-based Classifier**

# ML-based Predictor

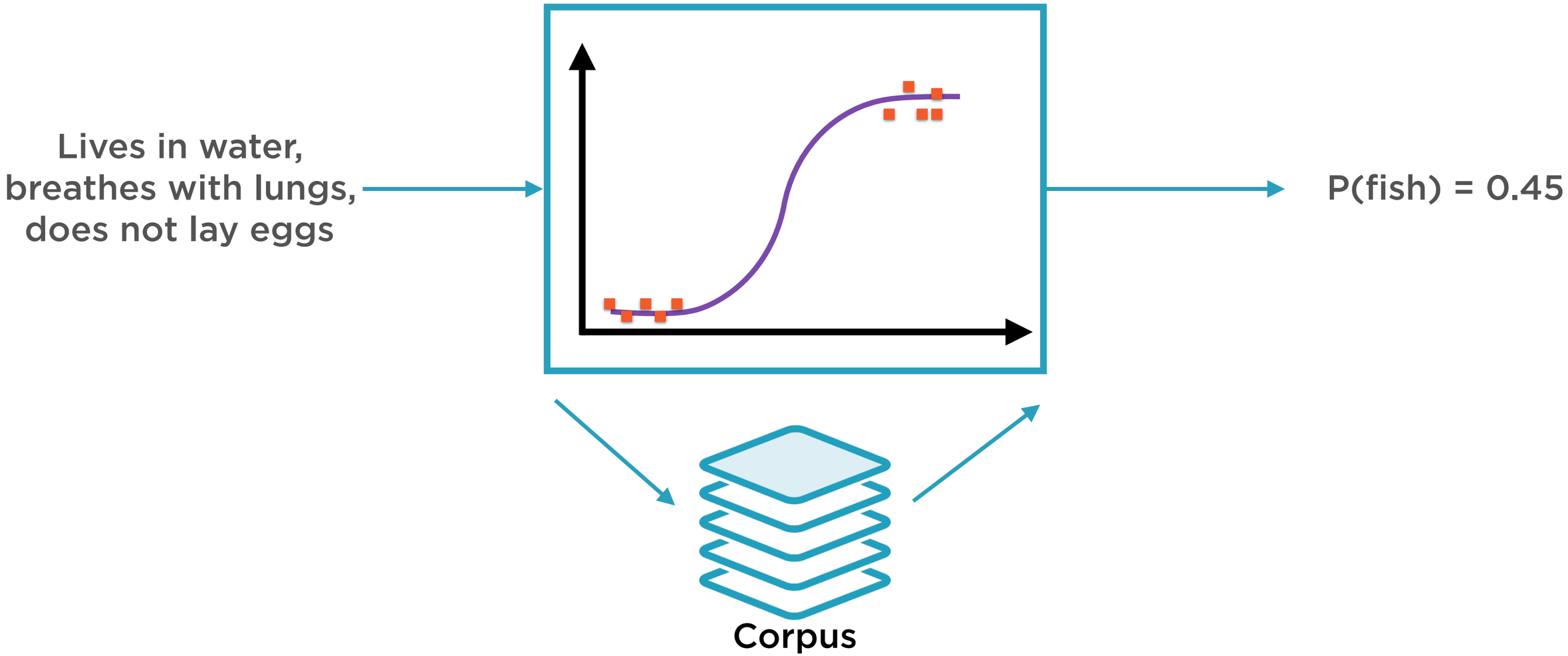
Corpus

Logistic Regression

ML-based Predictor

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

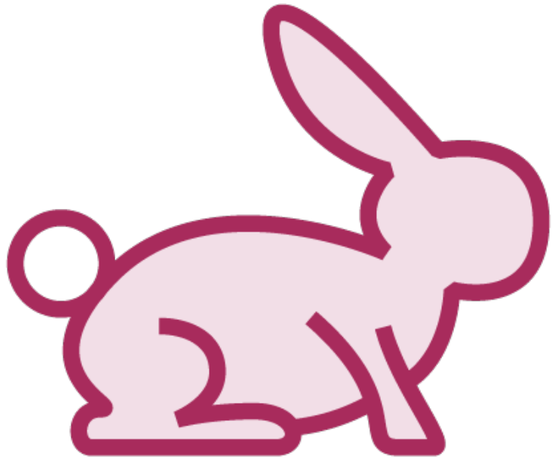
# ML-based Predictor



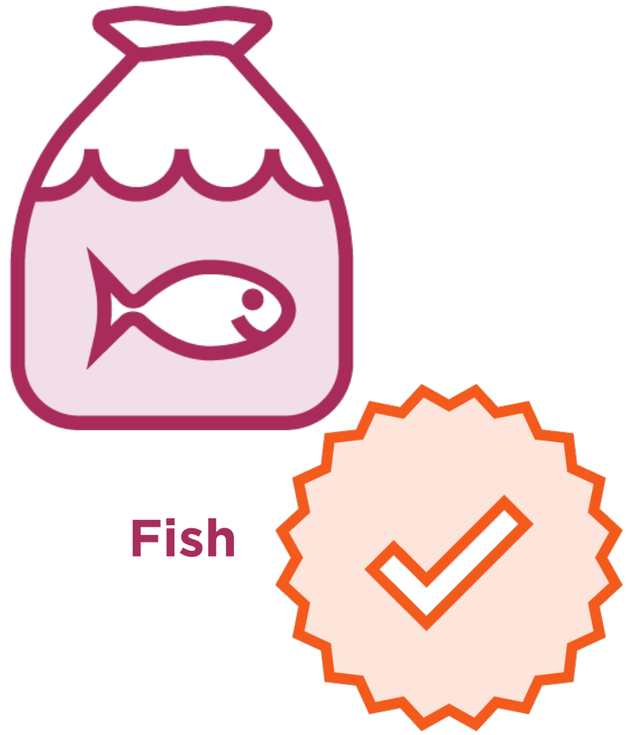
# Applying Logistic Regression


**Probability of whales being fish < $P_{threshold}$**

# Applying Logistic Regression



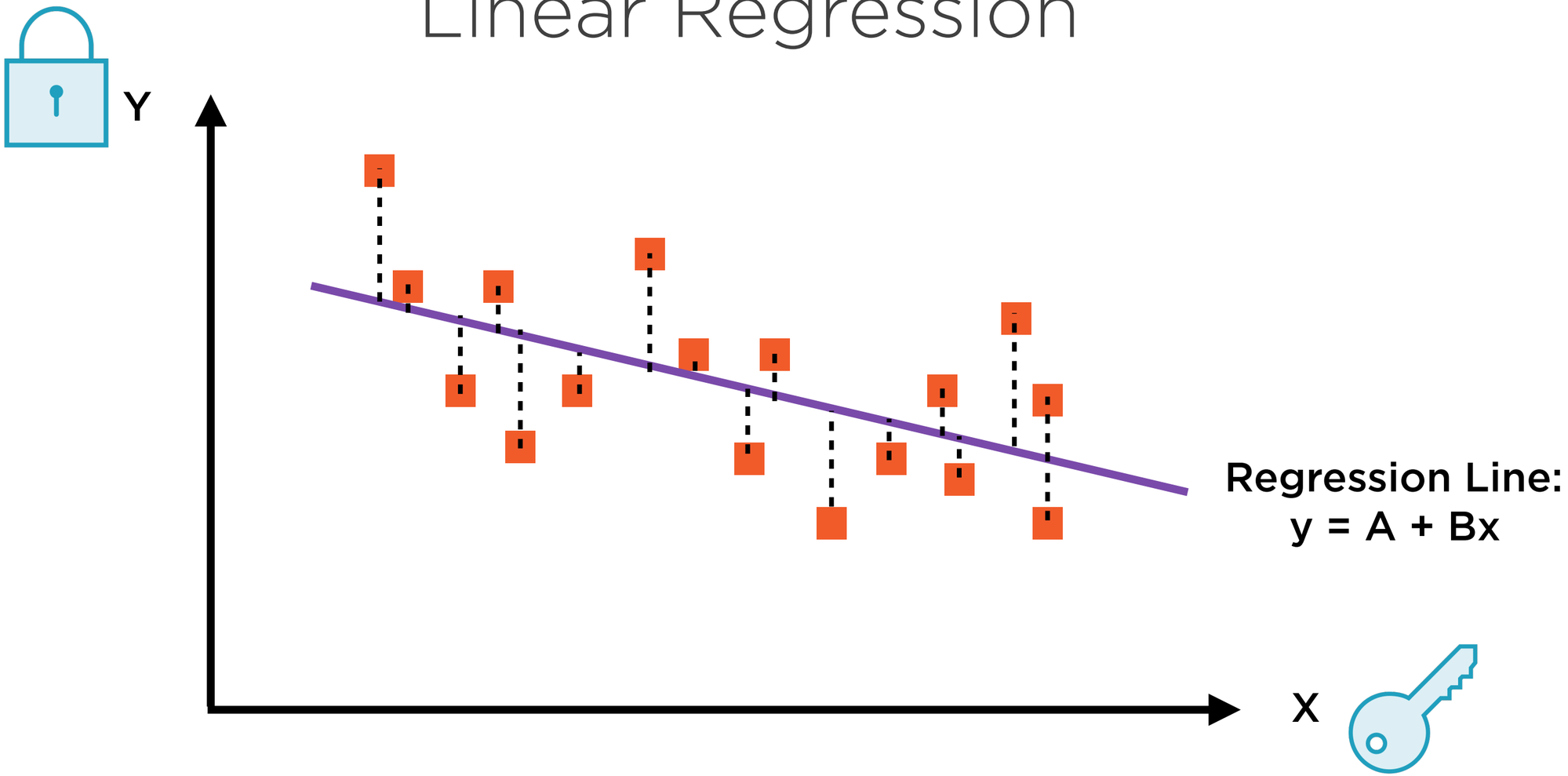
**Mammal**



**Fish**

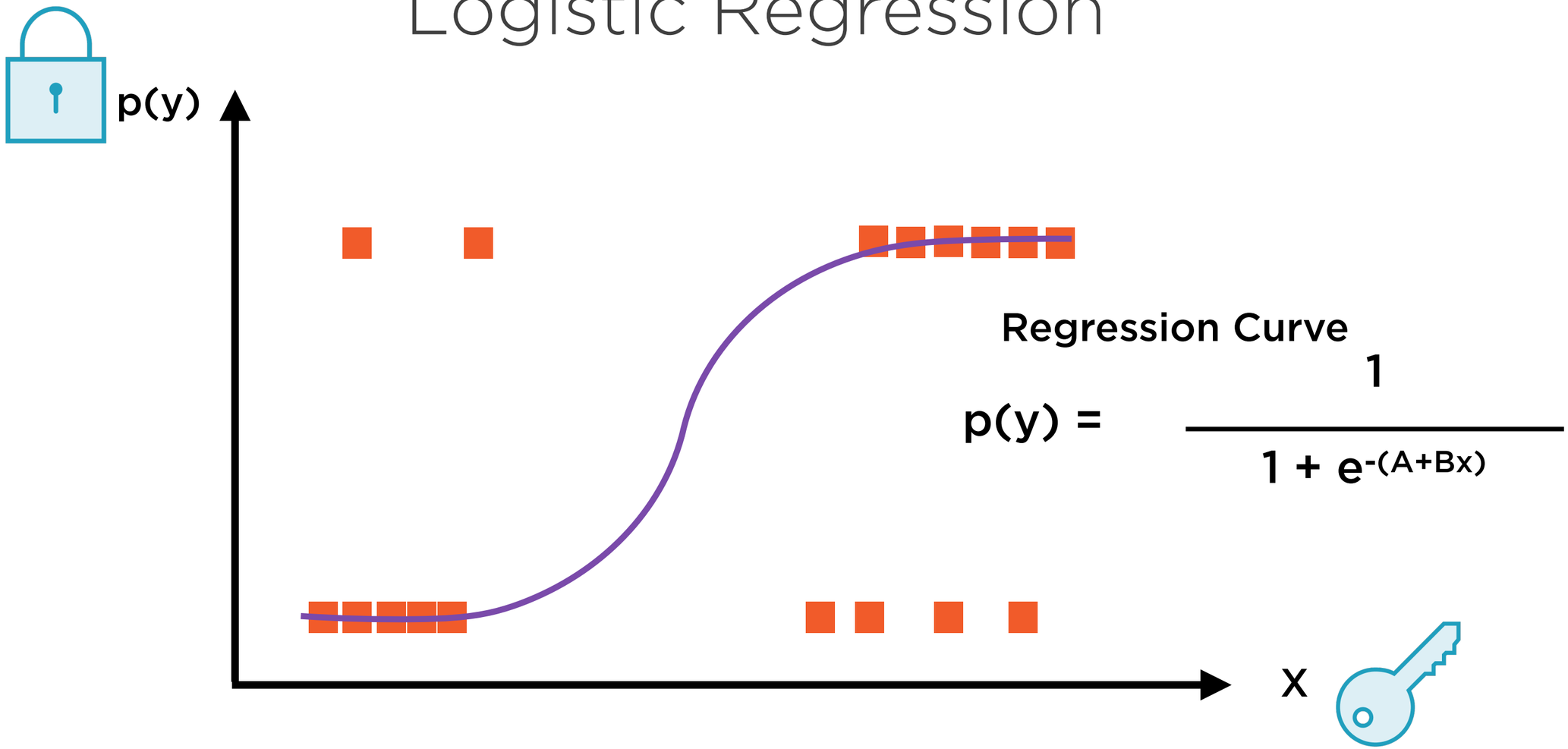
**Probability of whales being fish > $P_{threshold}$**

# Linear Regression




Finding the best fit line through these points

# Logistic Regression




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

# 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

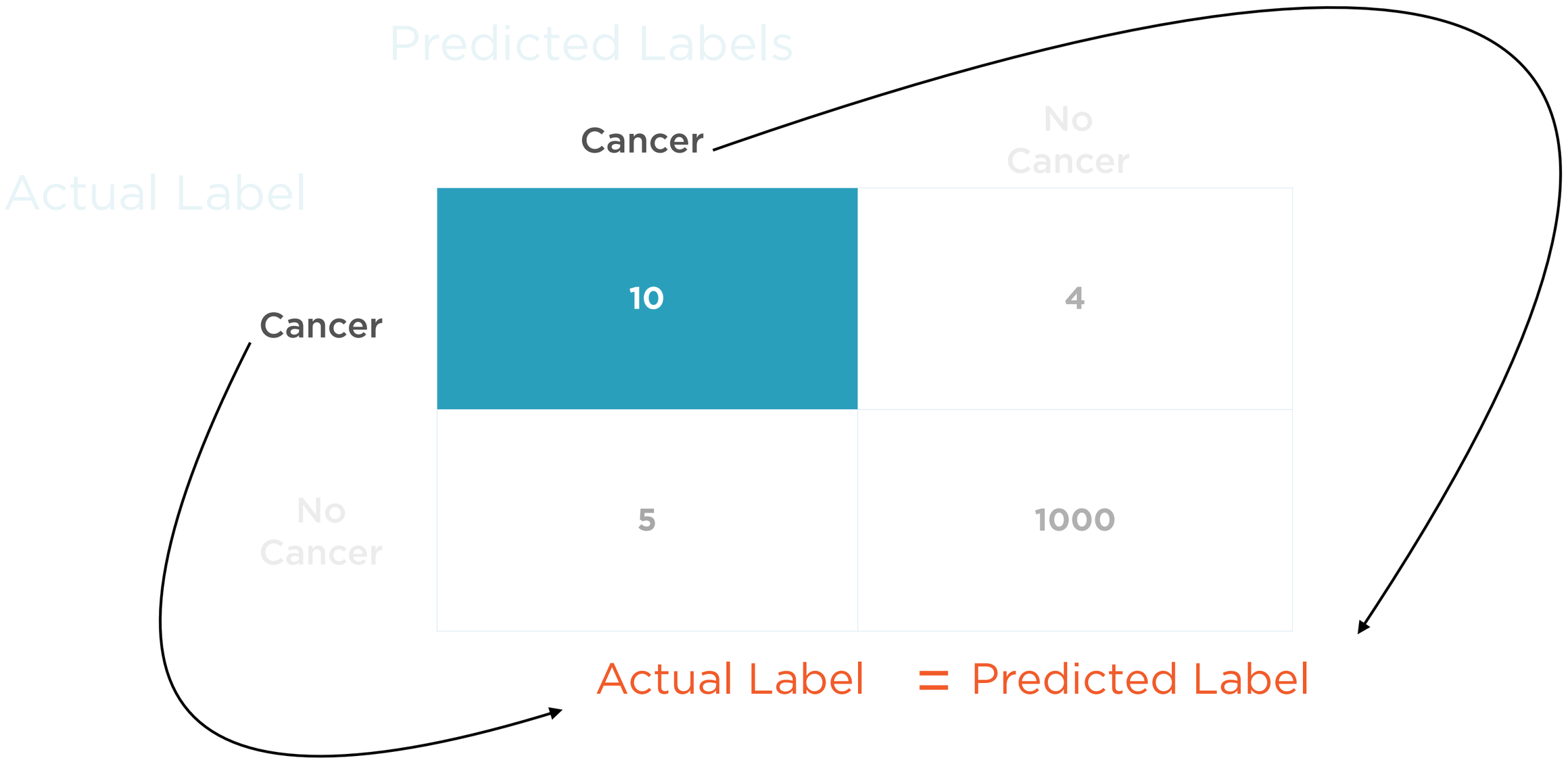
| | Cancer | No Cancer |
|---|---|---|
| Cancer | **10 instances** | **4 instances** |
| No Cancer | **5 instances** | **1000 instances** |

# Confusion Matrix

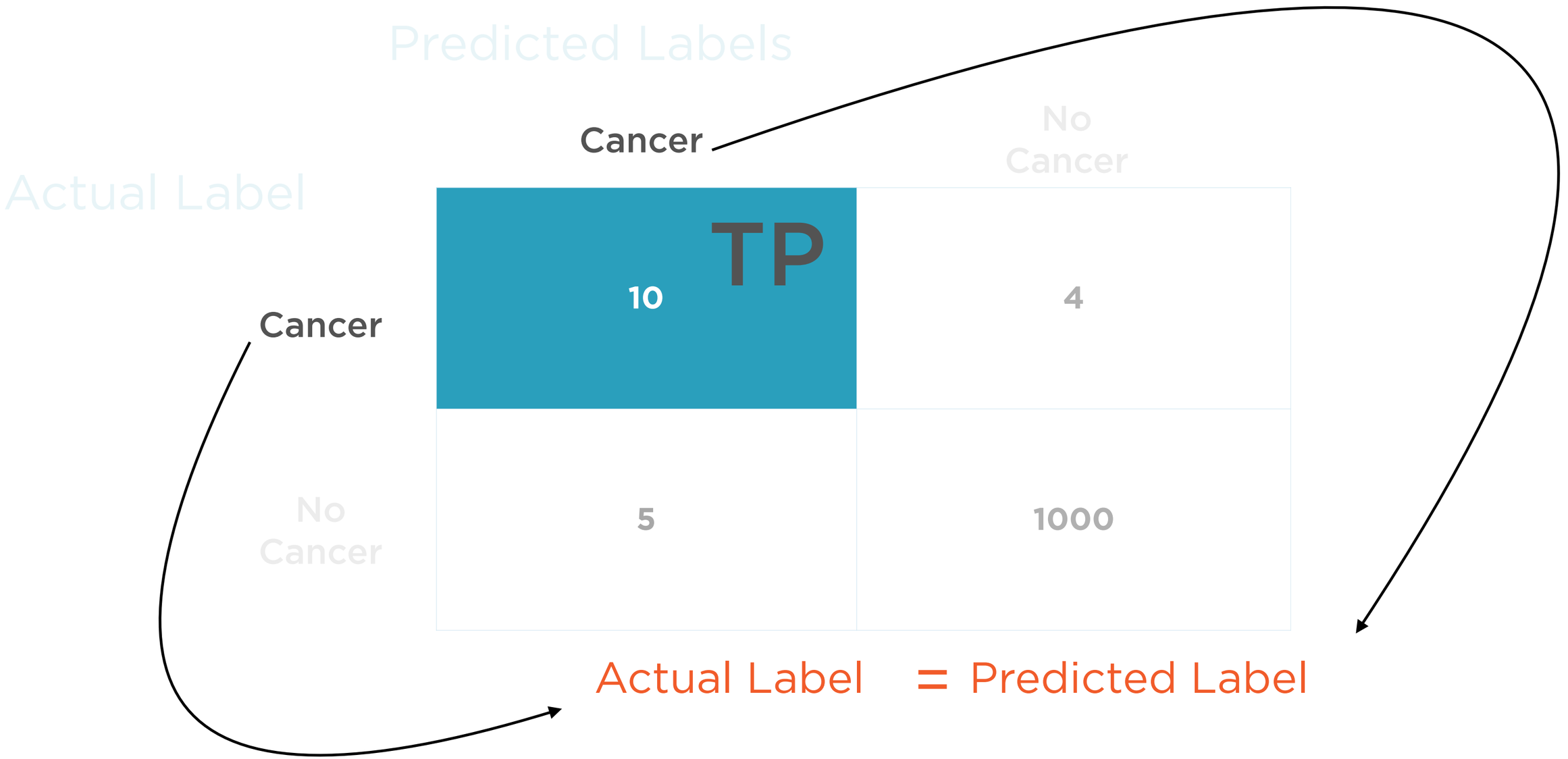
## Predicted Labels

| Actual Label | Cancer | No Cancer |
|---|---|---|
| Cancer | **10** | **4** |
| No Cancer | **5** | **1000** |

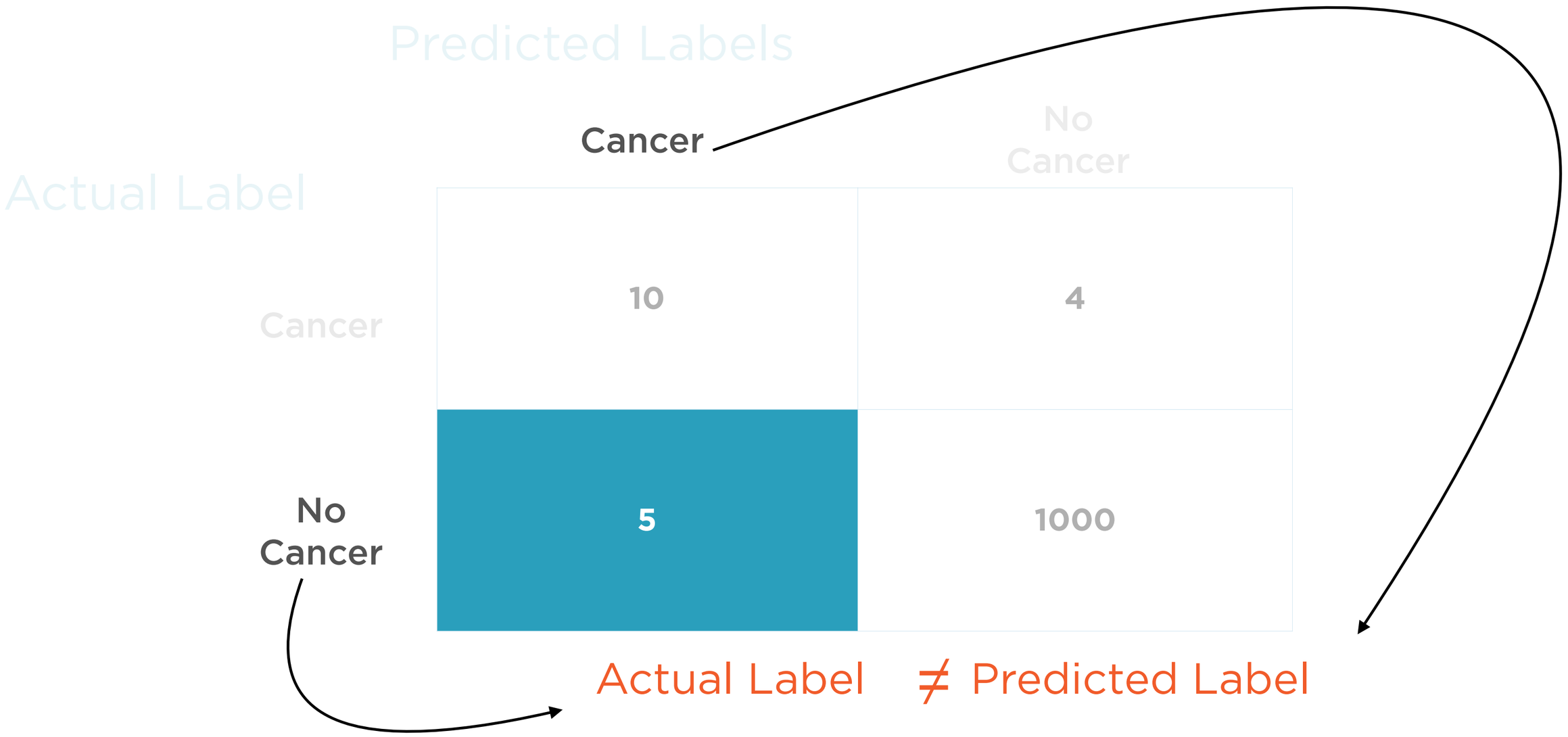
# True Positive



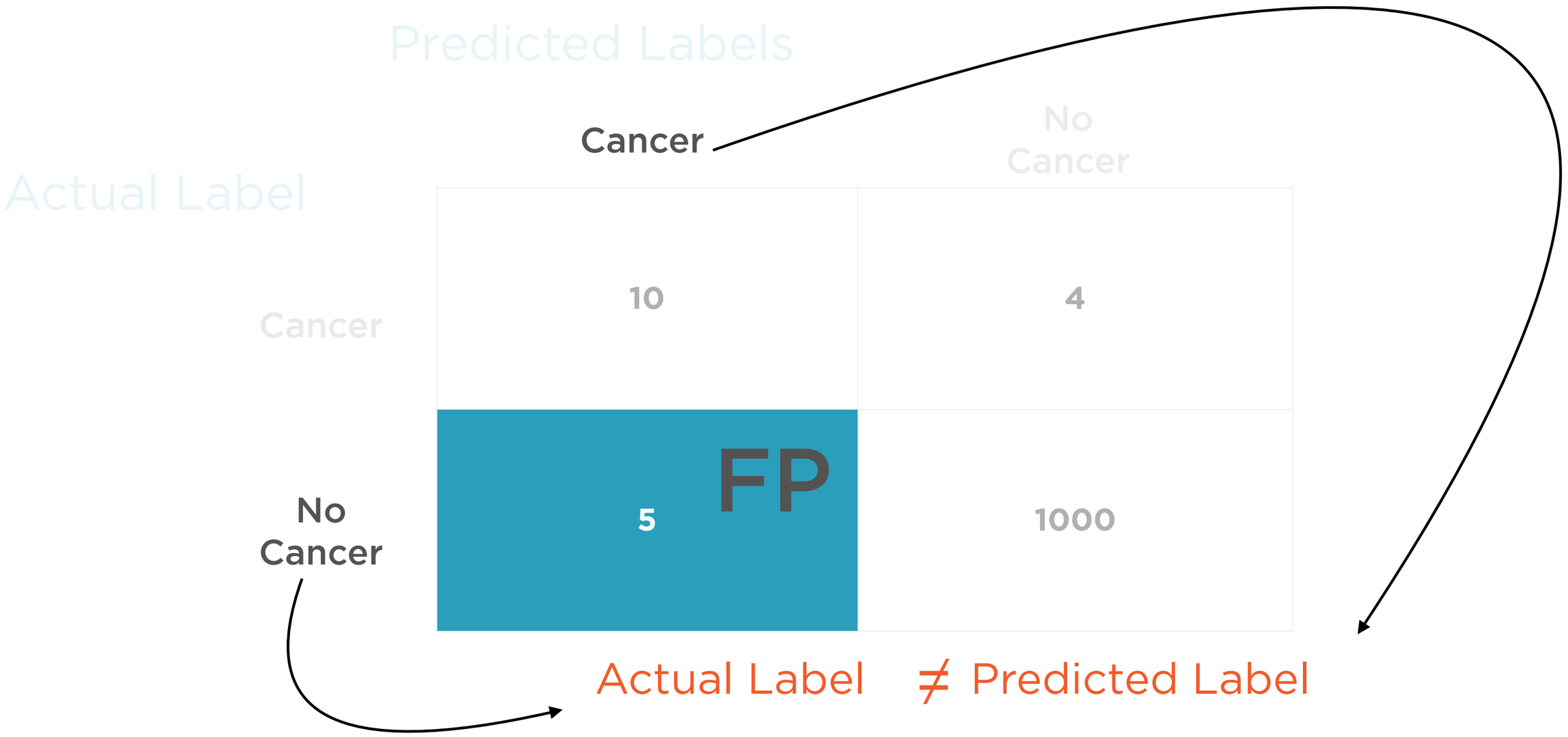
# True Positive



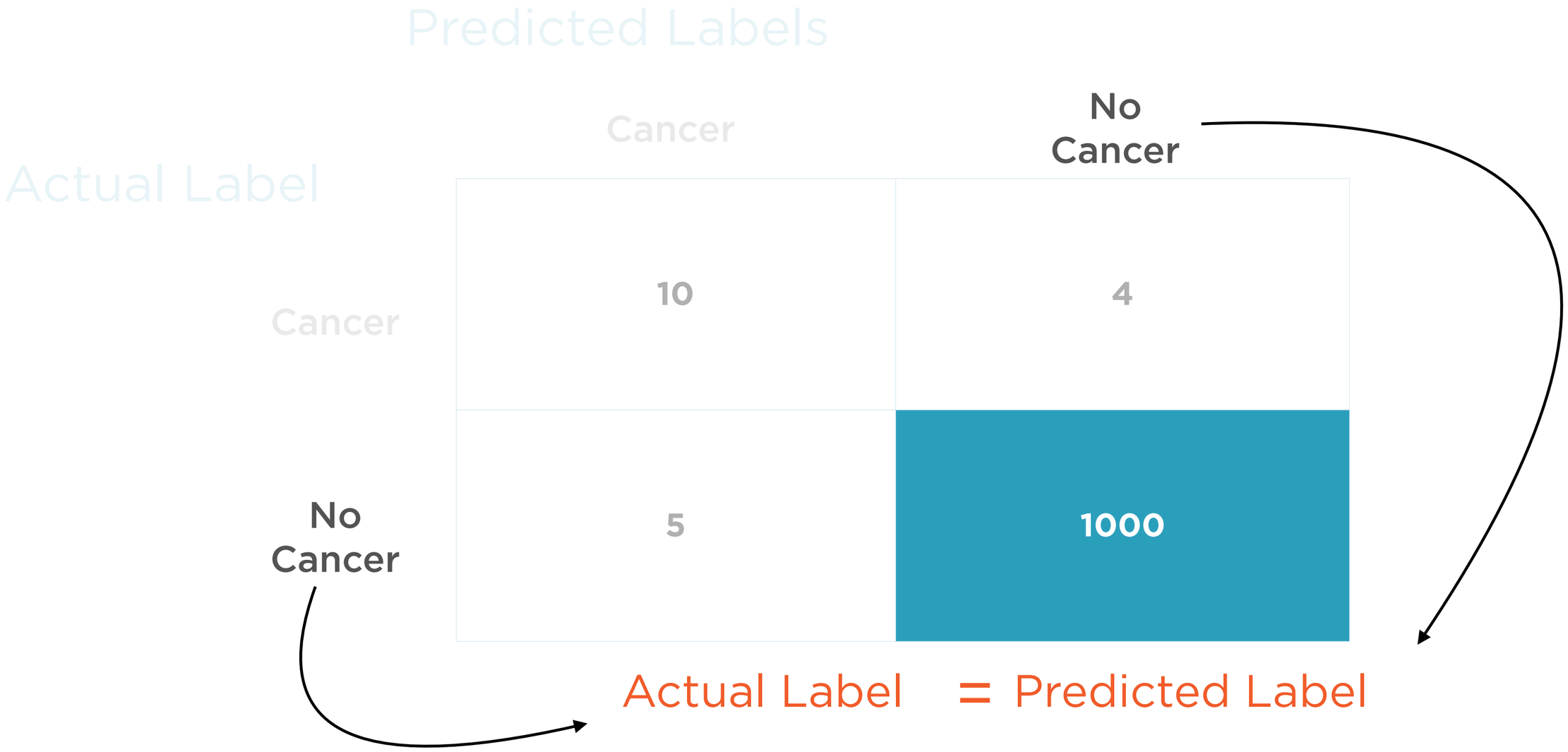
# False Positive



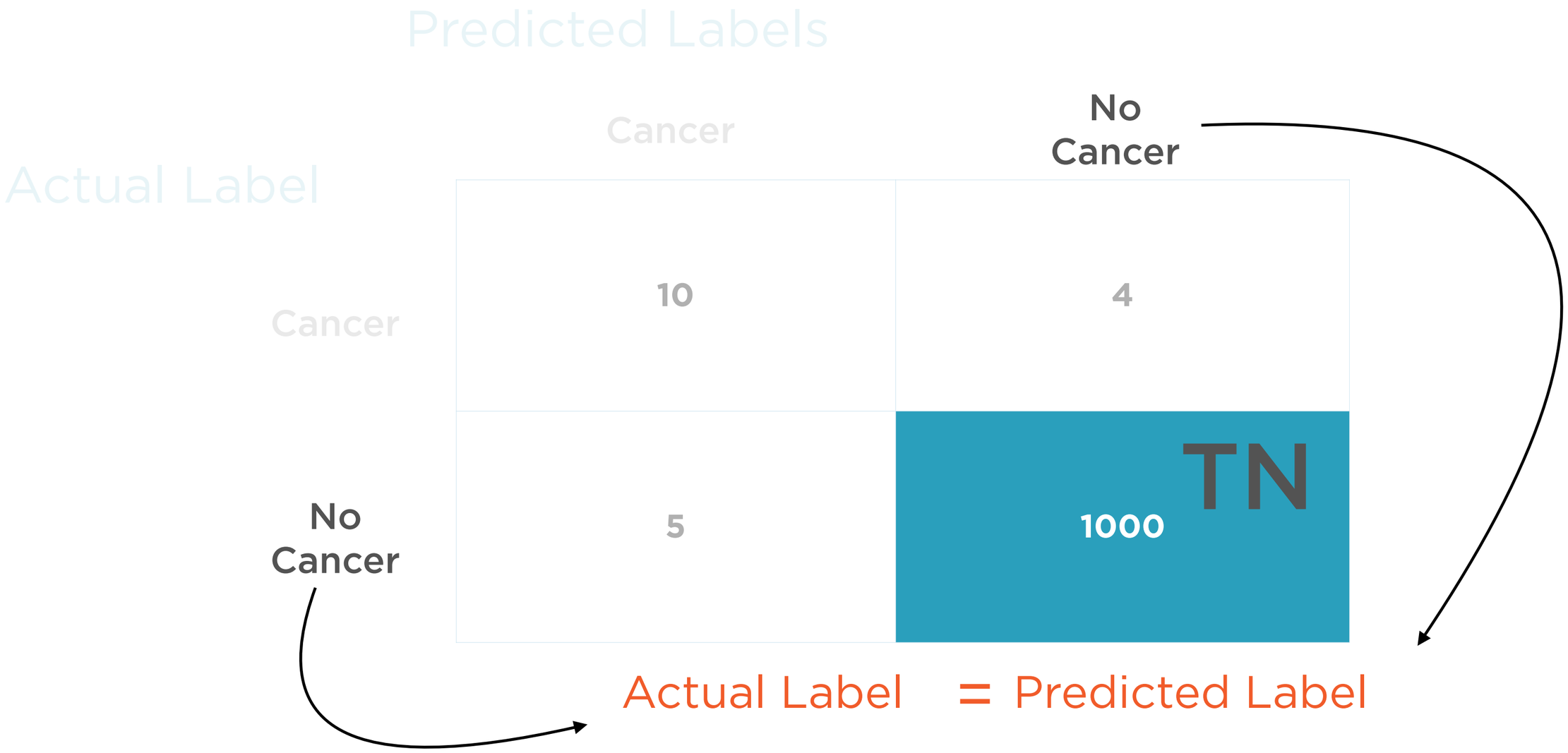
# False Positive



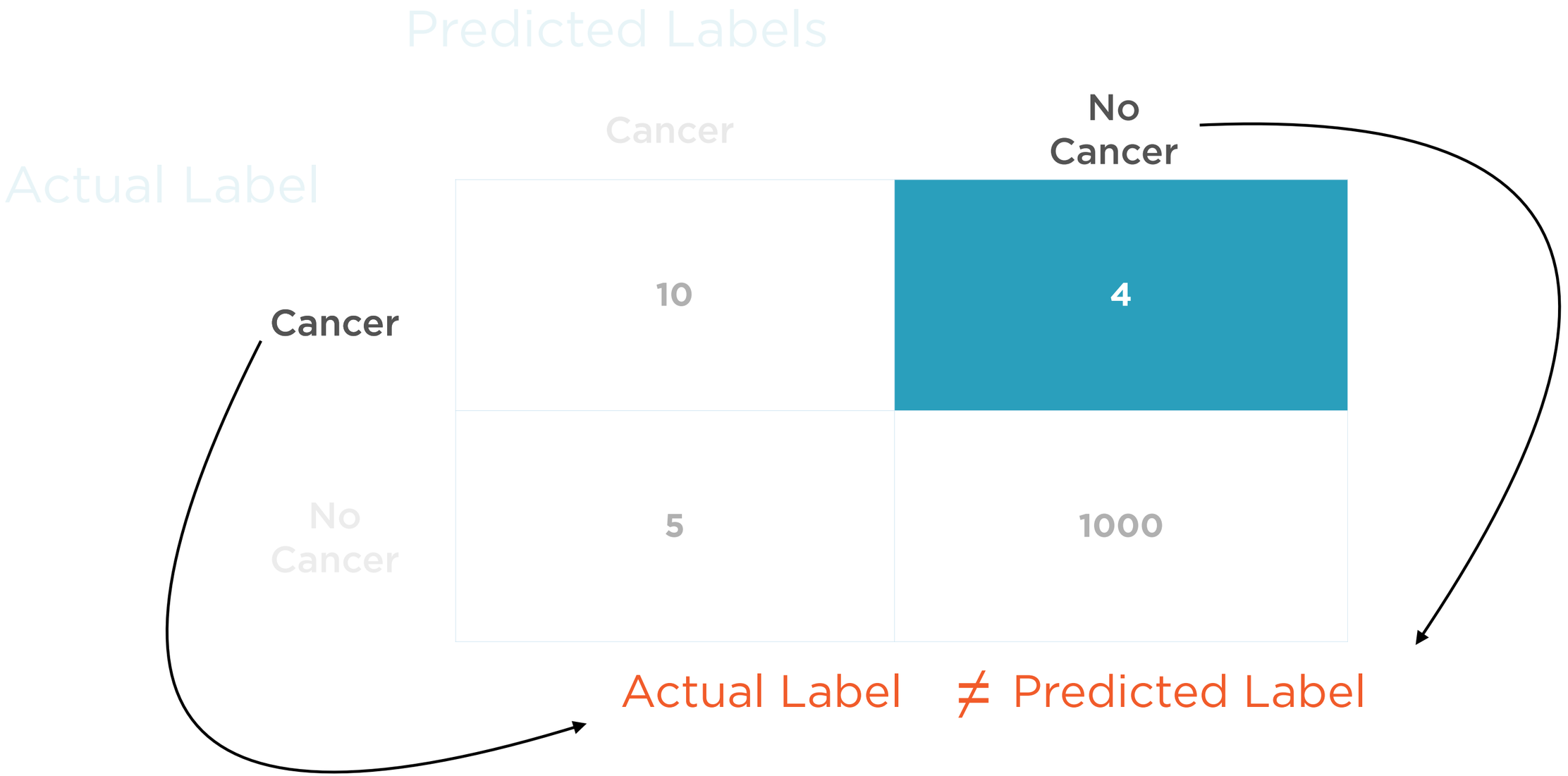
# True Negative



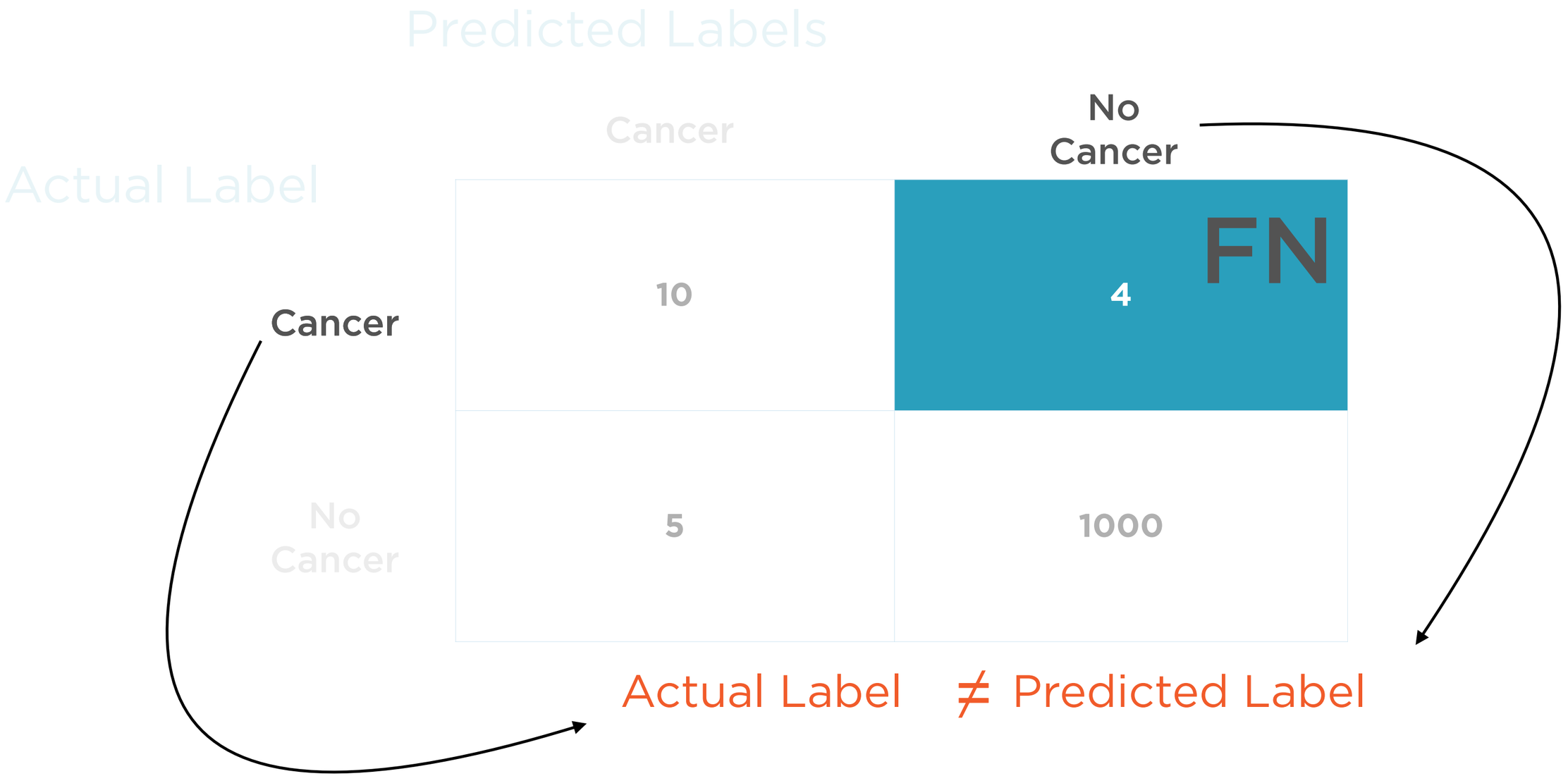
# True Negative



# False Negative



# False Negative



# Confusion Matrix

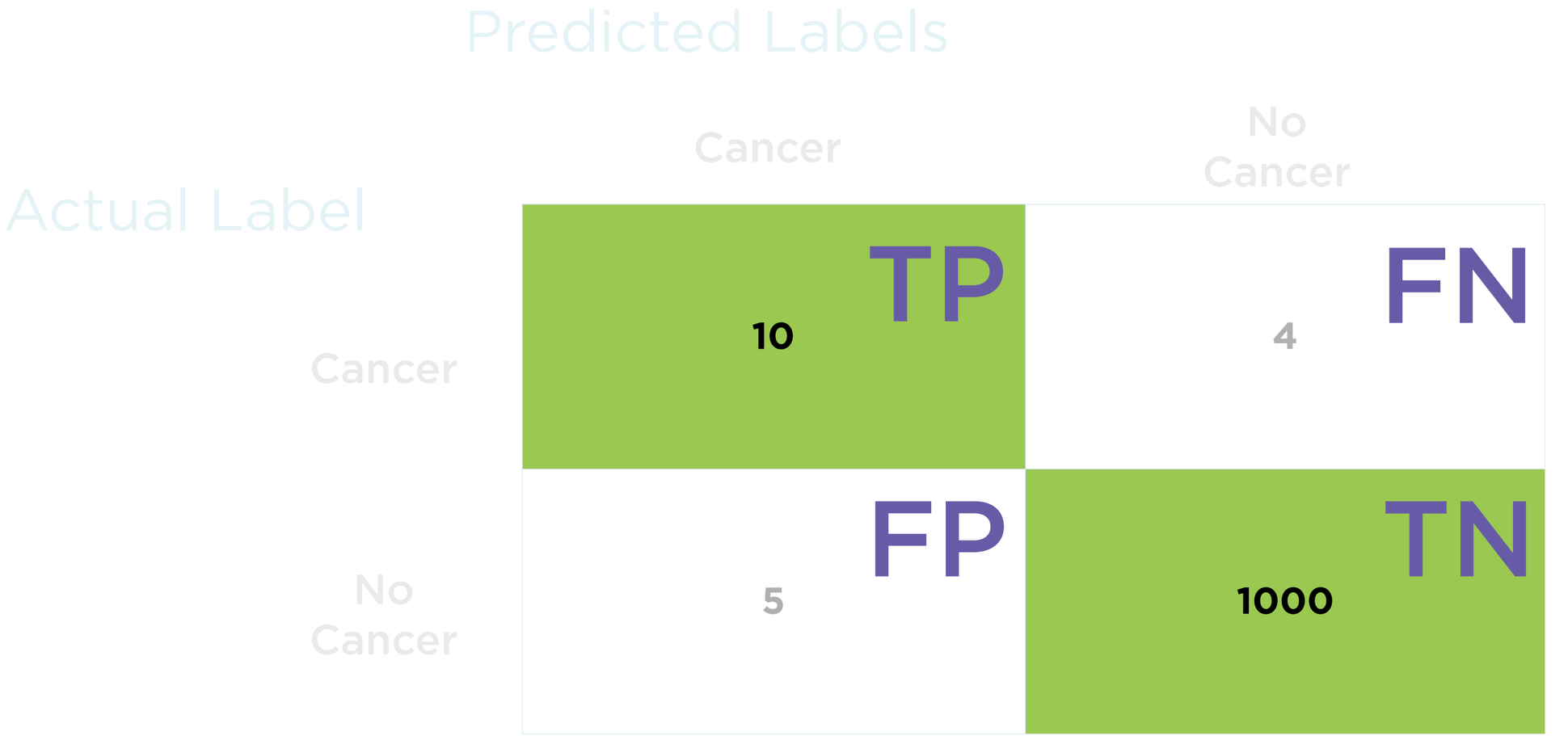
## Predicted Labels

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

# Accuracy

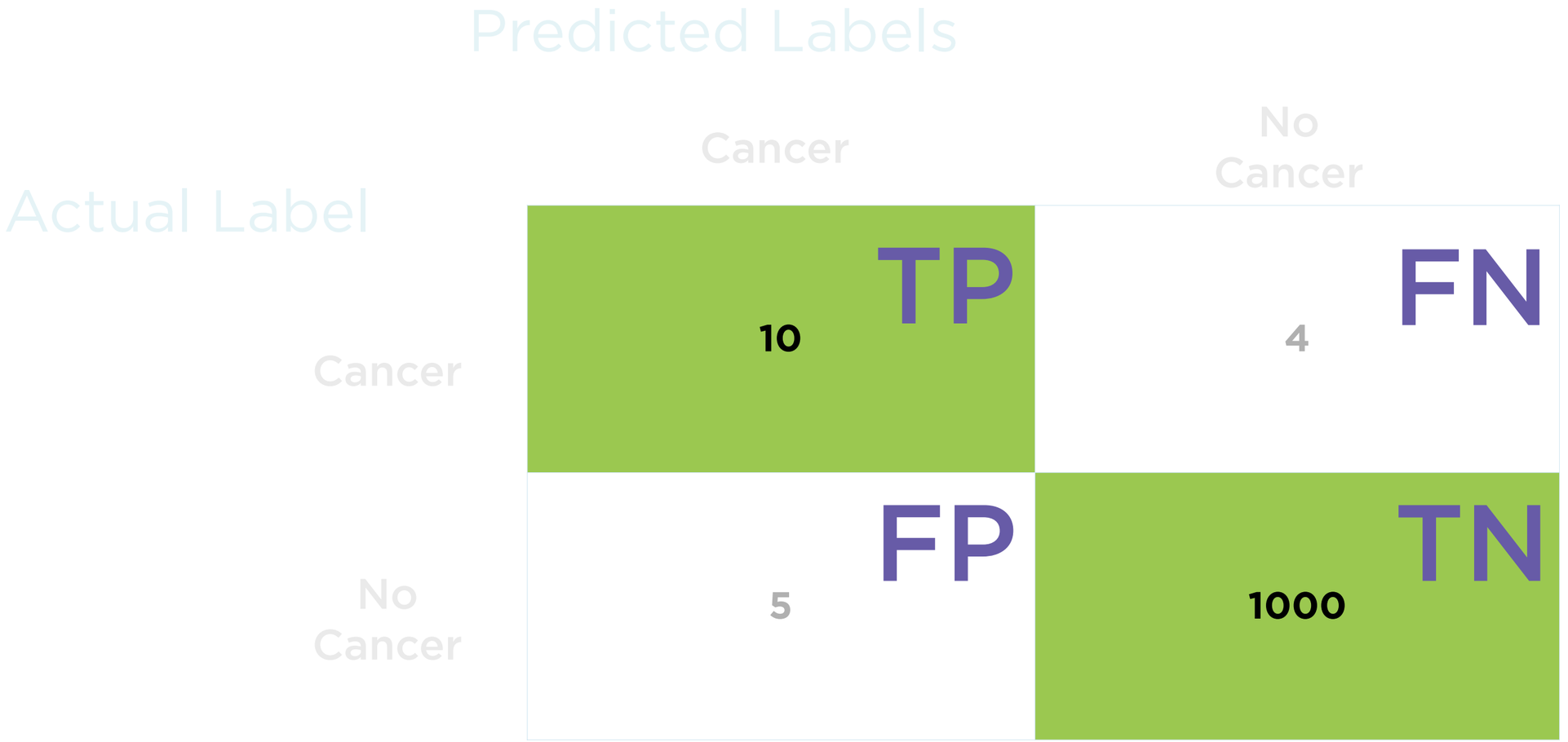
| Actual Label \ Predicted Labels | Cancer | No Cancer |
| --- | --- | --- |
| Cancer | 10 **TP** | 4 **FN** |
| No Cancer | 5 **FP** | 1000 **TN** |

# Accuracy




Actual Label = Predicted Label

# Accuracy




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

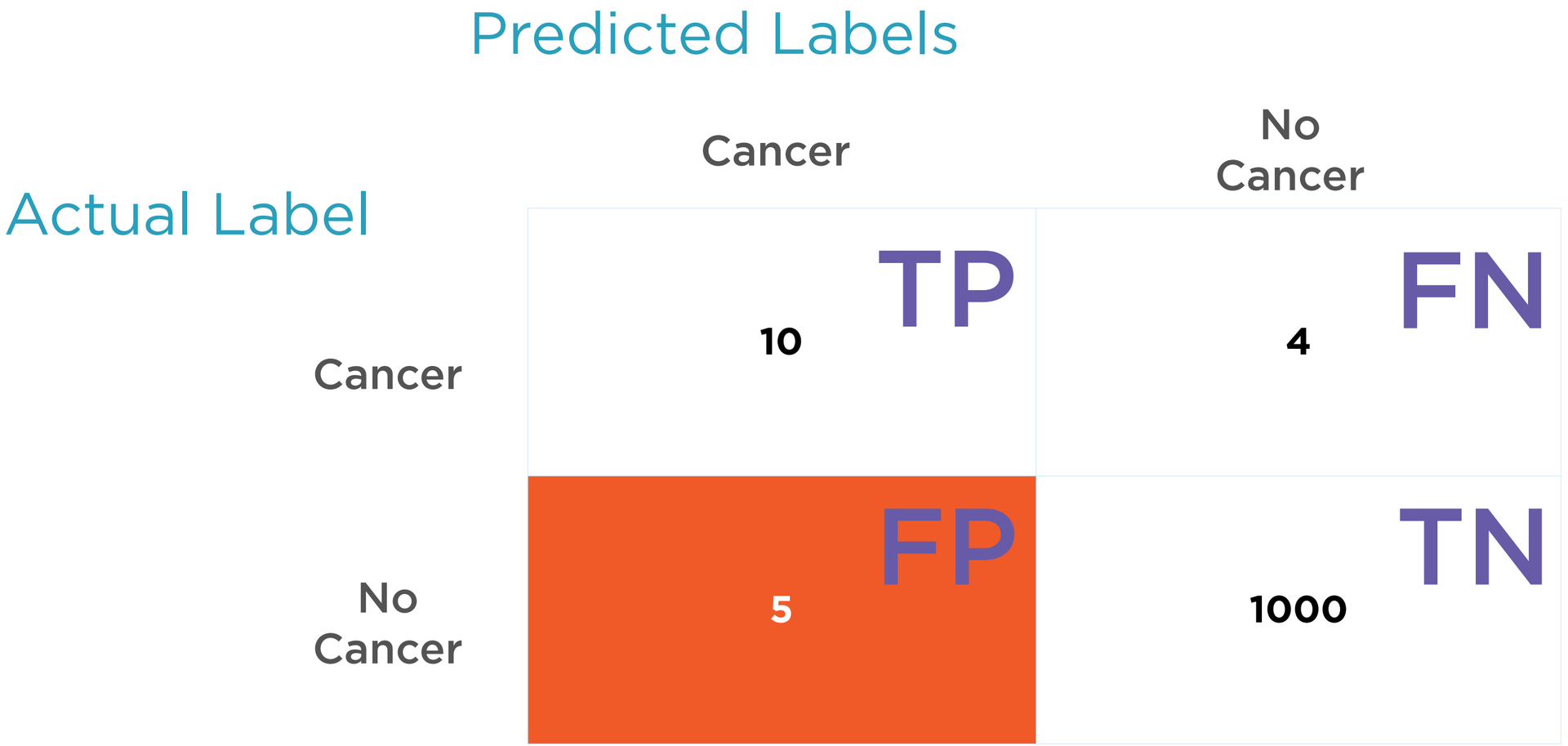
# Accuracy

**Accuracy = 99.12%**

**Classifier gets it right 99.12% of the time**

**But...**

# Accuracy

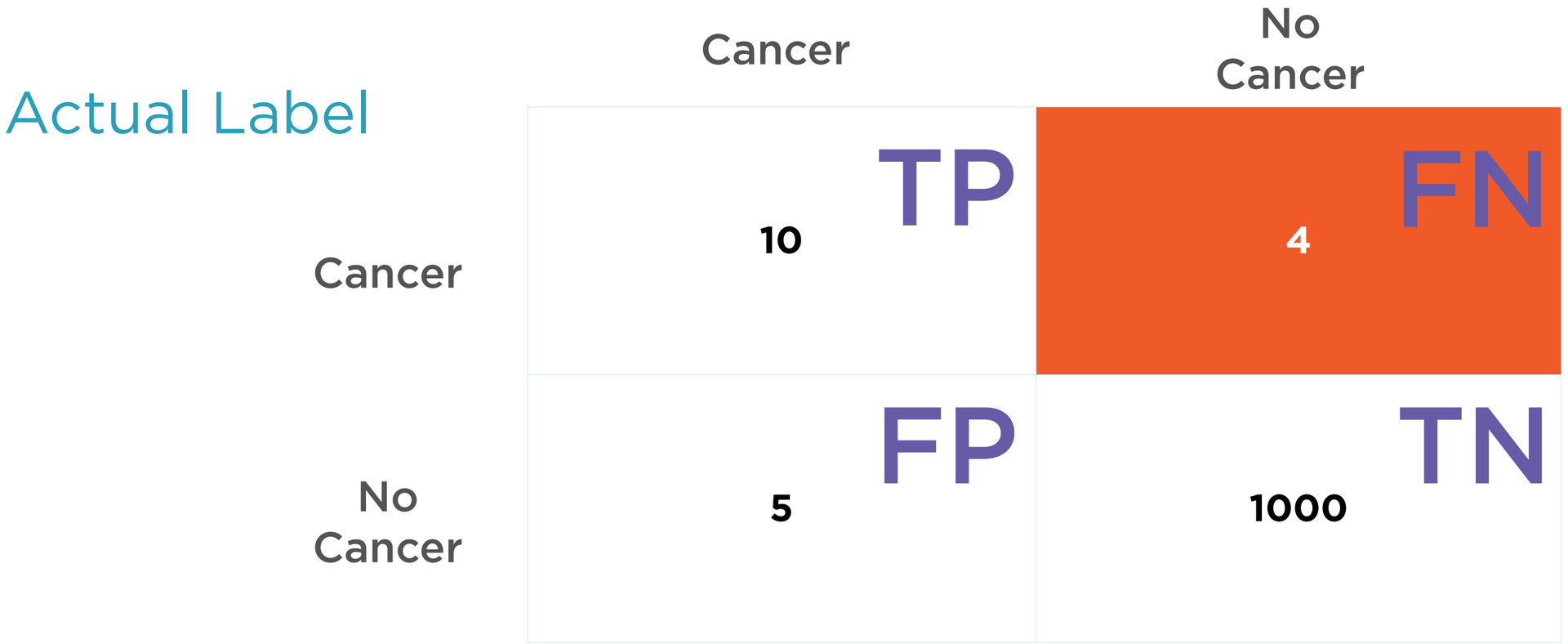



People on chemotherapy, radiation when not required

# Accuracy

Predicted Labels

Actual Label



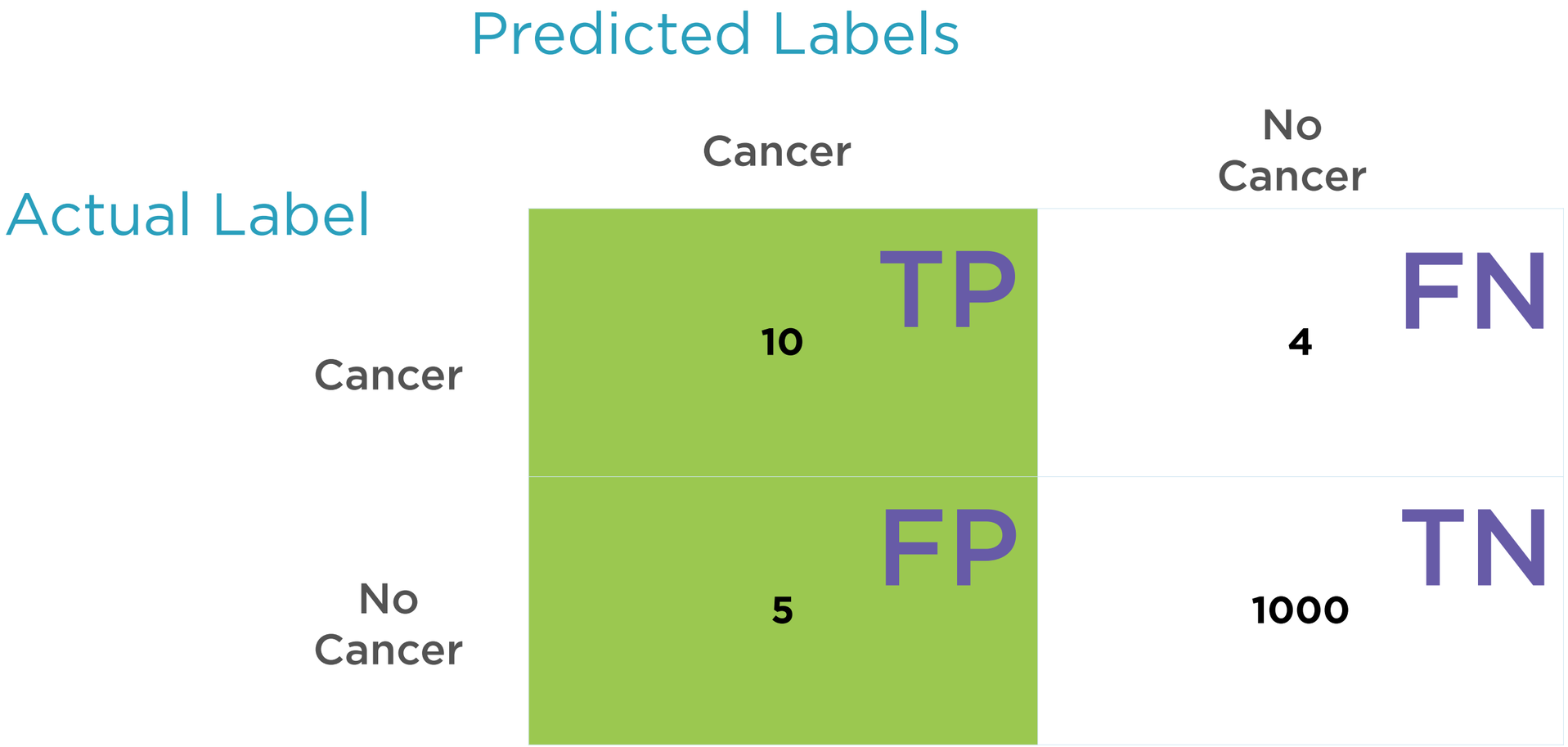

Cancer not detected, no treatment prescribed

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

# Precision

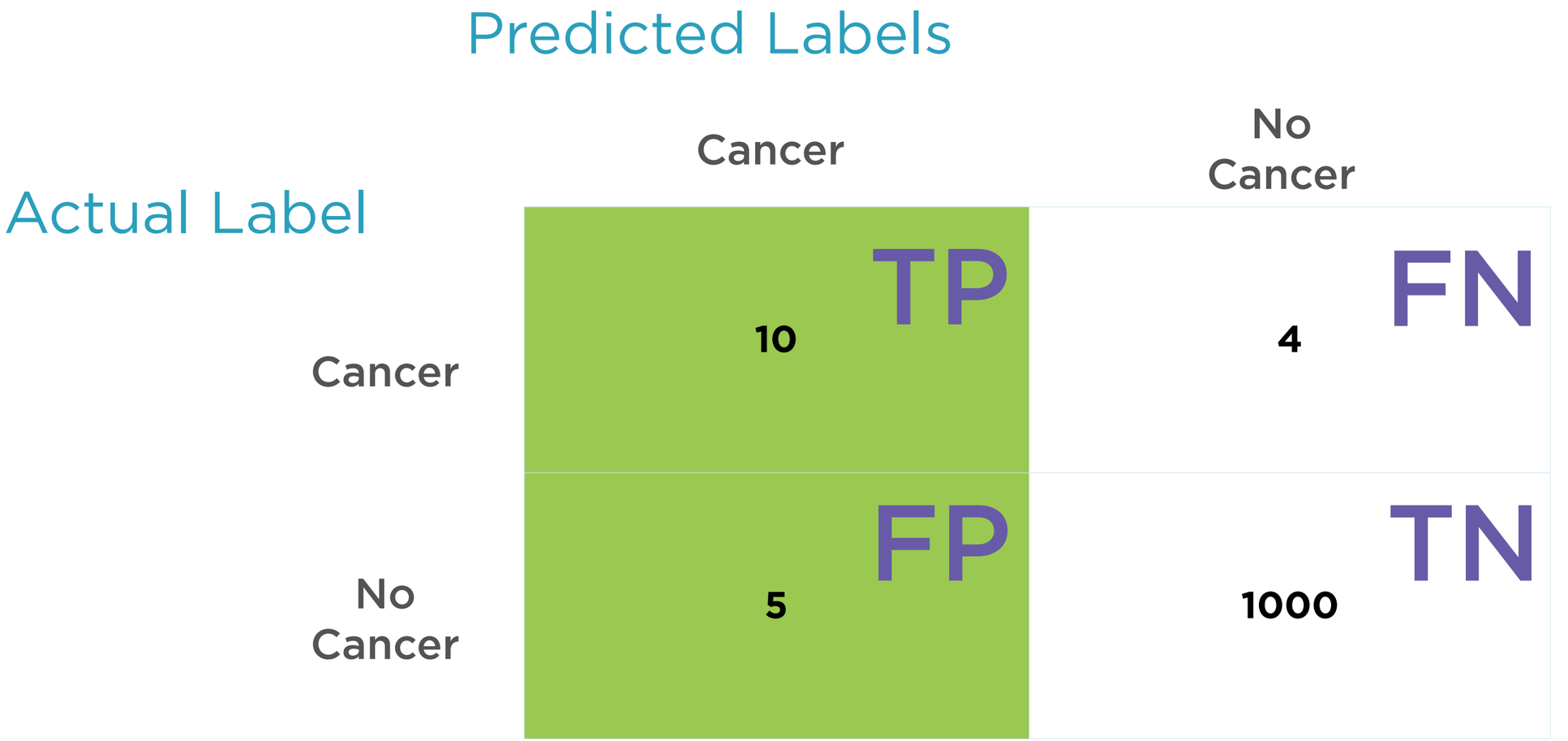
| Actual Label \ Predicted Labels | Cancer | No Cancer |
| --- | --- | --- |
| Cancer | 10 **TP** | 4 **FN** |
| No Cancer | 5 **FP** | 1000 **TN** |

# Precision




Precision = Accuracy when classifier flags cancer

# Precision




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

# Precision

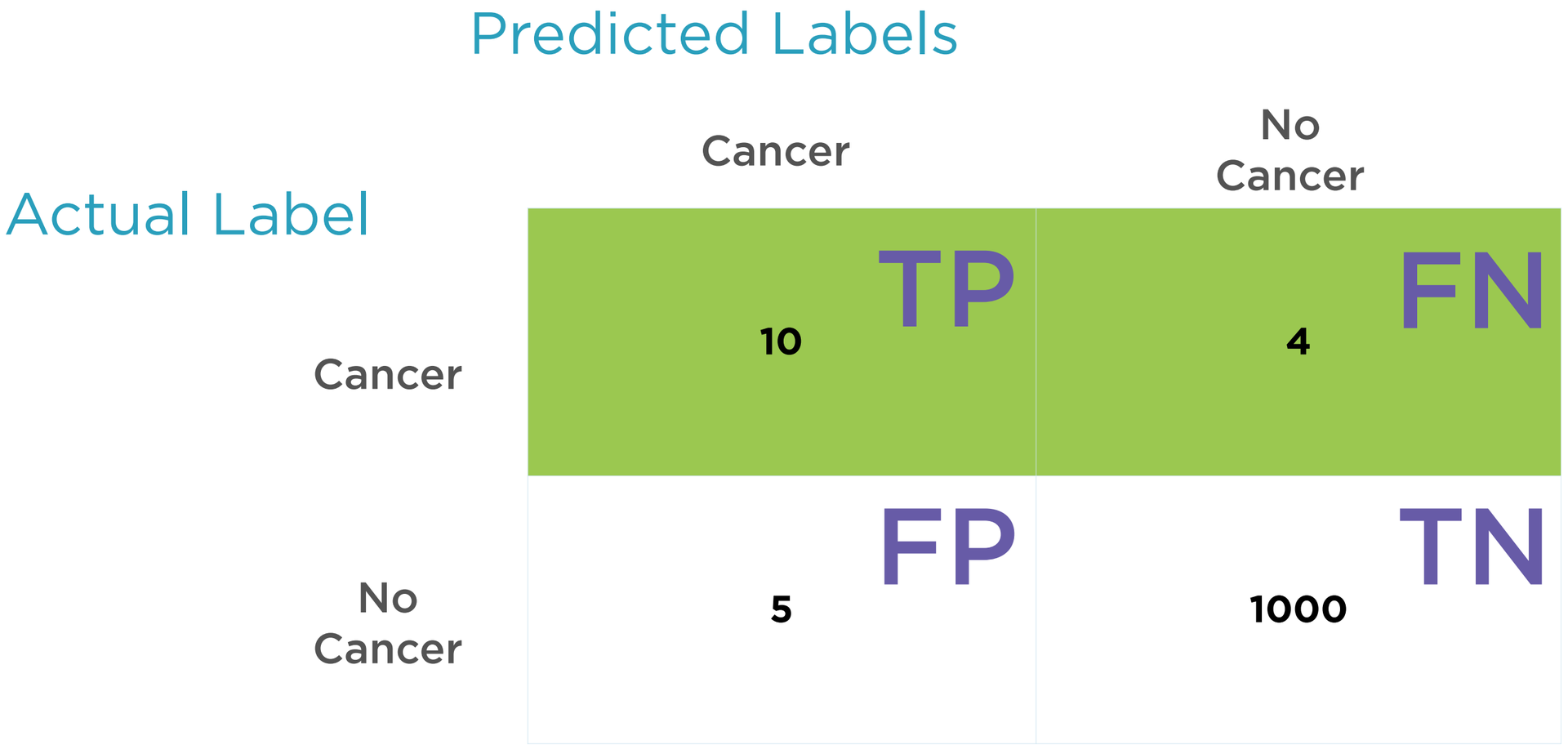
**Precision = 66.67%**

**1 in 3 cancer diagnoses is incorrect**

# Recall

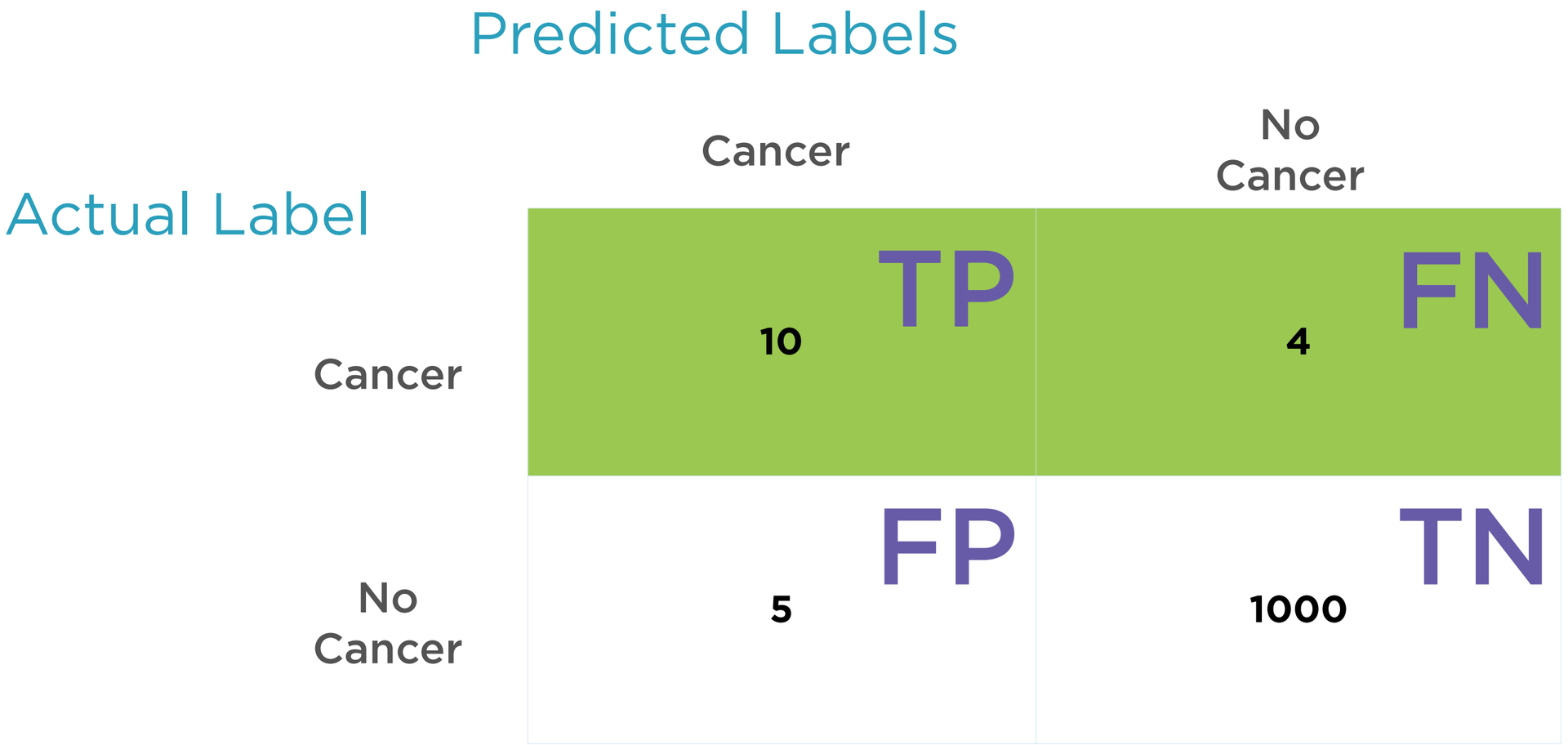
| Actual Label \ Predicted Labels | Cancer | No Cancer |
| --- | --- | --- |
| Cancer | 10 **TP** | 4 **FN** |
| No Cancer | 5 **FP** | 1000 **TN** |

# Recall




Recall = Accuracy when cancer actually present

# Recall




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

# Recall

**Recall = 71.42%**

**2 in 7 cancer cases missed**

# Demo

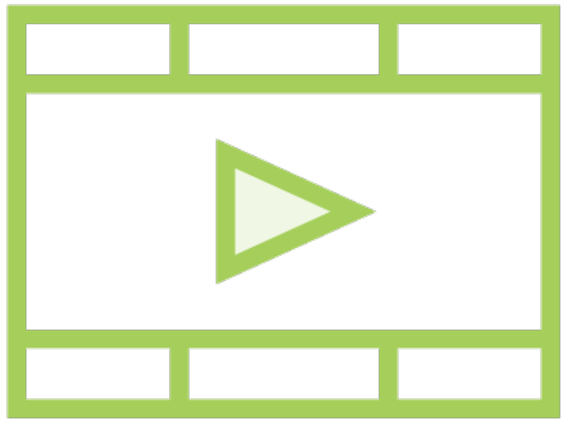
**Building a classification model using logistic regression**

**Selecting relevant features to build classifier using statistical techniques**

# Summary

- Classification to predict categorical variables
- Intuition behind logistic regression
- Evaluating classifiers using accuracy, precision, and recall
- Building a classification model using logistic regression
- Selecting relevant features to build the classifier using statistical techniques

# Related Courses



**Building Regression Models with scikit-learn**

**Building Classification Models with scikit-learn**

**Finding Relationships in Data with Python**