Creating Machine Learning Models

UNDERSTANDING APPROACHES TO MACHINE LEARNING

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

Overview

Machine learning vs. rule-based learning

Choosing the right model based on data

Supervised and unsupervised learning

Regression and classification

Clustering and dimensionality reduction

Transfer learning - cold-start vs. warm-start learning

Popular ML frameworks and their niches

Prerequisites and Course Outline

Prerequisites

Basic Python programming

Built and trained simple machine learning models

Basic understanding of the machine learning workflow

Prerequisites

Python Fundamentals

Understanding Machine Learning

Building Your First scikit-learn Solution

Course Outline

Approaches to machine learning

Regression models

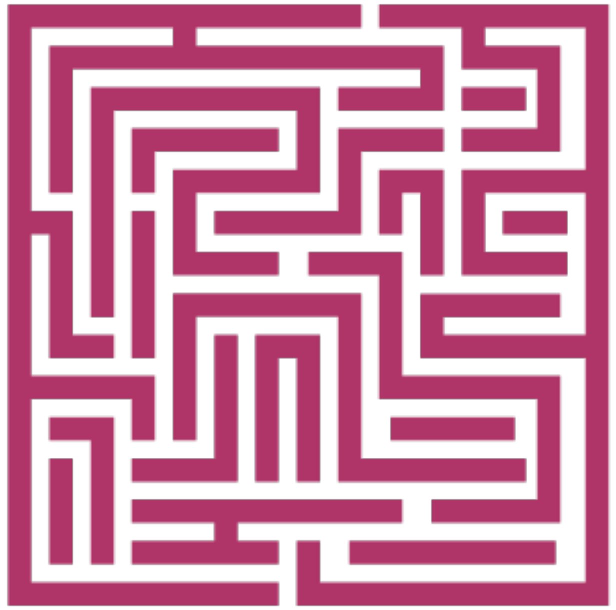
Classification models

Clustering models

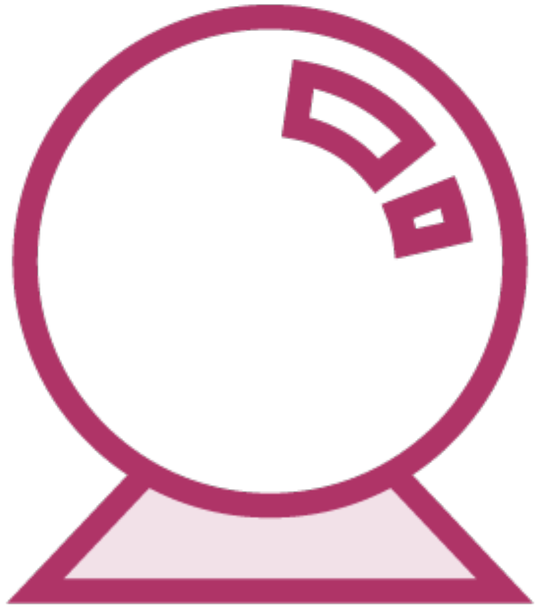
Rule-based vs. ML-based Learning

A machine learning algorithm is an algorithm that is able to learn from data

Machine Learning



Work with a huge maze of data



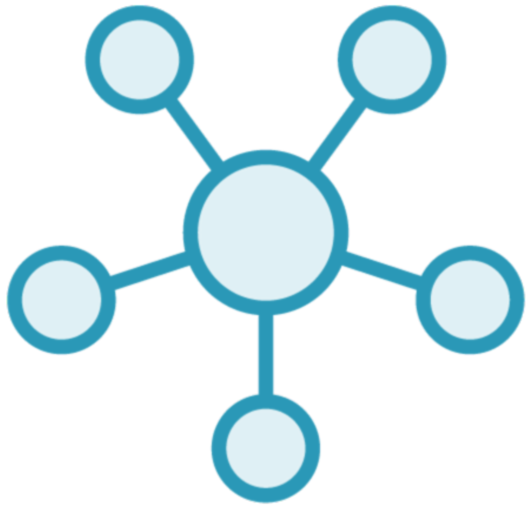
Find patterns

Make intelligent decisions

Broad Problem Categories

Classification

Regression



Clustering

Dimensionality reduction

Broad Problem Categories

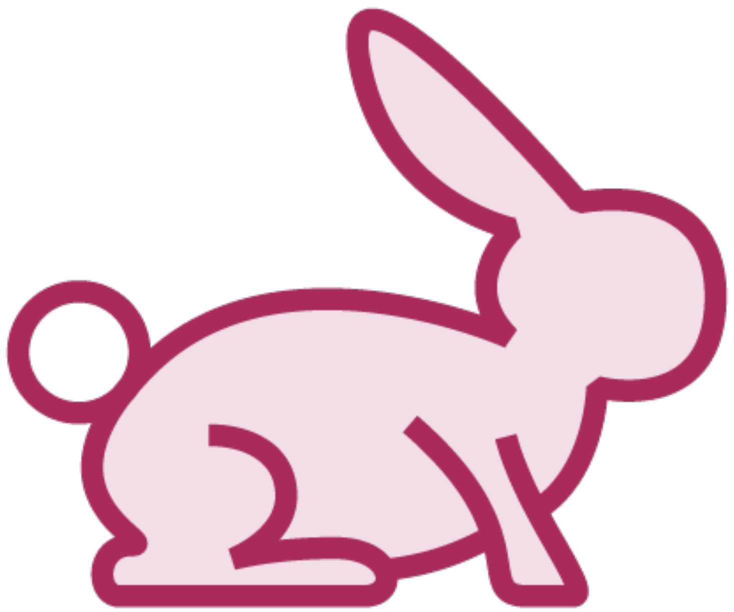
Classification

Regression

Clustering

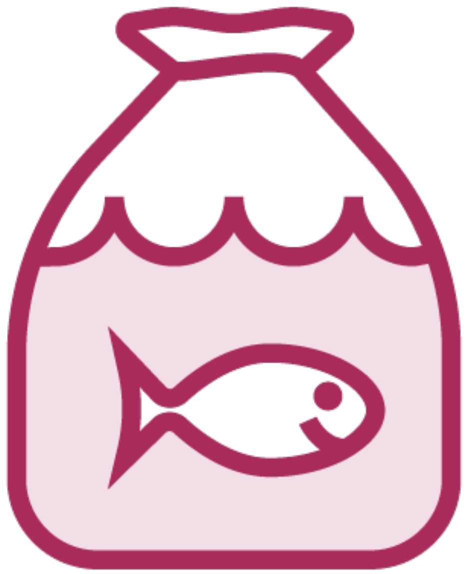
Dimensionality reduction

Whales: Fish or Mammals?



Mammals

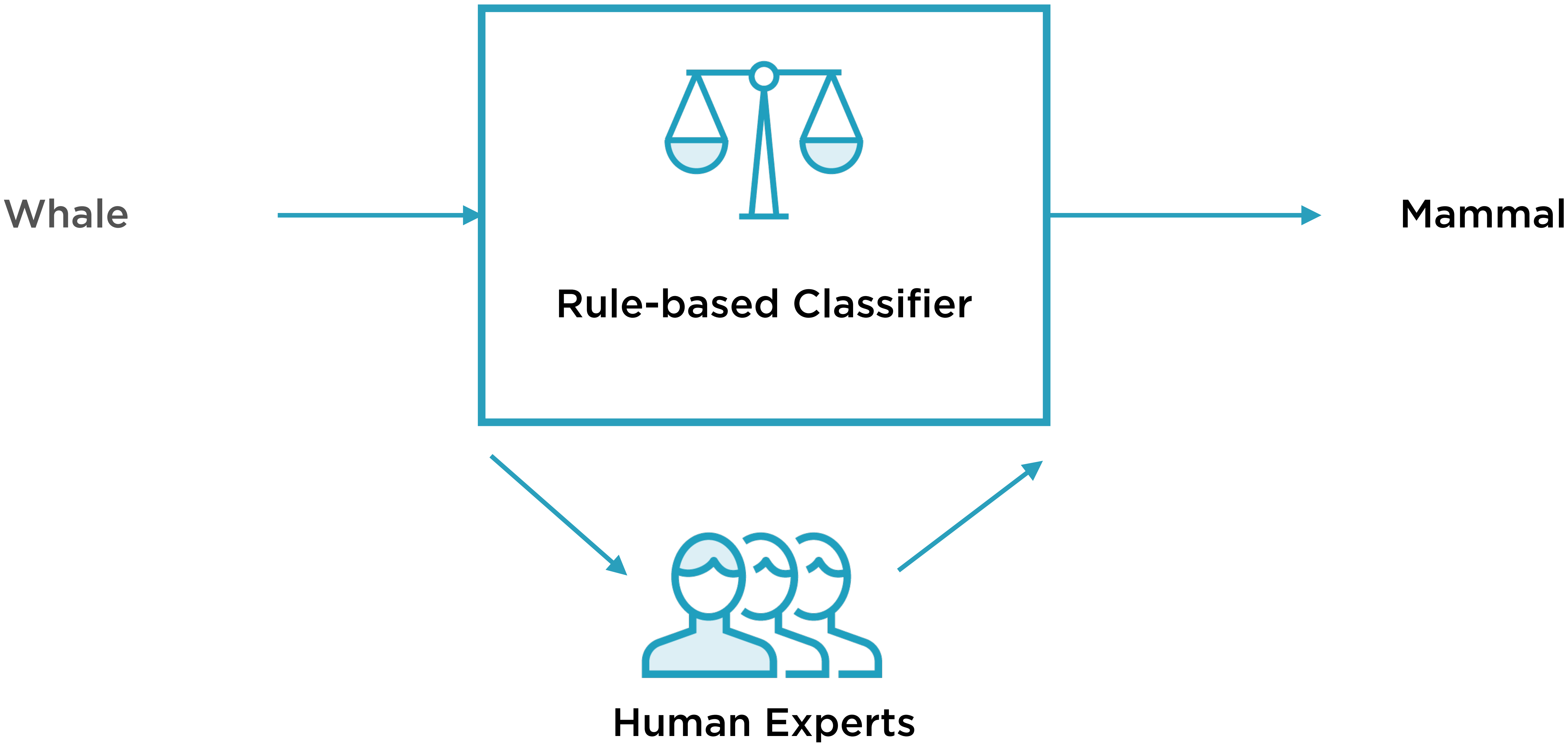
Members of the infraorder *Cetacea*



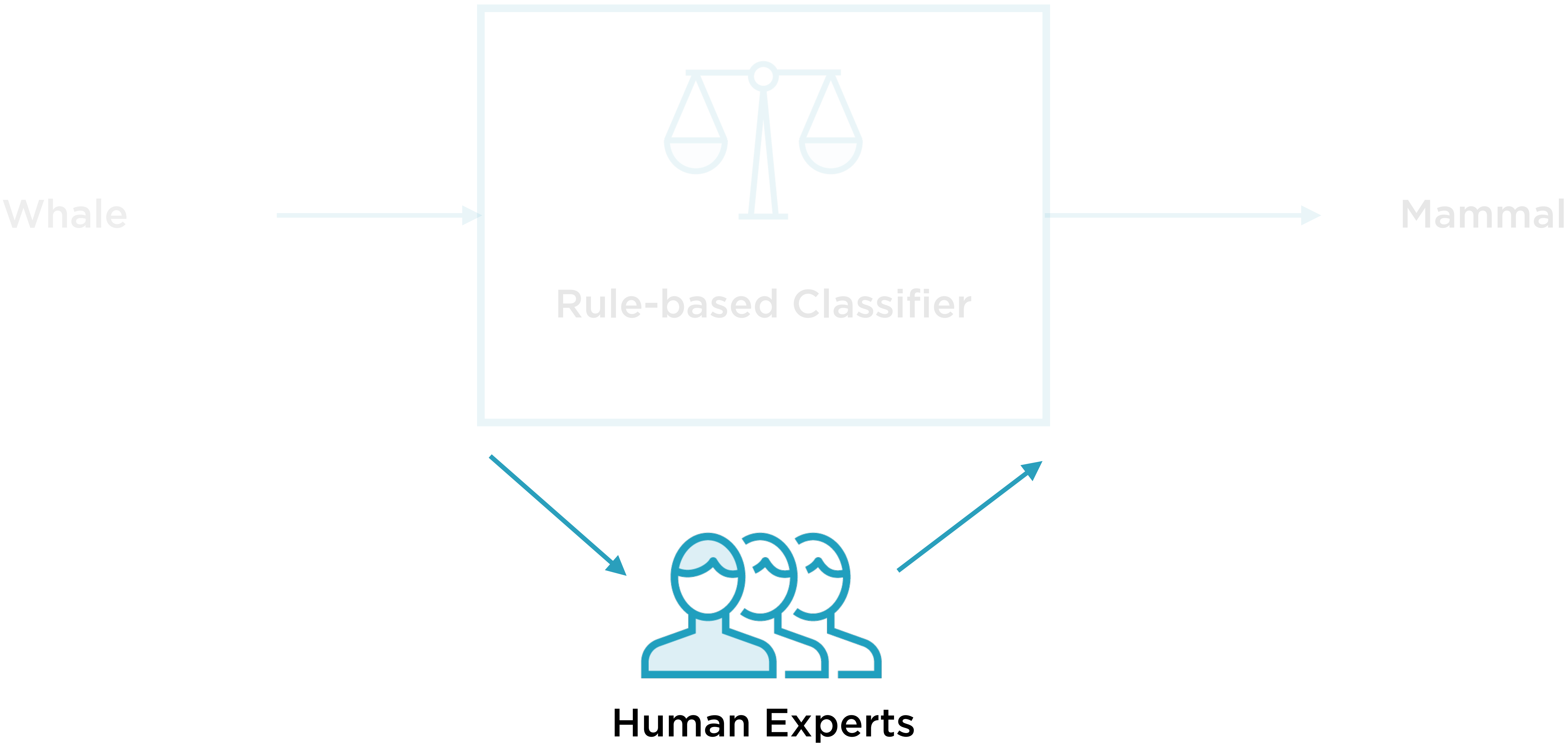
Fish

Look like fish, swim like fish, move with fish

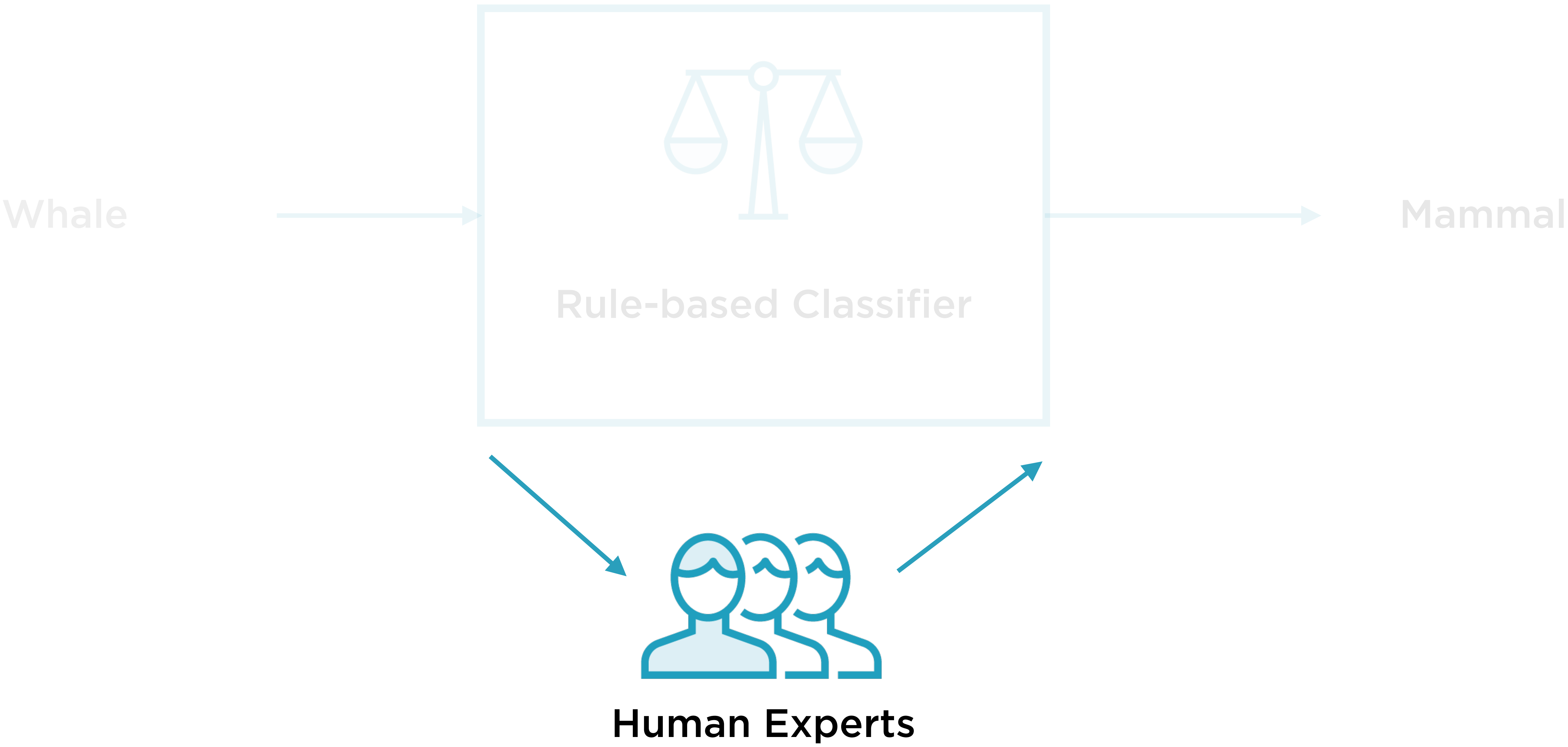
Rule-based Binary Classifier



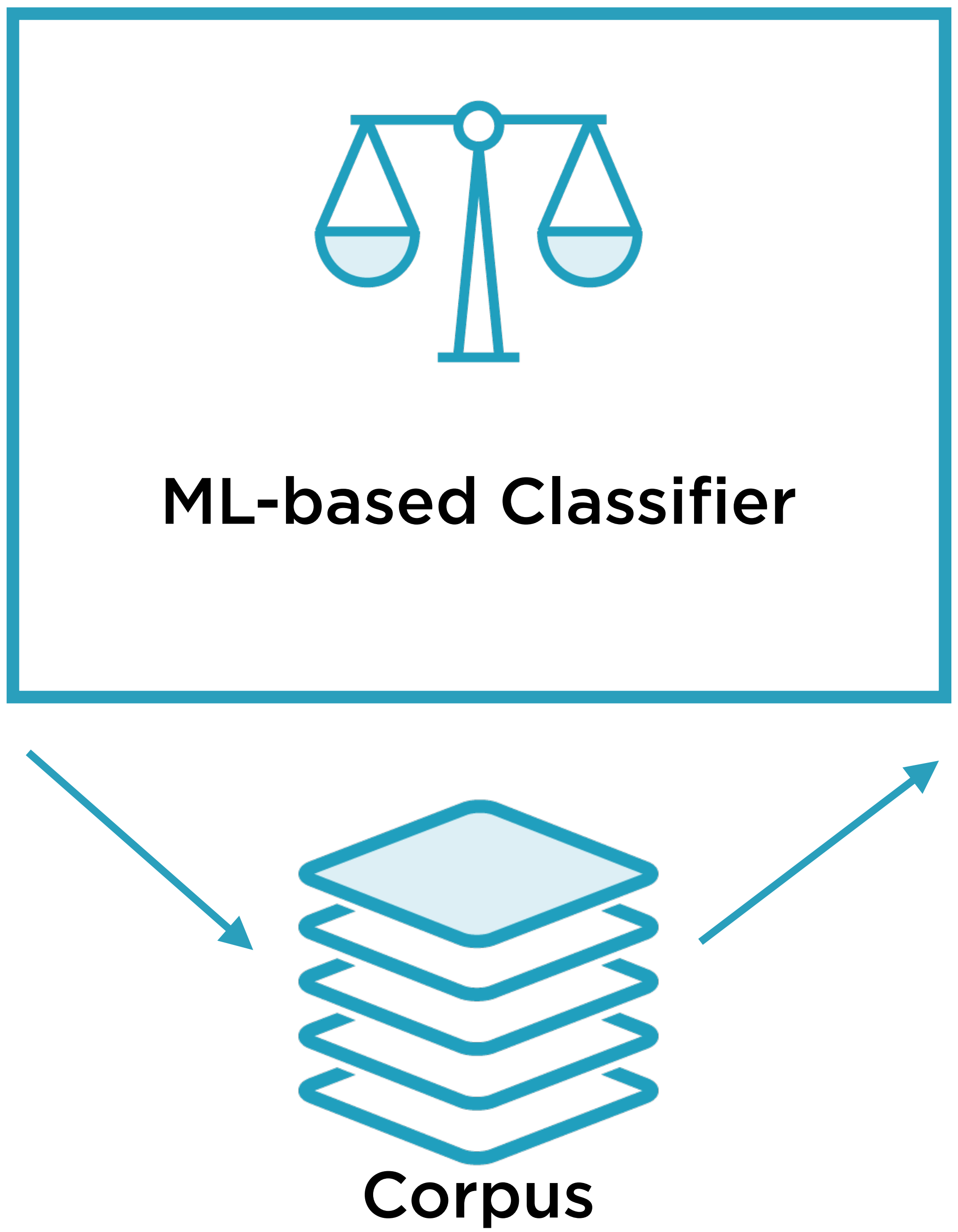
Human Experts Formulate Rules



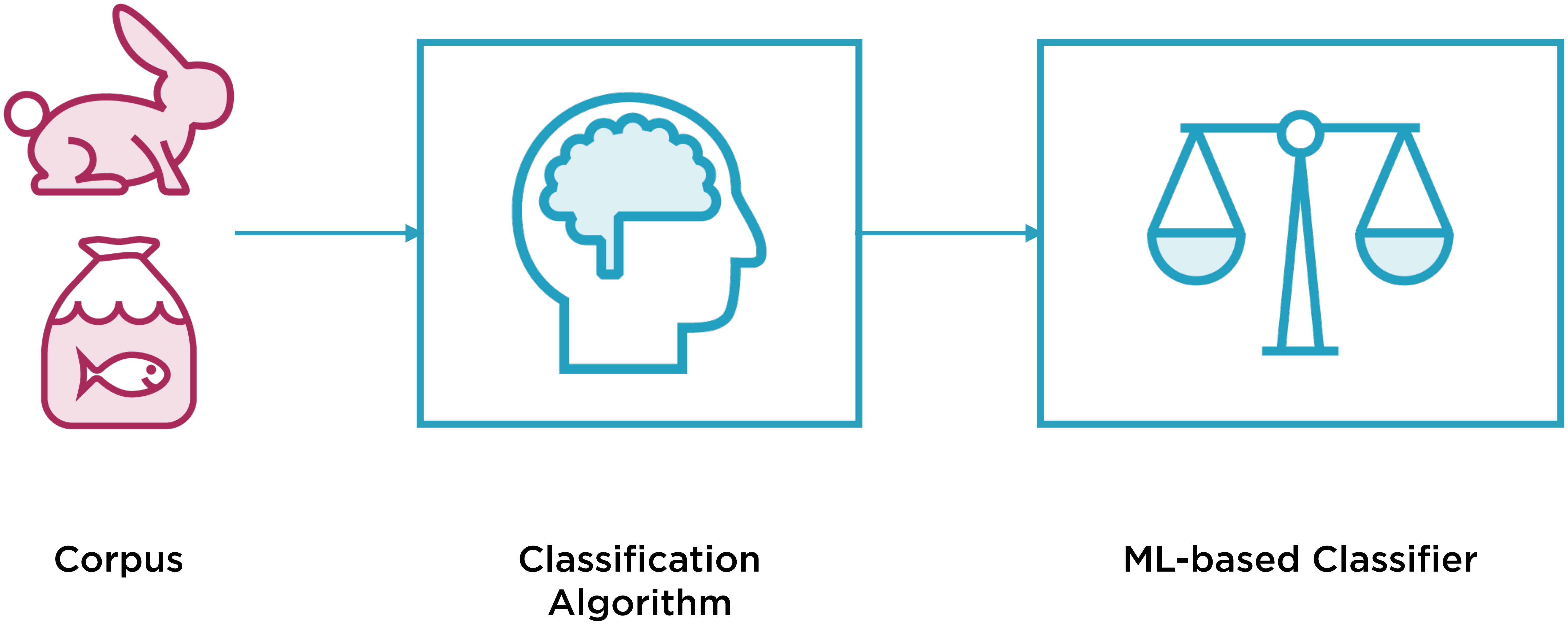
Rules Specific to Problem and Data



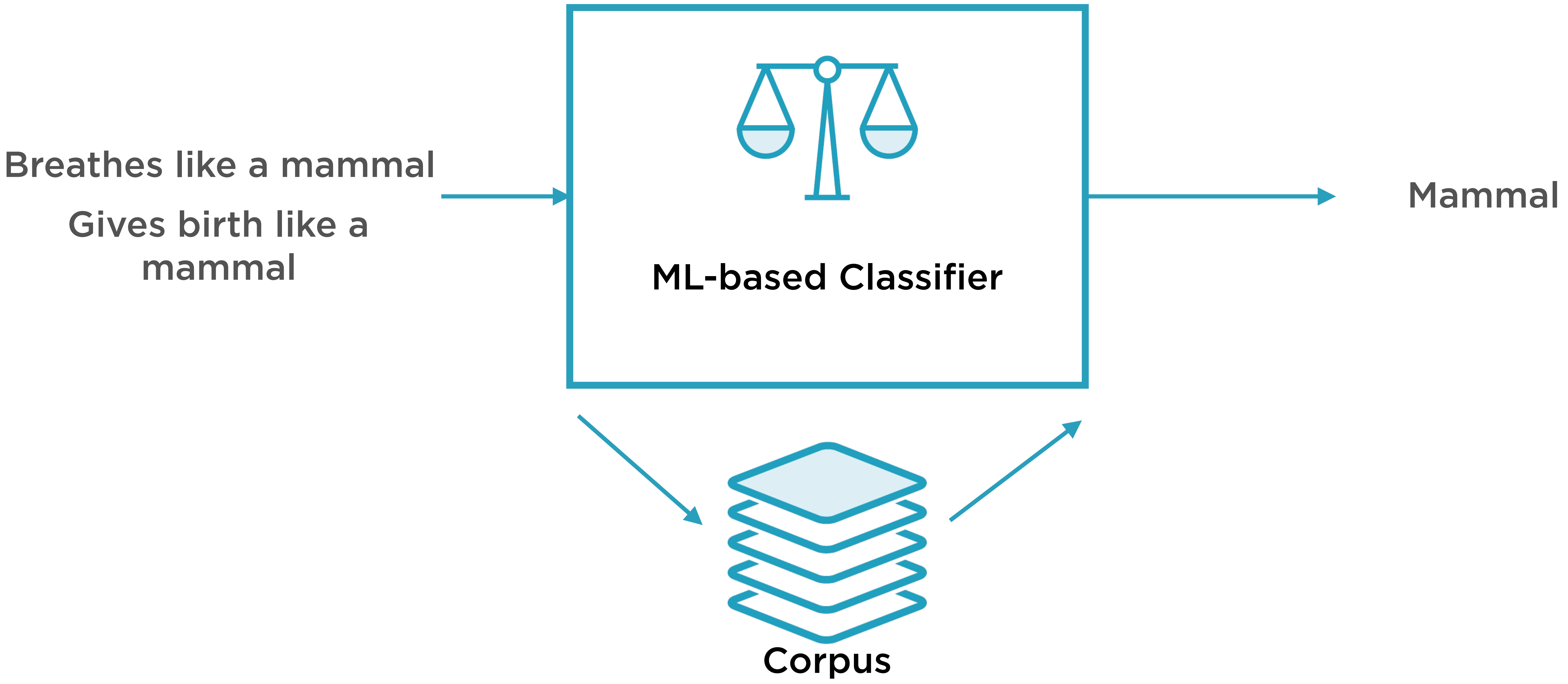
ML-based Binary Classifier



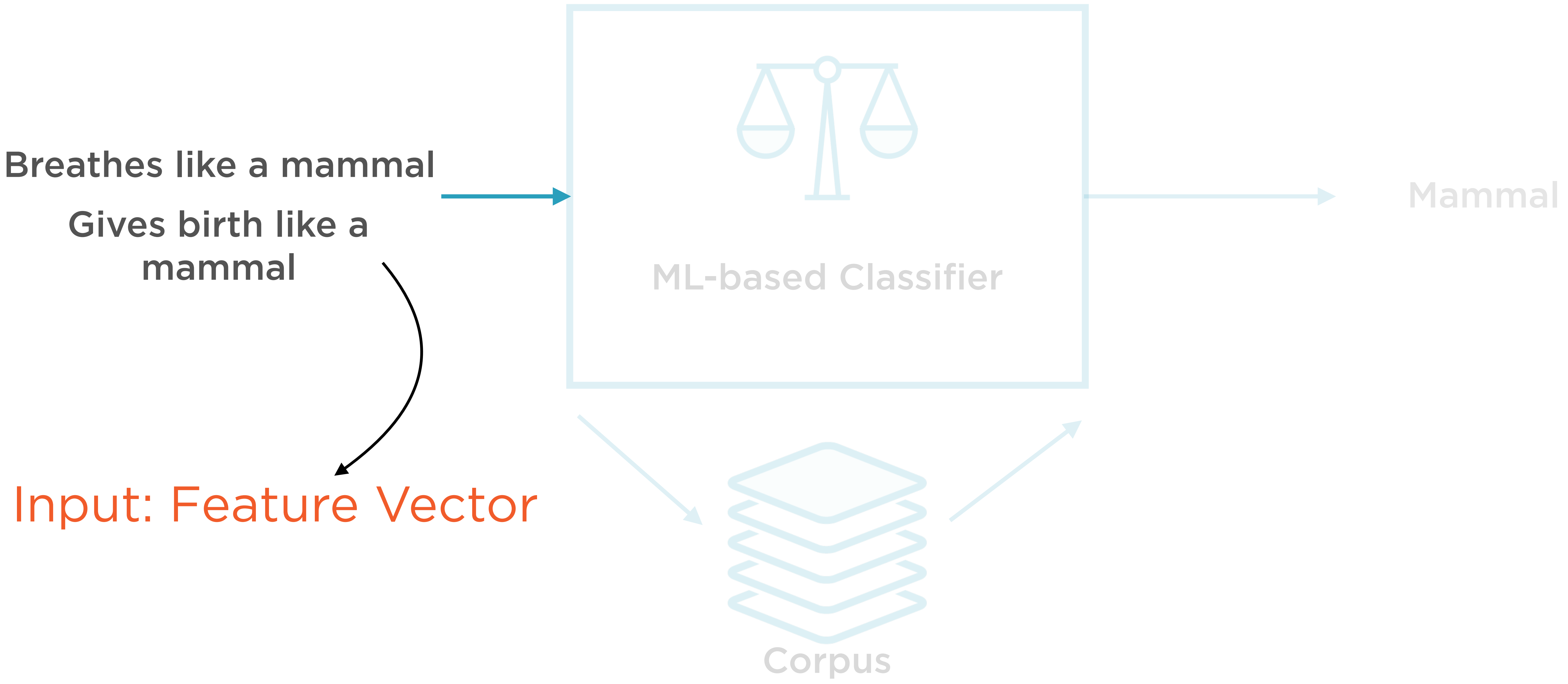
ML-based Binary Classifier



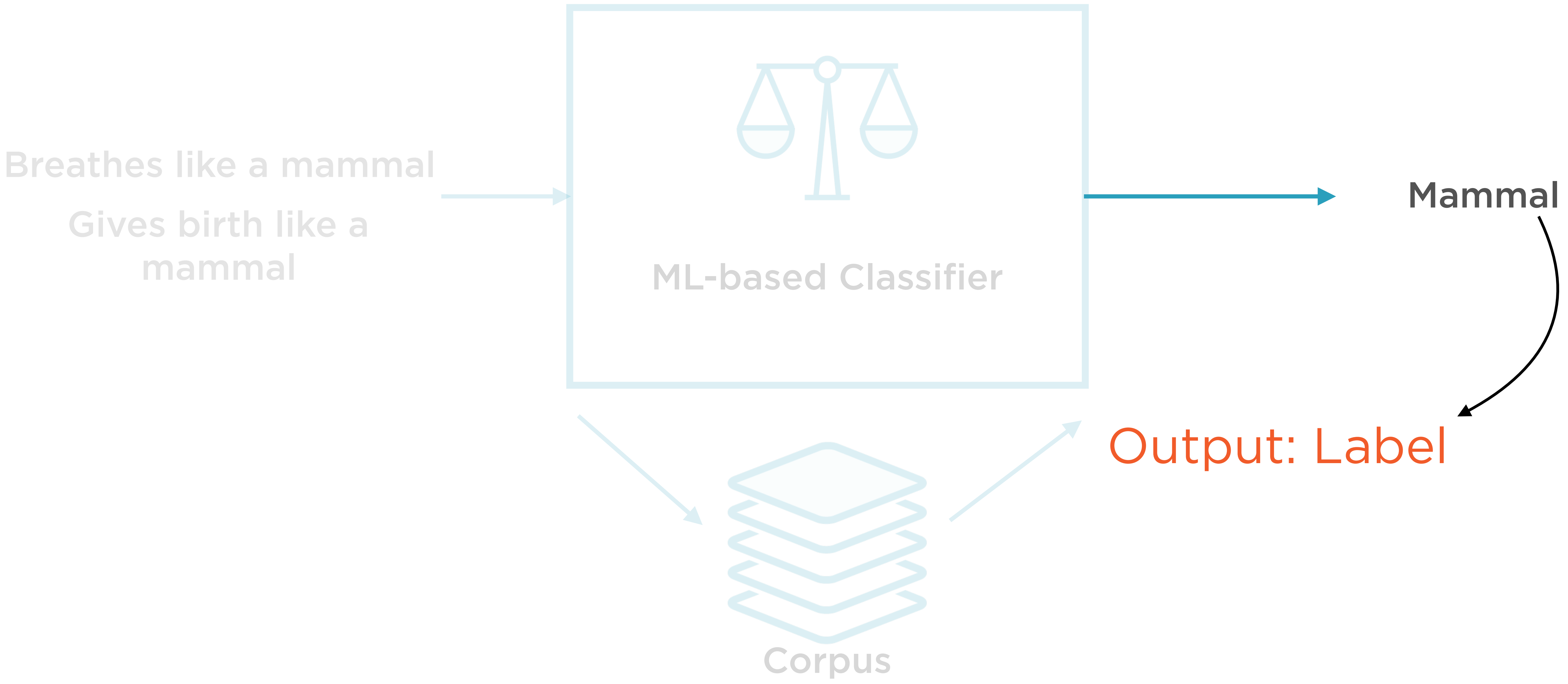
ML-based Binary Classifier



ML-based Binary Classifier



ML-based Binary Classifier



Rule-based Analysis

Problem statement is fairly simple

Rules are straightforward and can be easily codified

Rules change infrequently

Few problem instances to train ML models

ML-based Analysis

Problem statement is reasonably complex

Hard to find patterns using visualizations and other exploratory tools

Decision variables sensitive to data, need to change as new information is received

Large corpus available to train models

ML-based and Rule-based Models

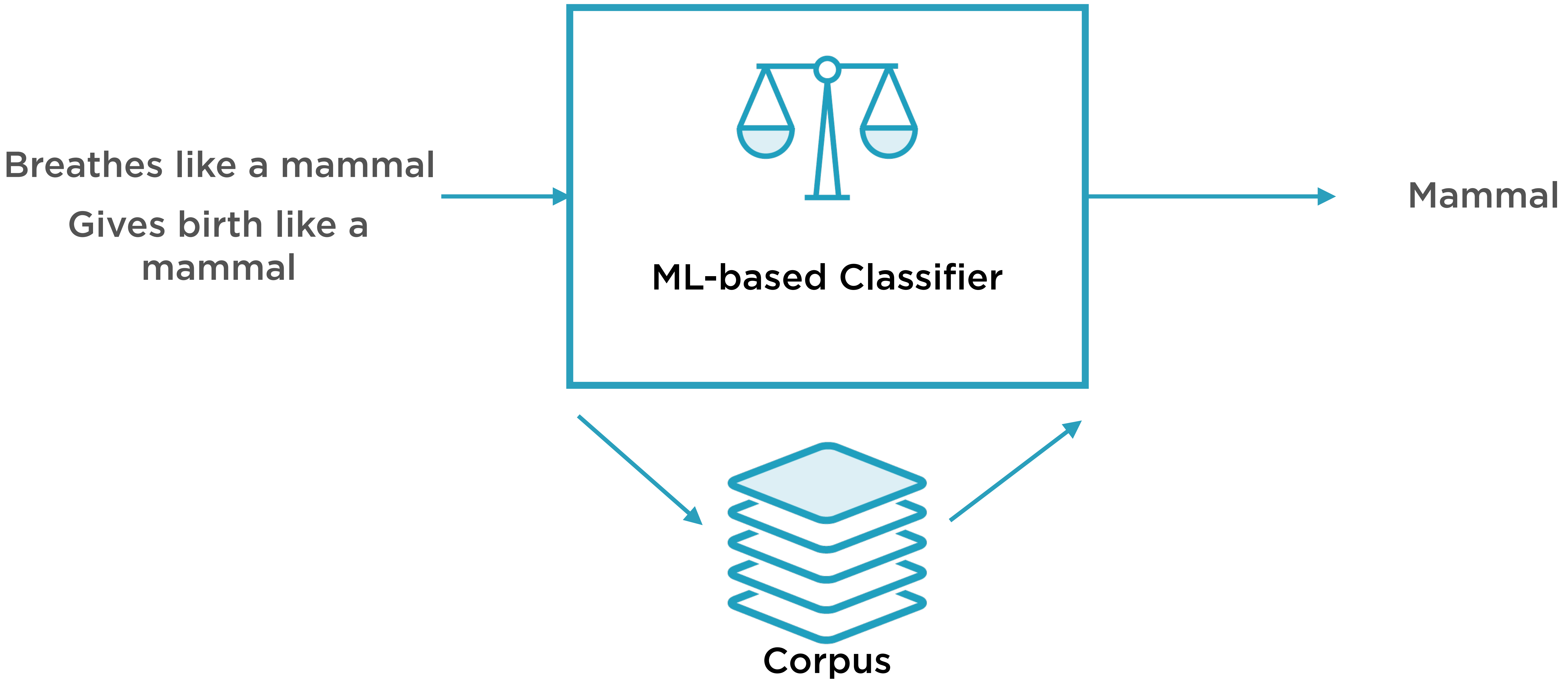
ML-based	Rule-based
Dynamic - alter output based on patterns in data	Static - rules are applied independent of data
Expert skill not needed, need an intuition for how models work	Experts vital for formulating rules, experts based on problem
To update model, update corpus	To update model, need to update rules i.e. recode model

ML-based and Rule-based Models

ML-based	Rule-based
Large, high-quality data corpus	No corpus required
Can not operate on a single problem instance	Can operate on isolated problem instances
Explicit training step	No training step required

Traditional ML vs. Representation ML

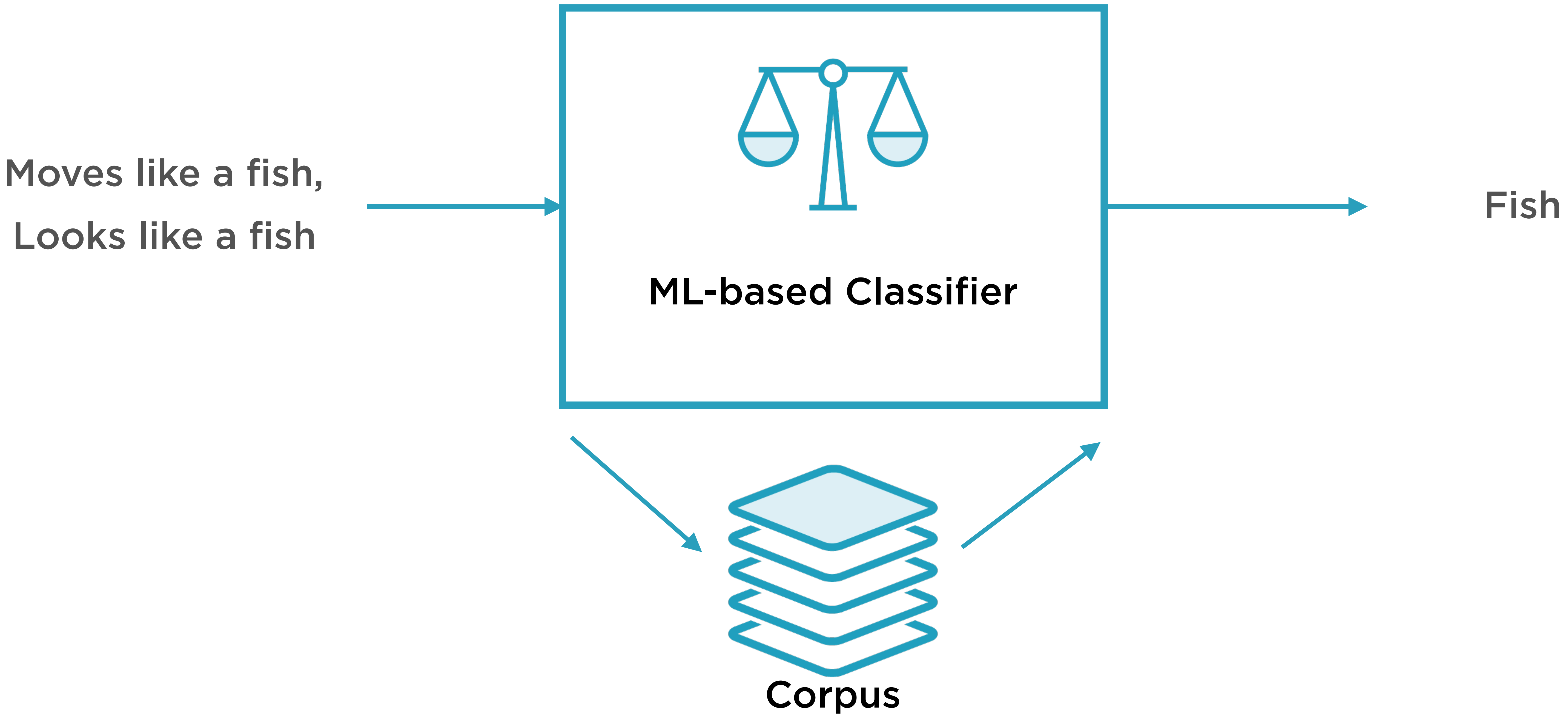
ML-based Binary Classifier



Garbage In, Garbage Out

If data fed into an ML model is of poor quality, the model will be of poor quality

ML-based Binary Classifier

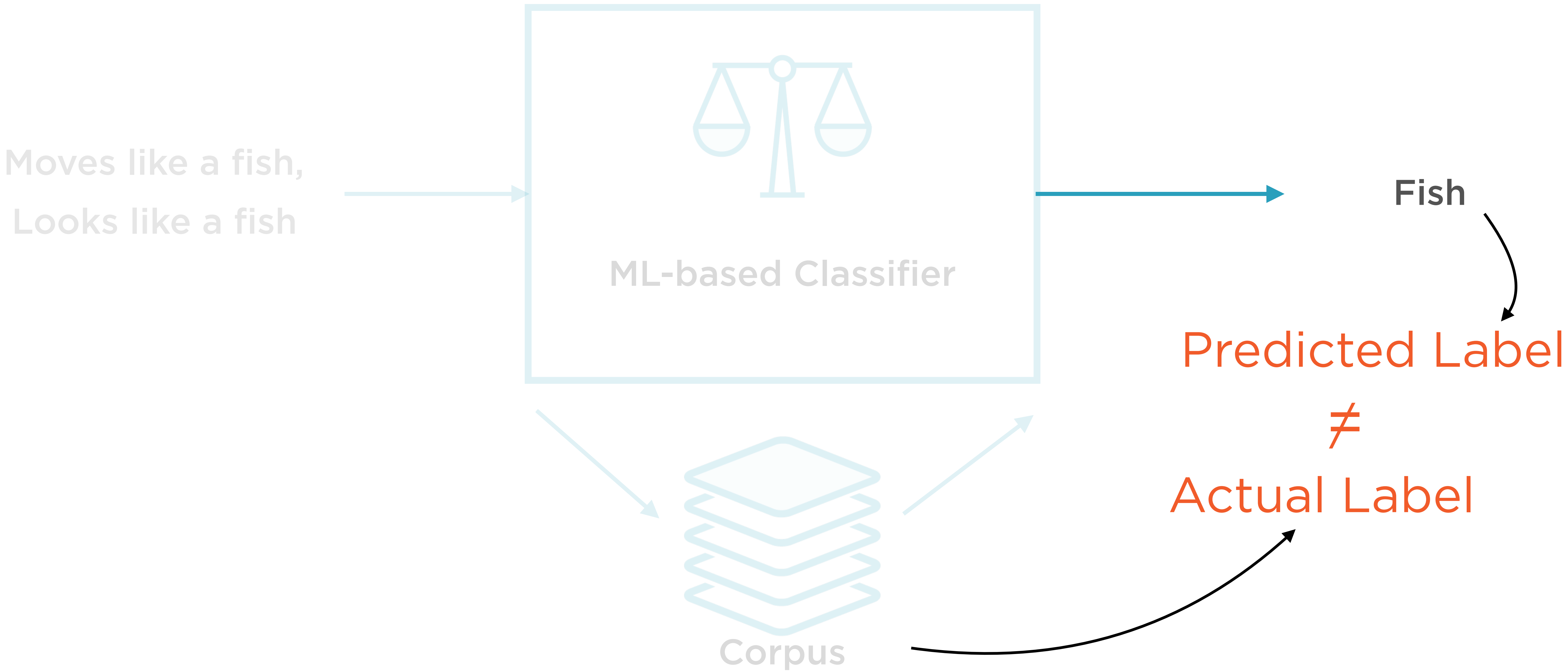


ML-based Binary Classifier

Moves like a fish,
Looks like a fish

Input: Poorly defined features

ML-based Binary Classifier



Traditional ML models require experts to specify the right features

Representation ML models extract the right features by themselves

Traditional ML Models

Regression models: Linear, Lasso, Ridge, SVR

Classification models: Naive Bayes, SVMs, Decision trees, Random forests

Dimensionality Reduction: Manifold learning, factor analysis

Clustering: K-means, DBSCAN, Spectral clustering

Representation ML Models

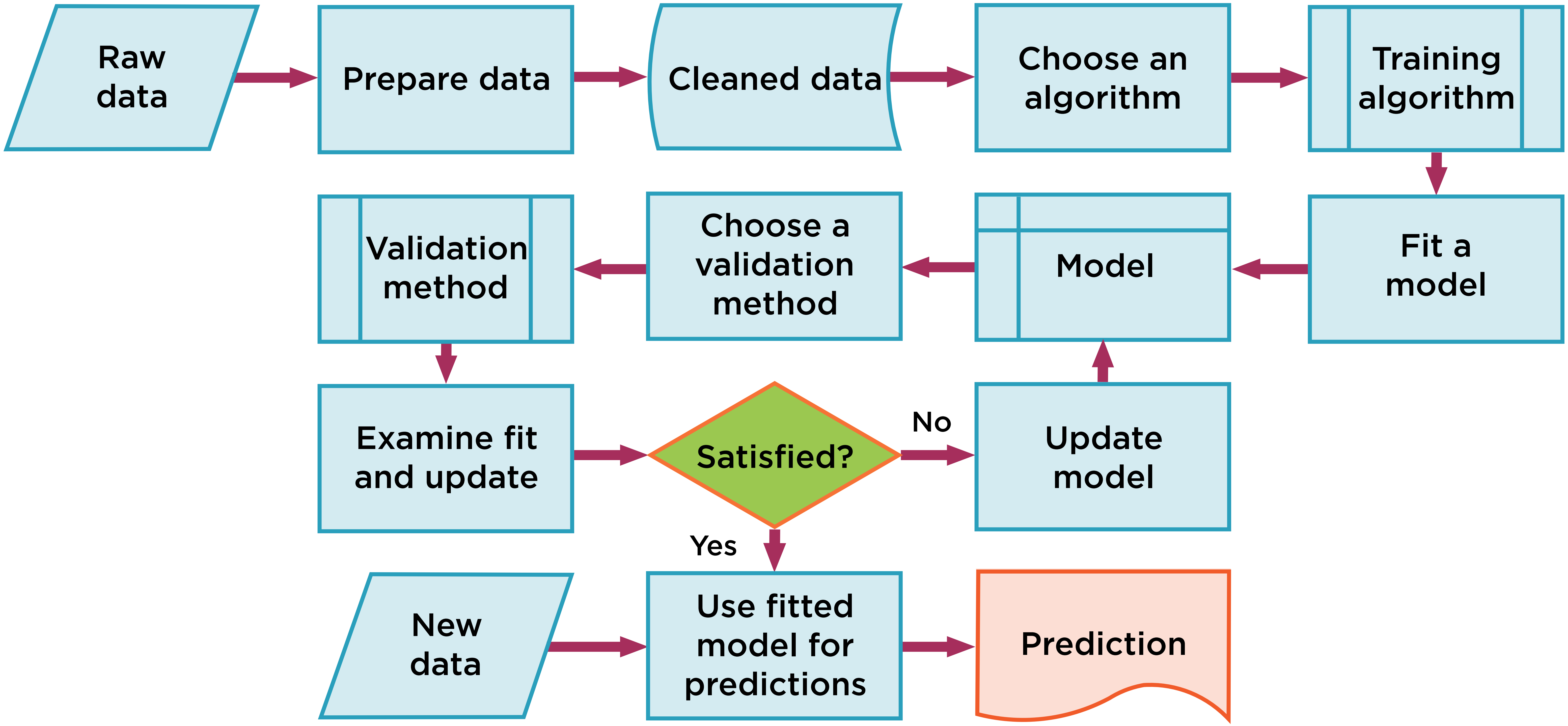
Deep learning models such as neural networks

Also used to solve classification, regression, clustering, dimensionality reduction

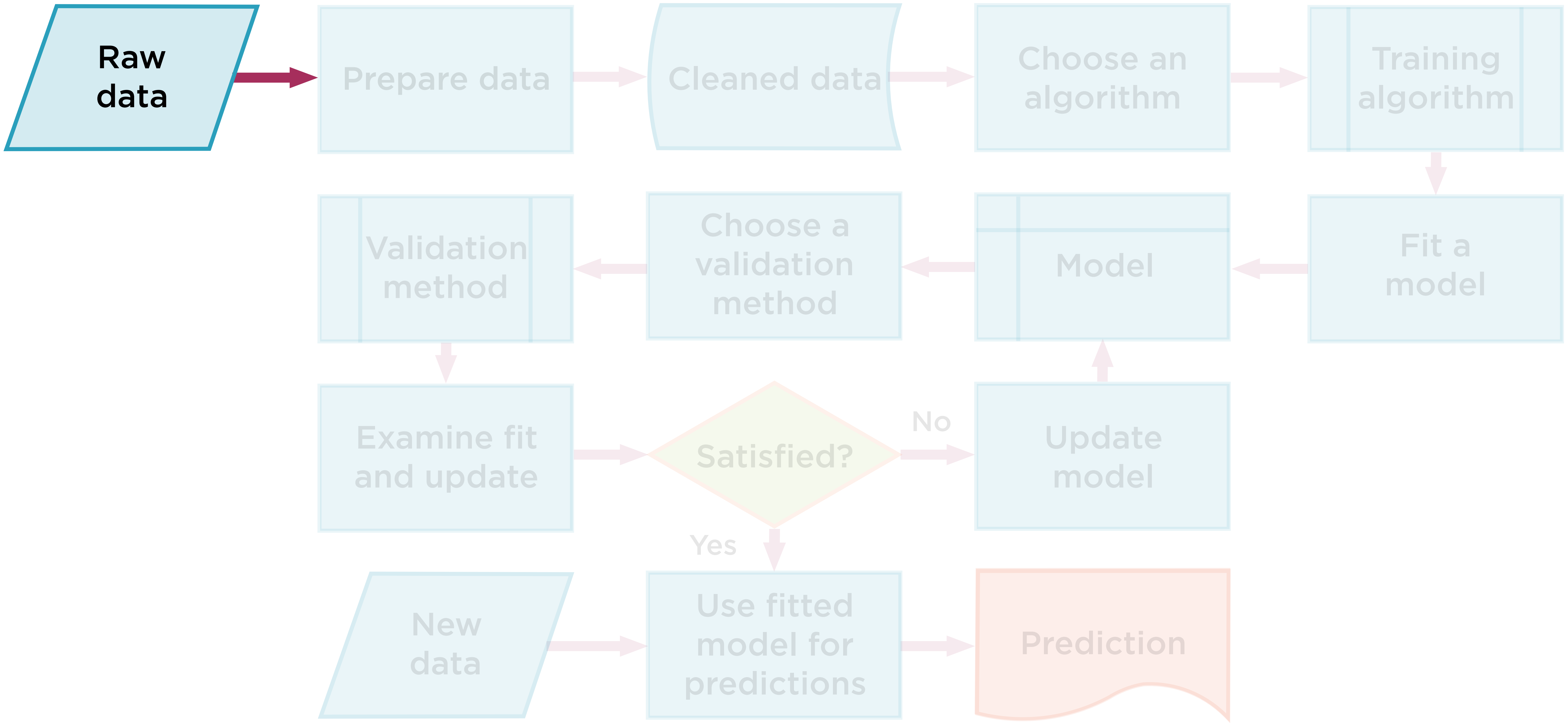
However internal workings rely on neural network architectures

The Machine Learning Workflow

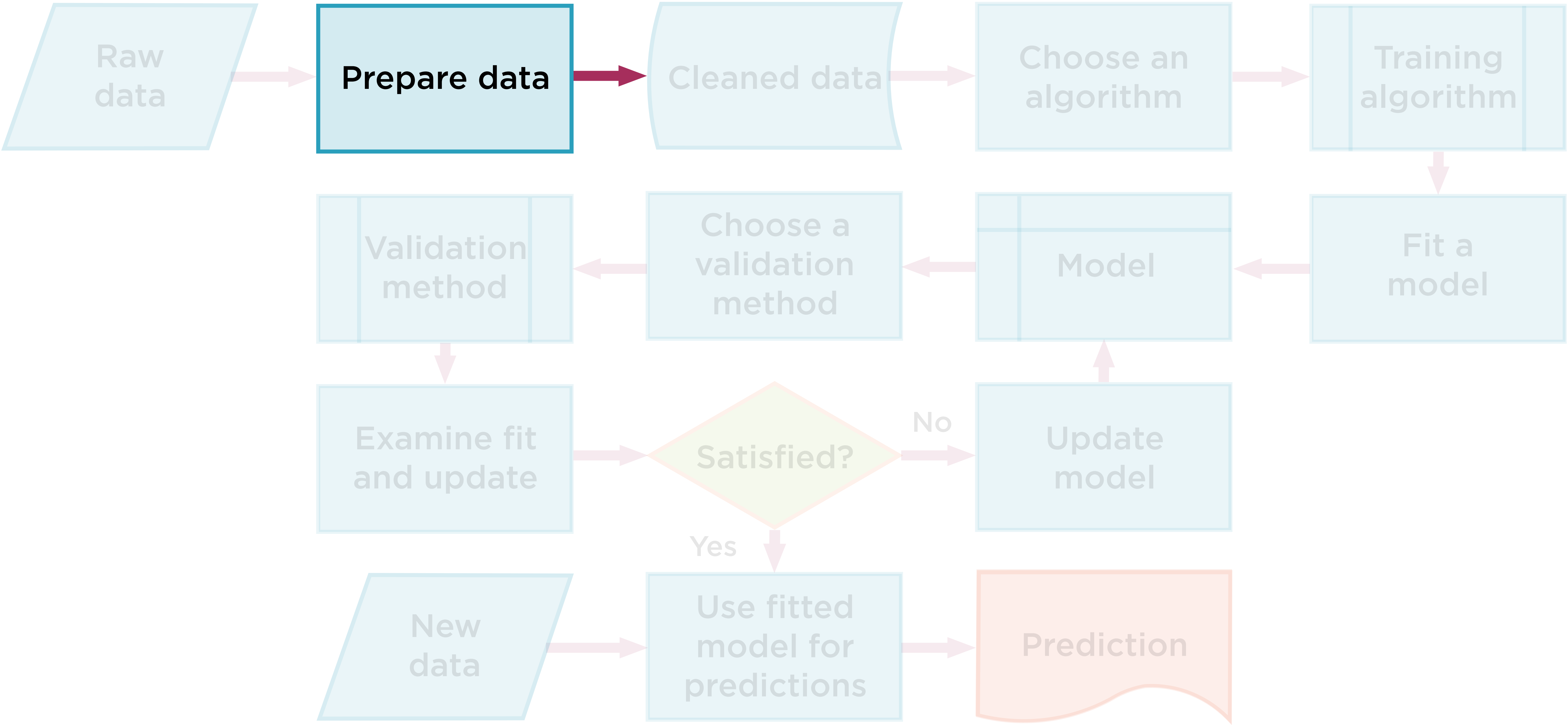
Basic Machine Learning Workflow



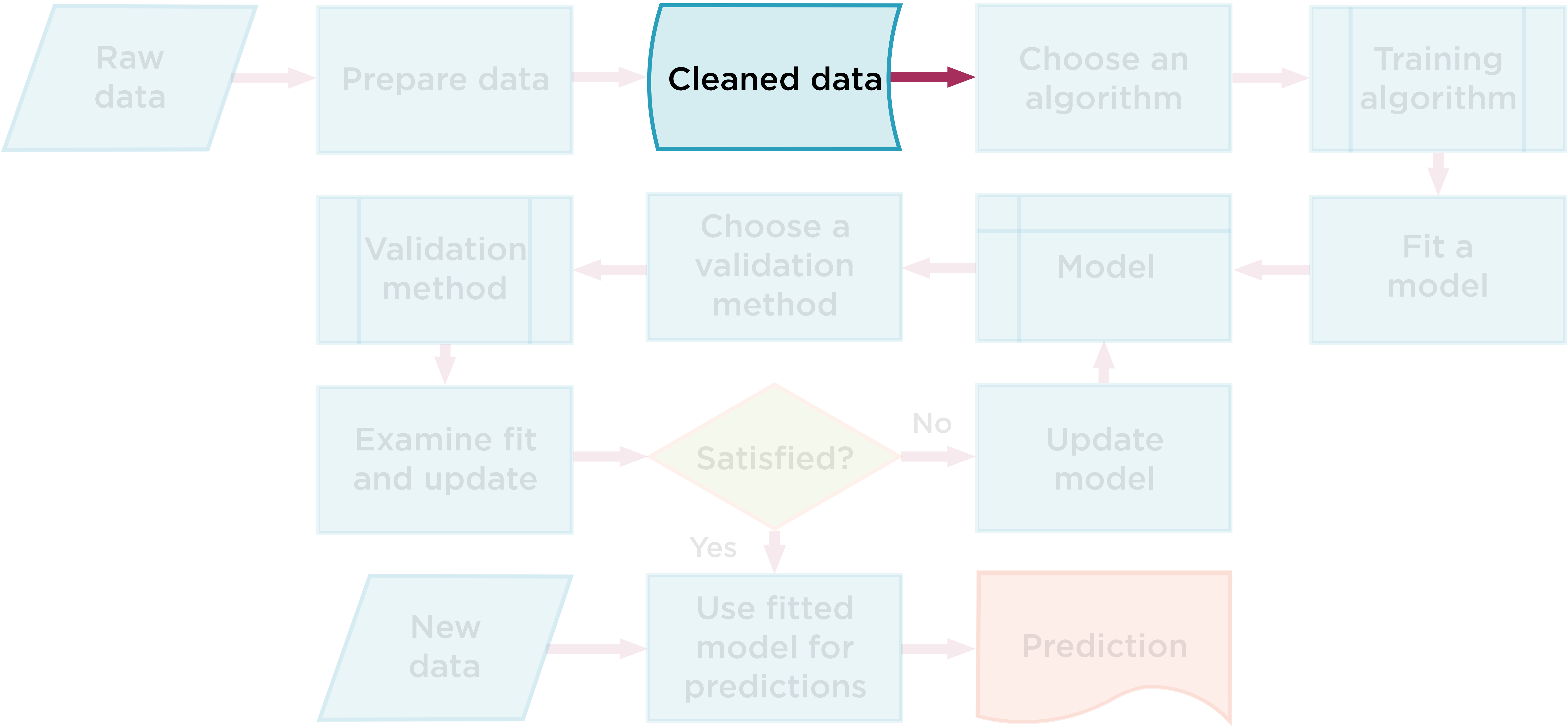
What Data Do You Have to Work With?



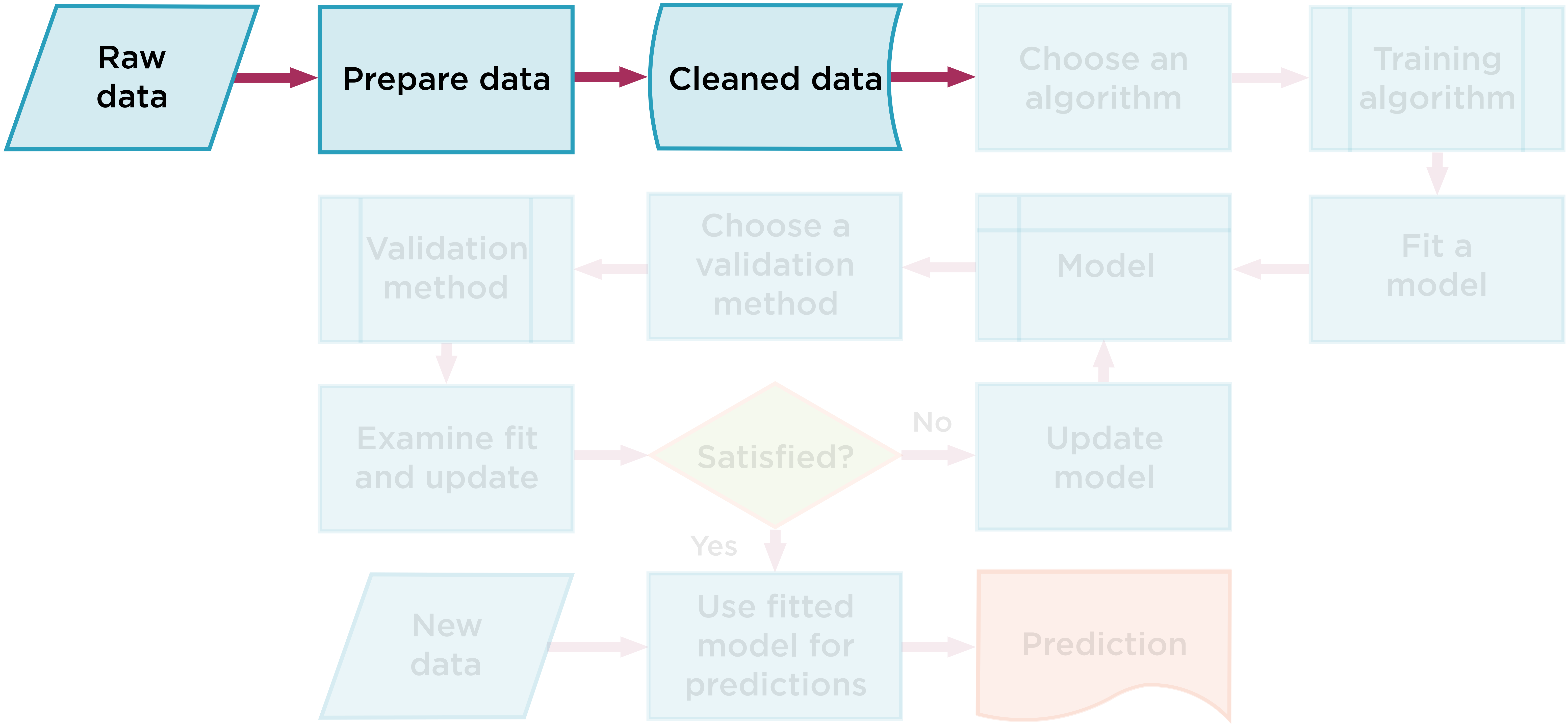
Load and Store Data



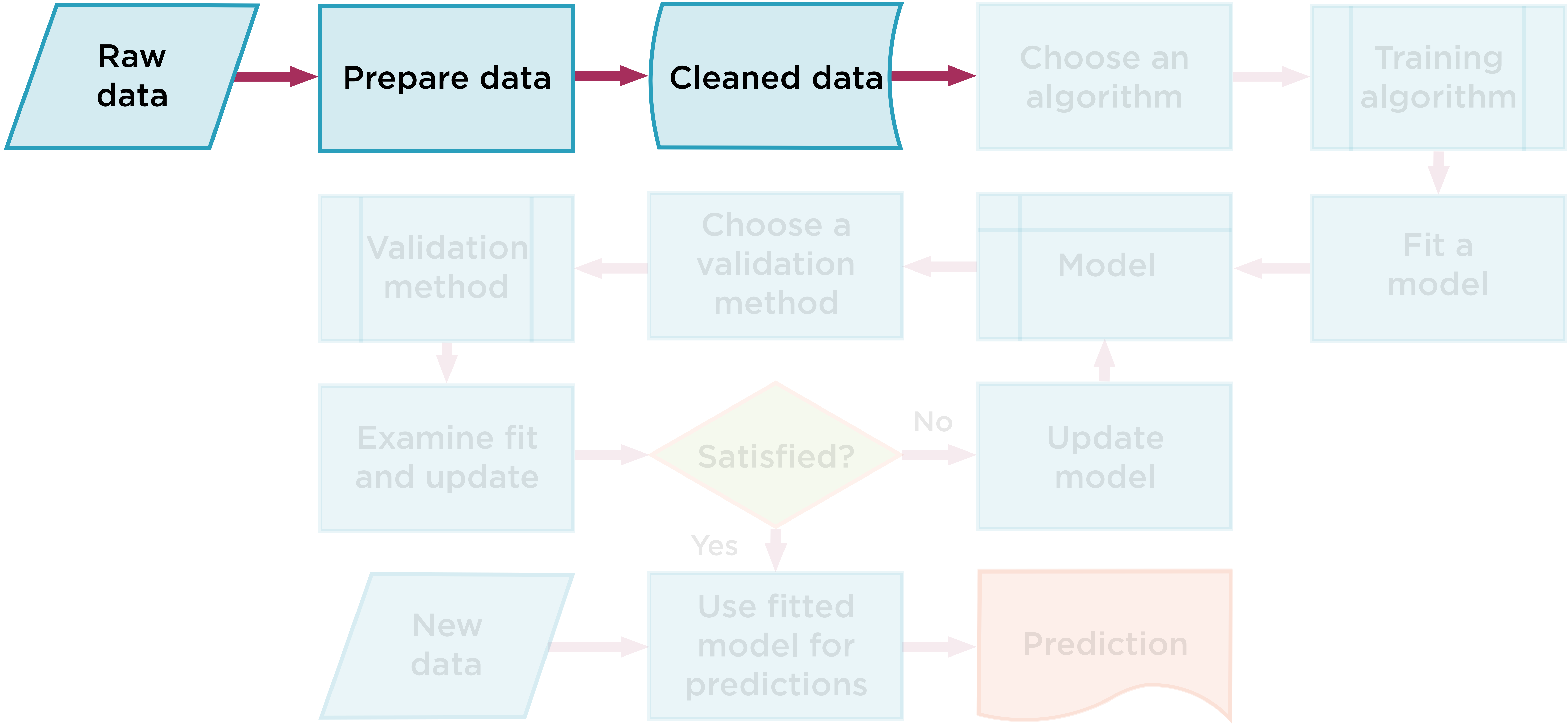
Data Preprocessing



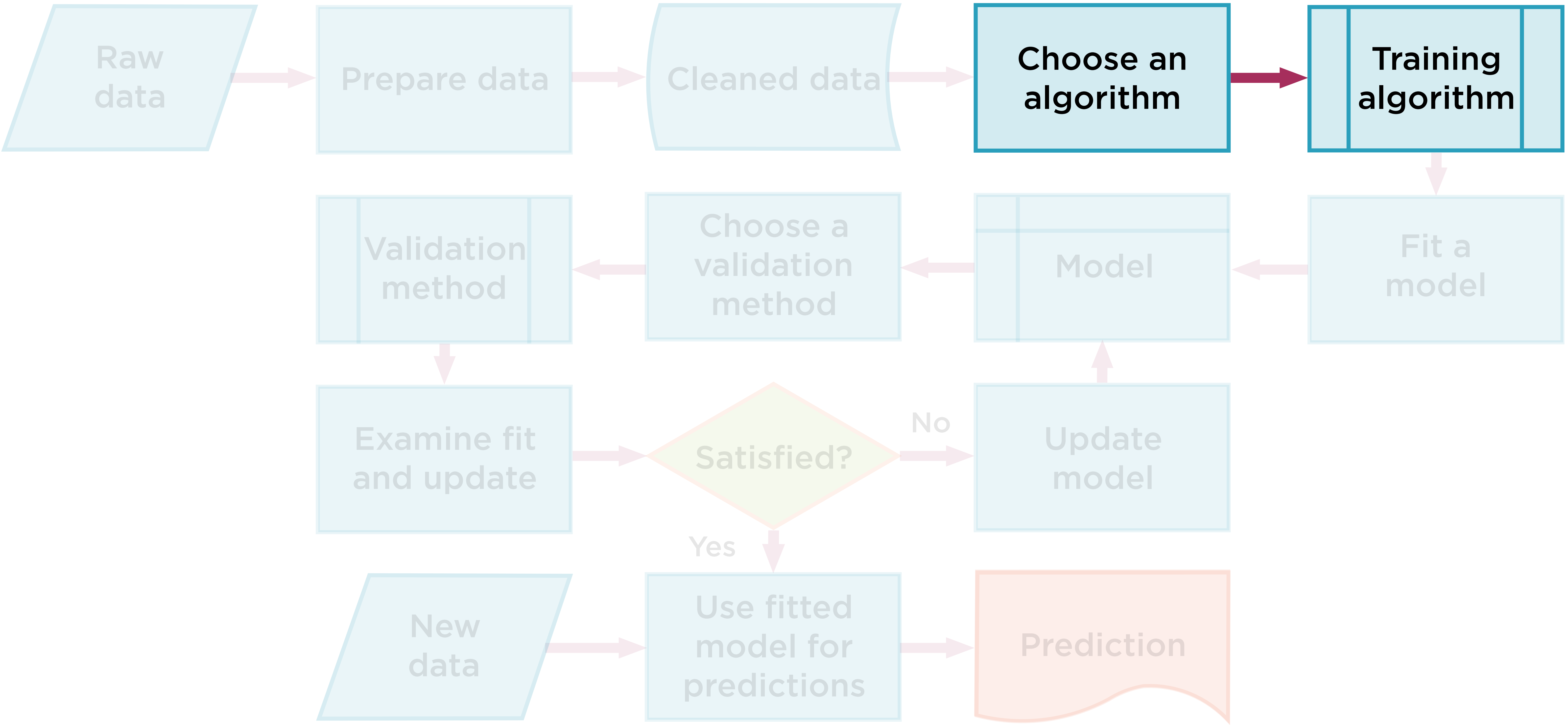
Selecting and Extracting Features



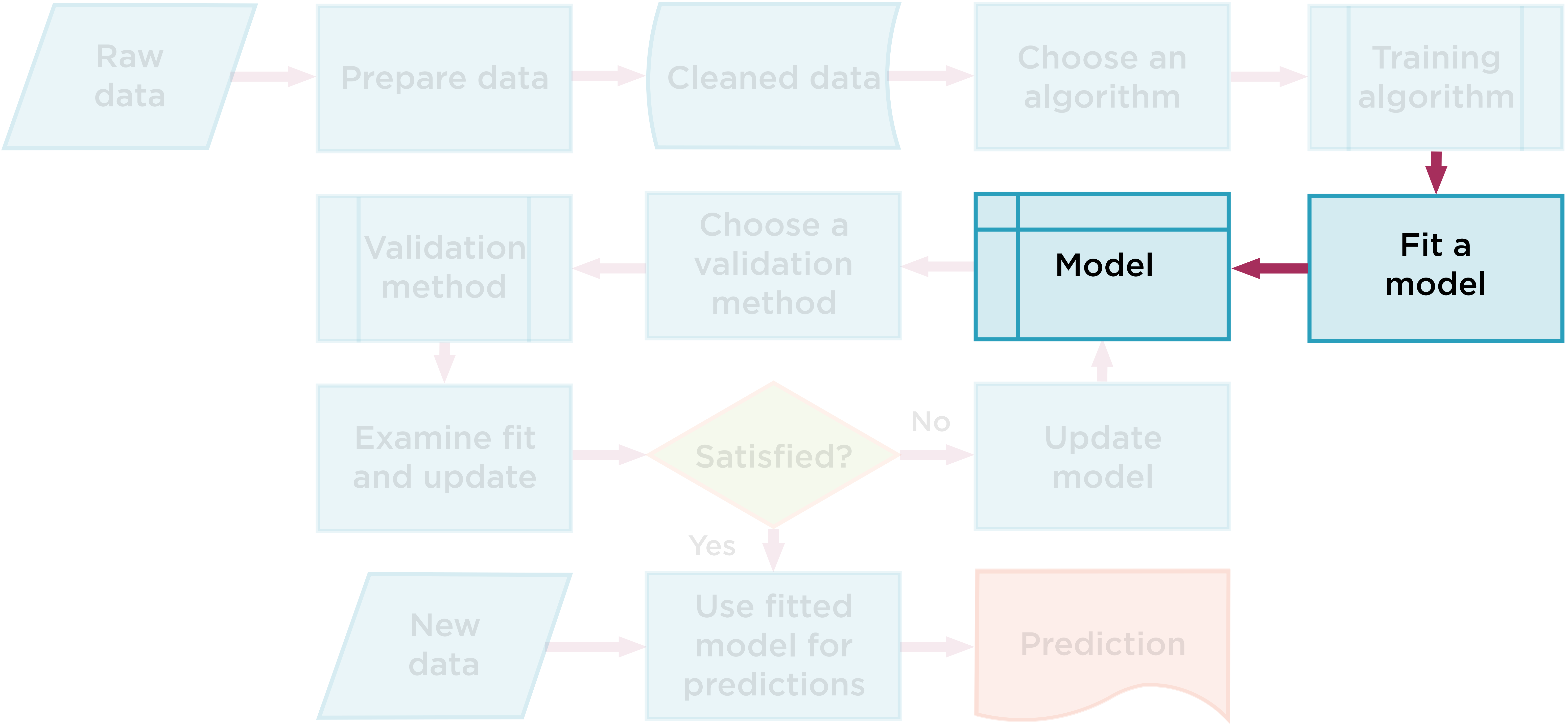
Critical and Time-consuming Steps



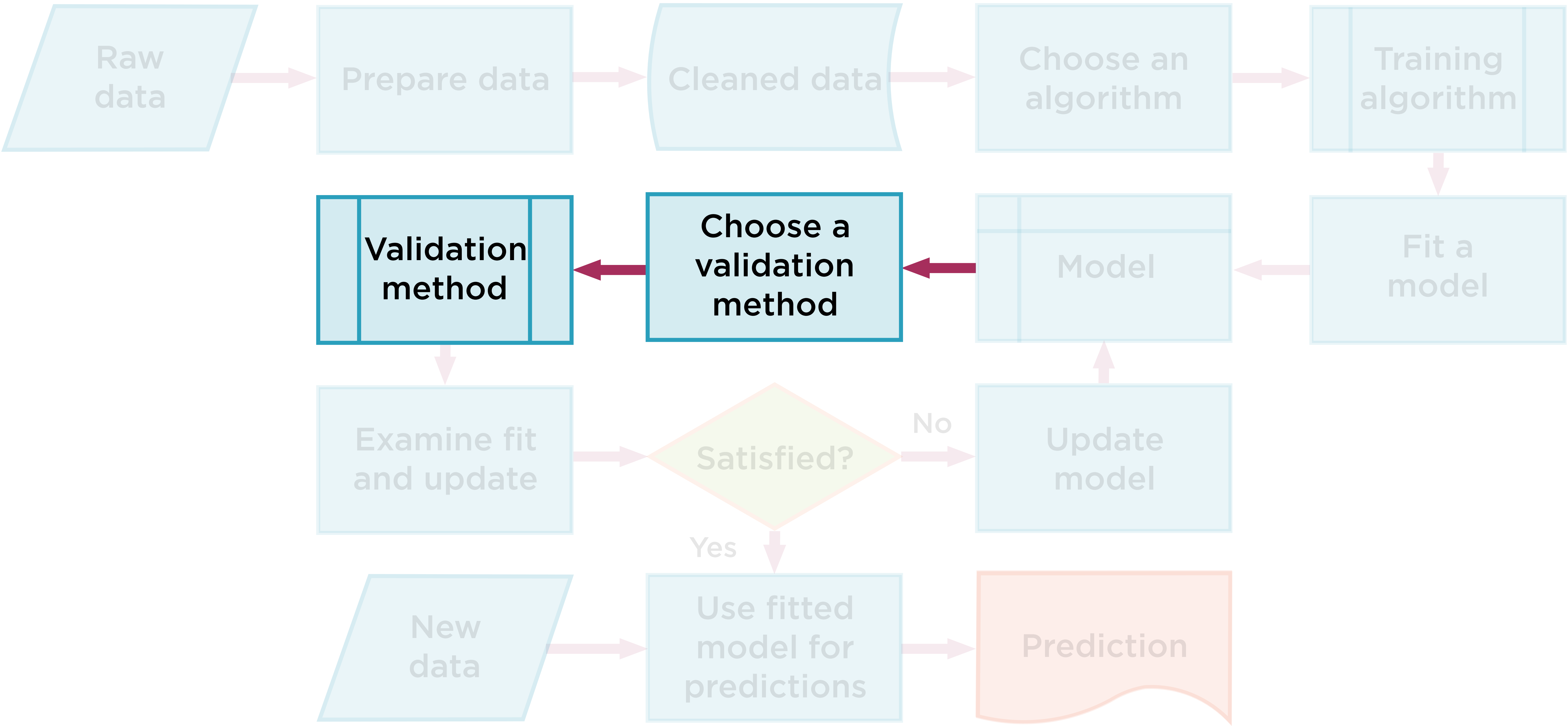
Decision Trees, Support Vector Machines?



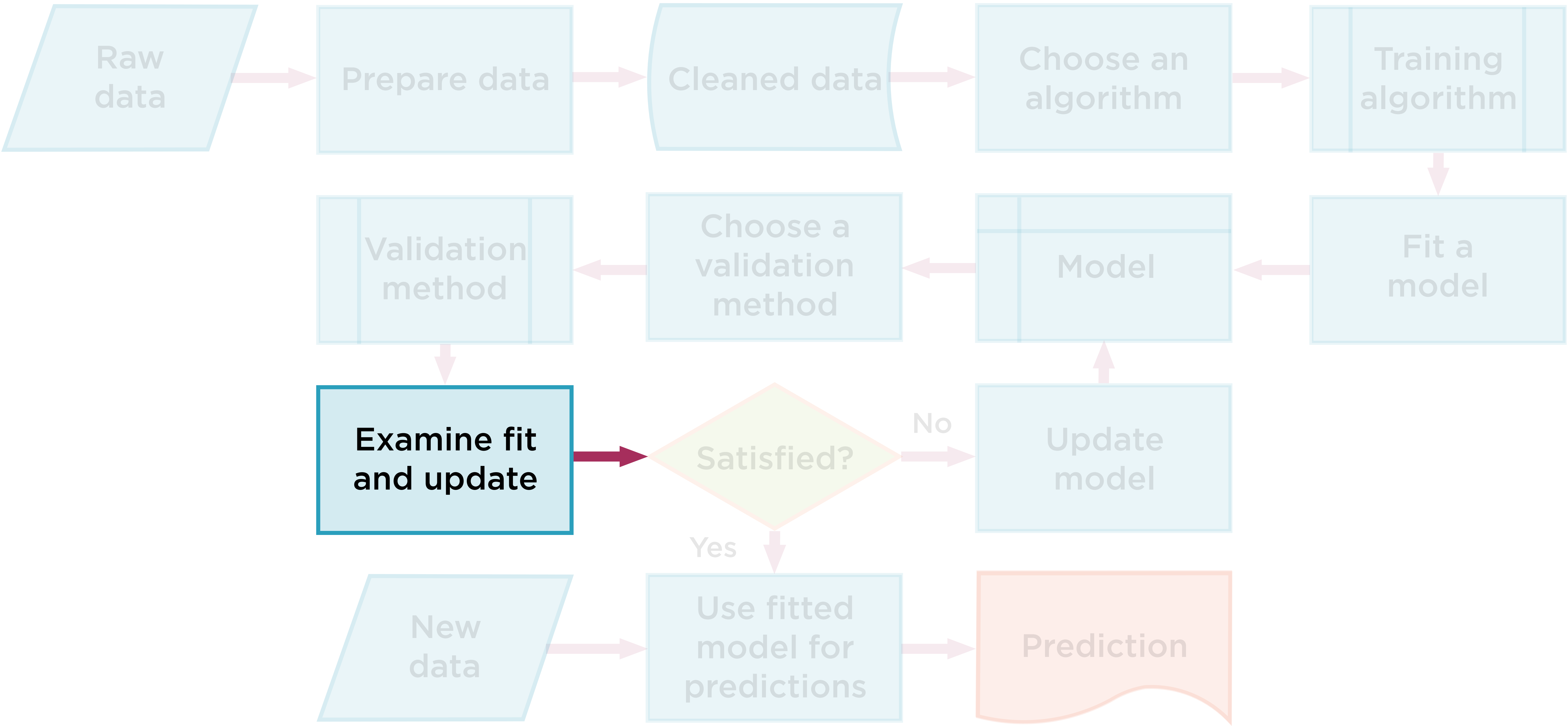
Training to Find Model Parameters



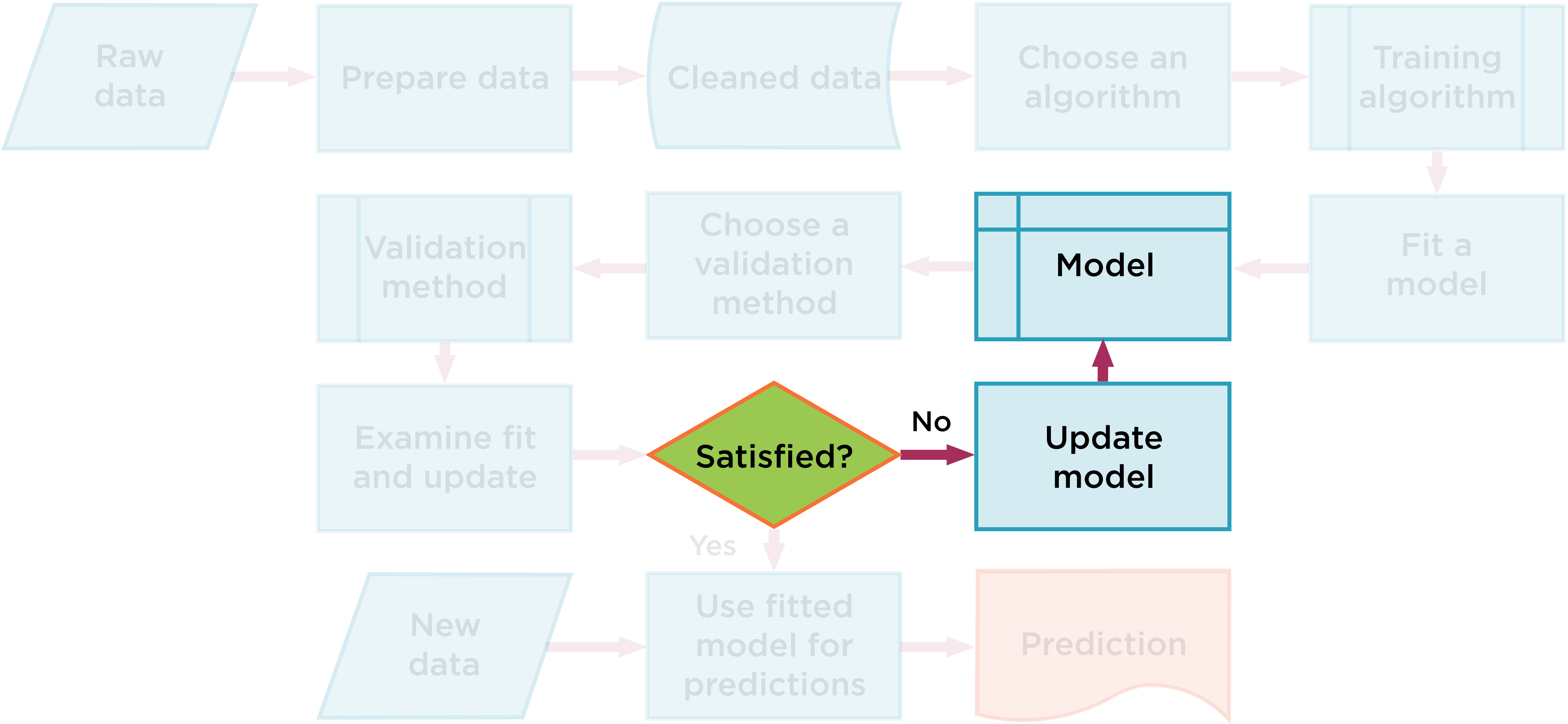
Evaluate the Model



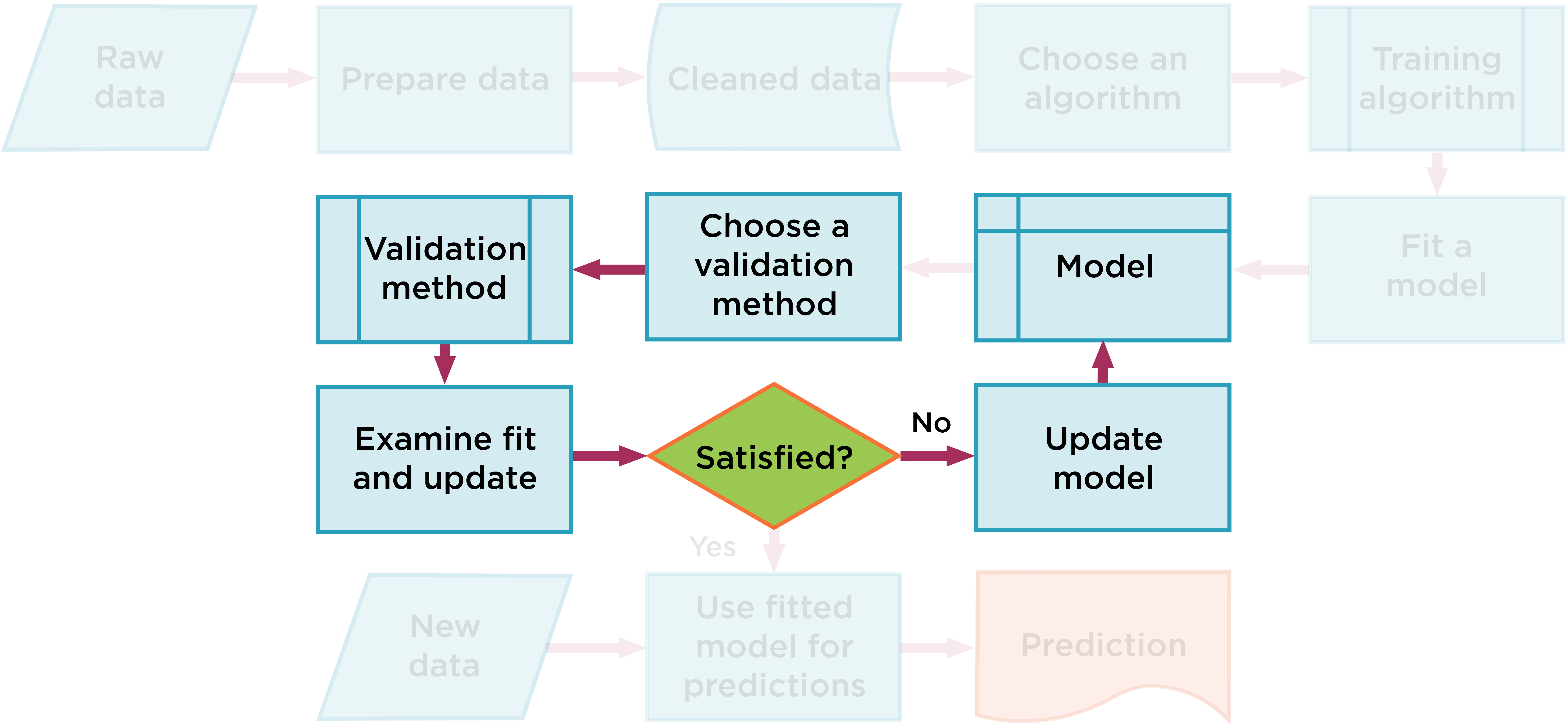
Score the Model



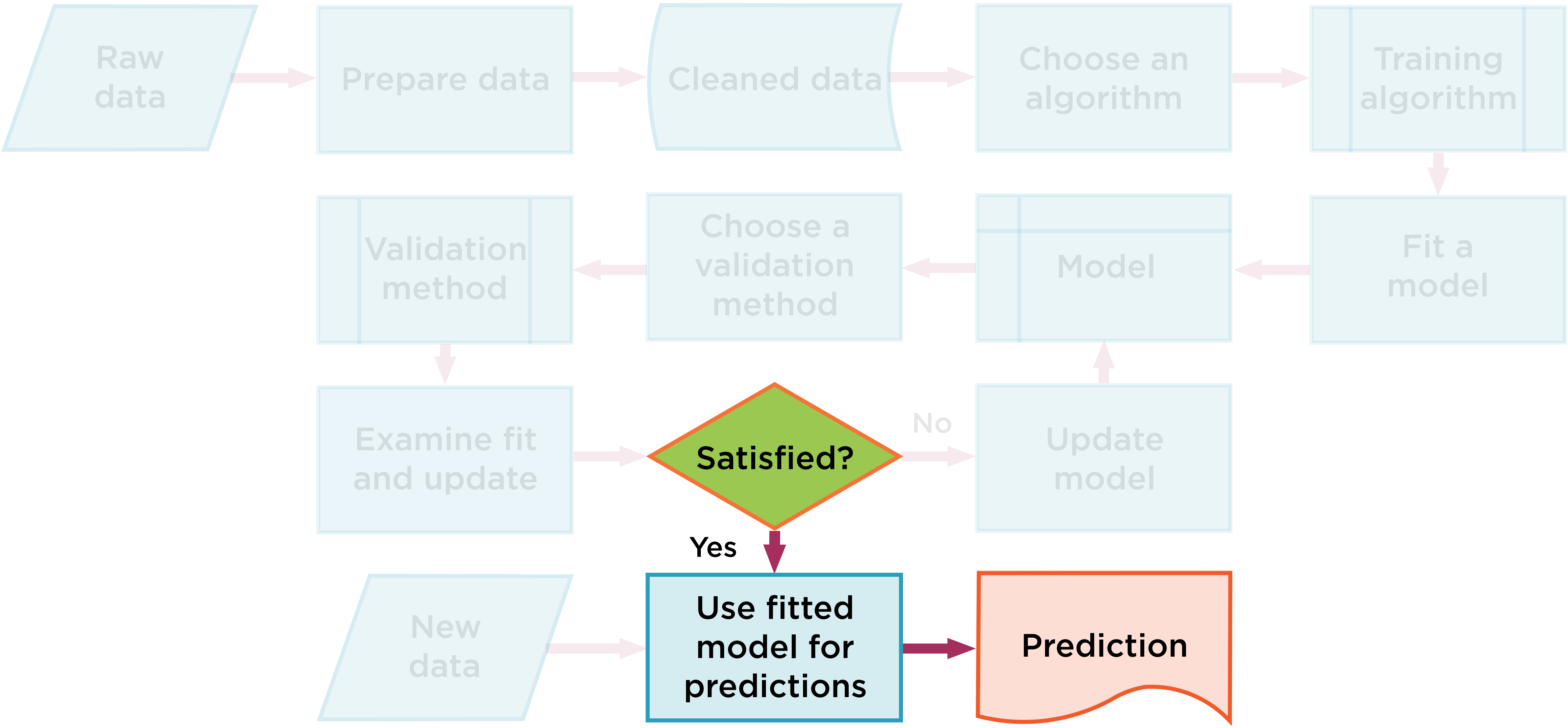
Different Algorithm, More Data, More Training?



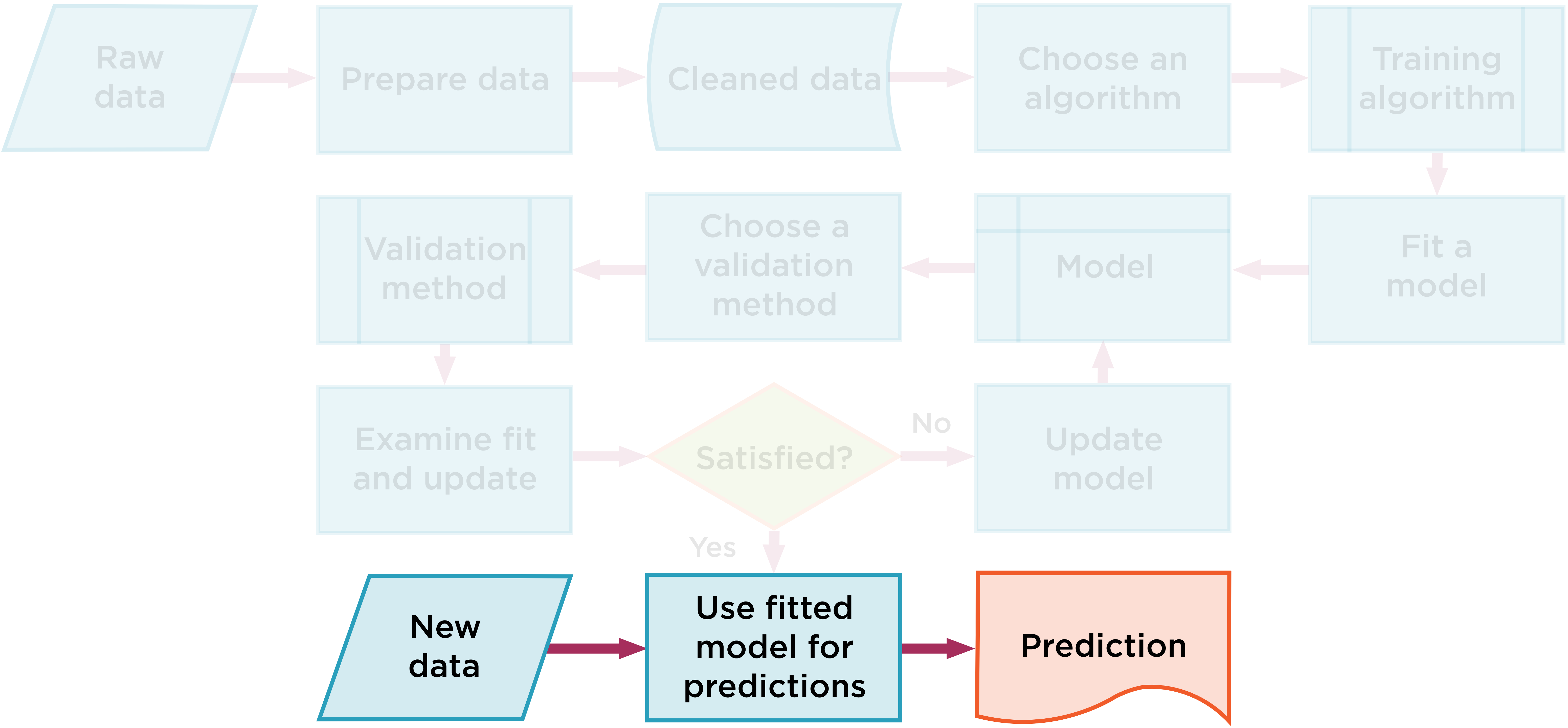
Iterate Till Model Finalized



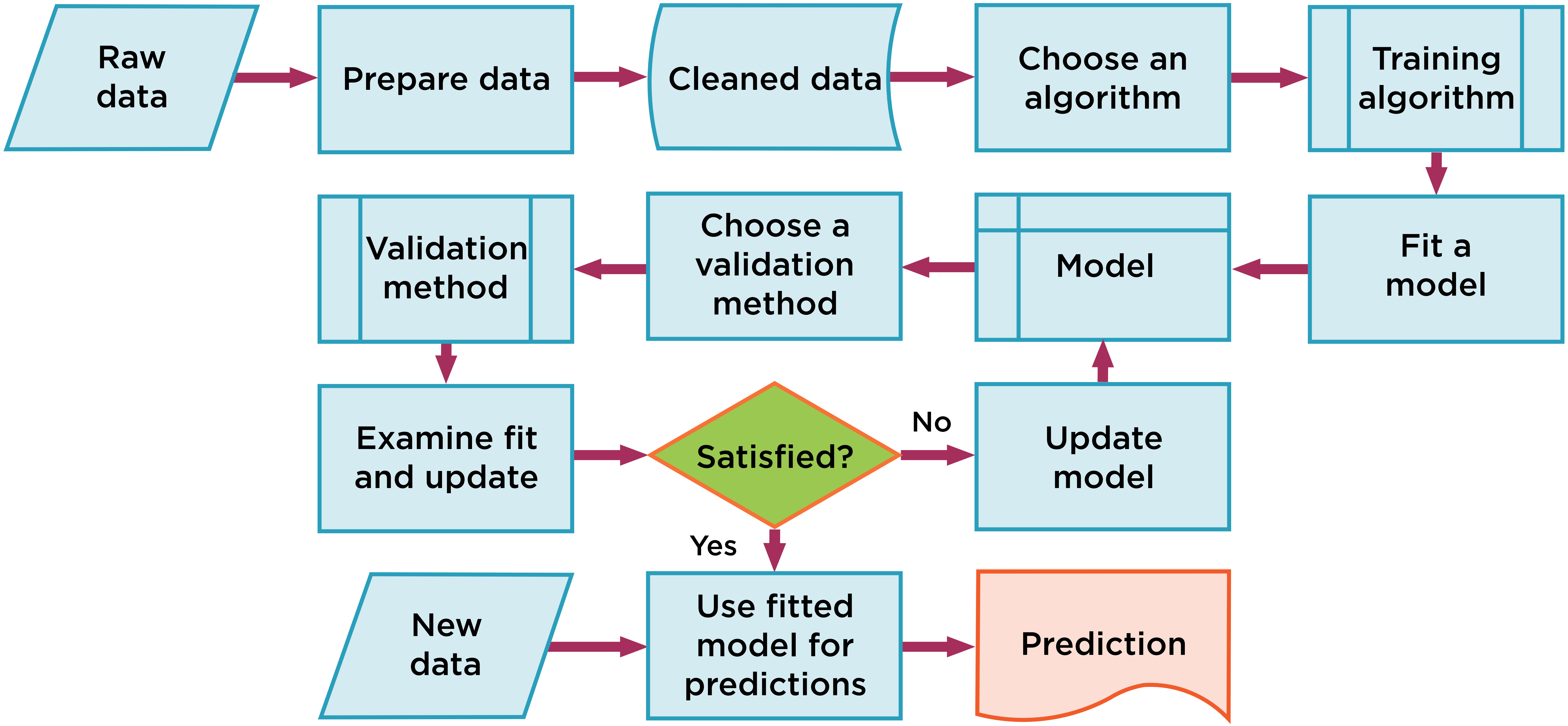
Model Used for Predictions



Retrained Using New Data



Basic Machine Learning Workflow

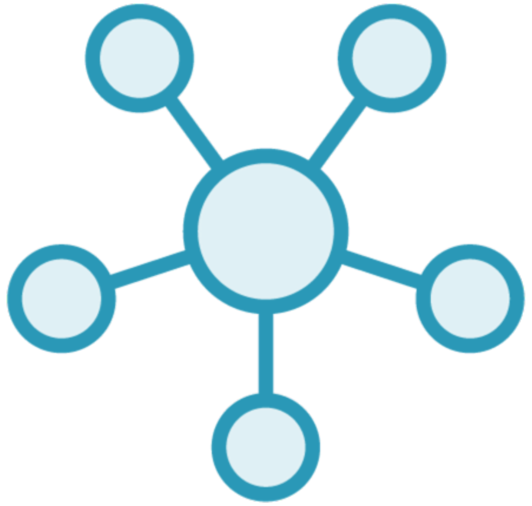


Choosing the Right Problem Based on Data

Broad Problem Categories

Classification

Regression



Clustering

Dimensionality reduction

Broad Problem Categories

Classify input data into categories

Regression

Clustering

Dimensionality reduction

Classification Use Cases

Predict categories

Email: spam or ham?

Stocks: Buy, sell or hold?

Images: Cat, dog or mouse?

Text: Positive, negative or neutral sentiment?

Broad Problem Categories

Classification

Regression

Clustering

Dimensionality reduction

Broad Problem Categories

Classification

Predict continuous numeric values

Clustering

Dimensionality reduction

Regression Use Cases

Given past stock data predict price tomorrow

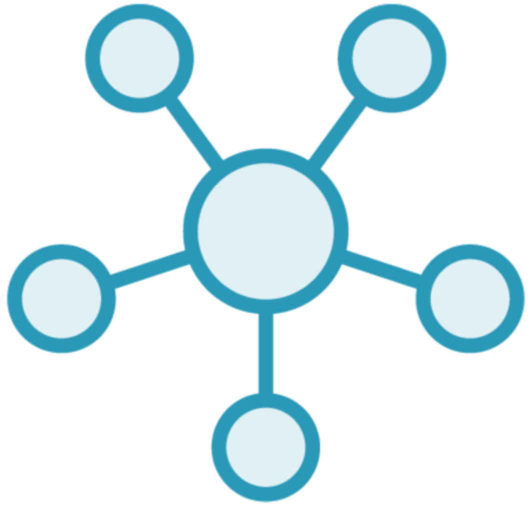
Given characteristics of a car predict mileage

Given location and attributes of a home predict price

Broad Problem Categories

Classification

Regression



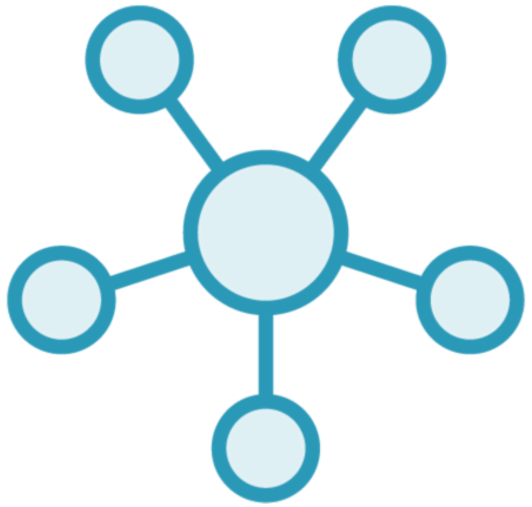
Clustering

Dimensionality reduction

Broad Problem Categories

Classification

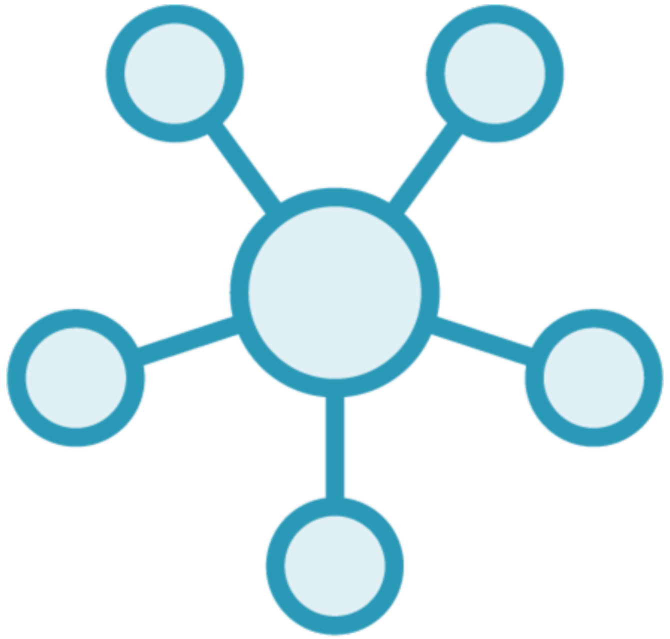
Regression



Discover patterns and groupings in data

Dimensionality reduction

Clustering Use Cases



Document discovery - find all documents related to homicide cases

Social media ad targeting - find all users who are interested in sports

Broad Problem Categories

Classification

Regression

Clustering

Dimensionality reduction

Broad Problem Categories

Classification

Regression

Clustering

Find latent or significant features in data

Dimensionality Reduction Use Cases

Find latent drivers of stock movements

Pre-process data to build more robust machine learning models

Improve performance of models in training

Supervised Learning

Classification

Regression

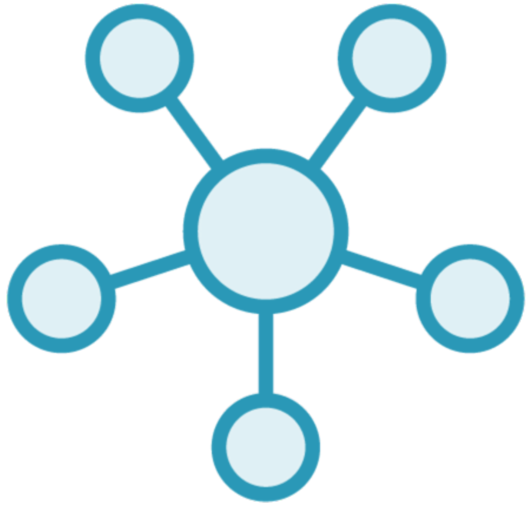
Clustering

Dimensionality reduction

Unsupervised Learning

Classification

Regression



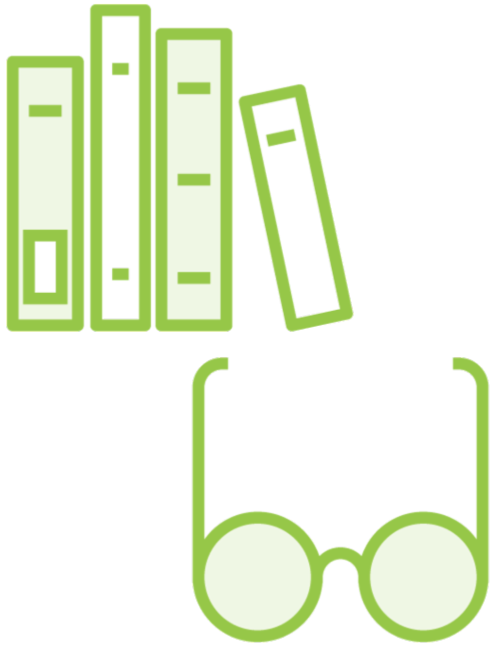
Clustering

Dimensionality reduction

Specialized Problem Categories

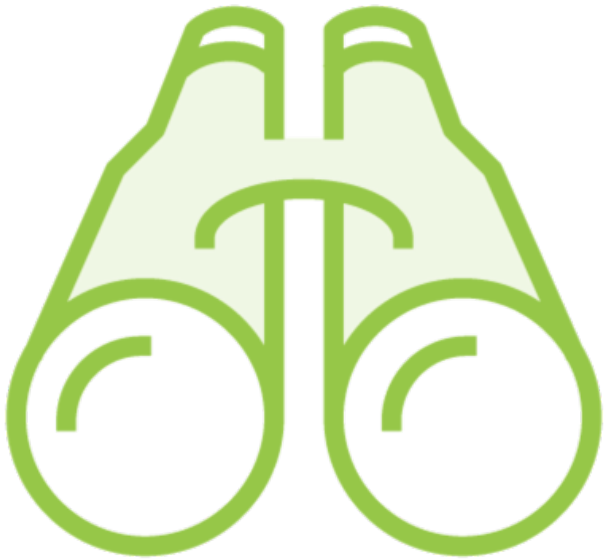
Recommendation Systems

Recommend products to users



Association Rules Detection

Detect transactions that occur together



Reinforcement Learning

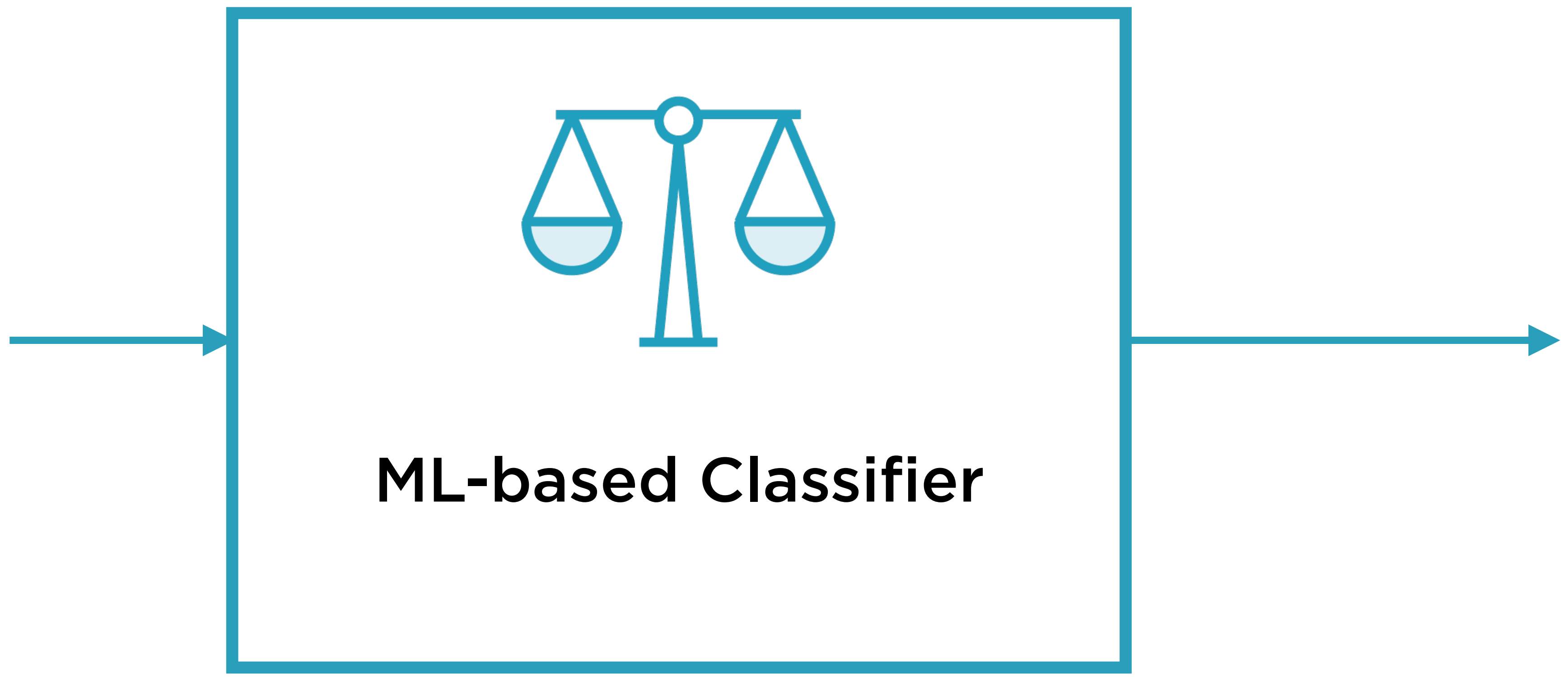
Train agent to navigate an uncertain environment

Broad Solution Categories

Use-case	Problem
Image data	Convolutional Neural Networks
Complex textual data	Recurrent Neural Networks
Sequential or time series data	Recurrent Neural Networks
Linear x-variables	Linear and logistic regression, PCA
Twisted data (S-curves, Swiss Rolls)	Manifold learning
Large numbers of x-variables	Decision trees

Supervised and Unsupervised Learning

Whales: Fish or Mammals?



ML-based Classifier

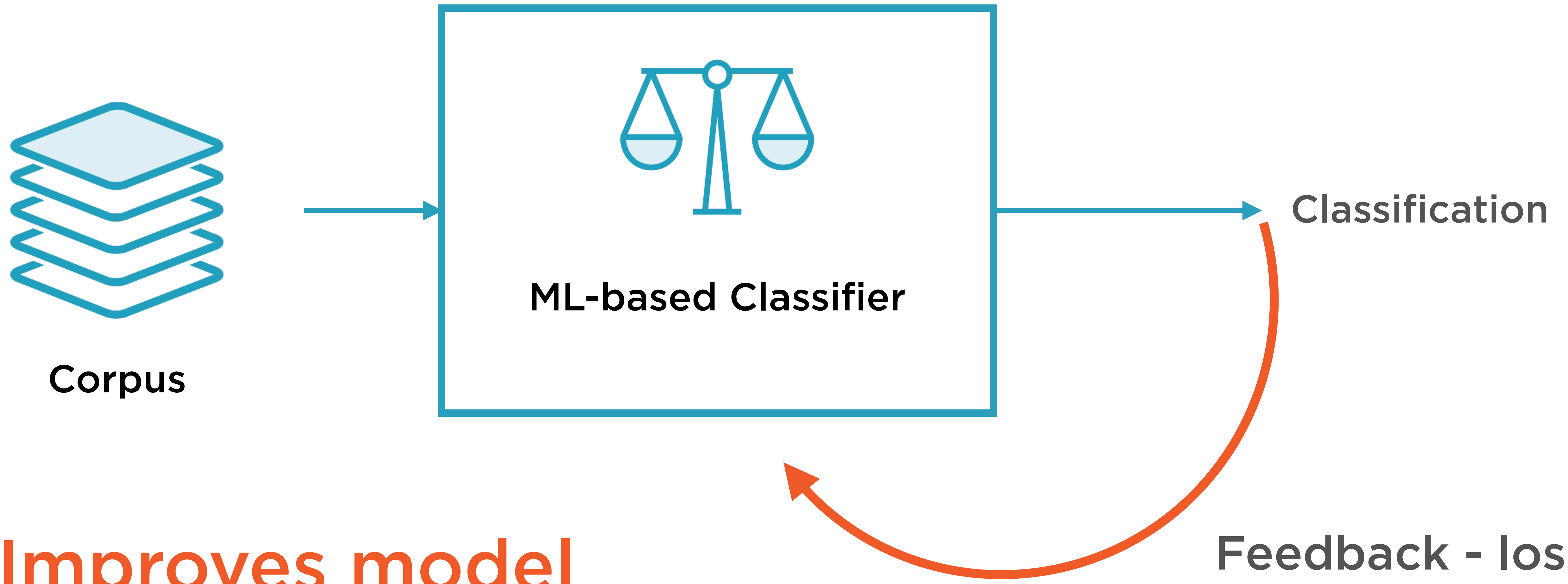
Training

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



Improves model parameters

Feedback - loss function or cost function

```
y = f(x)
```

Supervised Machine Learning

Most machine learning algorithms seek to “learn” the function f that links the features and the labels

```
y = Wx + b
```

$$f(x) = Wx + b$$

Linear regression specifies, up-front, that the function f is linear

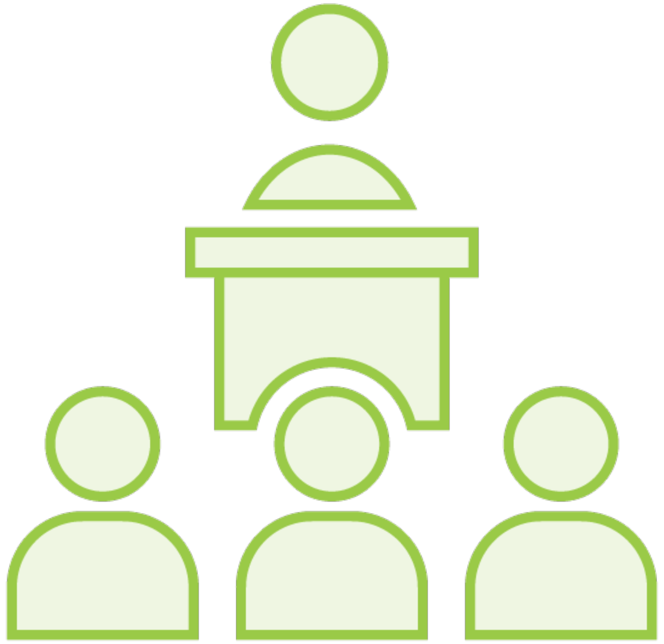
```
def doSomethingReallyComplicated(x1,x2…):
  …
  …
  …
  return complicatedResult
```

f(x) = doSomethingReallyComplicated(x)

ML algorithms such as neural network can “learn” (reverse-engineer) pretty much anything given the right training data

Unsupervised Learning learns patterns in data **without a labeled corpus**

Types of ML Algorithms



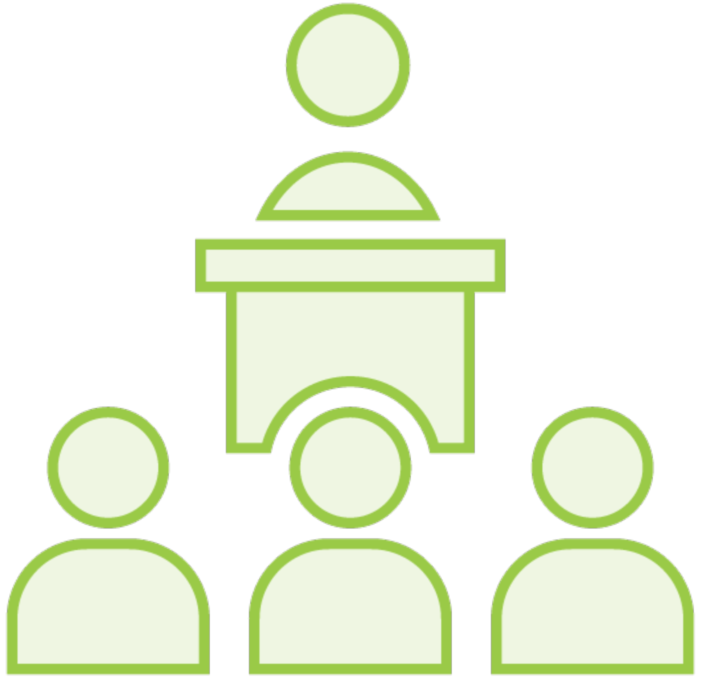
Supervised

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

Supervised Learning



Input variable x and output variable y

Learn the mapping function `y = f(x)`

Approximate the mapping function so for new values of x we can predict y

Use existing dataset to correct our mapping function approximation

Unsupervised Learning

Only have input data x – no output data

Model the underlying structure to learn more about data

Algorithms self discover the patterns and structure in the data

Unsupervised Learning Use-cases

ML Technique	Use-case
To make unlabelled data self-sufficient	Identify photos of a specific individual
Latent factor analysis	Find common drivers of 200 stocks
Clustering	Find relevant document in a corpus
Anomaly detection	Flag fraudulent credit card transactions
Quantization	Compress true color (24 bit) to 8 bit
Pre-training for supervised learning problems (classification, regression)	All of the above!

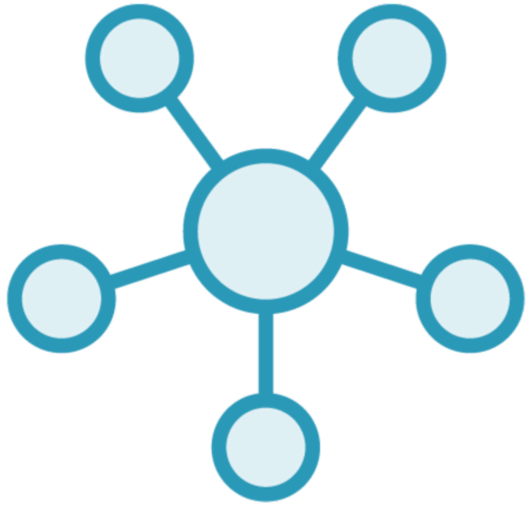
Unsupervised Learning Use-cases

What	How
To make unlabelled data self-sufficient	Autoencoder
Latent factor analysis	Autoencoder
Clustering	Clustering
Anomaly detection	Autoencoder
Quantization	Clustering
Pre-training for supervised learning problems (classification, regression)	All of the above!

Unsupervised Learning

Classification

Regression



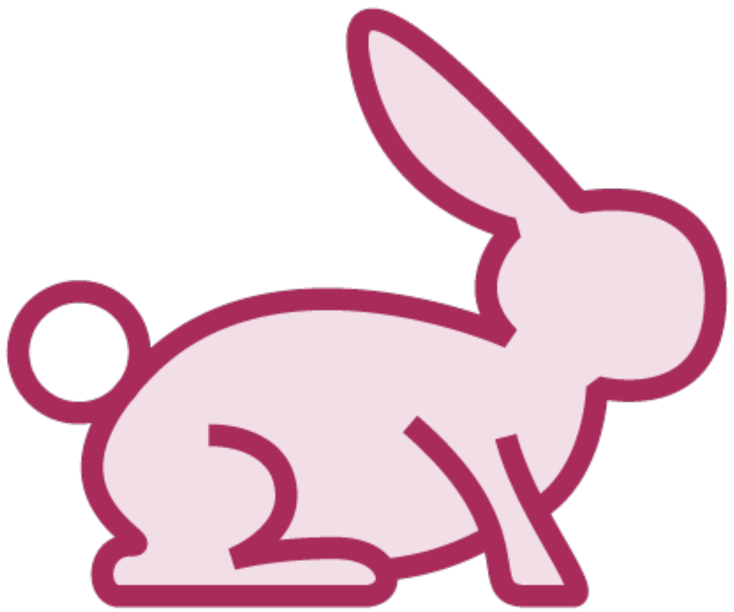
Clustering

Dimensionality reduction

“What lies behind us and what lies ahead of us are tiny matters compared to what lives within us”

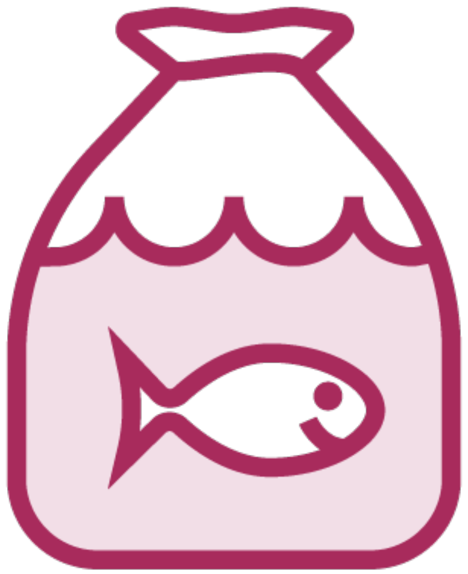
Henry David Thoreau

Whales: Fish or Mammals?



Mammals

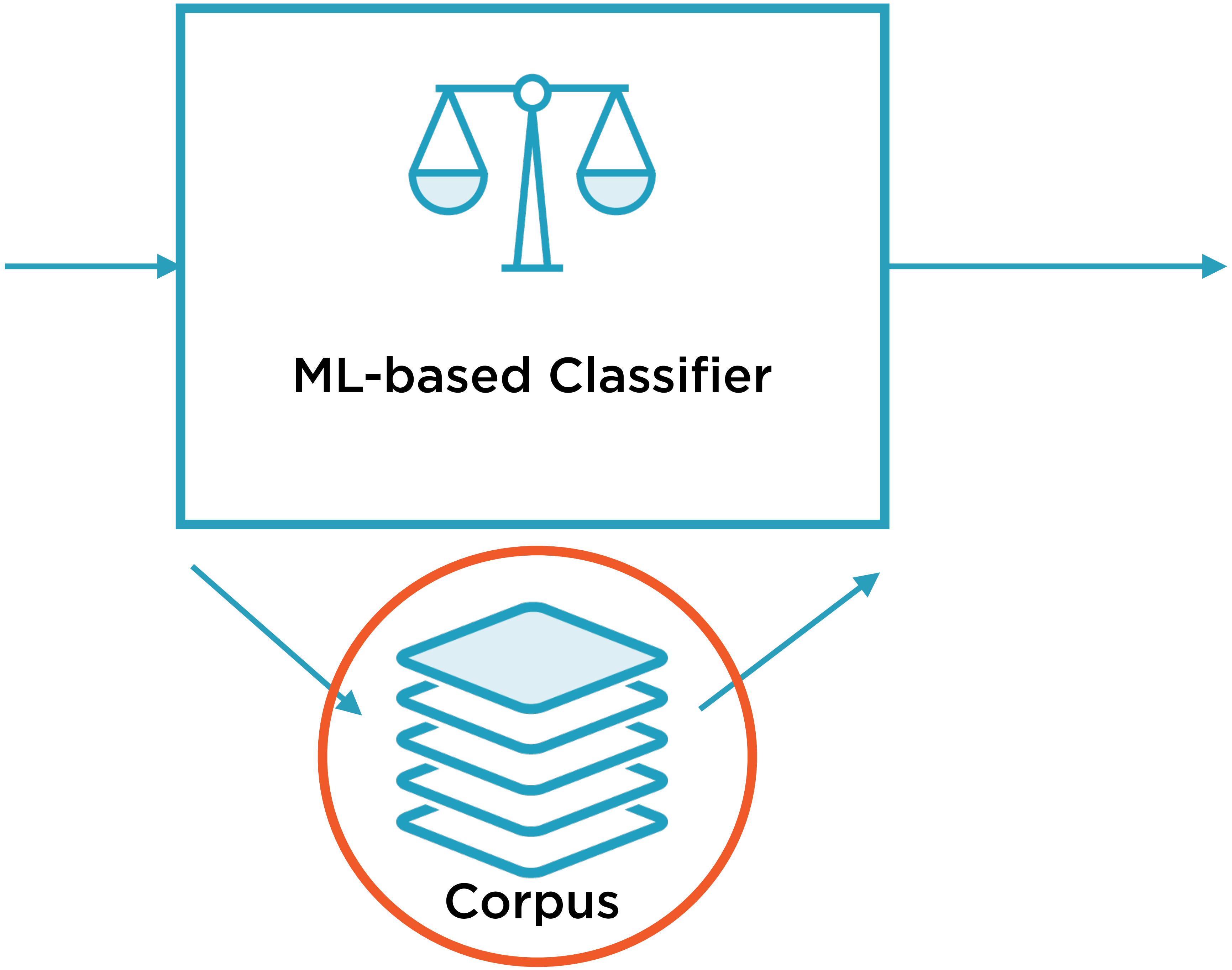
Members of the infraorder *Cetacea*



Fish

Look like fish, swim like fish, move with fish

No Labeled Training Data



Transfer Learning

Transfer Learning

The practice of re-using a trained neural network that solves a problem similar to yours, usually leaving the network architecture unchanged and re-using some or all of the model weights.

Avoid designing NN architecture from scratch

Transfer Learning

The practice of re-using a trained neural network that solves a problem similar to yours, usually leaving the network architecture unchanged and re-using some or all of the model weights.

Transfer Learning

The practice of re-using a trained neural network that solves a problem similar to yours, usually leaving the network architecture unchanged and re-using some or all of the model weights.

Only makes sense for common, widely studied use-cases

Transfer Learning

The practice of re-using a trained neural network that solves a problem similar to yours, usually leaving the network architecture unchanged and re-using some or all of the model weights.

In which basic problem structure stays same, but details vary

Transfer Learning

The practice of re-using a trained neural network that solves a problem similar to yours, usually leaving the network architecture unchanged and re-using some or all of the model weights.

Image recognition, language translation are classic examples

Transfer Learning

The practice of re-using a trained neural network that solves a problem similar to yours, usually leaving the network architecture unchanged and re-using some or all of the model weights.

Often the hardest part - allows us to "stand on the shoulders of giants"

Transfer Learning

The practice of re-using a trained neural network that solves a problem similar to yours, usually leaving the network architecture unchanged and re-using some or all of the model weights.

Re-train from scratch, fine-tune model weights, use entirely as-is

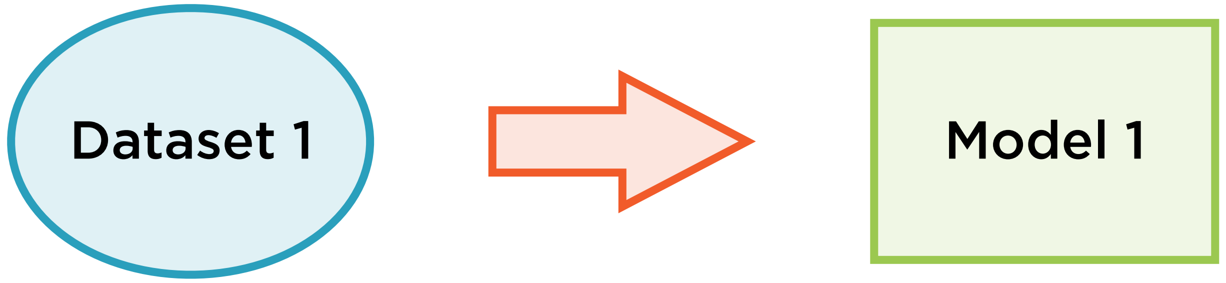
Transfer Learning

The practice of re-using a trained neural network that solves a problem similar to yours, usually leaving the network architecture unchanged and re-using some or all of the model weights.

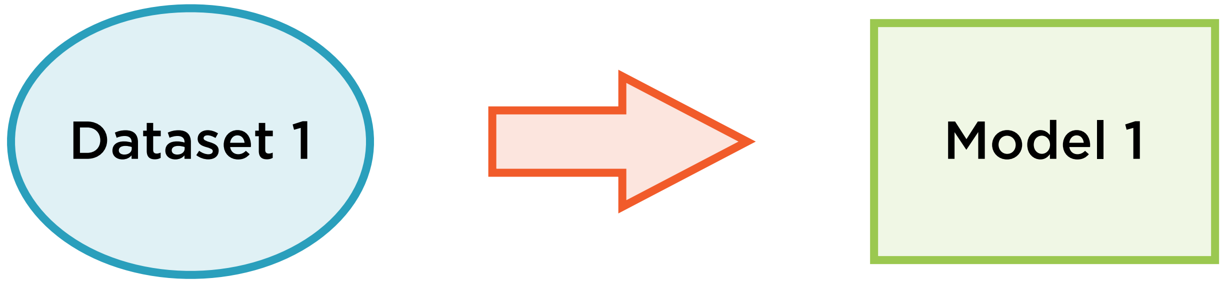
Several choices based on size and similarity of datasets

Cold-start ML vs. Transfer Learning

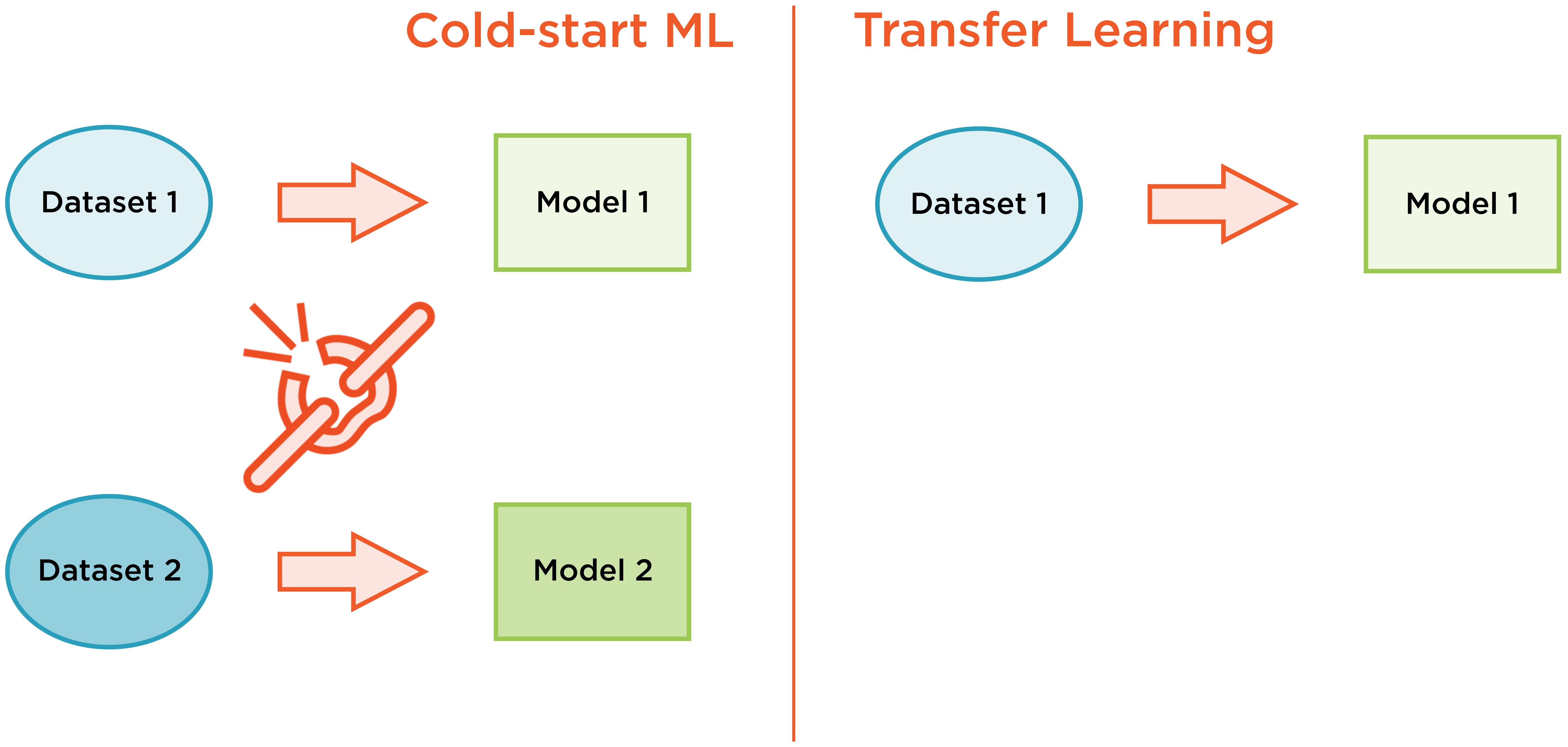
Cold-start ML



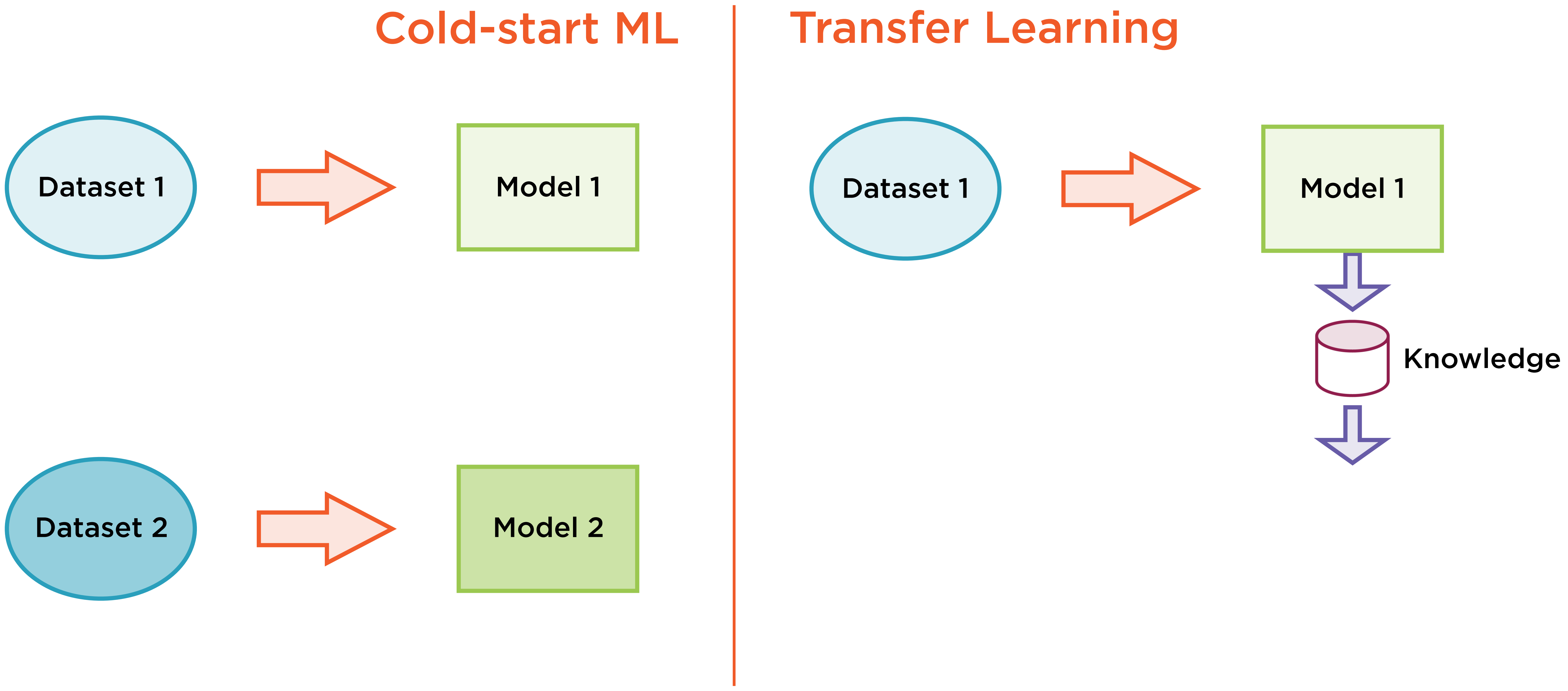
Transfer Learning



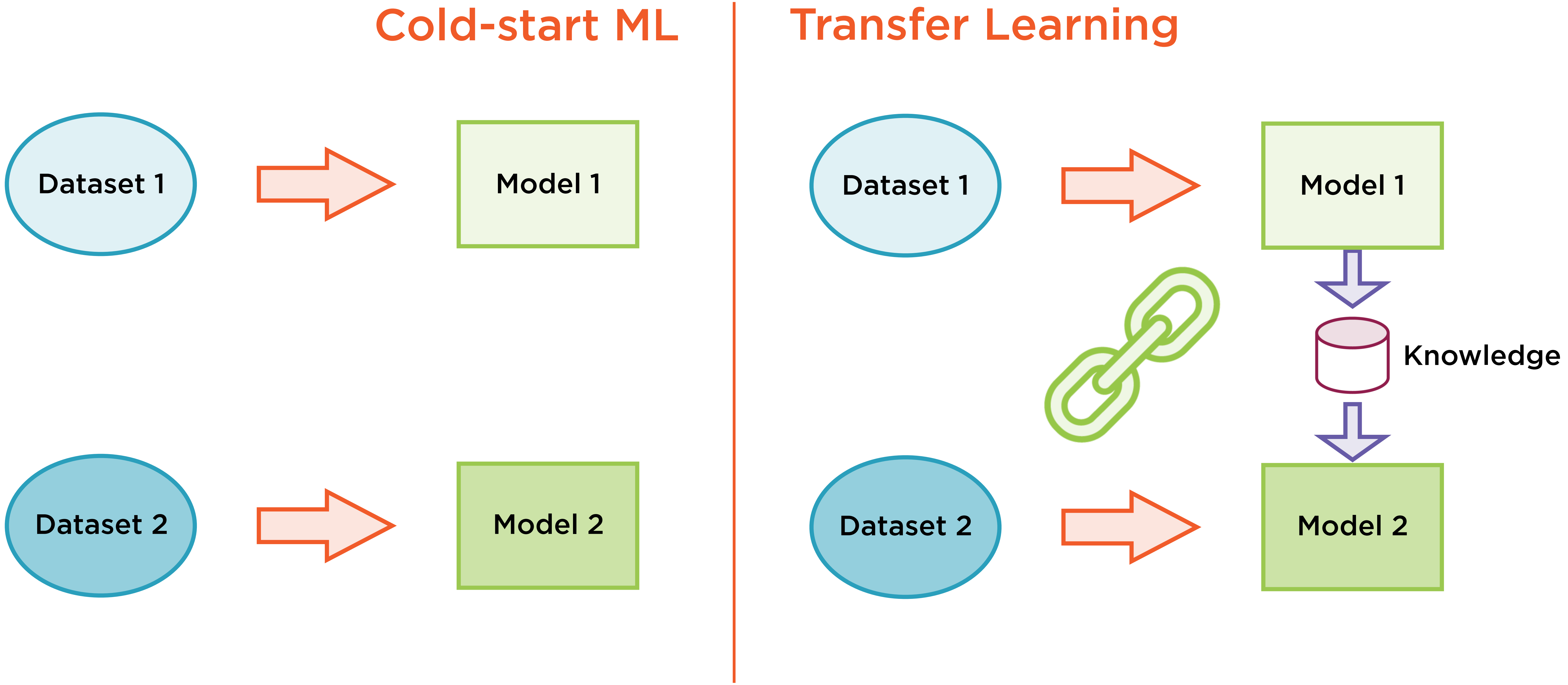
Cold-start ML vs. Transfer Learning



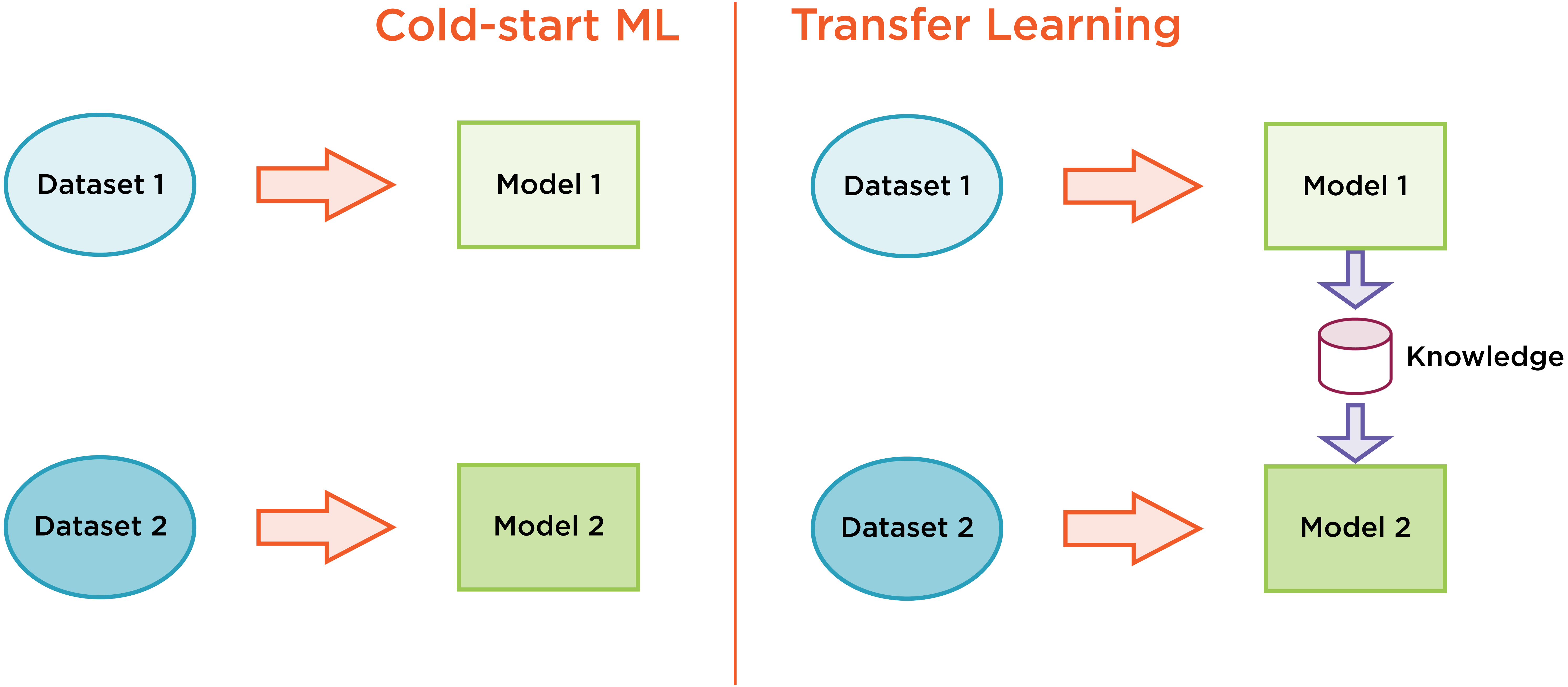
Cold-start ML vs. Transfer Learning



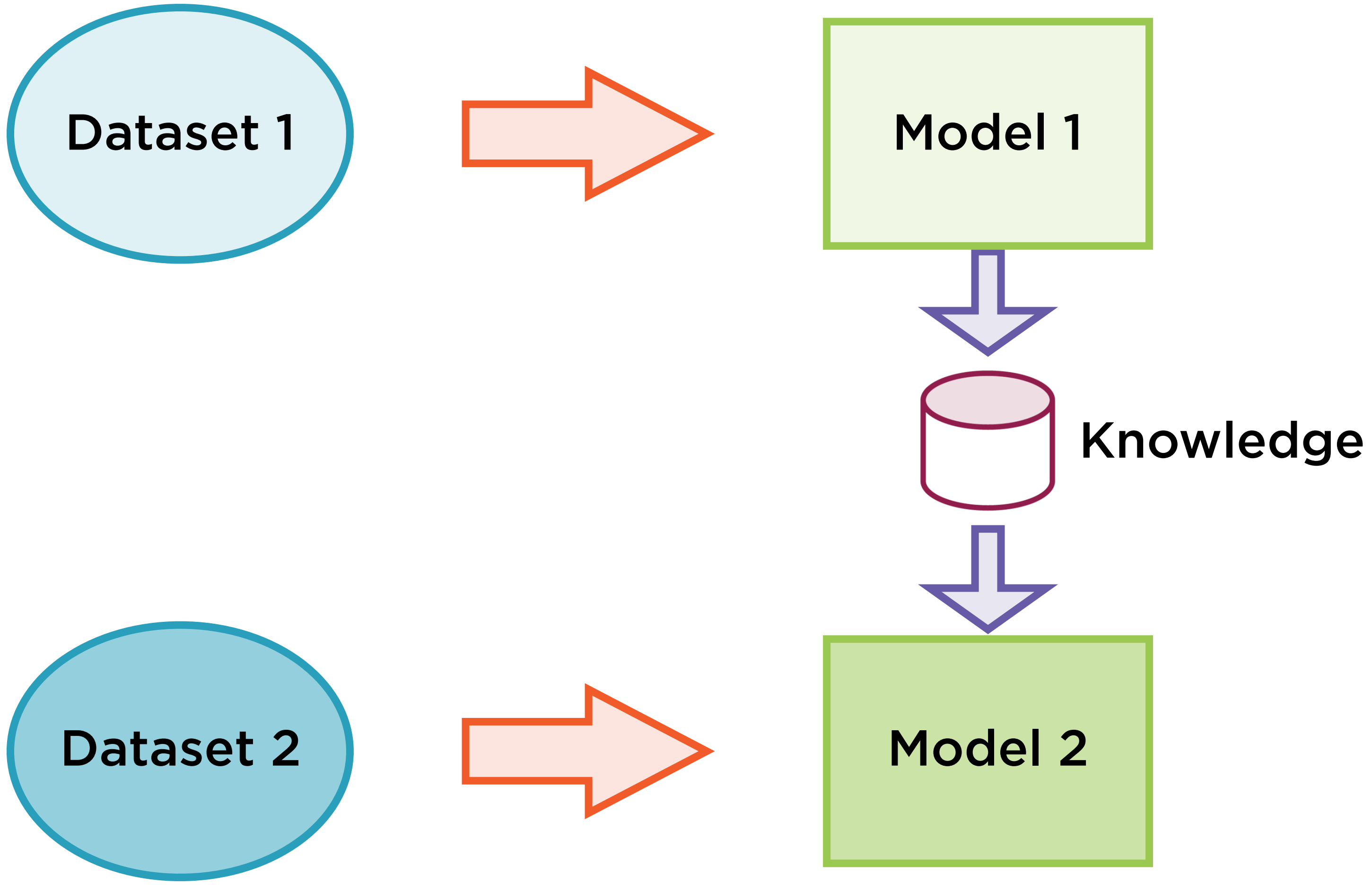
Cold-start ML vs. Transfer Learning



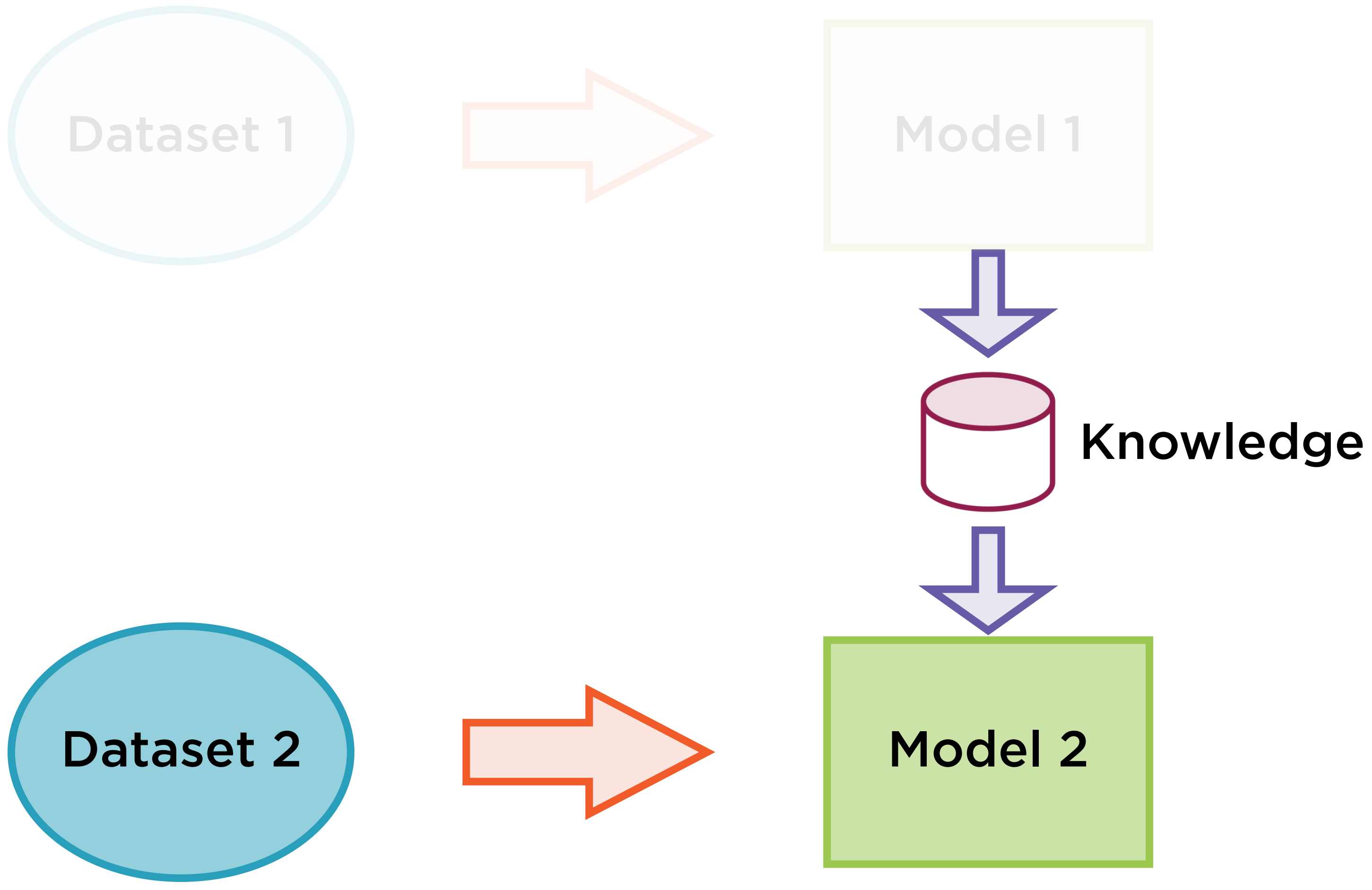
Cold-start ML vs. Transfer Learning



Transfer Learning



Transfer Learning



Transferred knowledge is especially useful when the new dataset is small and not sufficient to train a model from scratch

Warm-start ML

Use information gained from previous training runs to identify smarter starting points for the next training run.

Need not apply only to neural networks - other models support these as well

Warm-start ML

Use information gained from previous training runs to identify smarter starting points for the next training run.

Warm-start ML

Use information gained from previous training runs to identify smarter starting points for the next training run.

Add individual learners to an ensemble model

Warm-start ML

Use information gained from previous training runs to identify smarter starting points for the next training run.

Retain learnings from previous set of learners

Popular Machine Learning Frameworks

scikit-learn

Easy-to-use, very comprehensive and efficient Python library for traditional ML models

Attractions of scikit-learn

Easy-to-use

Comprehensive

Efficient

Attractions of scikit-learn

Easy-to-use

Comprehensive

Efficient

Ease of Use

Estimator API for consistent interface

Create a model object

Fit to training data

Predict for new data

Pipelines for complex operations

Attractions of scikit-learn

Easy-to-use

Comprehensive

Efficient

Completeness

All common families of ML models

Cross-validation

Feature extraction and selection

Data pre-processing

Data generation

- Swiss rolls, S-curves

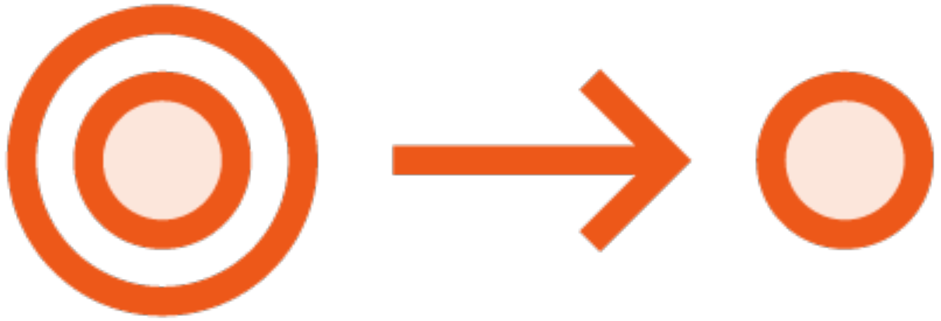
Attractions of scikit-learn

Easy-to-use

Comprehensive

Efficient

Efficiency



Highly optimized implementations

Built on SciPy, hence scikit prefix

Inter-operates with

- NumPy
- SciPy
- Matplotlib
- Pandas

PyTorch

A deep learning framework for fast, flexible experimentation.

https://pytorch.org/

TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. A comprehensive, flexible ecosystem of tools, libraries and community resources to easily build and deploy ML powered applications.

https://tensorflow.org/

Keras

A high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. However, multi-backend Keras is superseded by tf.keras.

https://keras.io/

Other Popular ML Frameworks

Apache MXNet	Microsoft CNTK
XGBoost	Theano

TensorFlow vs. PyTorch

TensorFlow	PyTorch
Originally developed at Google by the Google Brain team	Originally developed by AI researchers at Facebook
First released in November 2015	First released in October 2016
Tensors as fundamental data structures for computation	Tensors as fundamental data structures for computation
CUDA support for GPUs	CUDA support for GPUs

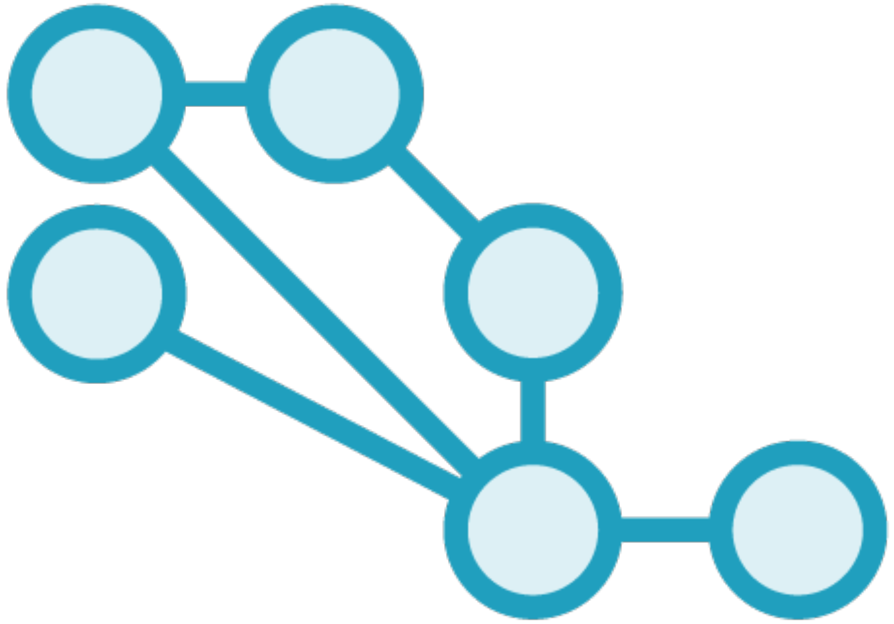
TensorFlow vs. PyTorch

TensorFlow	PyTorch
Computation graph is static	Computation graph is dynamic
Must be defined before being run	Can be defined and run as you go
tf.Session for separation from Python	Tightly integrated with Python

TensorFlow vs. PyTorch

TensorFlow	PyTorch
Debugging via tfdbg	Debugging with PyCharm, pdb
Visualization using built-in TensorBoard	Visualization using matplotlib, seaborn
Deployment using TF Serving	Need to set up REST API e.g. Flask
tf.device and tf.DeviceSpec to use GPUs (relatively hard)	torch.nn.DataParallel to use GPUs (relatively easy)

Learning From PyTorch



TensorFlow now has eager execution mode for dynamic graph execution

Higher level abstraction to build neural network layers using the Keras API

Demo

Exploring the environment and getting started with scikit-learn

Summary

Machine learning vs. rule-based learning

Choosing the right model based on data

Supervised and unsupervised learning

Regression and classification

Clustering and dimensionality reduction

Transfer learning - cold-start vs. warm-start learning

Popular ML frameworks and their niches