Using Built-in Algorithms in SageMaker

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

Overview

A variety of built-in models to deal with different ML problem types

ML algorithms available out-of-the-box, no need to write any code for the model

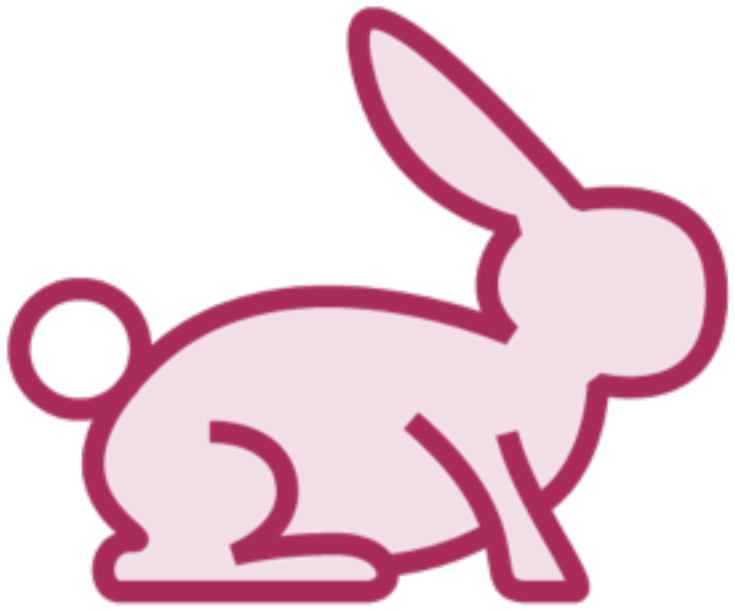
Not pre-trained, model is trained on your dataset

Format training data based on model specifications

Wide range of supervised and unsupervised learning models available

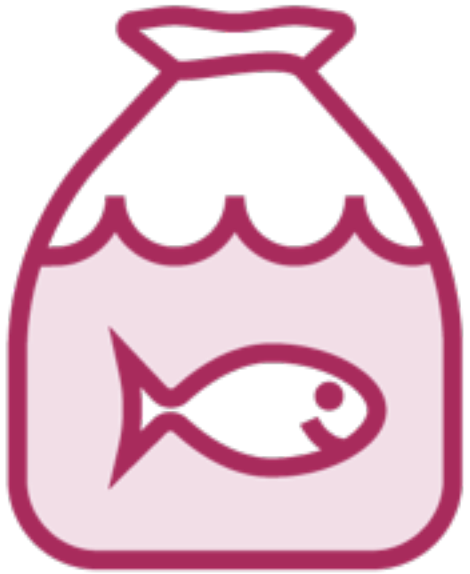
Built-in Algorithms

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

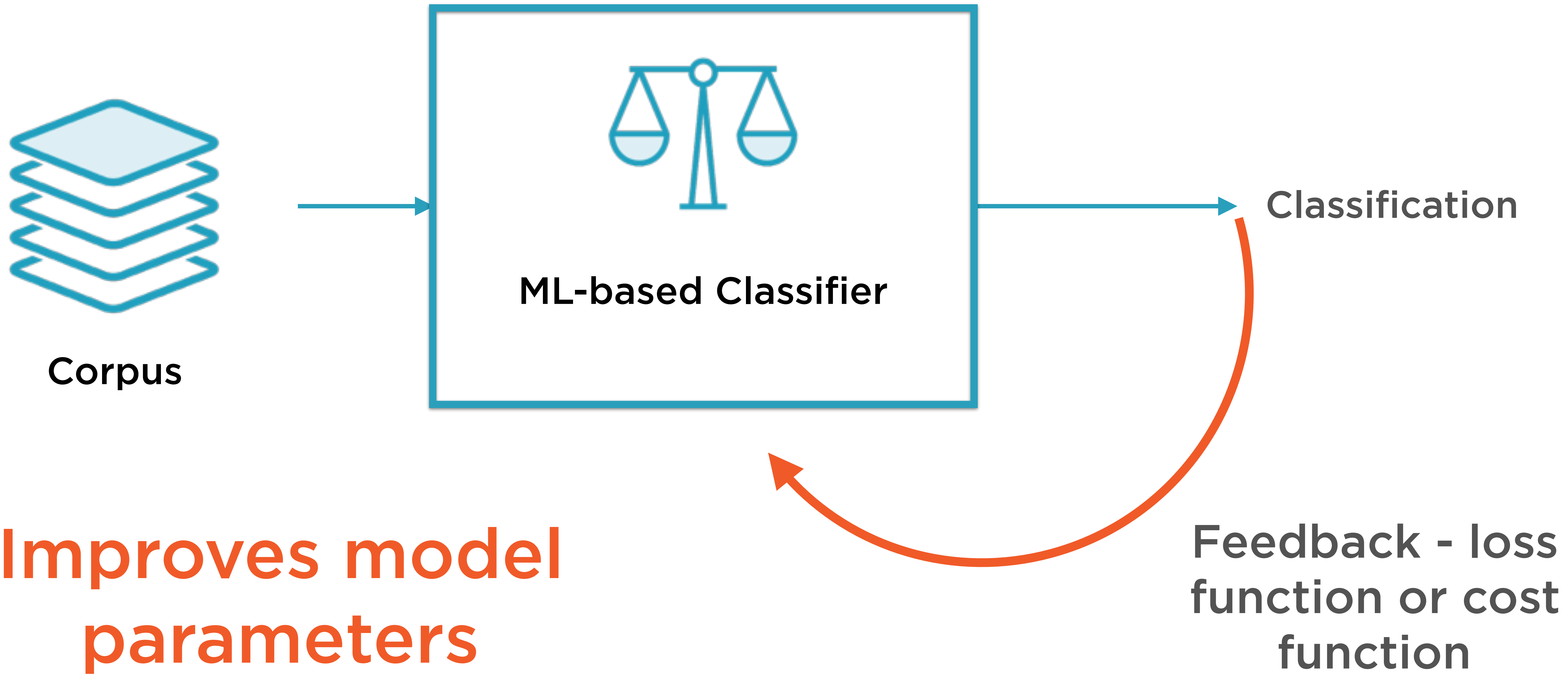
Training

Feed in a large corpus of data classified correctly

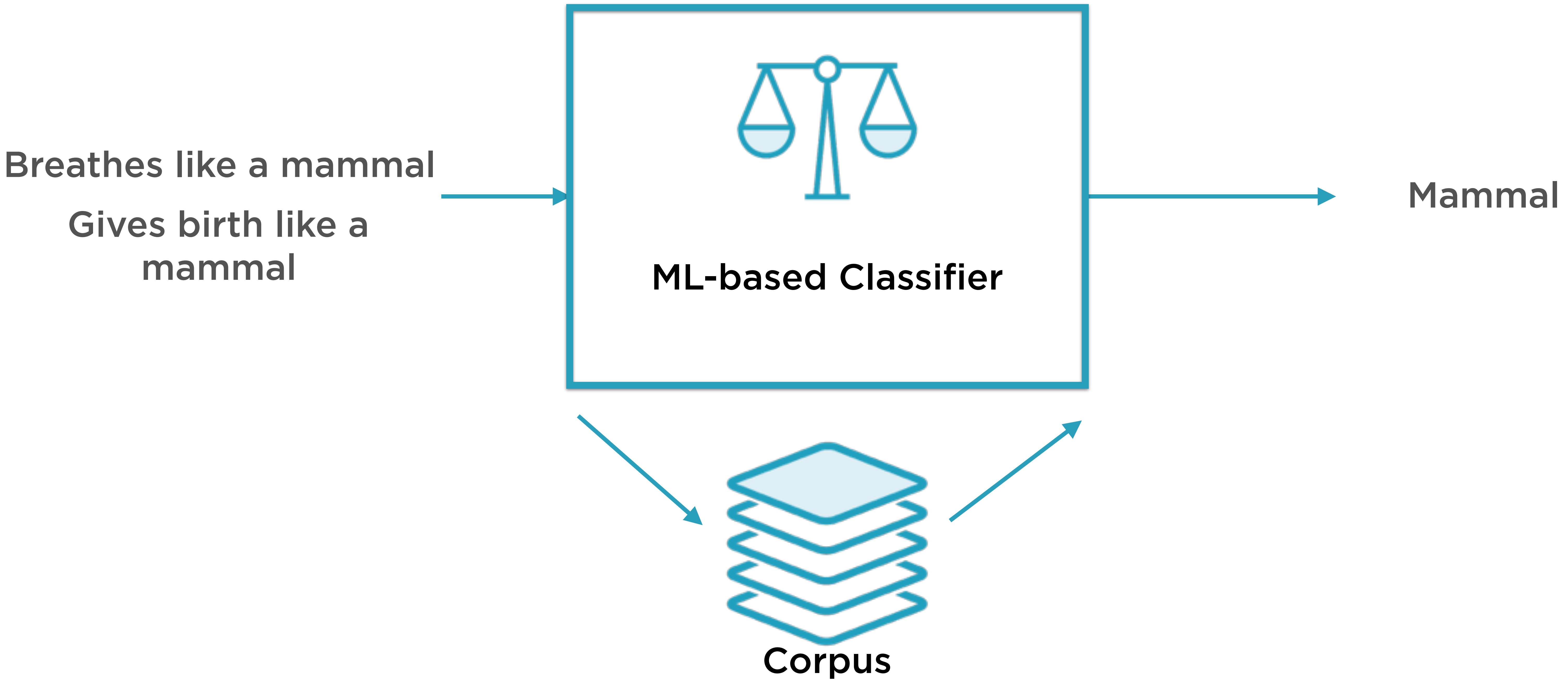
Prediction

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

Training the ML-based Classifier



ML-based Binary Classifier



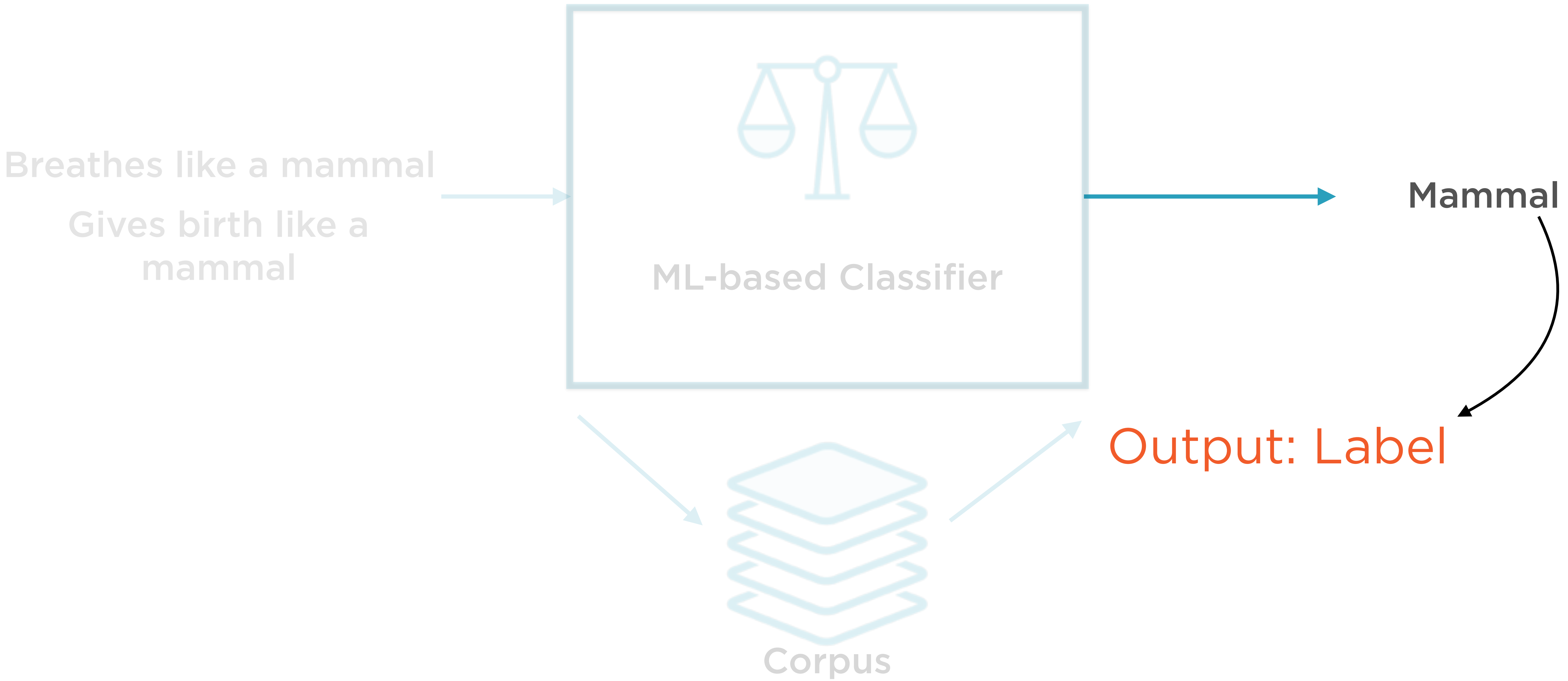
ML-based Binary Classifier

Breathes like a mammal

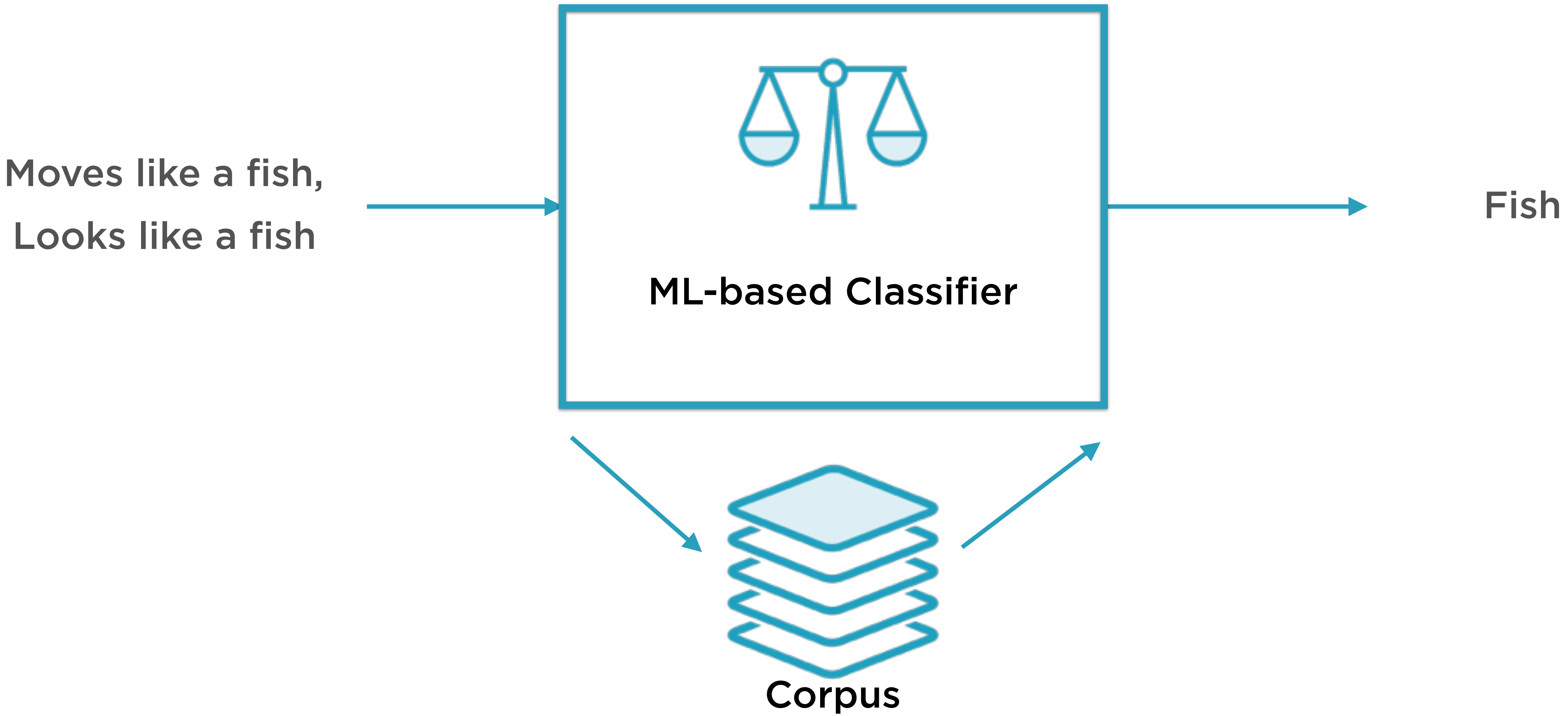
Gives birth like a mammal

Input: Feature Vector

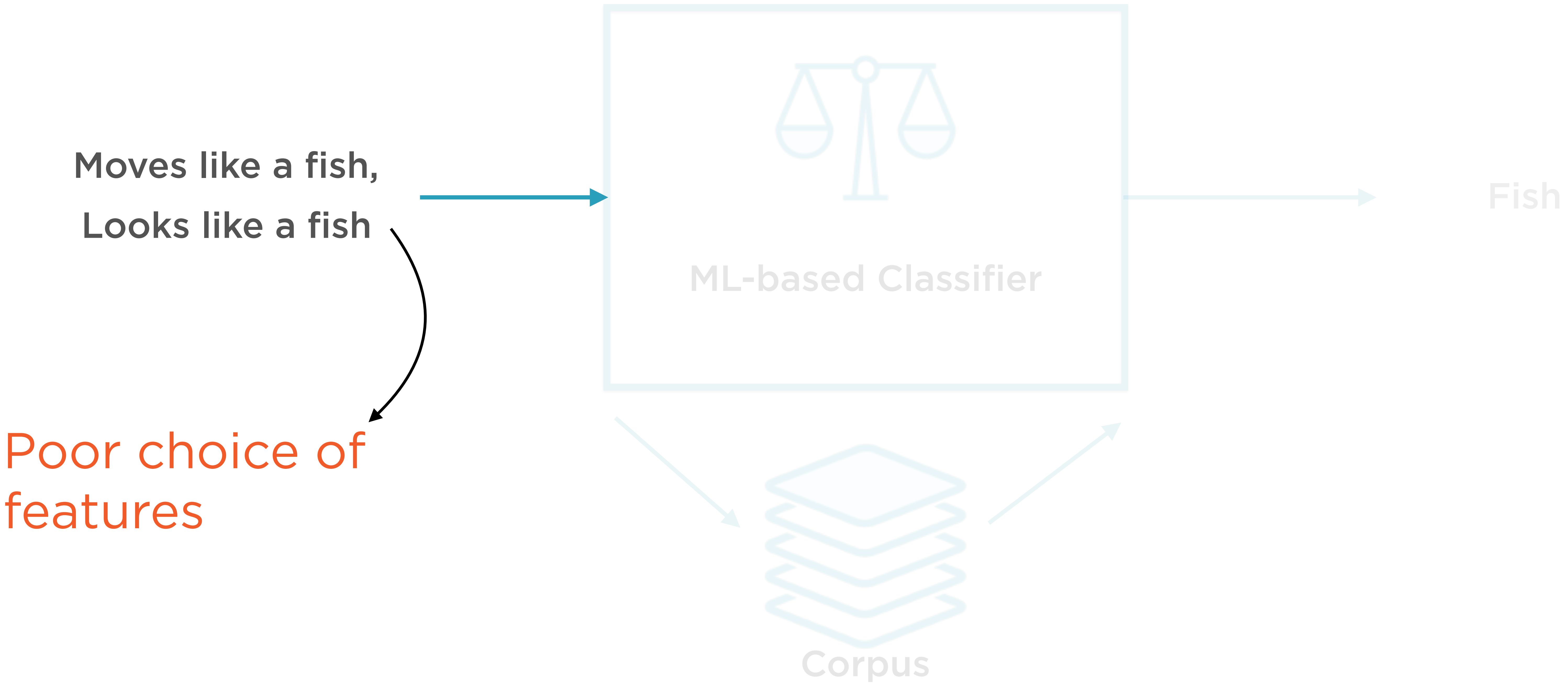
ML-based Binary Classifier



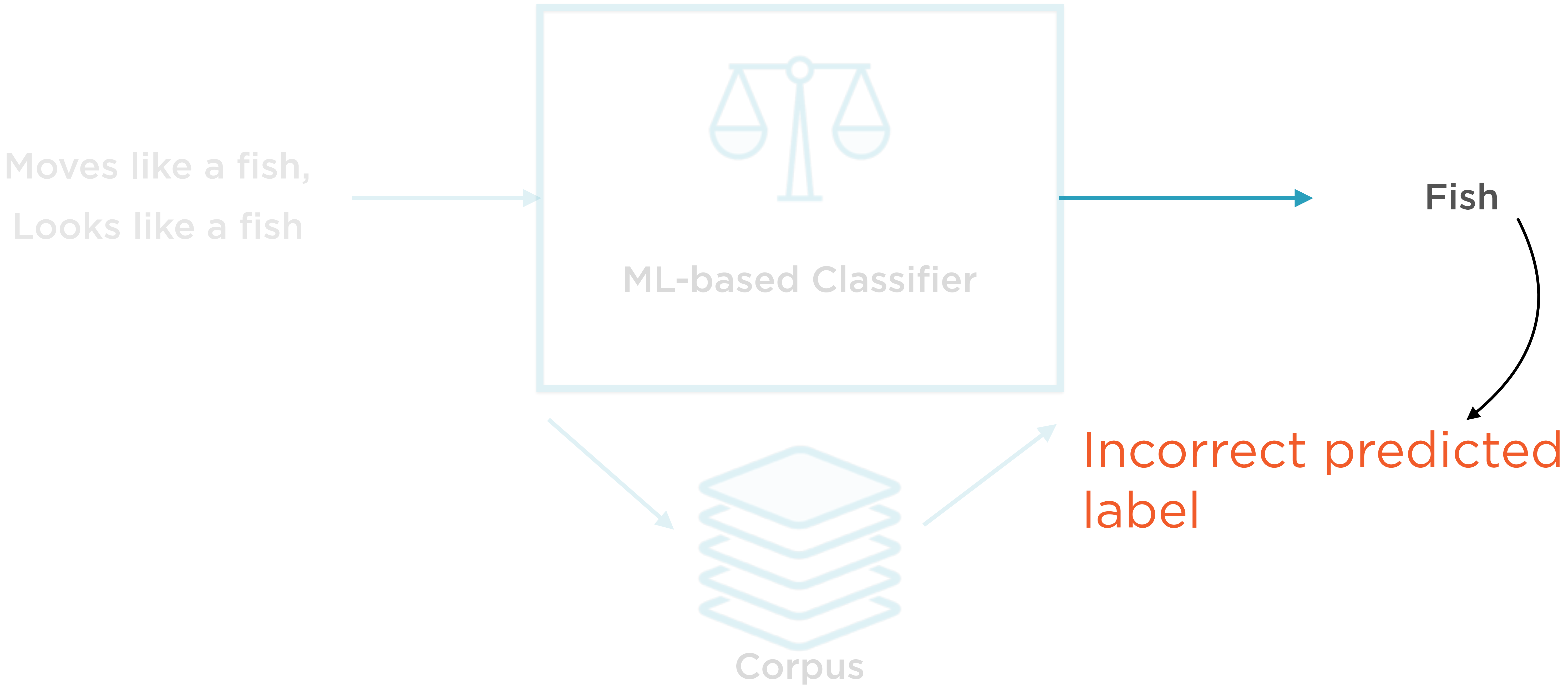
ML-based Binary Classifier



ML-based Binary Classifier



ML-based Binary Classifier



Machine Learning Model

Machine Learning Model

ML code written using scikit-learn, TensorFlow, Apache MXNet

SageMaker Built-in Algorithms

Provide out-of-the-box solutions for many common models

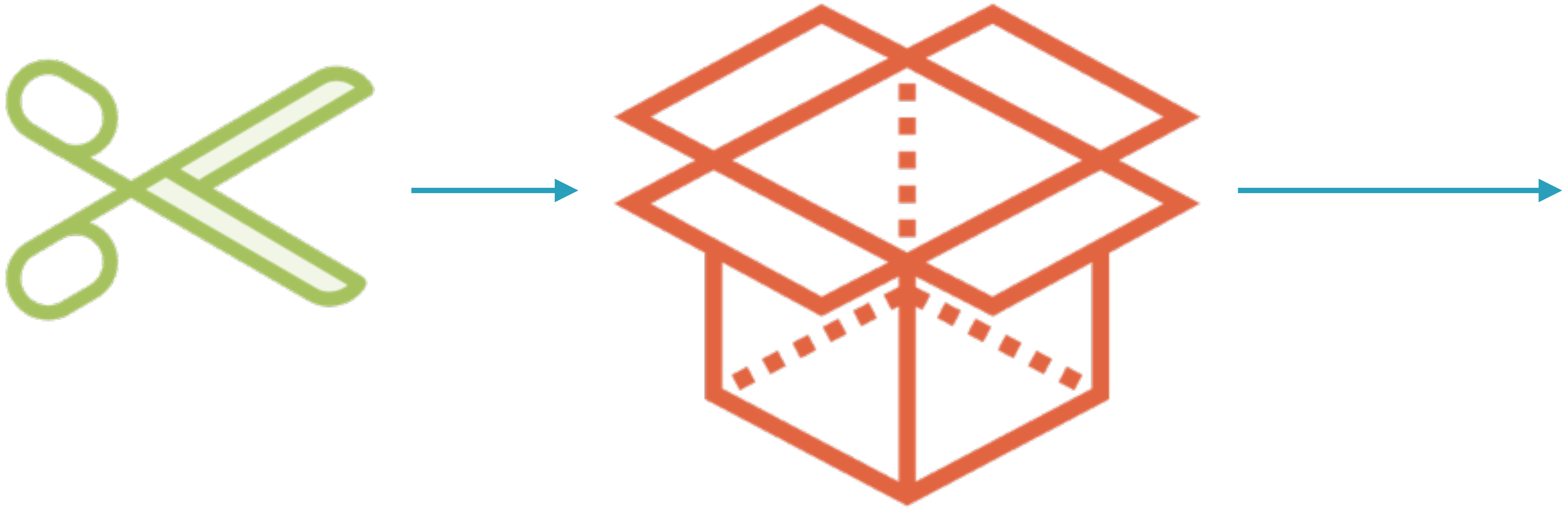
SageMaker Built-in Algorithms

Developer writes no code for the actual ML model

SageMaker Built-in Algorithms

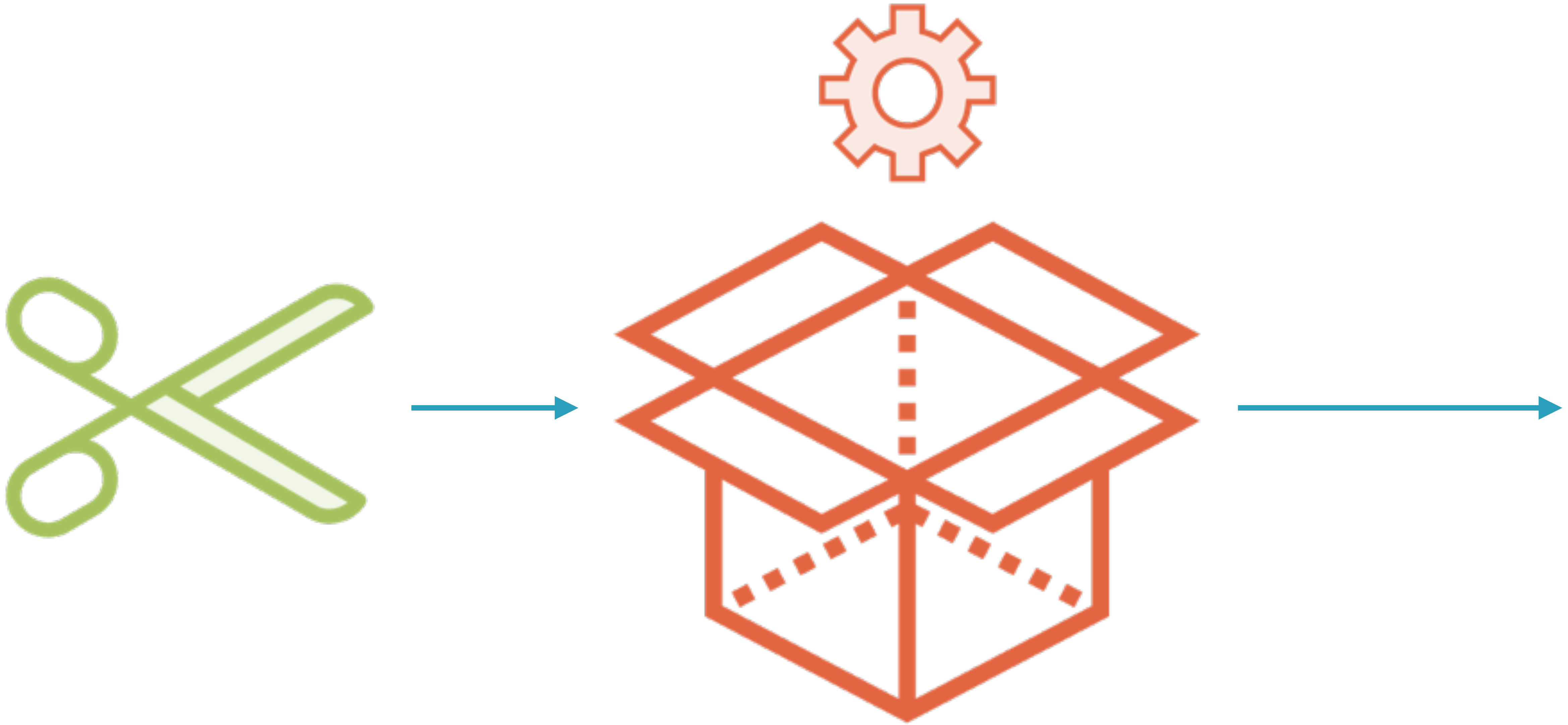
Model is hosted on Docker containers on AWS

SageMaker Built-in Algorithms



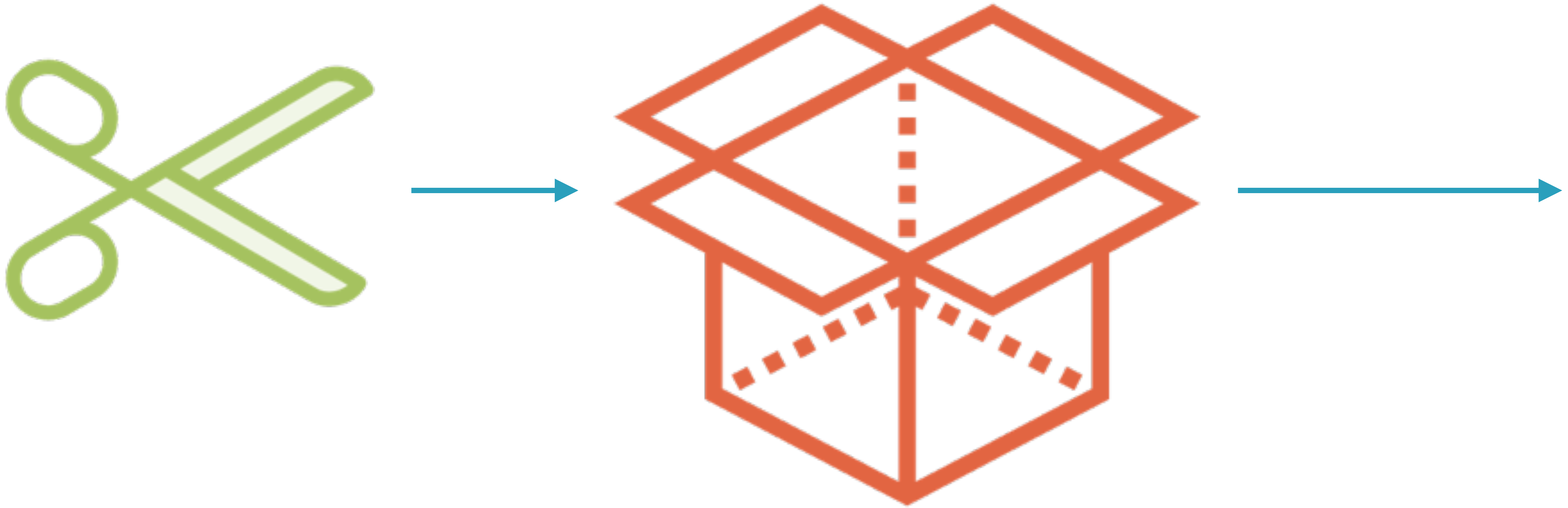
Developer formats the training data to fit the model input specifications

SageMaker Built-in Algorithms



Model runs training on AWS containers

SageMaker Built-in Algorithms



The model can then be deployed on compute instances

SageMaker Built-in Algorithms

And used for inference via endpoints

SageMaker Built-in Algorithms

Linear Learner

Classification and regression

Factorization Machines

Classification and regression

Seq2seq

Text summarization, speech to text

K-means Clustering

Clustering, grouping

Principal Components Analysis

Dimensionality reduction

SageMaker Built-in Algorithms

Linear Learner

Classification and regression

Factorization Machines

Classification and regression

Seq2seq

Text summarization, speech to text

K-means Clustering

Clustering, grouping

Principal Components Analysis

Dimensionality reduction

The Linear Learner

A **supervised** learning algorithm that can be used for both **regression** and **classification**

Types of ML Algorithms

Supervised

Labels associated with the training data is used to correct the algorithm

Unsupervised

The model has to be set up right to learn structure in the data

Types of ML Algorithms

Supervised

Labels associated with the training data is used to correct the algorithm

Unsupervised

The model has to be set up right to learn structure in the data

Linear Learner

Regression

Output prediction is a continuous real value

Classification

Output prediction is a categorical value - binary 0/1

Linear Learner

Regression

Output prediction is a continuous real value

Classification

Output prediction is a categorical value - binary 0/1

Simple Regression

Cause

Independent variable

Effect

Dependent variable

Simple Regression

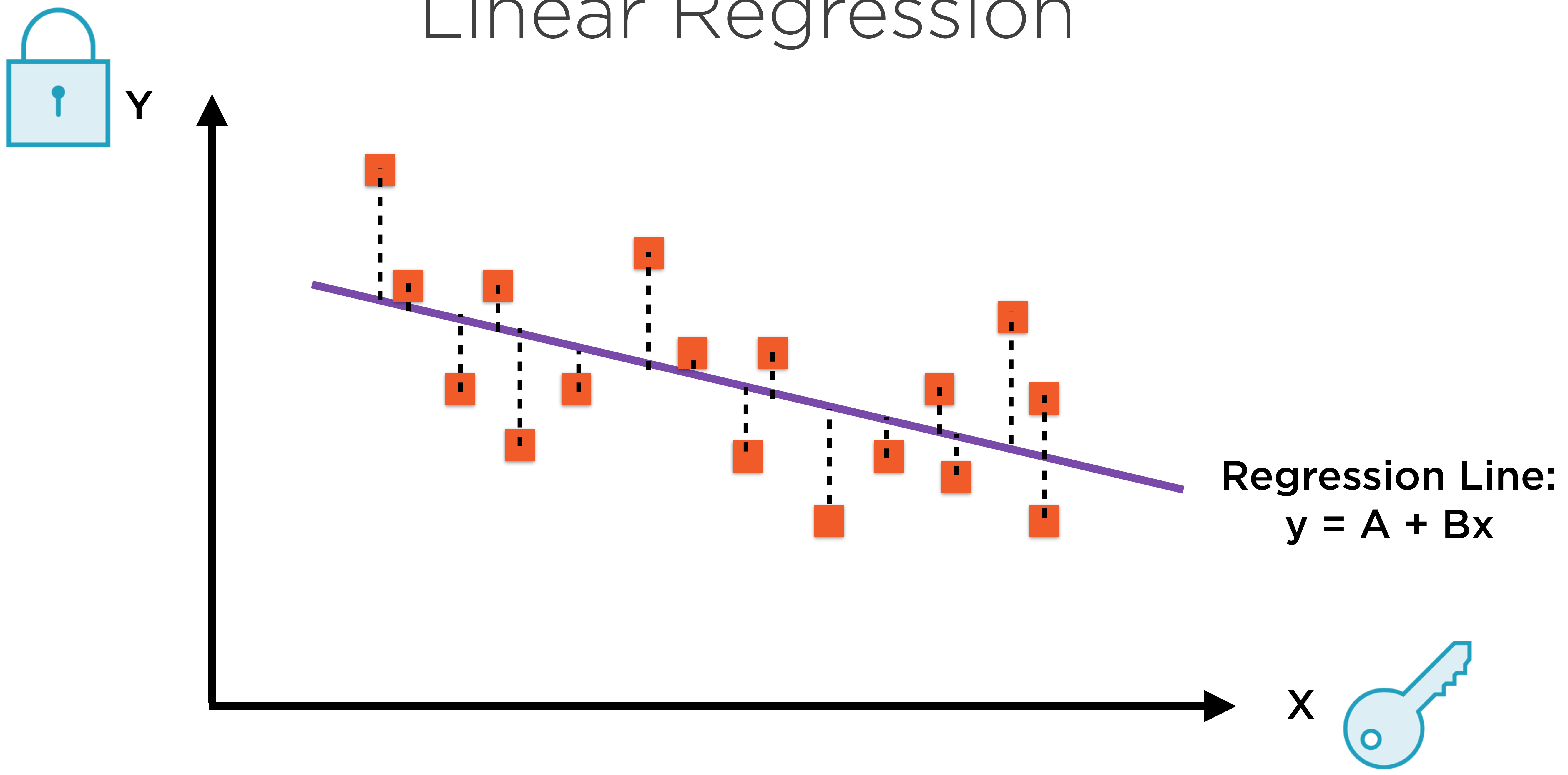
Cause

Distance from the city center

Effect

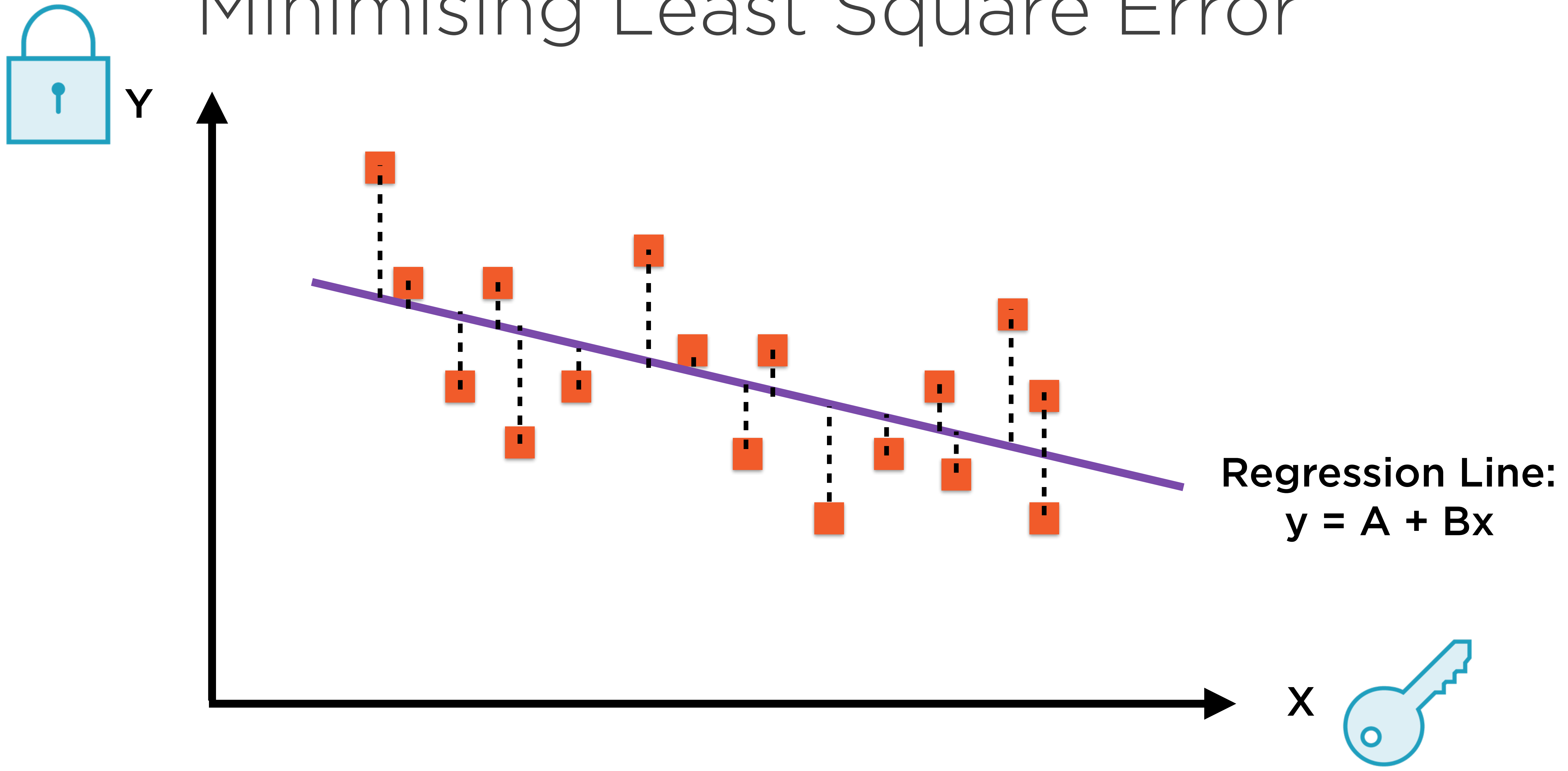
Changes in price per square foot of a house

Linear Regression



Finding the best fit line through these points

Minimising Least Square Error



The "best fit" line is called the regression line

Linear Learner

Regression

Output prediction is a continuous real value

Classification

Output prediction is a categorical value - binary 0/1

Two Approaches to Deadlines

Start 5 minutes before deadline

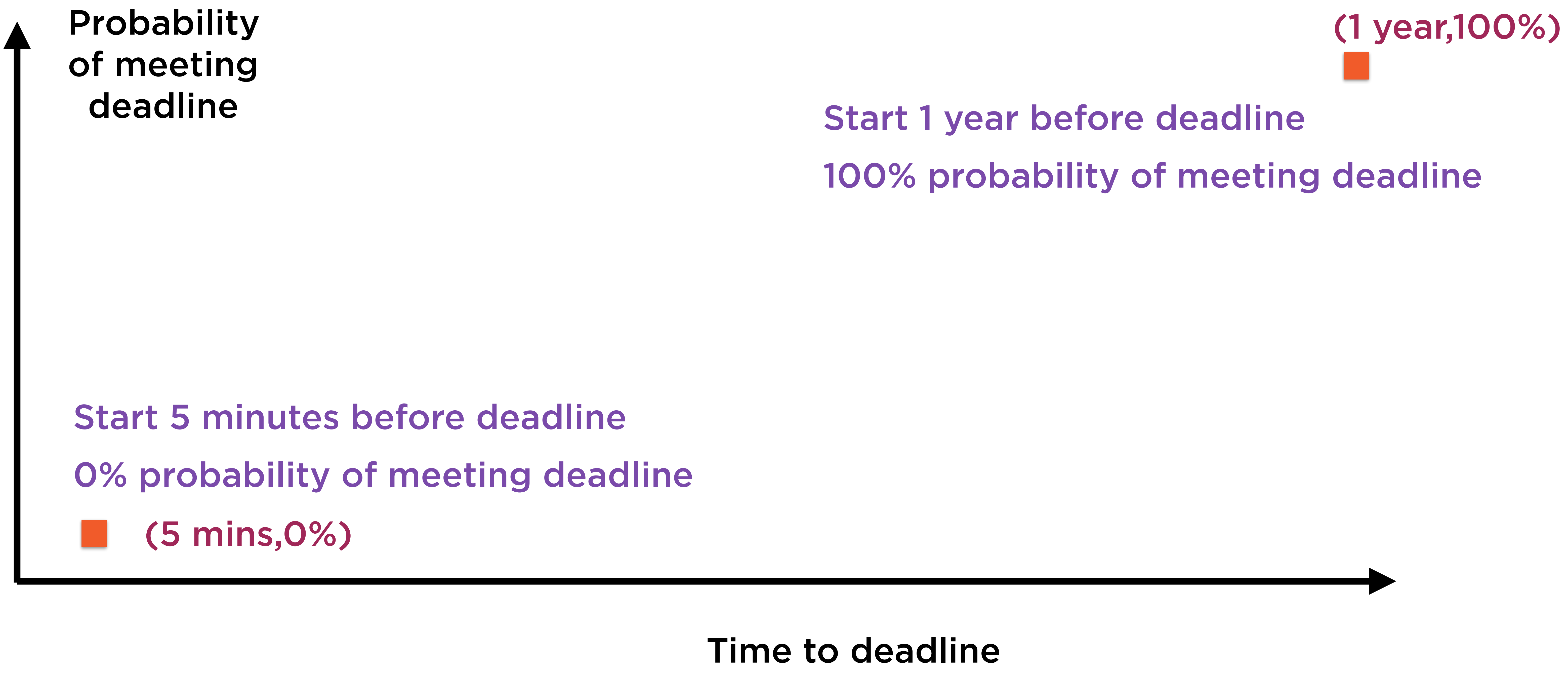
Good luck with that

Start 1 year before deadline

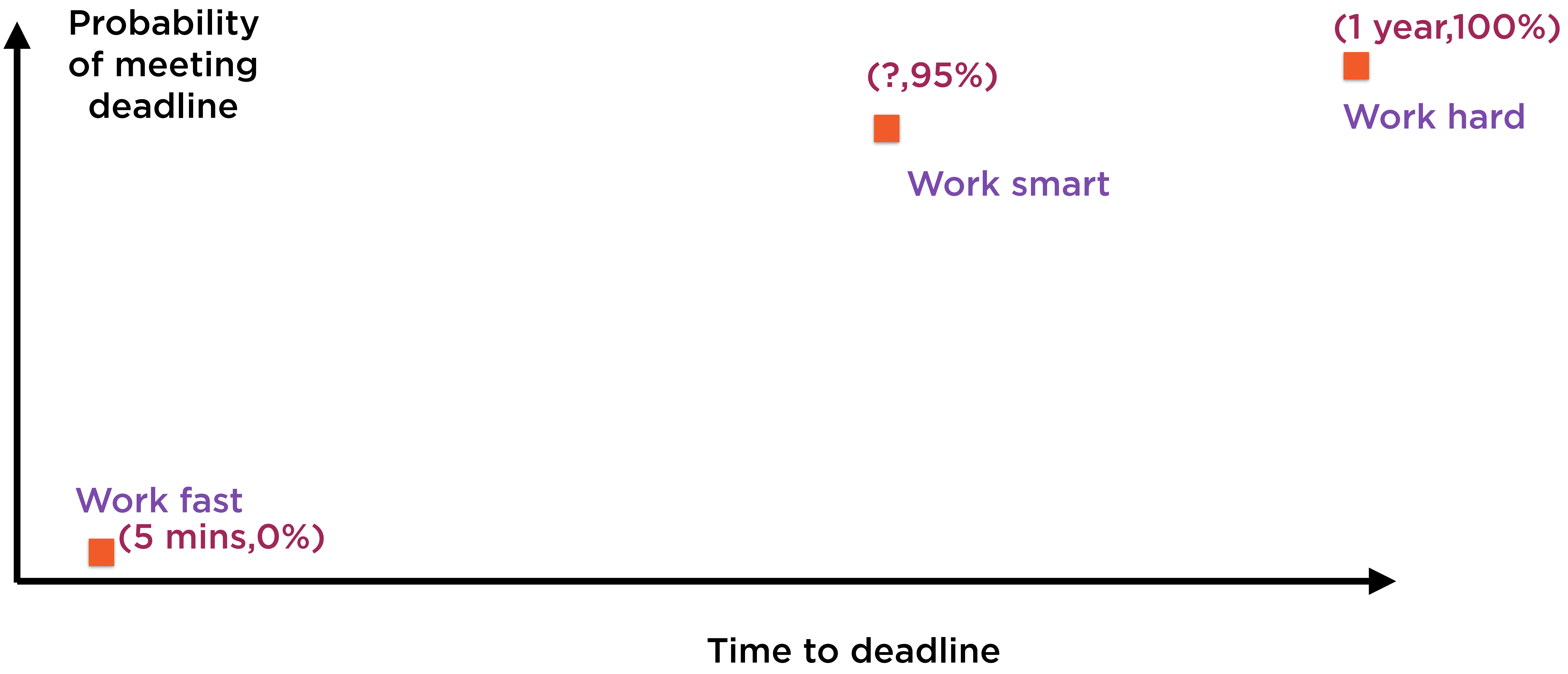
Maybe overkill

Neither approach is optimal

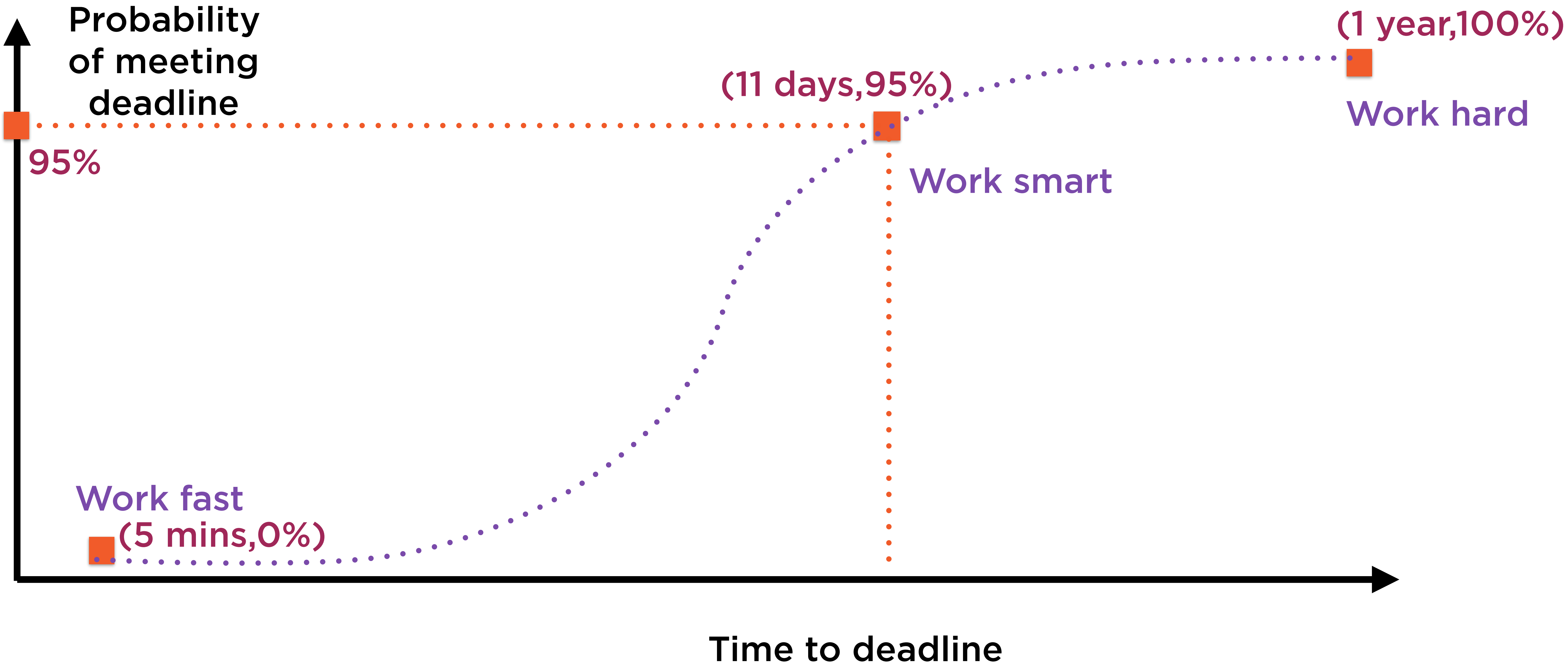
Logistic Regression



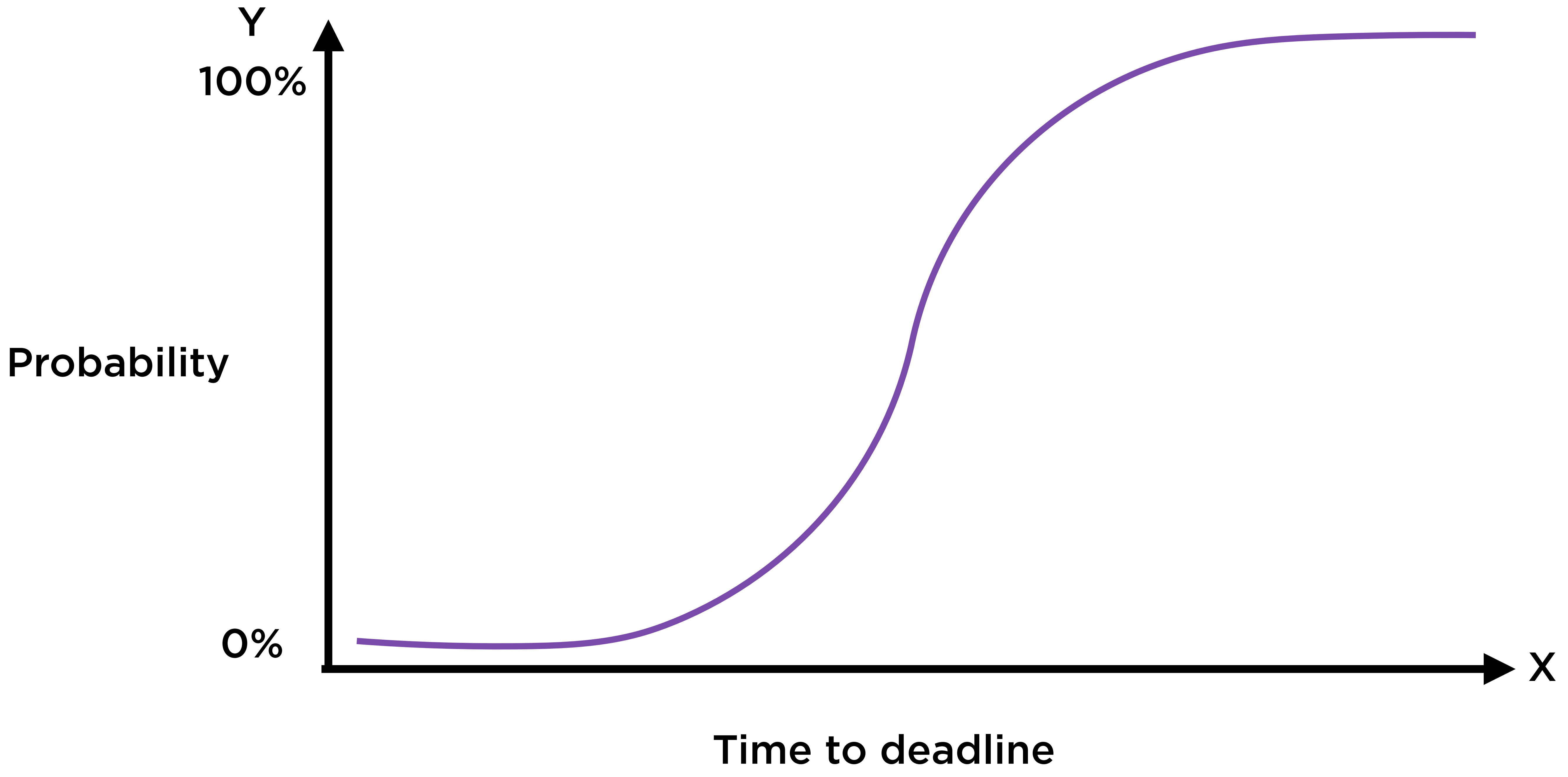
Working Hard, Fast, Smart



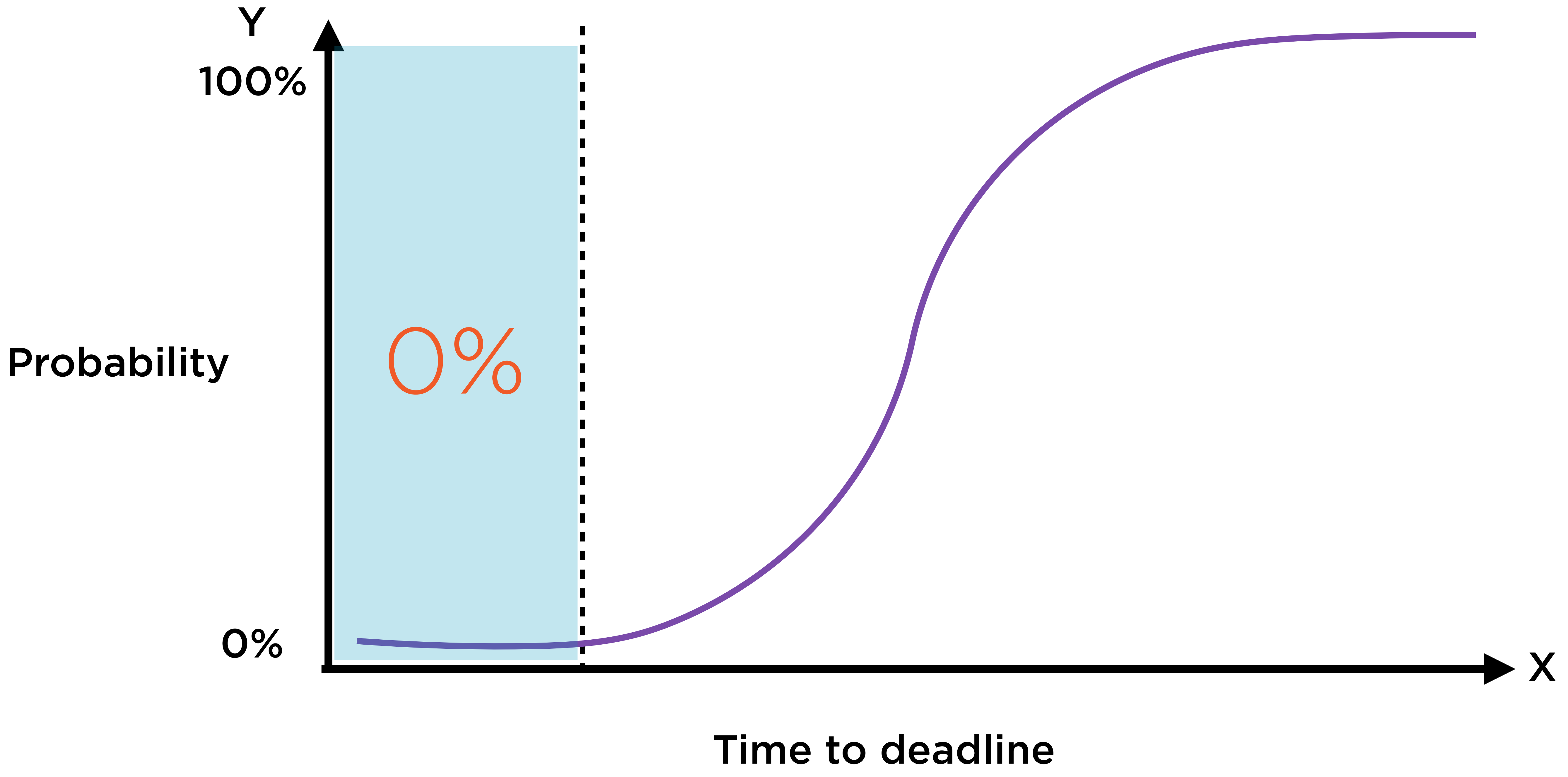
Working Hard, Fast, Smart



Working Smart with Logistic Regression

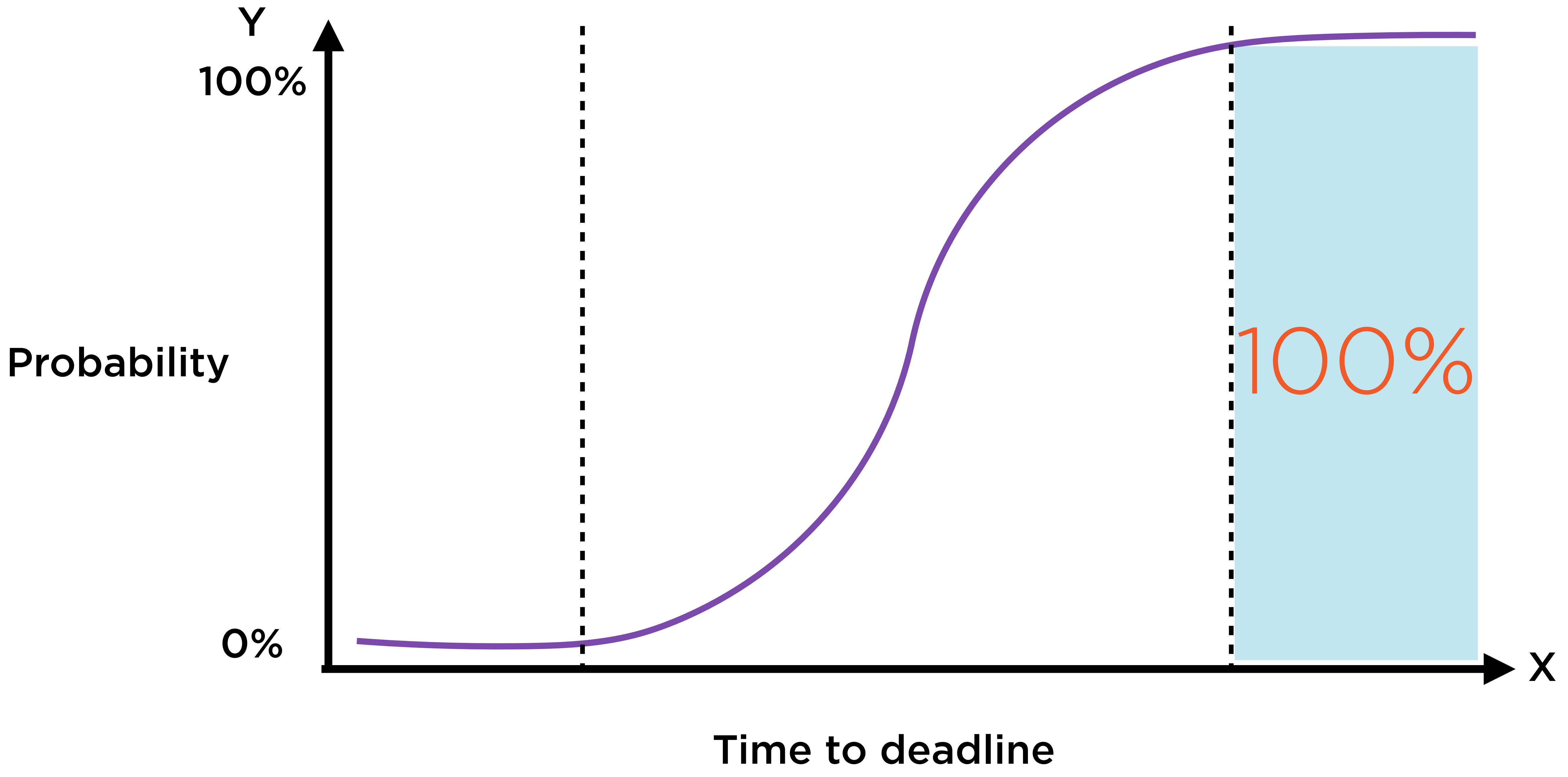


Working Smart with Logistic Regression



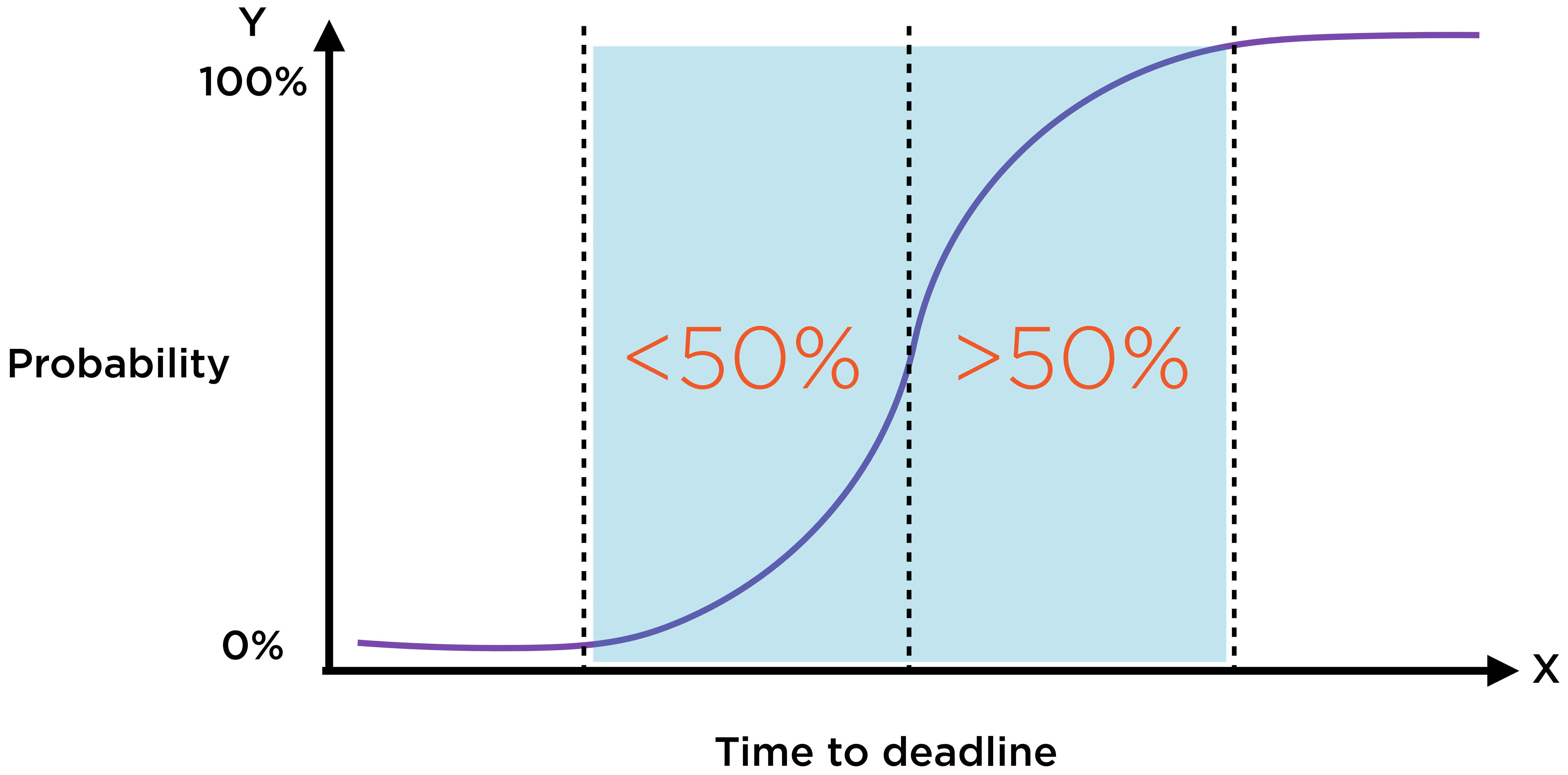
Start too late, and you'll definitely miss

Working Smart with Logistic Regression



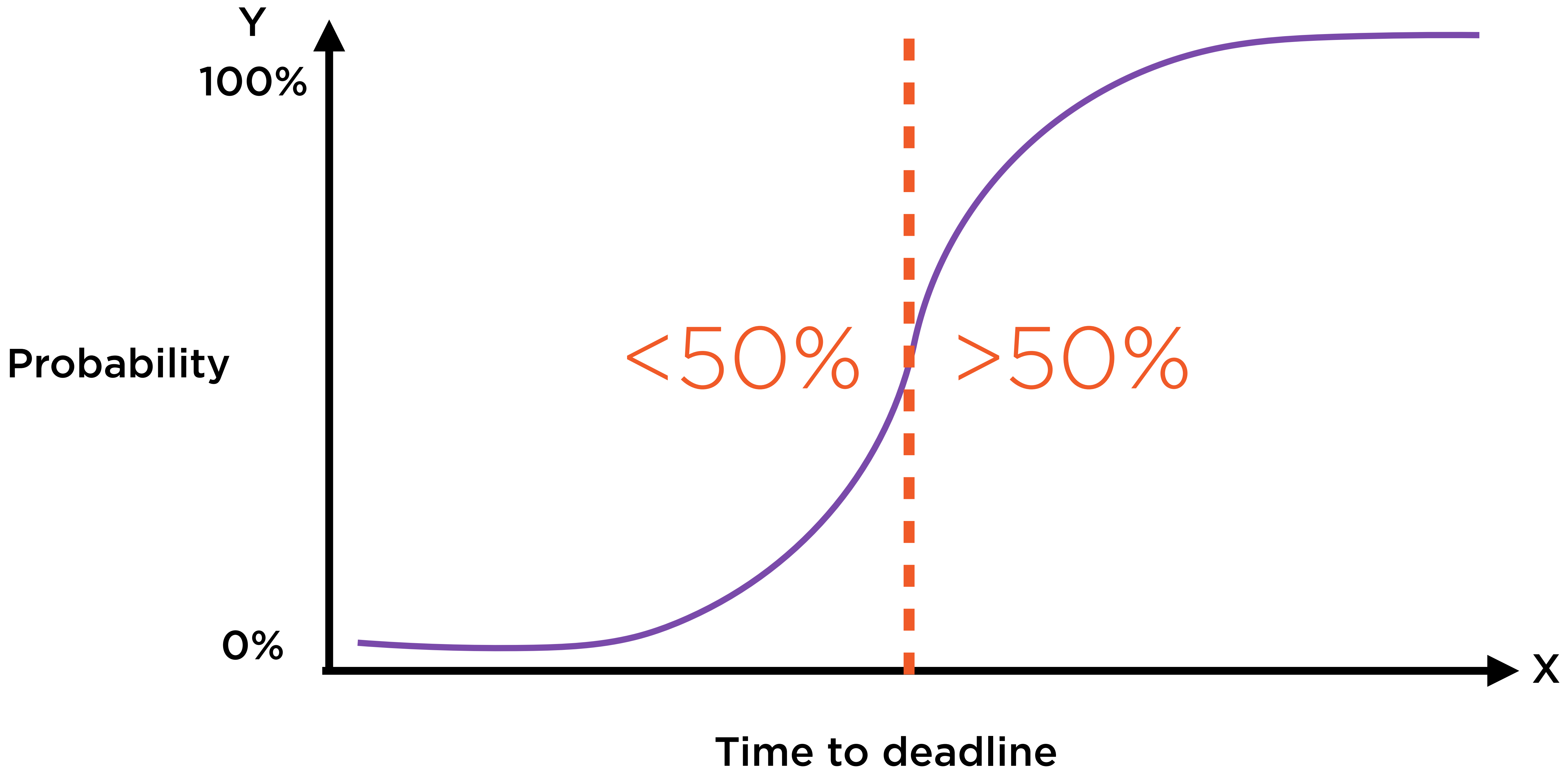
Start too early, and you'll definitely make it

Working Smart with Logistic Regression



Working smart is knowing when to start

Working Smart with Logistic Regression



This is the threshold probability value for classification

Linear Learner

Regression

Output prediction is a continuous real value

Classification

Output prediction is a categorical value - binary 0/1

Using Built-in Algorithms

Retrieve training data

Explore and clean data

Format and serialize input data

Set up data in the form accepted by the algorithm, upload to S3

Train with built-in algorithms

Stored in containers, set up estimators with containers as input, train with input data

Deploy model

Creates endpoint configuration and endpoint for prediction

Use endpoint for inference

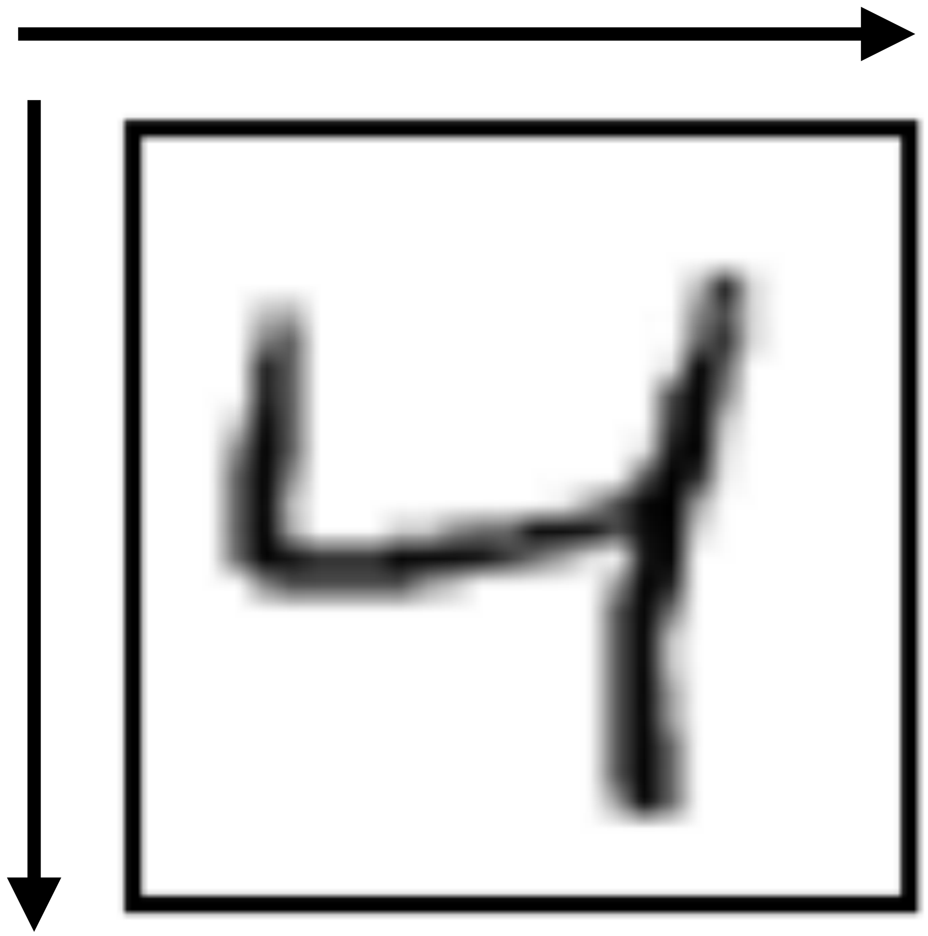
Predict using input data

Demo

Using the linear learner - a built-in algorithm provided by SageMaker for classification

Identify whether an MNIST digit is a 3 or not (binary classification)

MNIST Dataset



4

Every image is standardized to be of size 28x28

= 784 pixels

Representing Images

0	0	0	0	0	0
0.2	0.8	0	0.3	0.6	0
0.2	0.9	0	0.3	0.8	0
0.3	0.8	0.7	0.8	0.9	0
0	0	0	0.2	0.8	0
0	0	0	0.2	0.2	0

= 784 pixels

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

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 FP	1000

Actual Label ≠ Predicted Label

True Negative

Predicted Labels

Actual Label

	Cancer	No Cancer
Cancer	10	4
No Cancer	5	1000 TN

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

Accuracy

Predicted Labels

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

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

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

Recall

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

Recall = Accuracy when cancer actually present

Principal Components Analysis

Types of ML Algorithms

Supervised

Labels associated with the training data is used to correct the algorithm

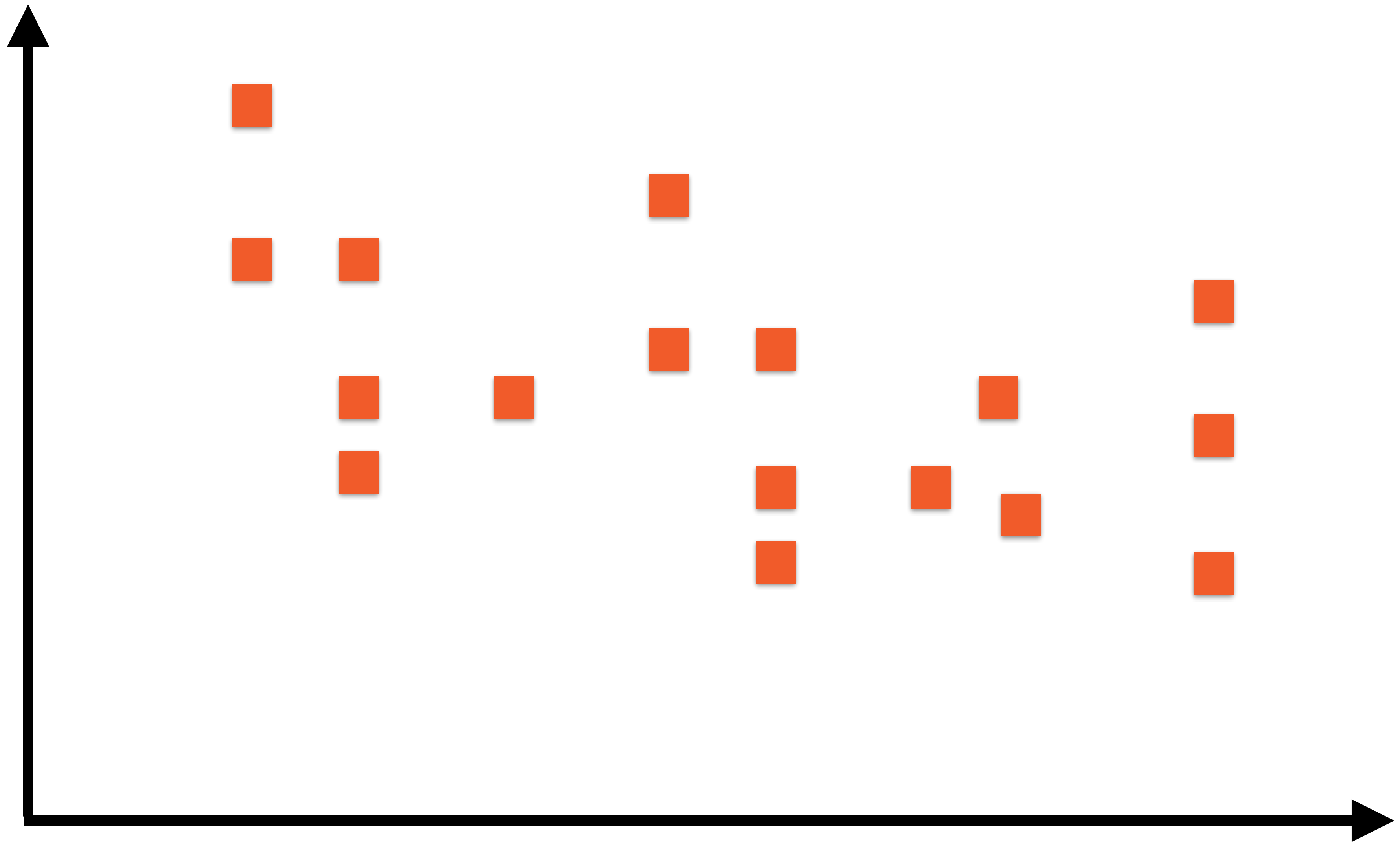
Unsupervised

The model has to be set up right to learn structure in the data

Data in One Dimension

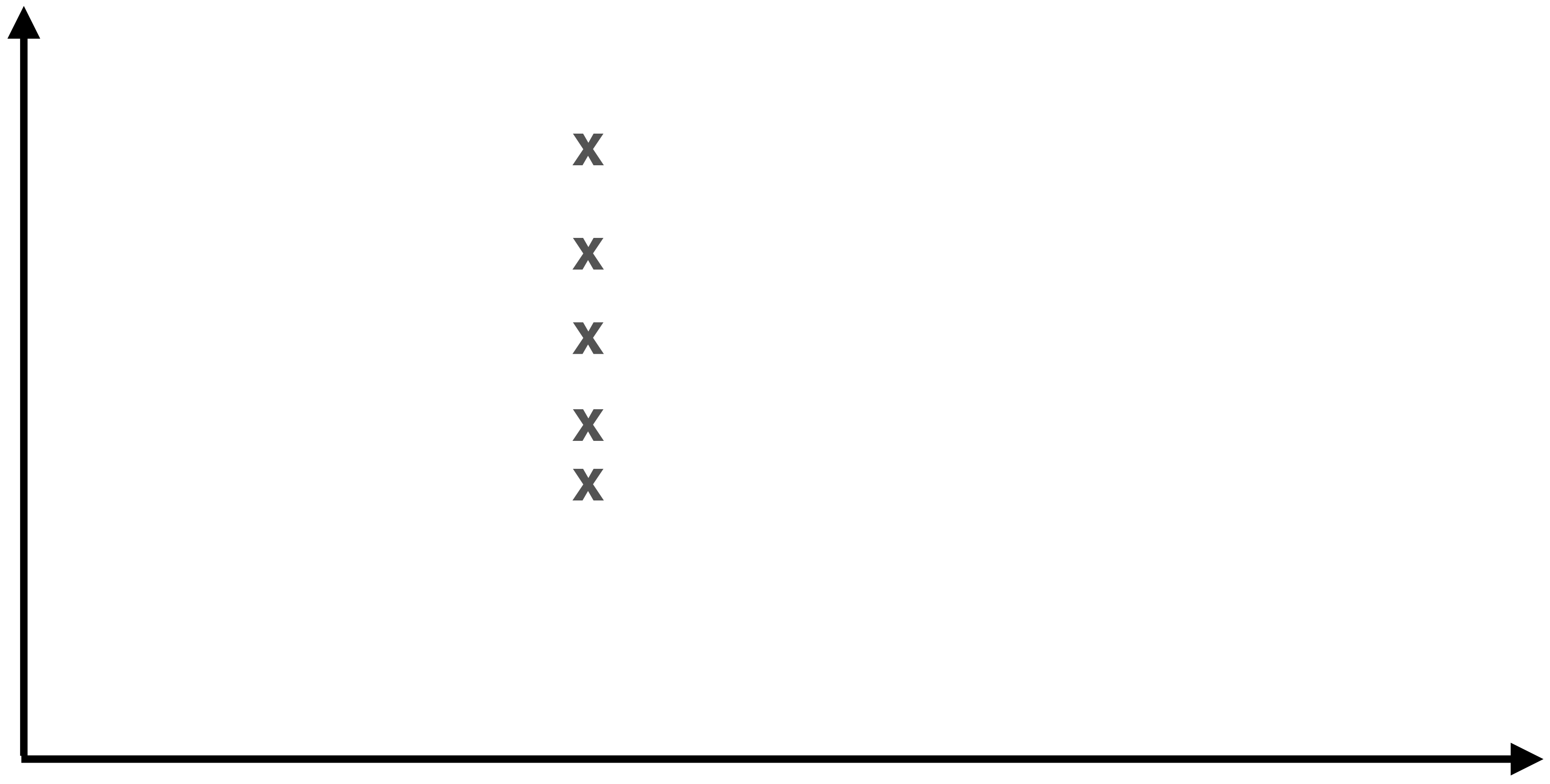
Unidimensional data points can be represented using a line, such as a number line

Data in Two Dimensions



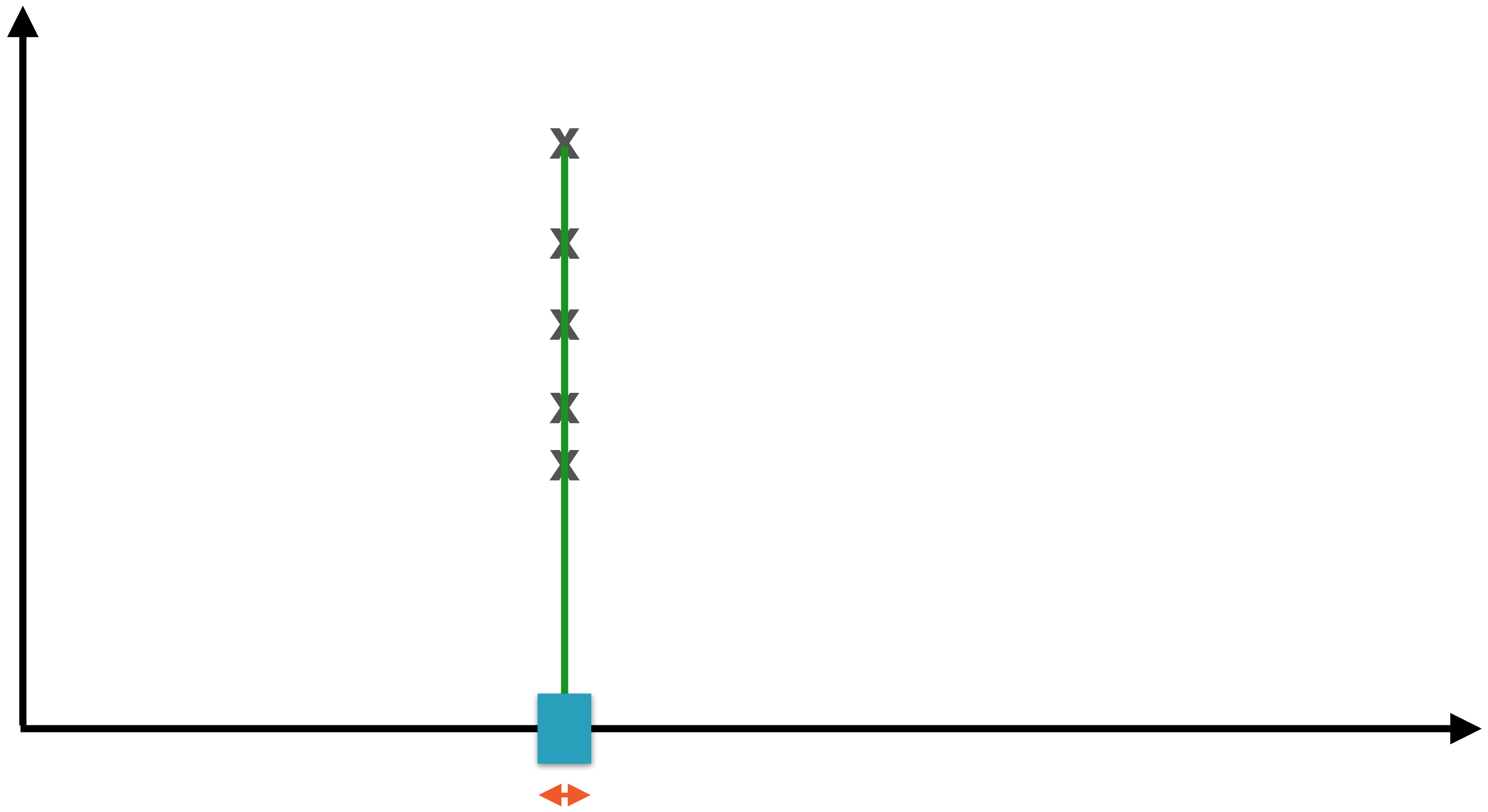
It's often more insightful to view data in relation to some other, related data

A Question of Dimensionality



Pop quiz: Do we really need two dimensions to represent this data?

Bad Choice of Dimensions



If we choose our axes (dimensions) poorly then we do need two dimensions

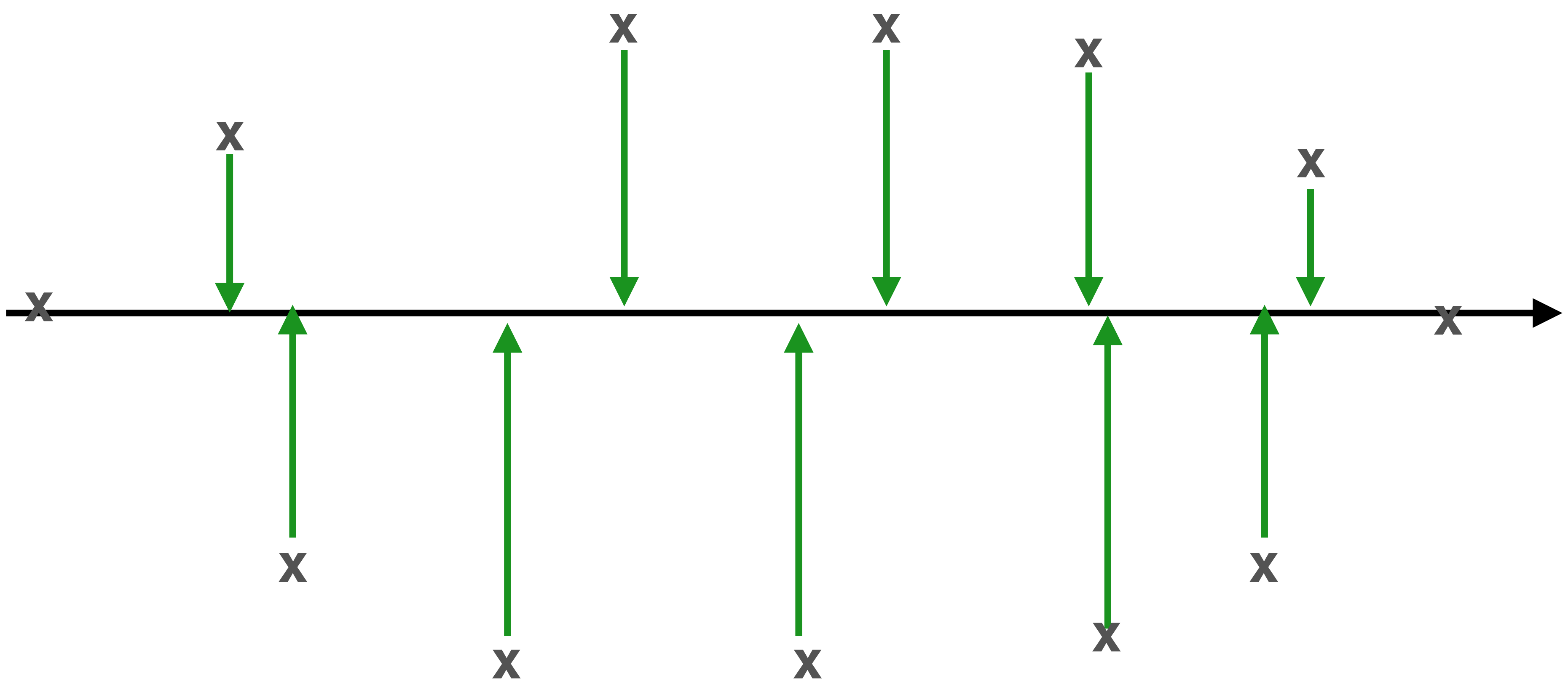
Good Choice of Dimensions

If we choose our axes (dimensions) well then one dimension is sufficient

Intuition Behind PCA

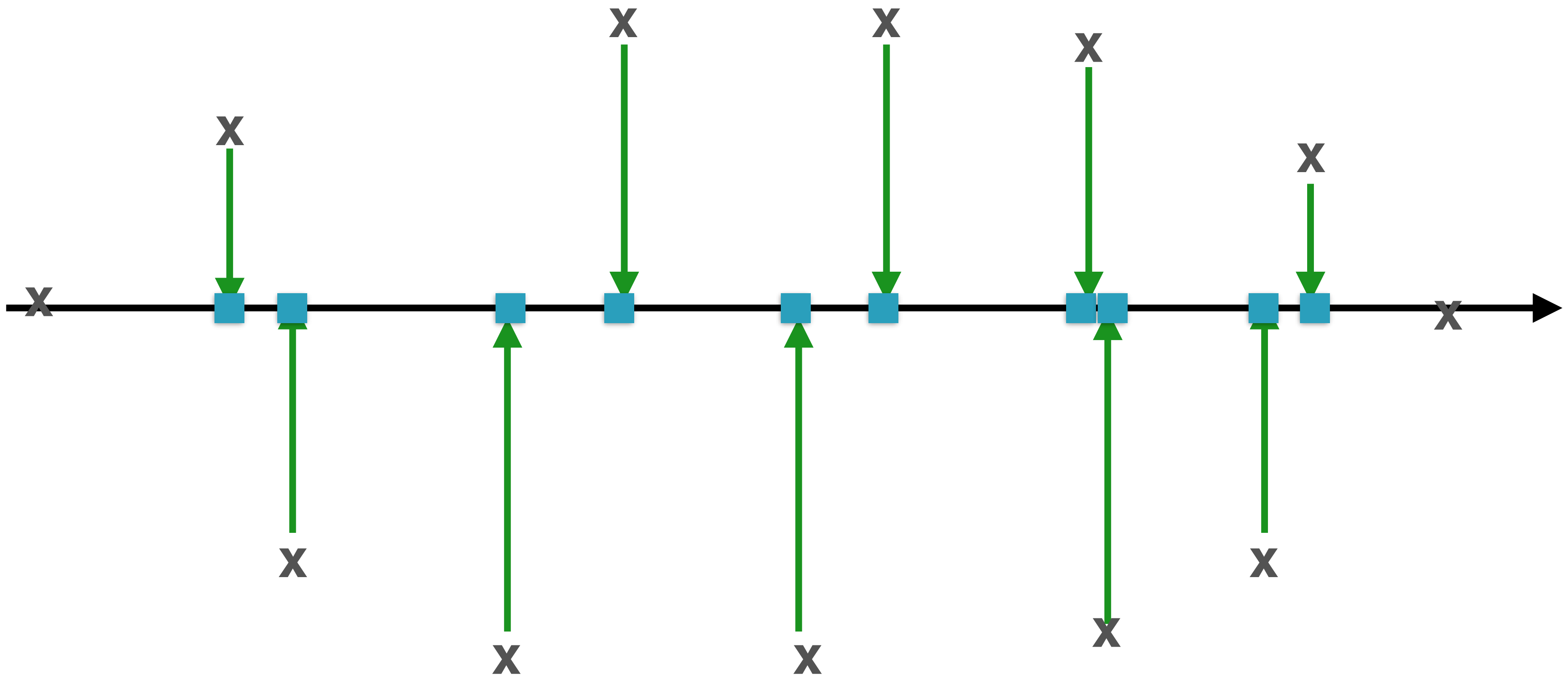
Objective: Find the “best” directions to represent this data

Intuition Behind PCA



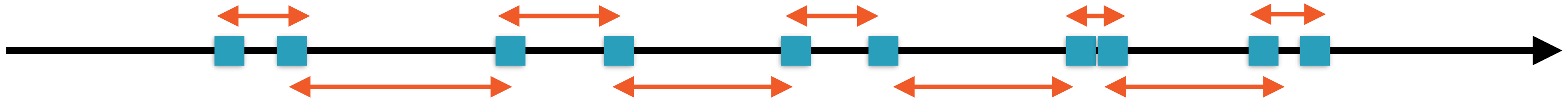
Start by "projecting" the data onto a line in some direction

Intuition Behind PCA



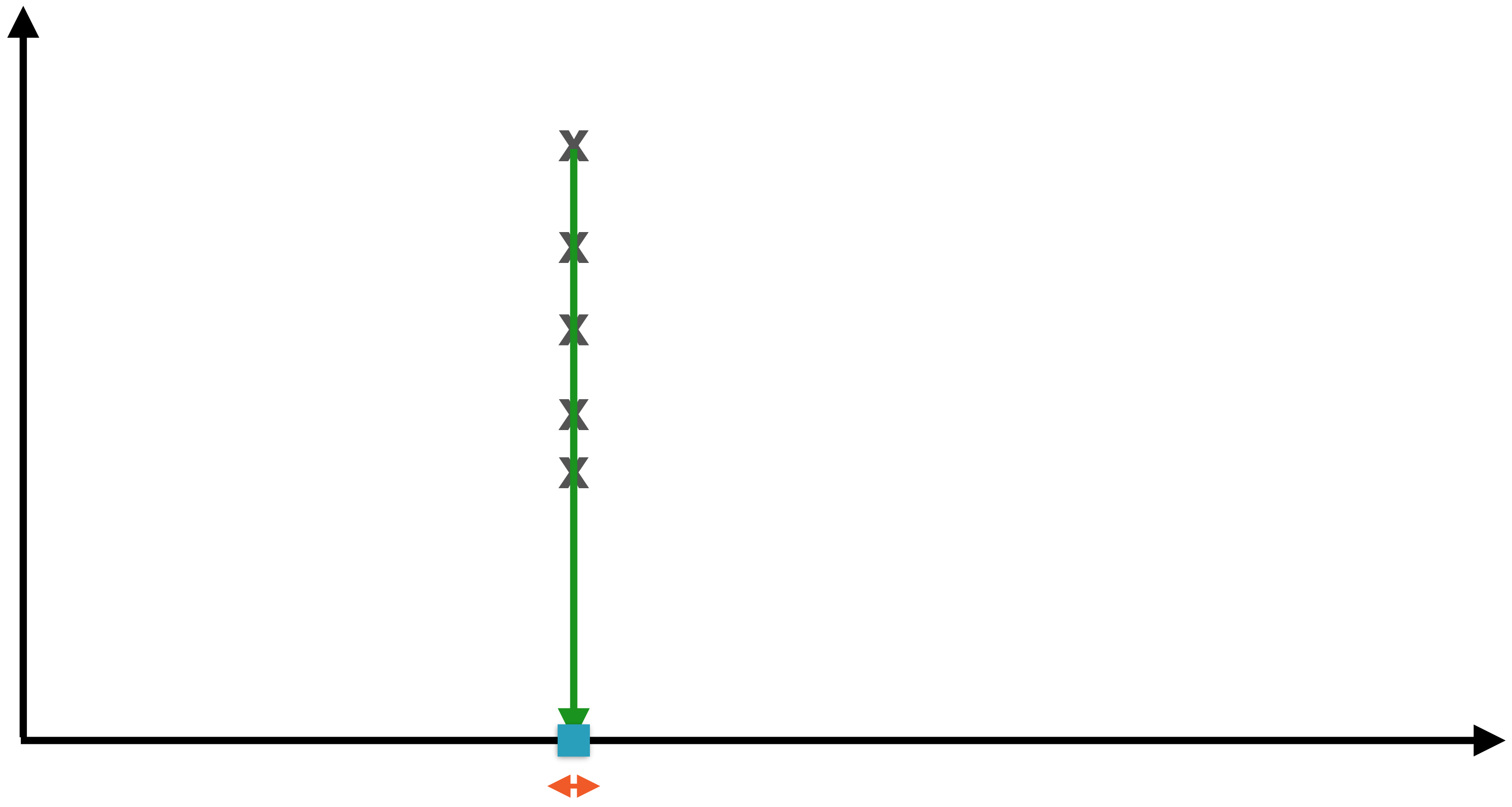
Start by "projecting" the data onto a line in some direction

Intuition Behind PCA



The greater the distances between these projections, the "better" the direction

Bad Projection

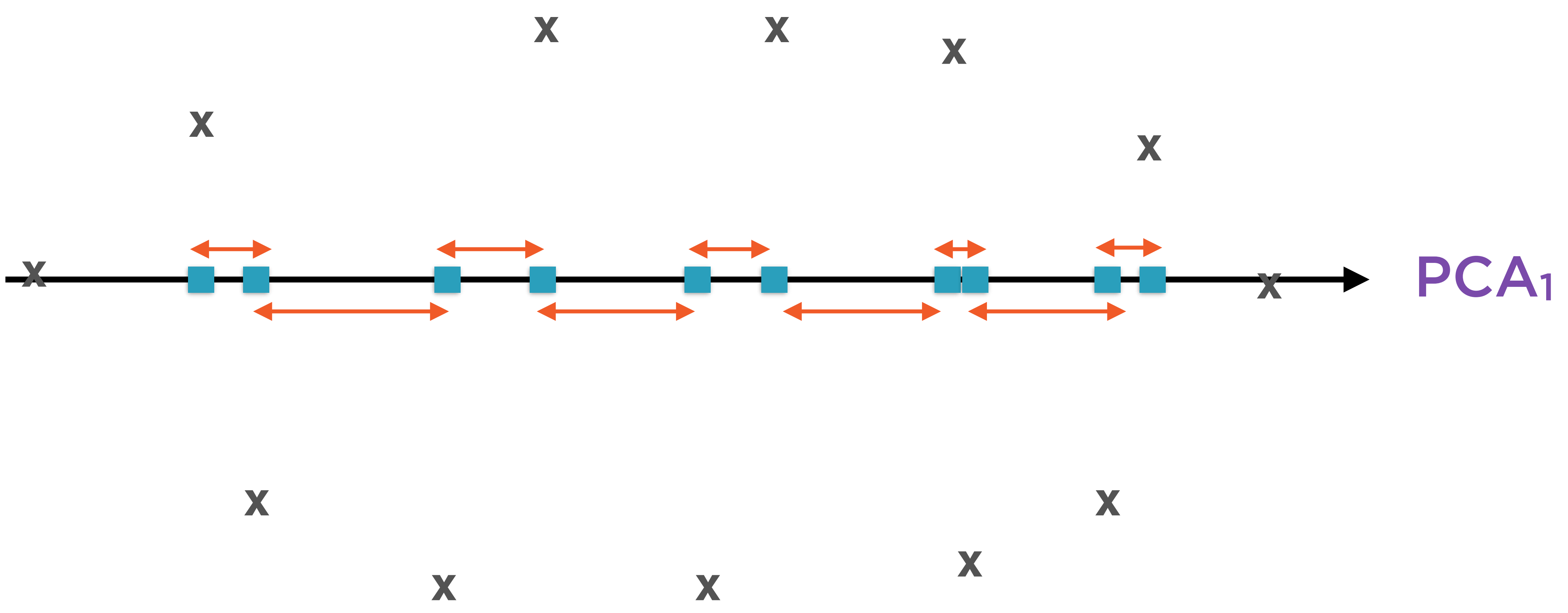


A projection where the distances are minimised is a bad one - information is lost

Good Projection

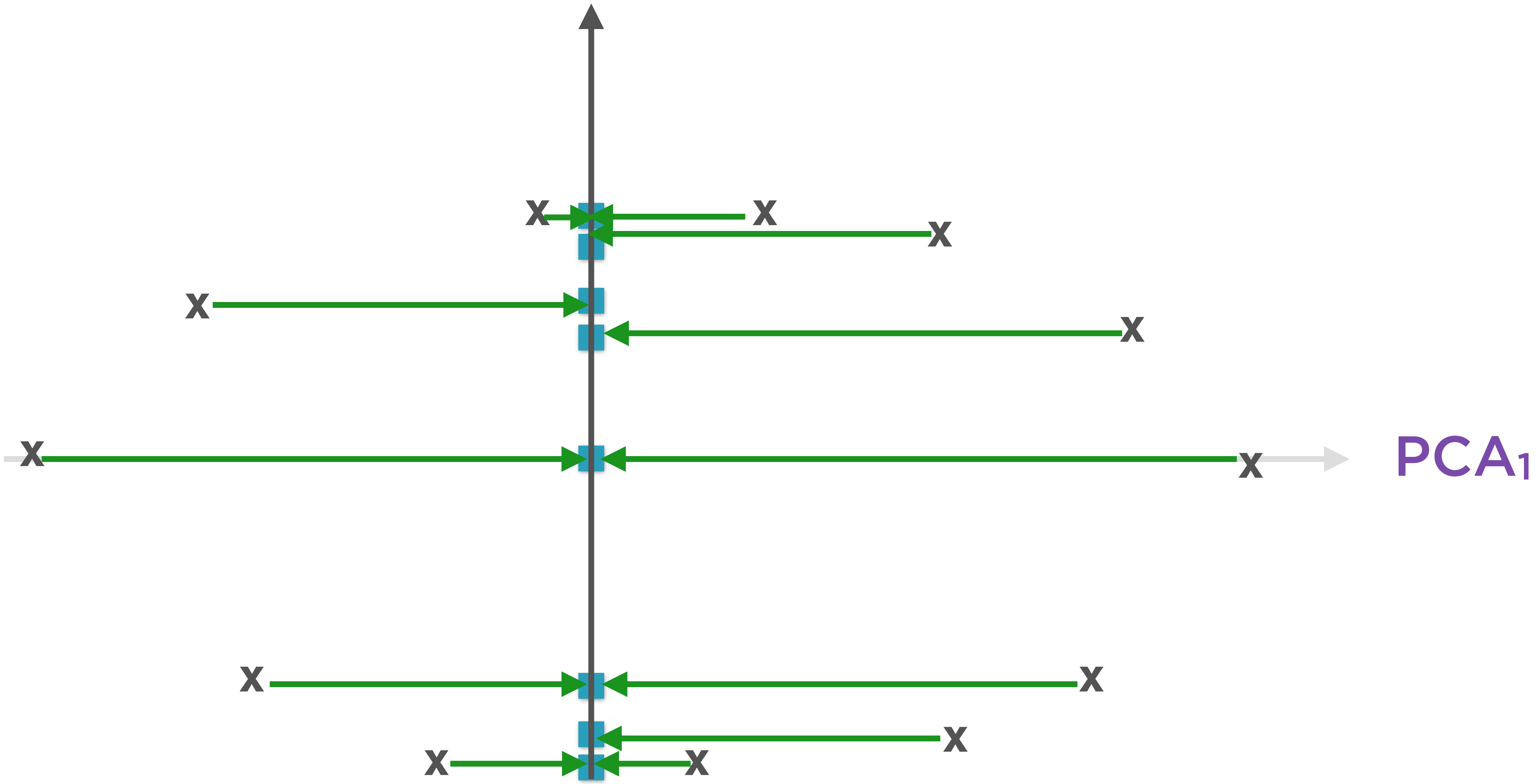
A projection where the distances are maximised is a good one - information is preserved

Intuition Behind PCA



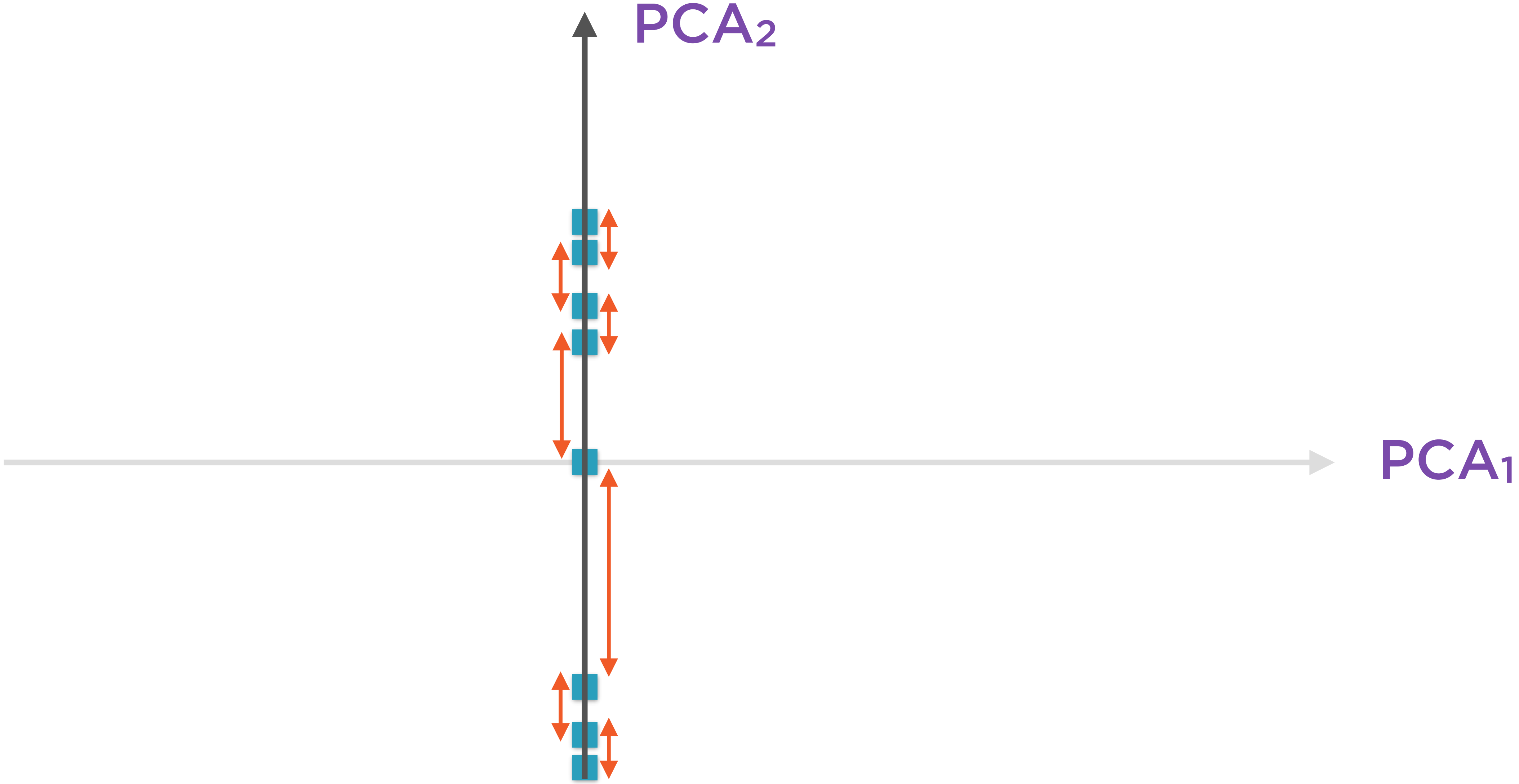
The direction along which this variance is maximised is the **first principal component** of the original data

Intuition Behind PCA



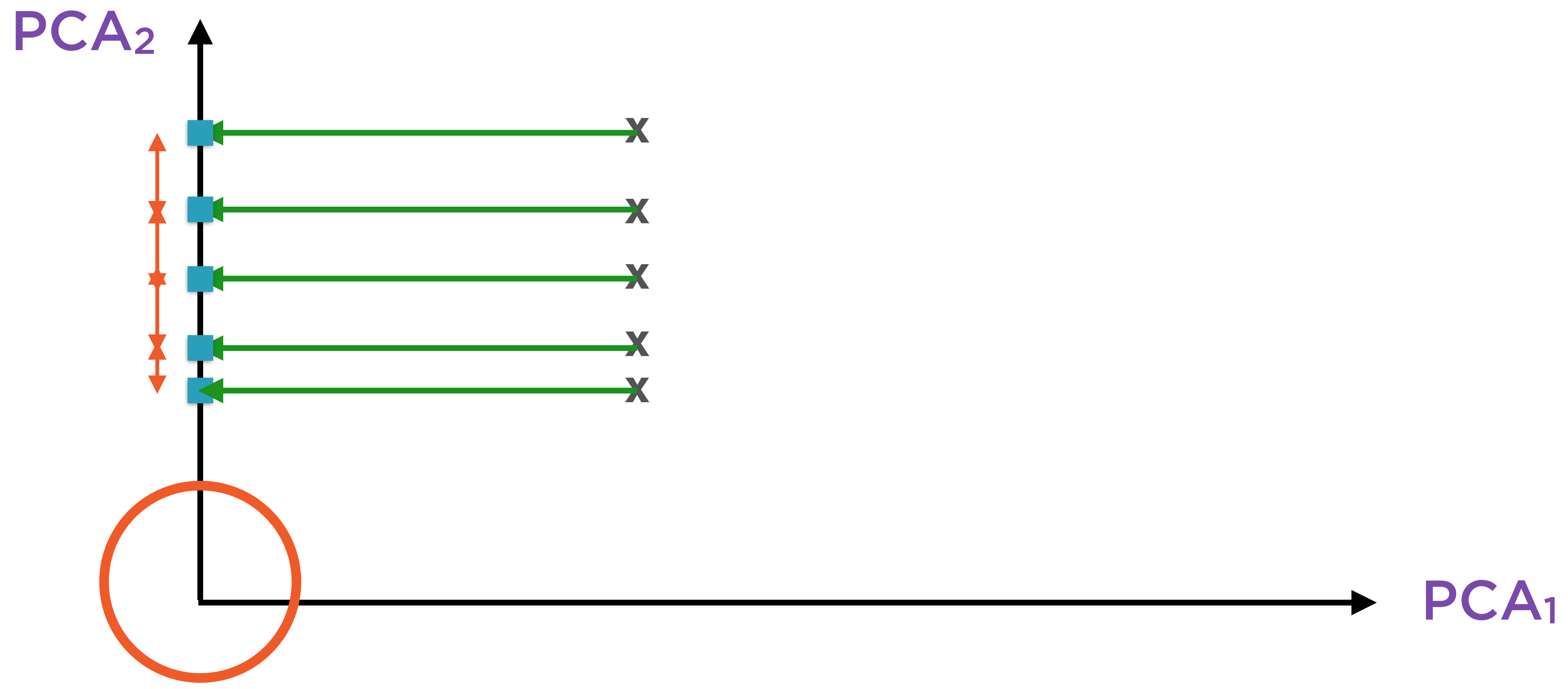
Find the next best direction, the **second principal component**, which must be at right angles to the first

Intuition Behind PCA



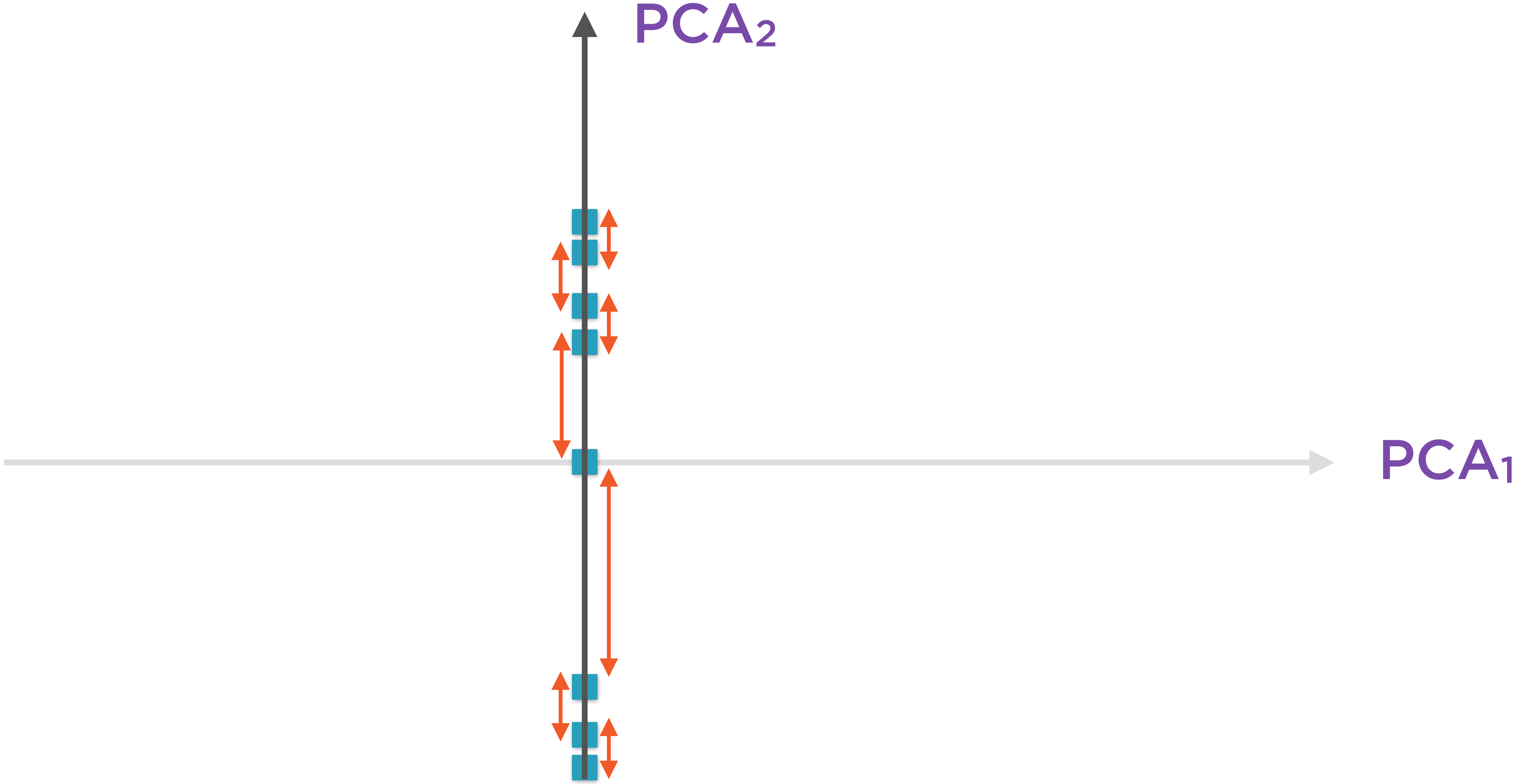
Find the next best direction, the **second principal component**, which must be at right angles to the first

Principal Components at Right Angles



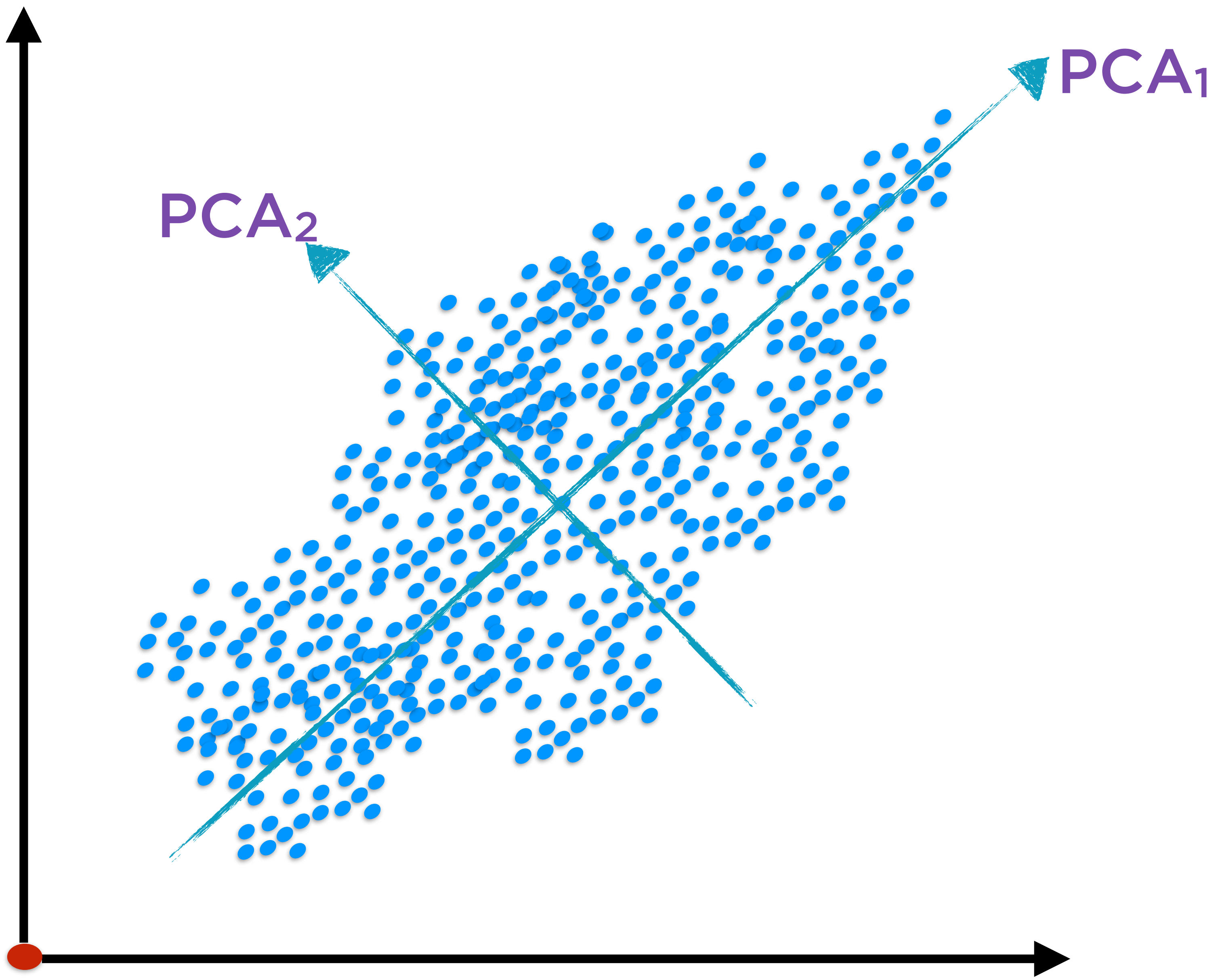
Directions at right angles help express the most variation with the smallest number of directions

Intuition Behind PCA



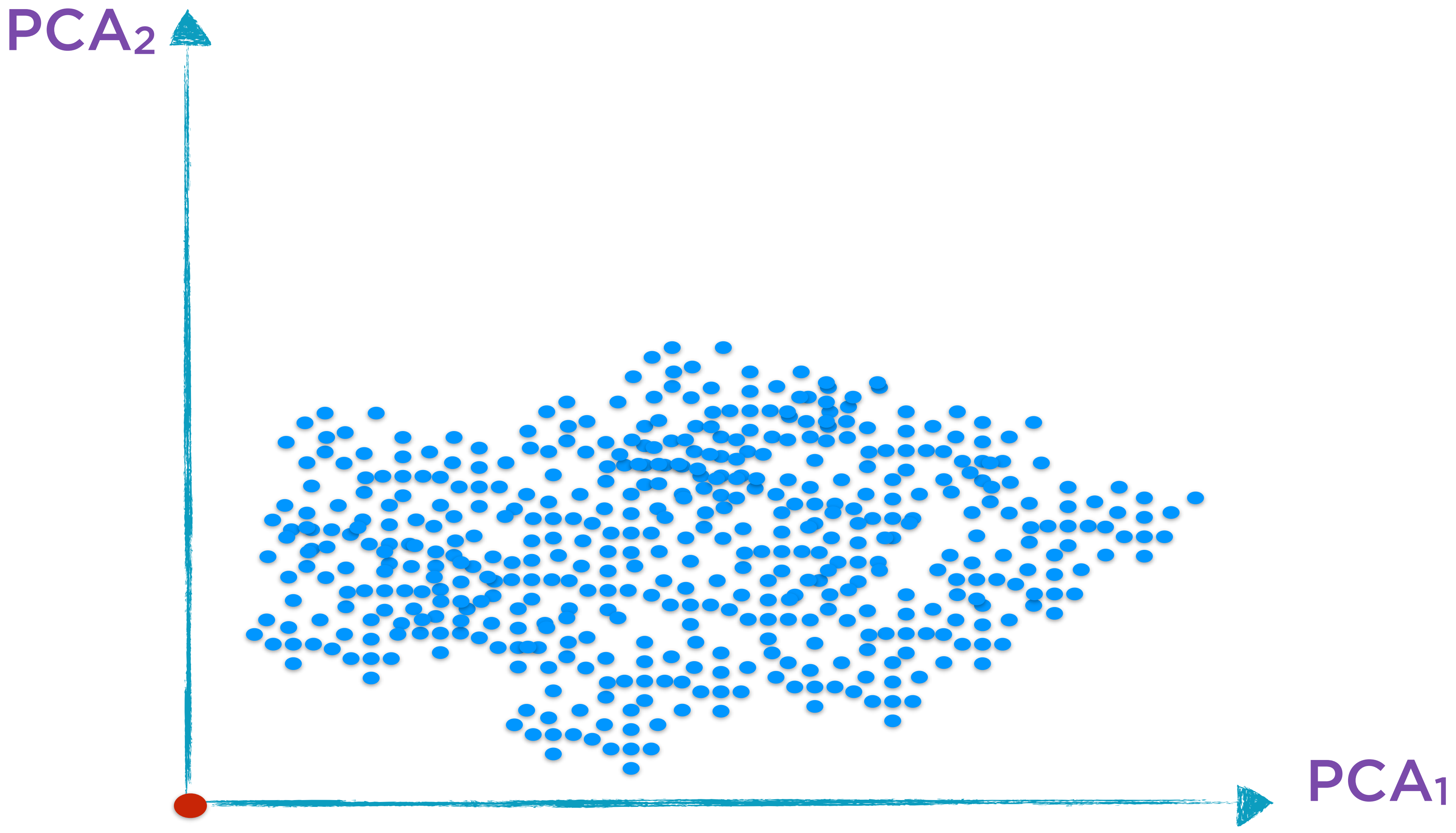
The variances are clearly smaller along this second principal component than along the first

Intuition Behind PCA



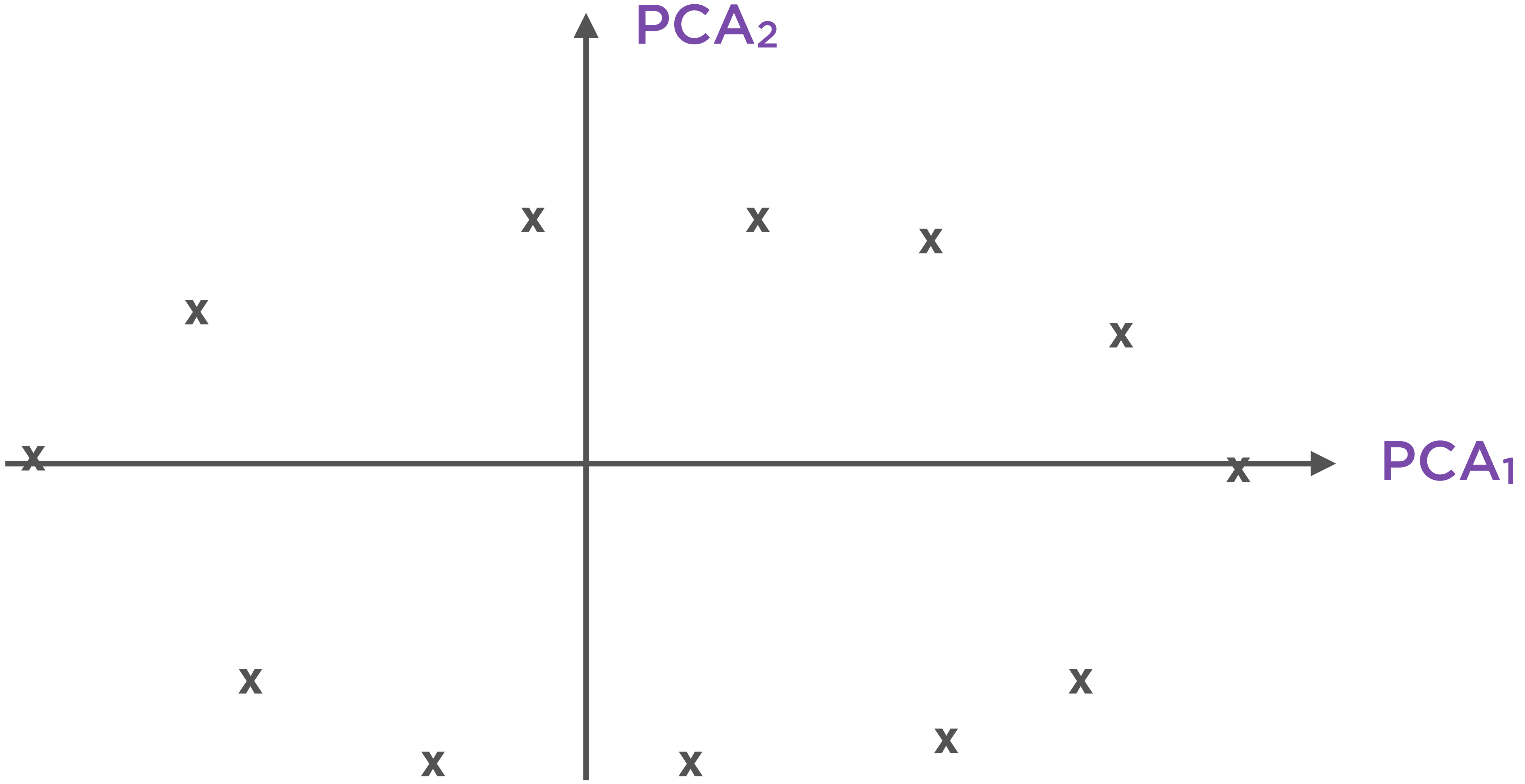
In general, there are as many principal components as there are dimensions in the original data

Intuition Behind PCA



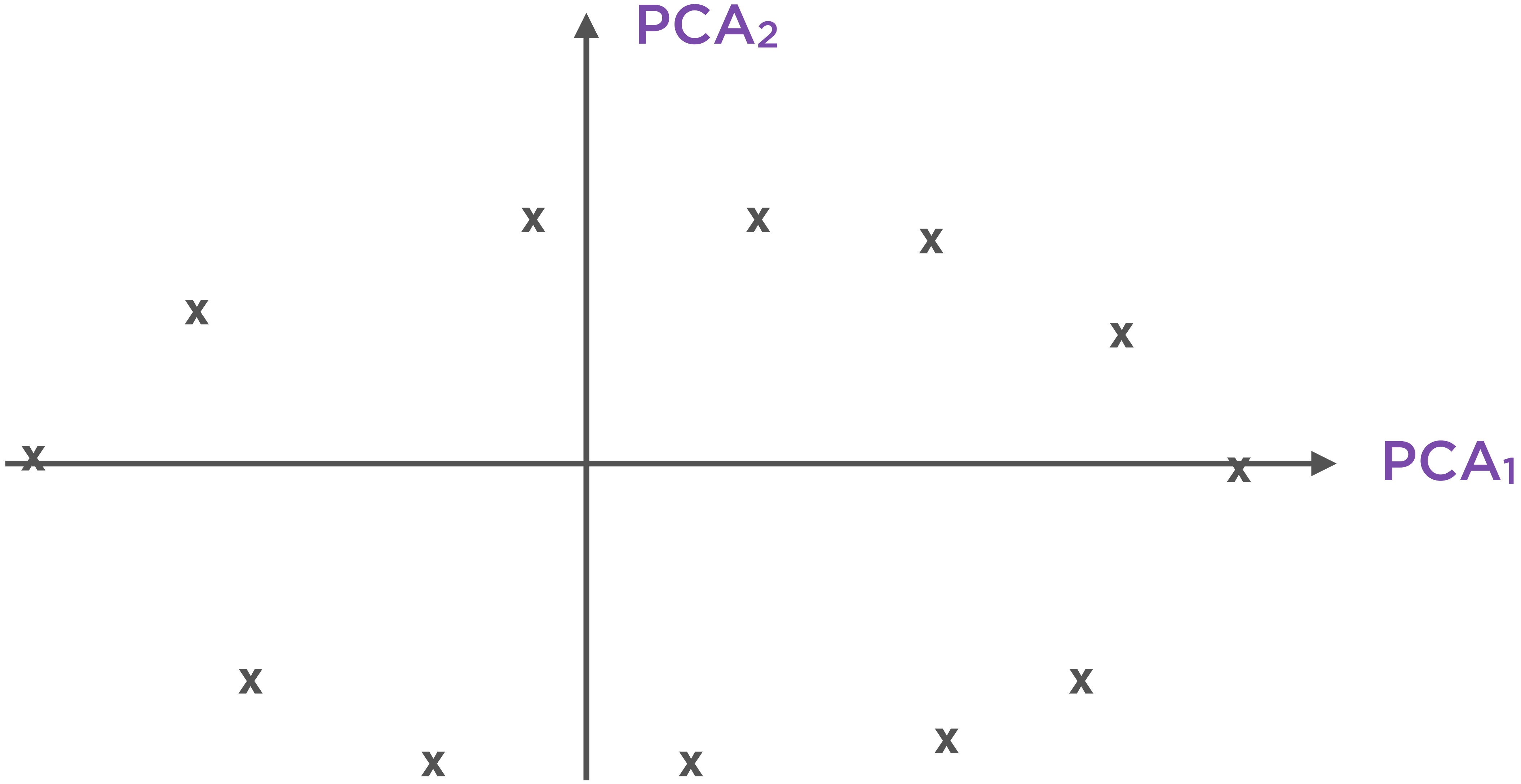
Re-orient the data along these new axes

Dimensionality Reduction



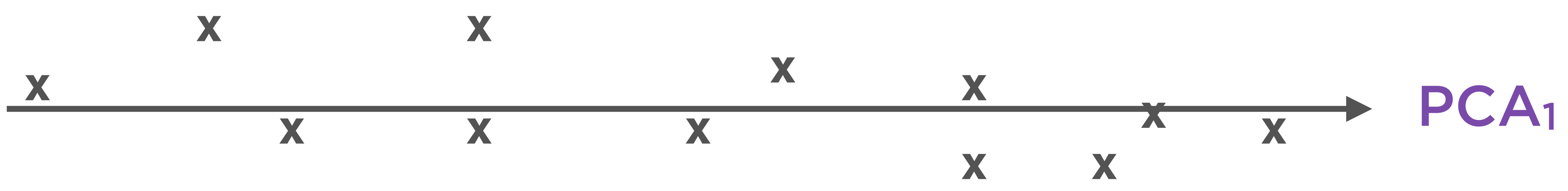
If the variance along the second principal component is small enough, we can just ignore it and use just 1 dimension to represent the data

Dimensionality Reduction



Variation along 2 dimensions: 2 principal components required

Dimensionality Reduction



Variation along 1 dimension: 1 principal component is sufficient

PCA is used for dimensionality reduction i.e. use fewer attributes to represent the same information

Choose the most **important** attributes

Demo

Use SageMaker's built-in PCA algorithm for dimensionality reduction

Represent the information in 50000 MNIST images using 10 principal components

10 images which contain the most important information from the original 50000

Summary

ML algorithms available out-of-the-box, no need to write any code for the model

Not pre-trained, model is trained on your dataset

Linear learner and PCA are examples of supervised and unsupervised models available