Designing Machine Learning Workflows

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

Overview

End-to-end machine learning workflows

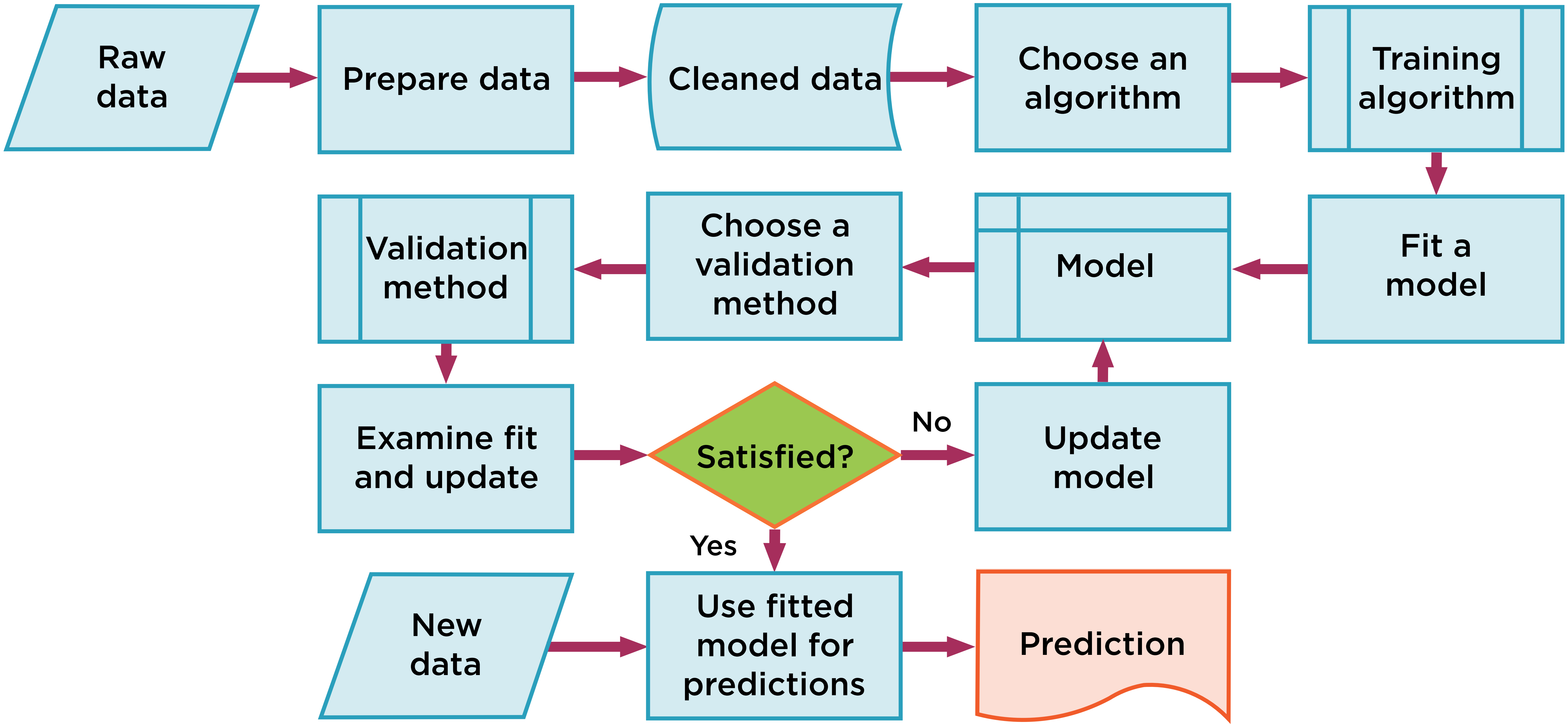
Local, distributed, and cloud-based training and prediction

Understanding the need for ensemble techniques

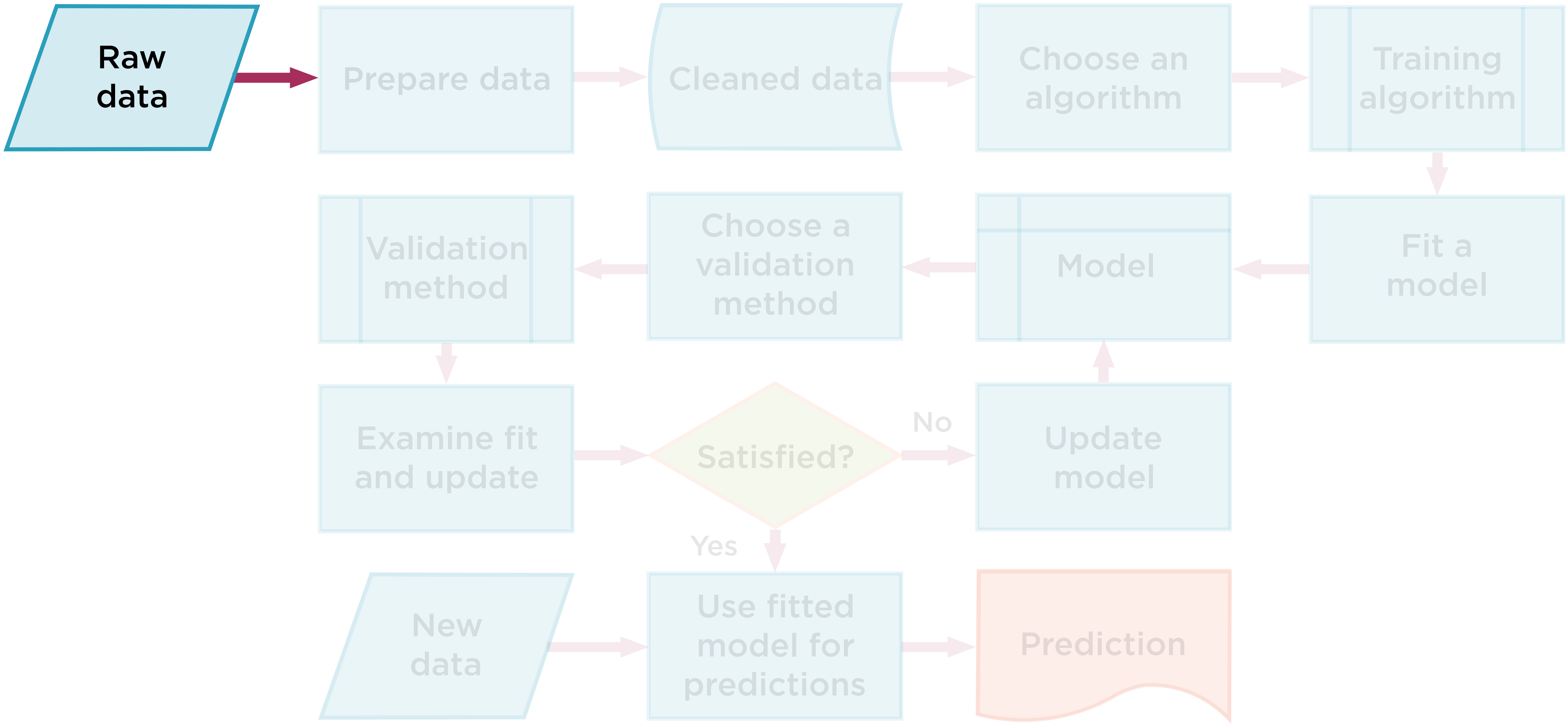
Choosing the right neural network based on the problem

Machine Learning Workflow

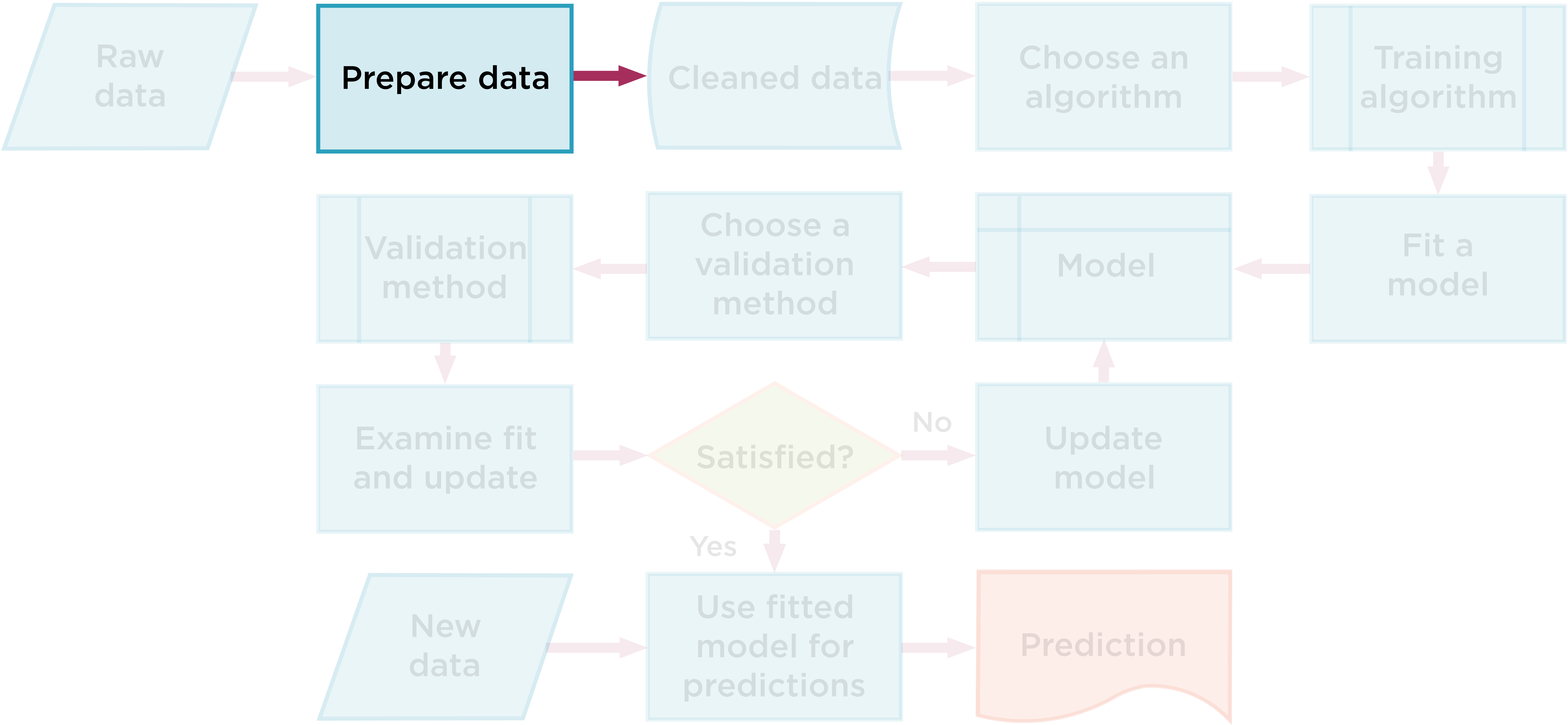
Basic Machine Learning Workflow



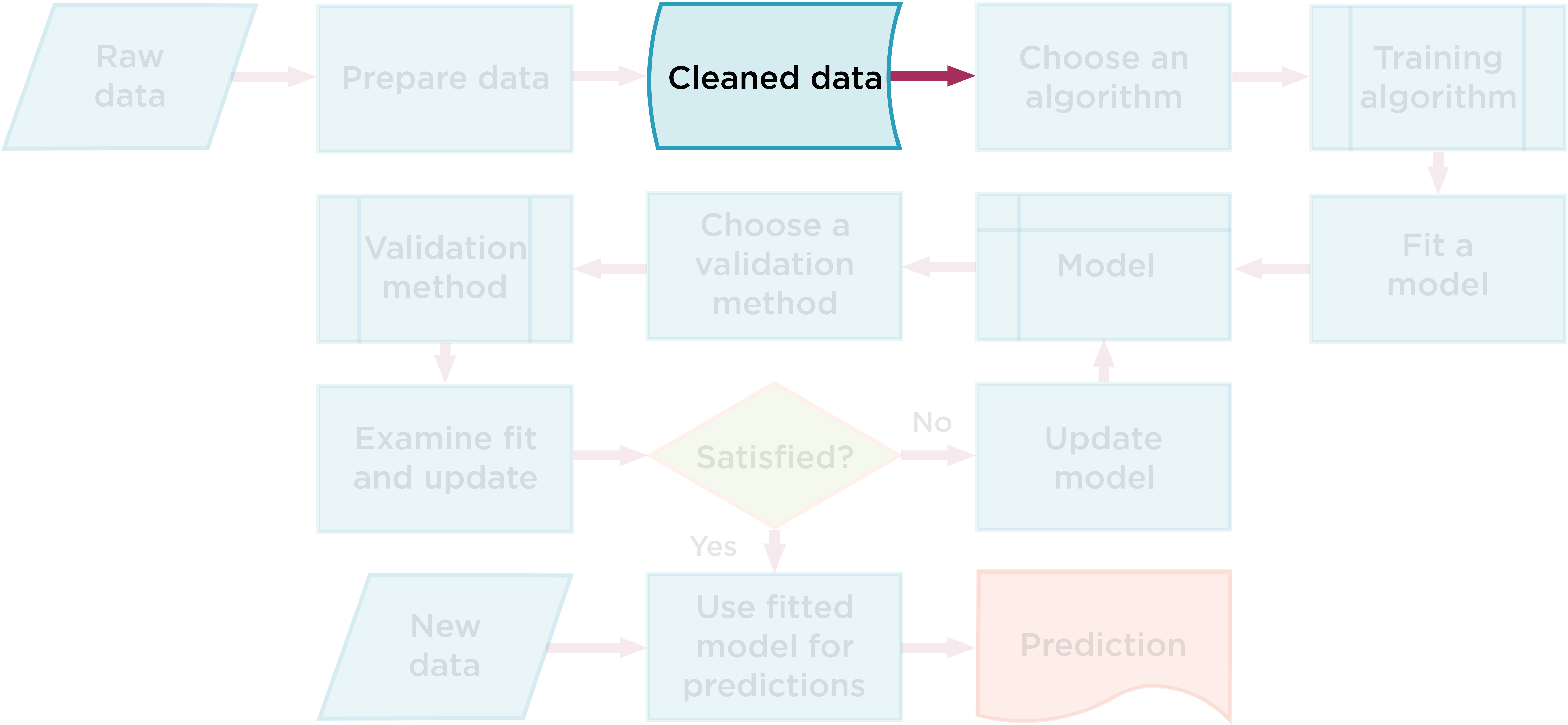
What Data Do You Have to Work With?



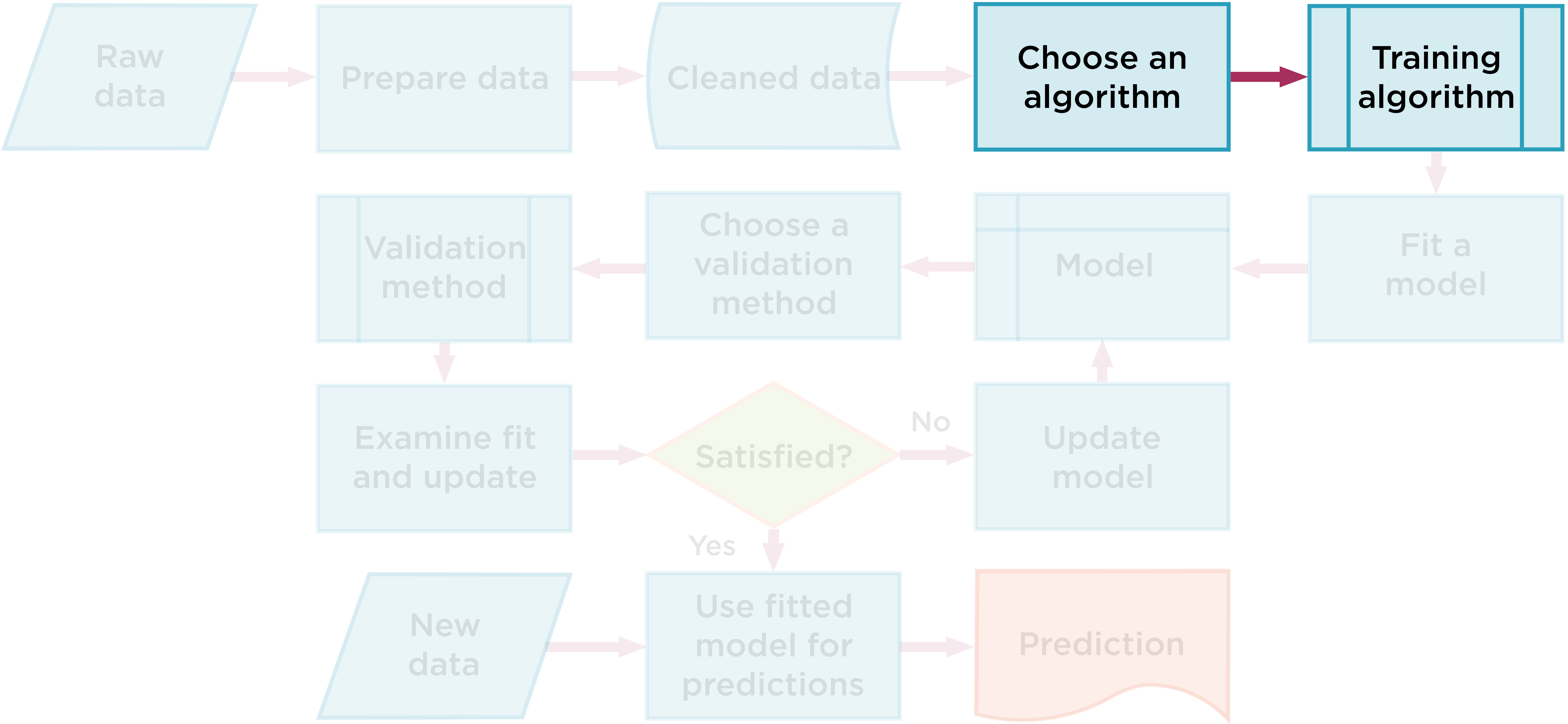
Load and Store Data



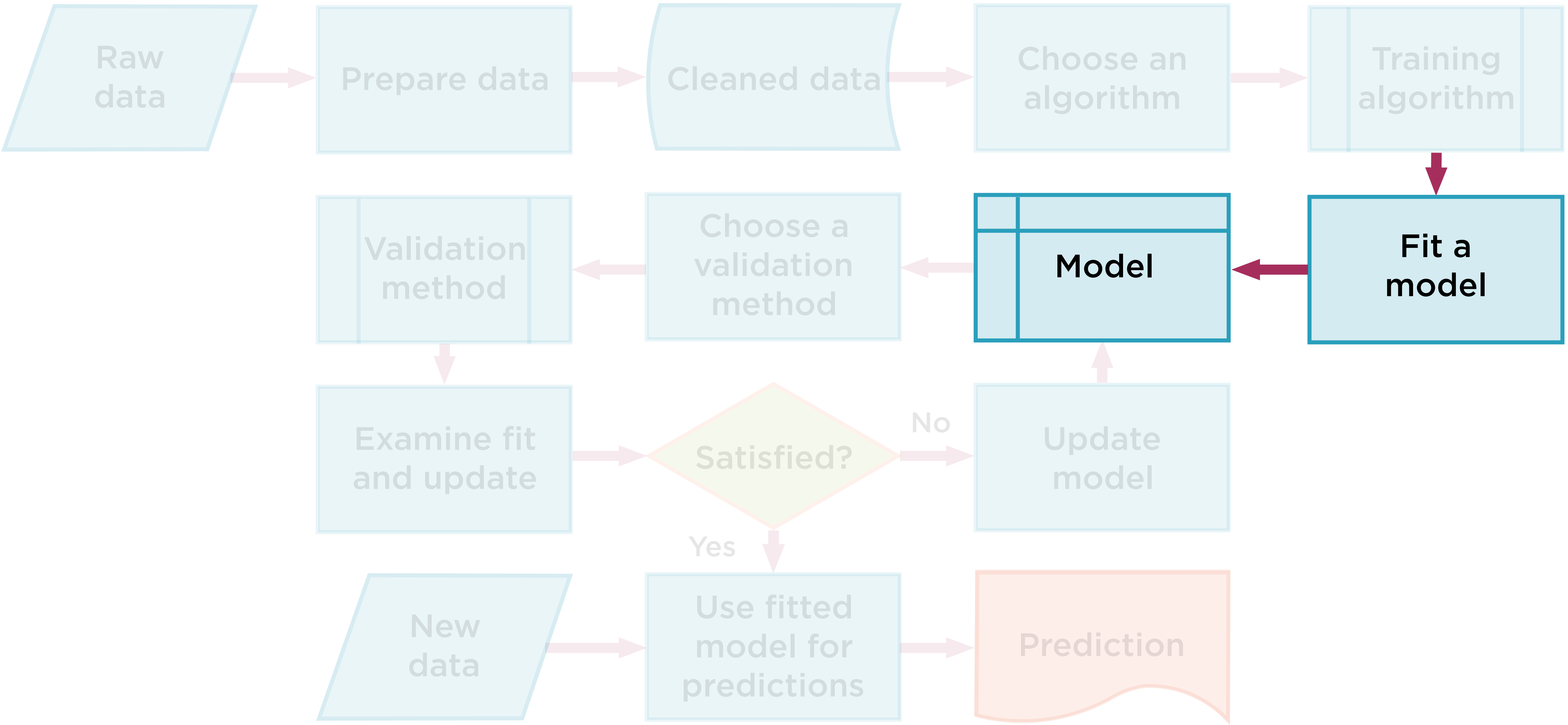
Data Preprocessing



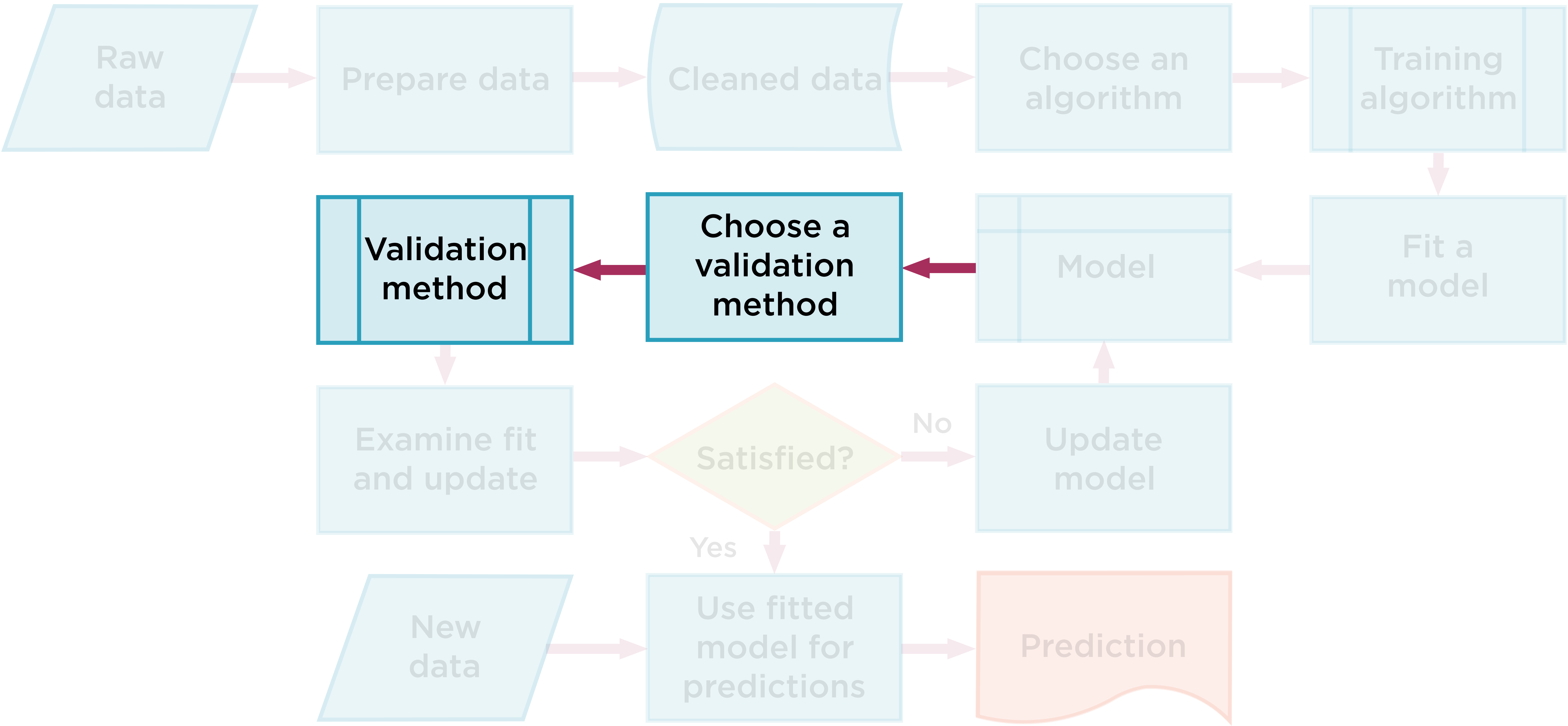
Decision Trees, Support Vector Machines?



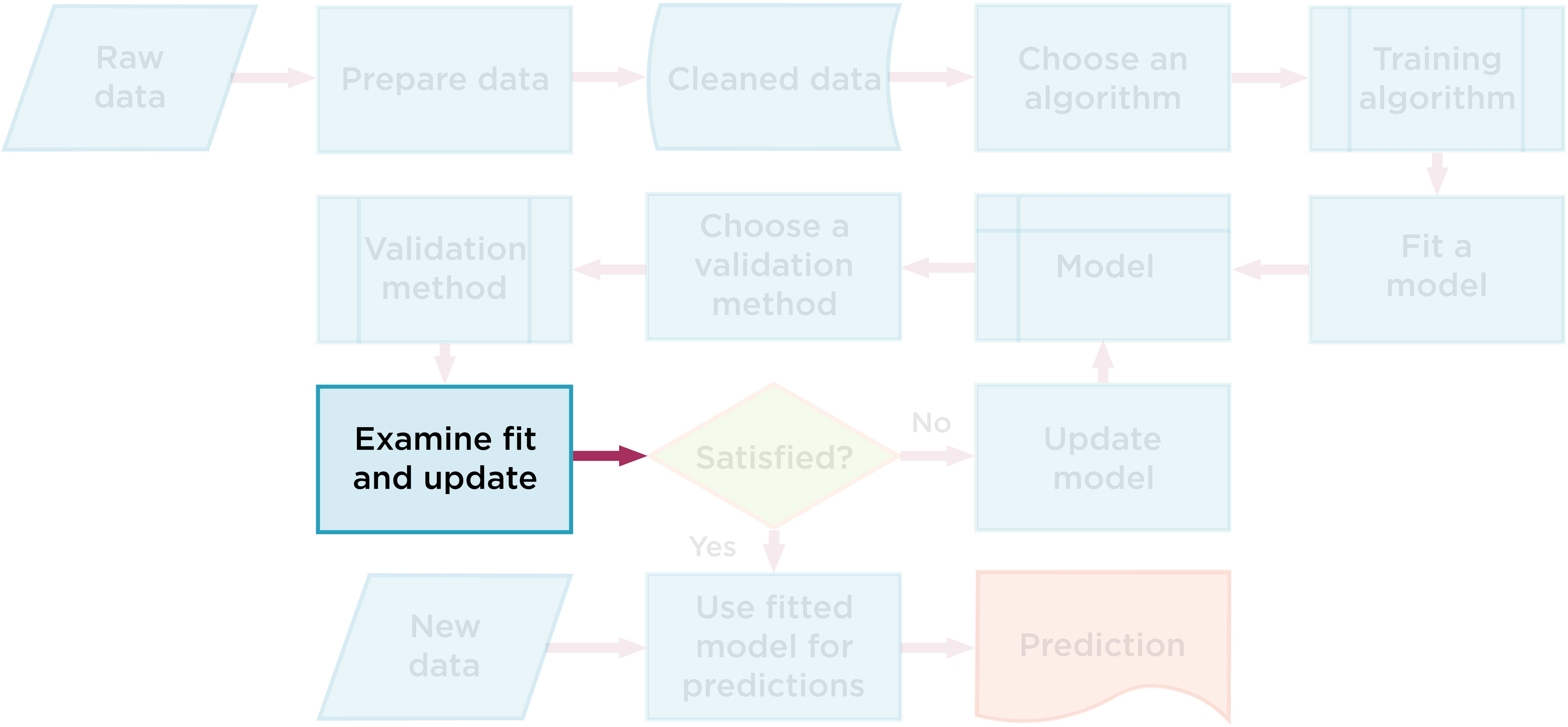
Training to Find Model Parameters



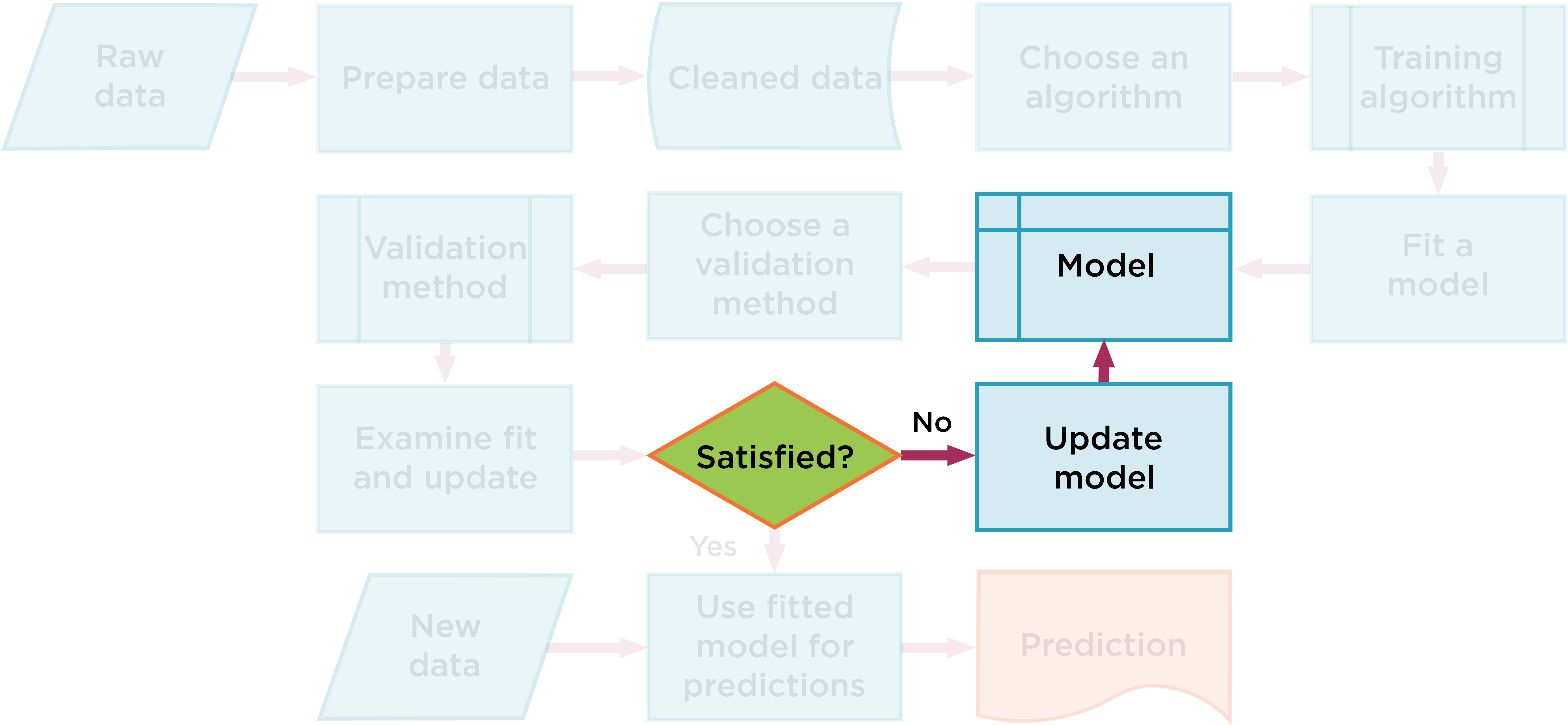
Evaluate the Model



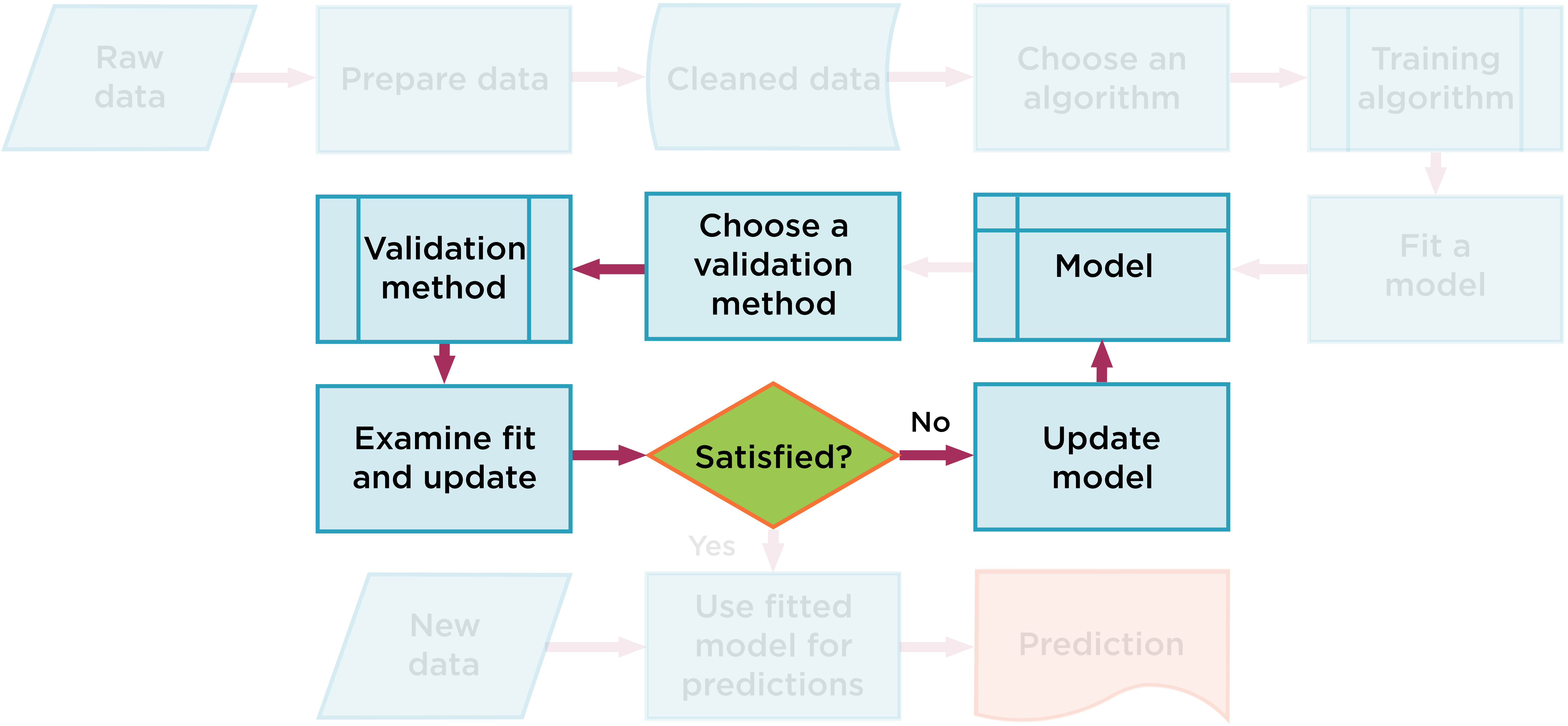
Score the Model



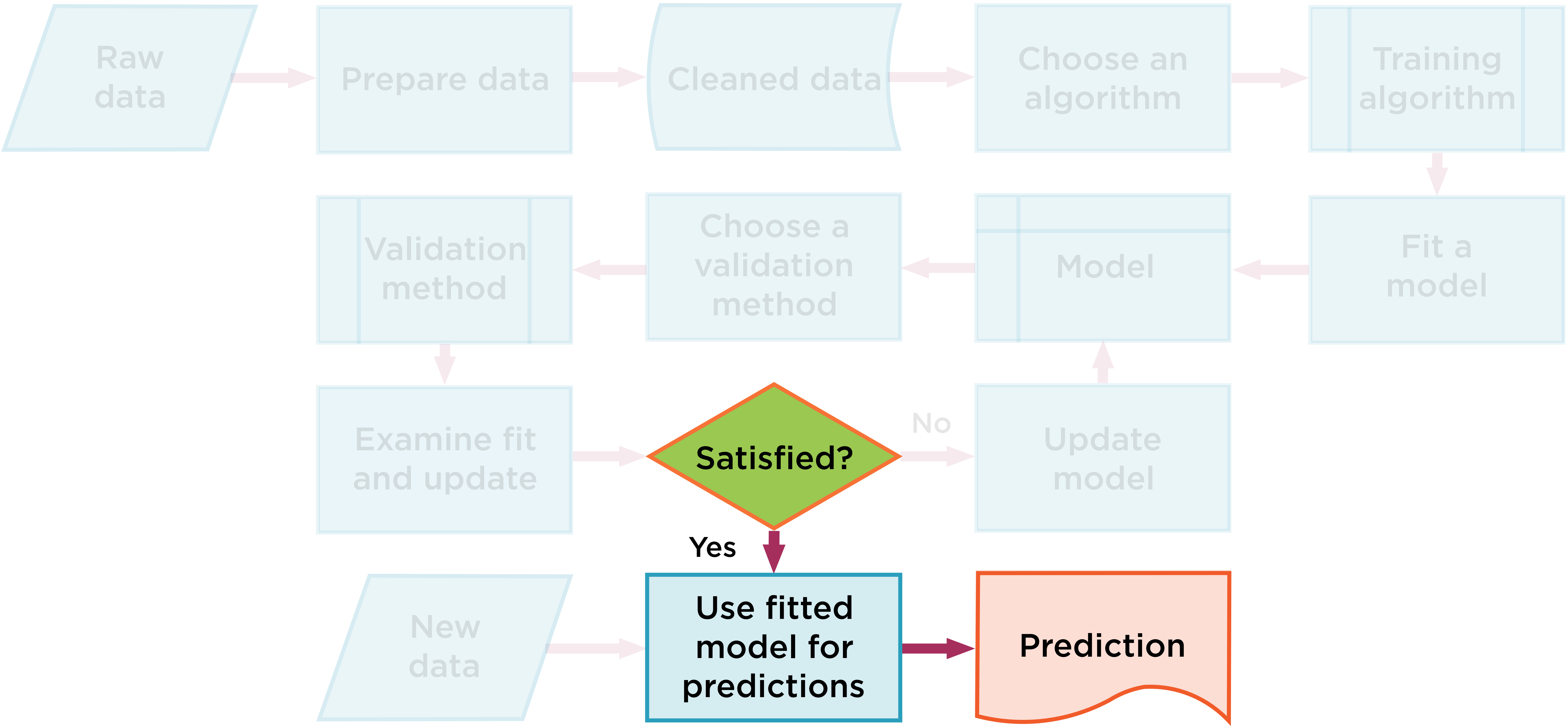
Different Algorithm, More Data, More Training?



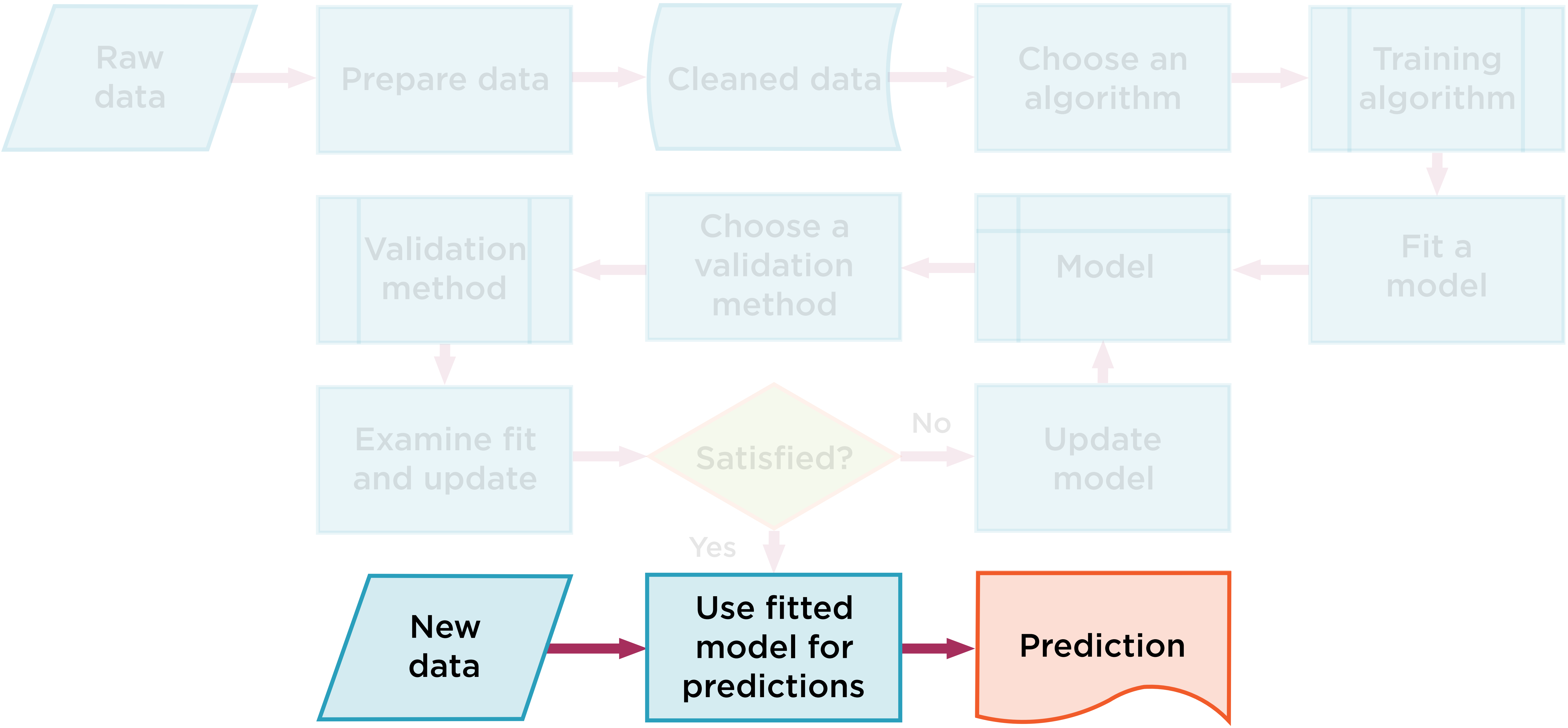
Iterate Till Model Finalized



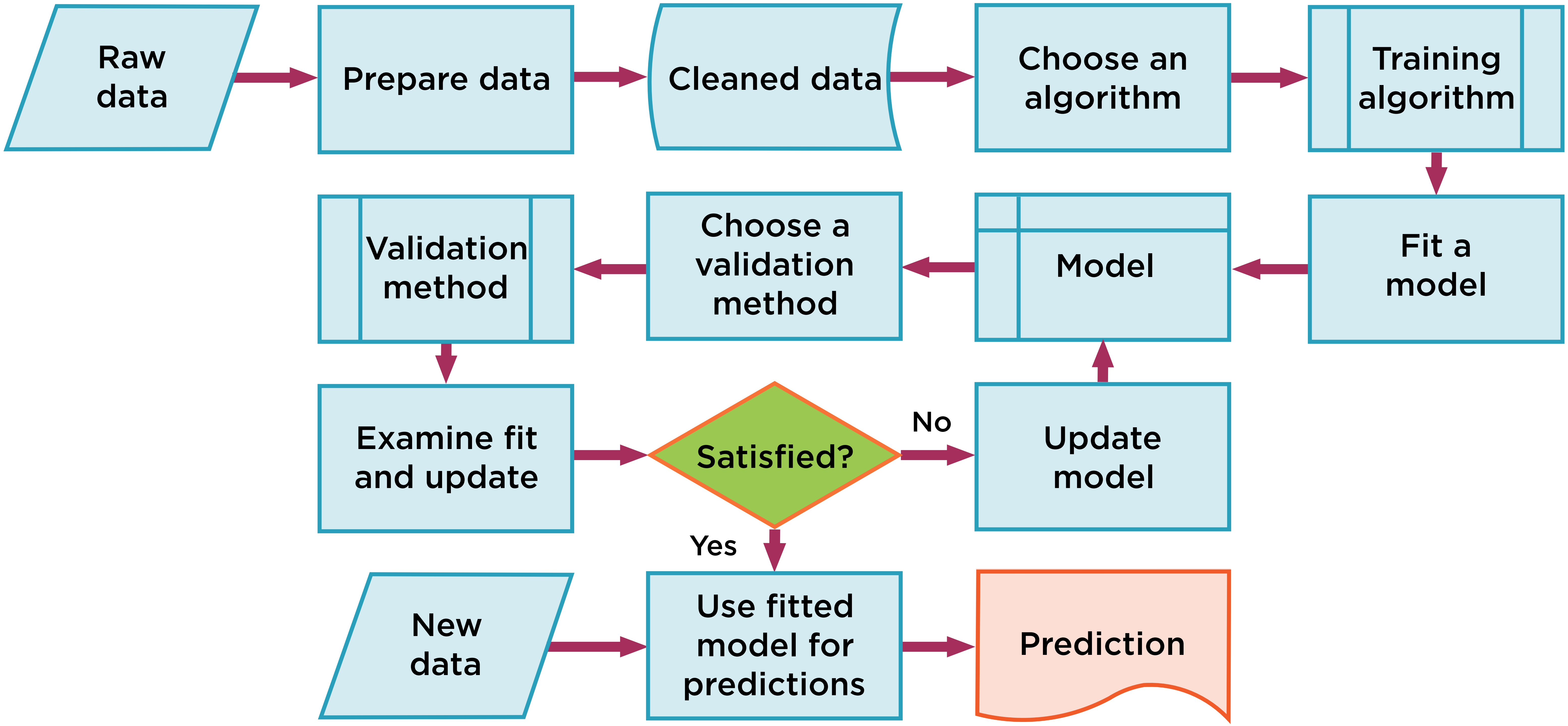
Model Used for Predictions



Retrained Using New Data



Basic Machine Learning Workflow



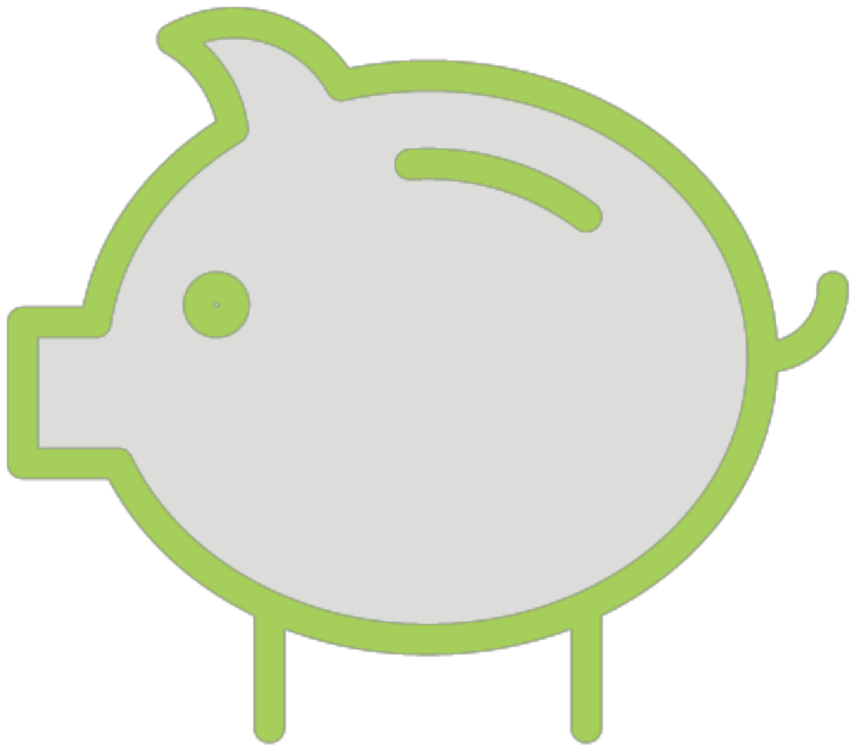
Case Study: PyTorch on the Cloud

PyTorch

A deep learning framework for fast, flexible experimentation.

https://pytorch.org/

Tight Python Integration



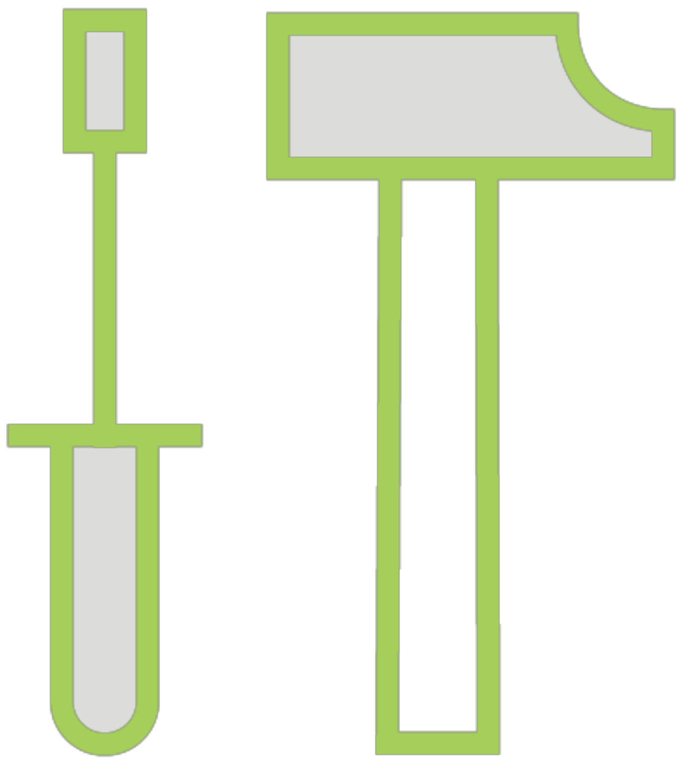
Deeply tied to Python

Approach similar to NumPy/scikit-learn

Create neural networks in Python

Use existing Python libraries and debuggers

GPU-ready Tensor Library



Tensors for either CPU or GPU

Powerful, fast NumPy-like functionality

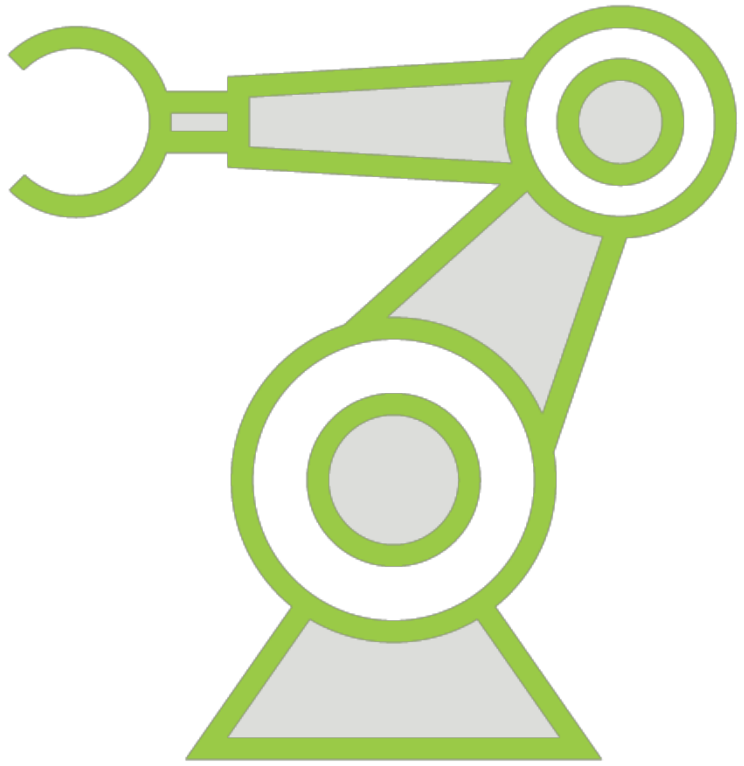
- slicing
- indexing
- reductions
- linear algebra

GPU (Graphics Processing Unit)

Specialized chips with highly parallel architecture that makes them an order of magnitude faster than CPUs for some deep learning applications

PyTorch Tensors have been architected to make optimal use of GPUs for massively parallel computations

GPUs for ML

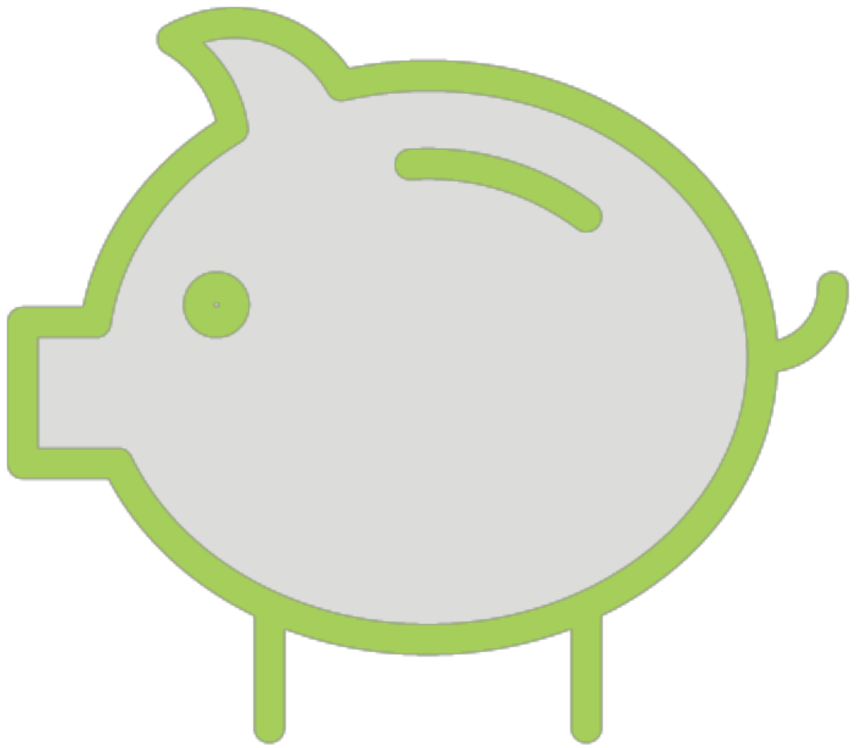


Usage of GPUs has gone far beyond video/graphics processing

Widely used in Big Data and Machine Learning applications

Speedup of 10-50X where parallelization yields big wins

Training Options



Local training as starter option

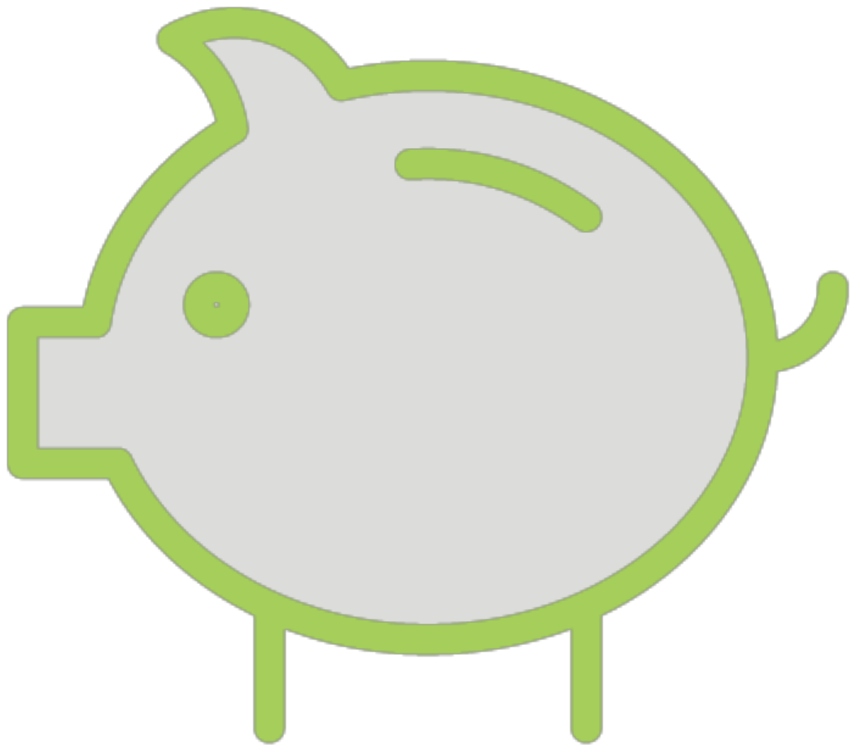
Fine for prototyping

Can not leverage GPUs

Out-of-memory issues

Hours and hours for training

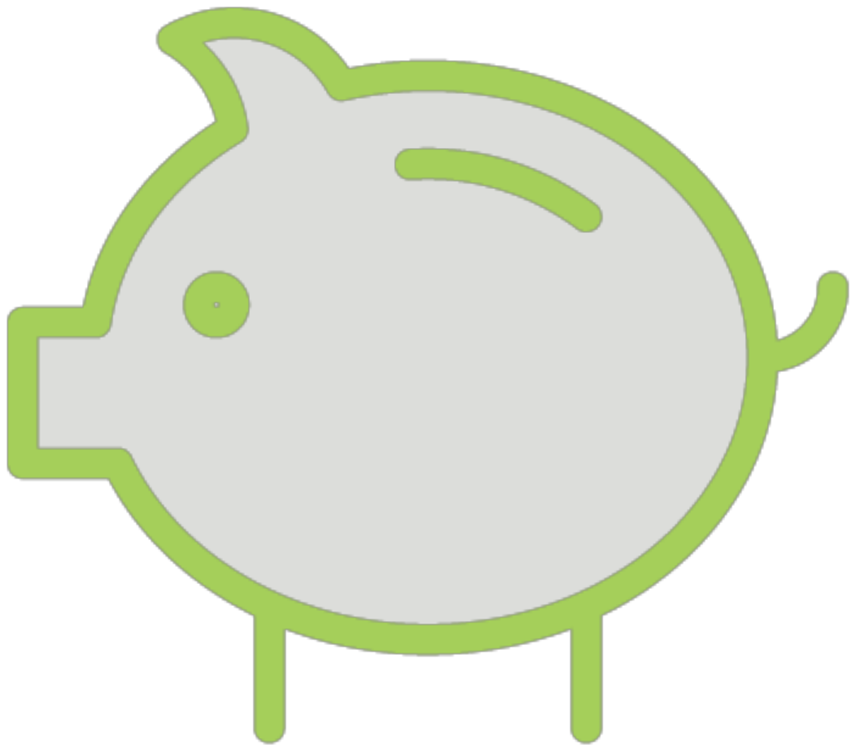
Training Options



Distributed training

- More epochs
- Performance dramatically rises
- Scaling needs additional hardware

Training Options



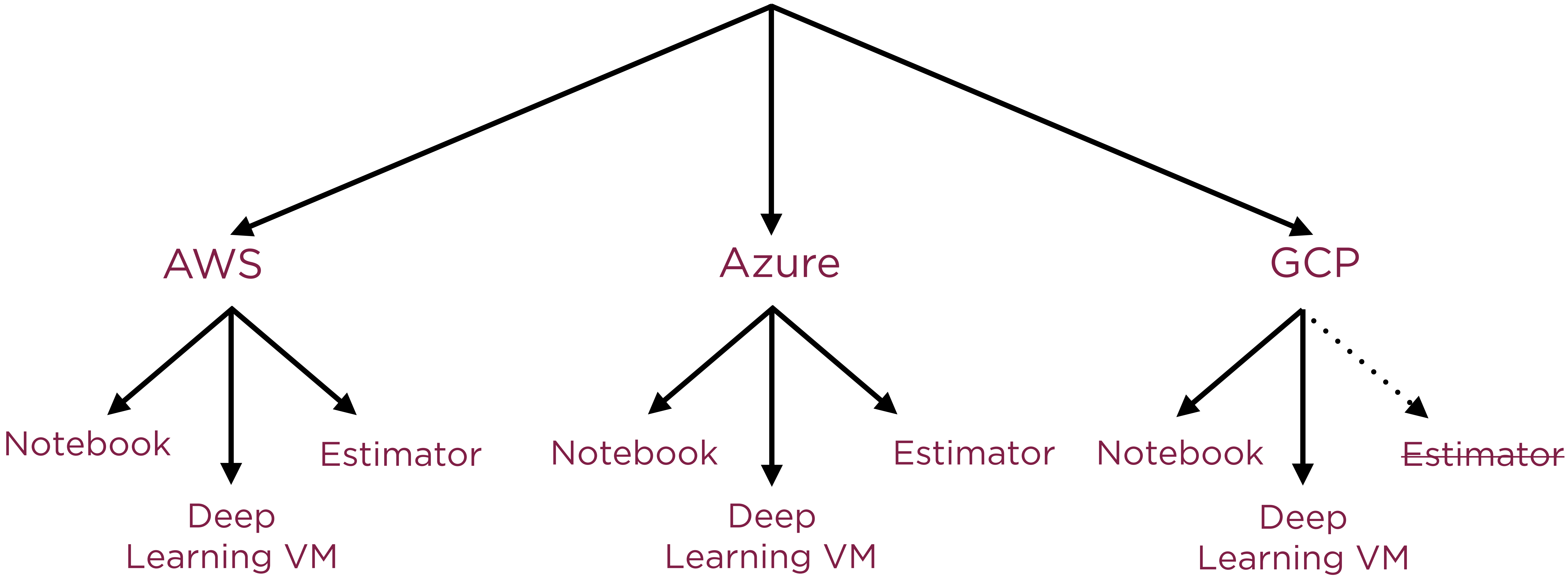
On-cloud training

- Pay-as-you-go
- Elastic and scalable
- CUDA and GPU support
- Frameworks and platform support

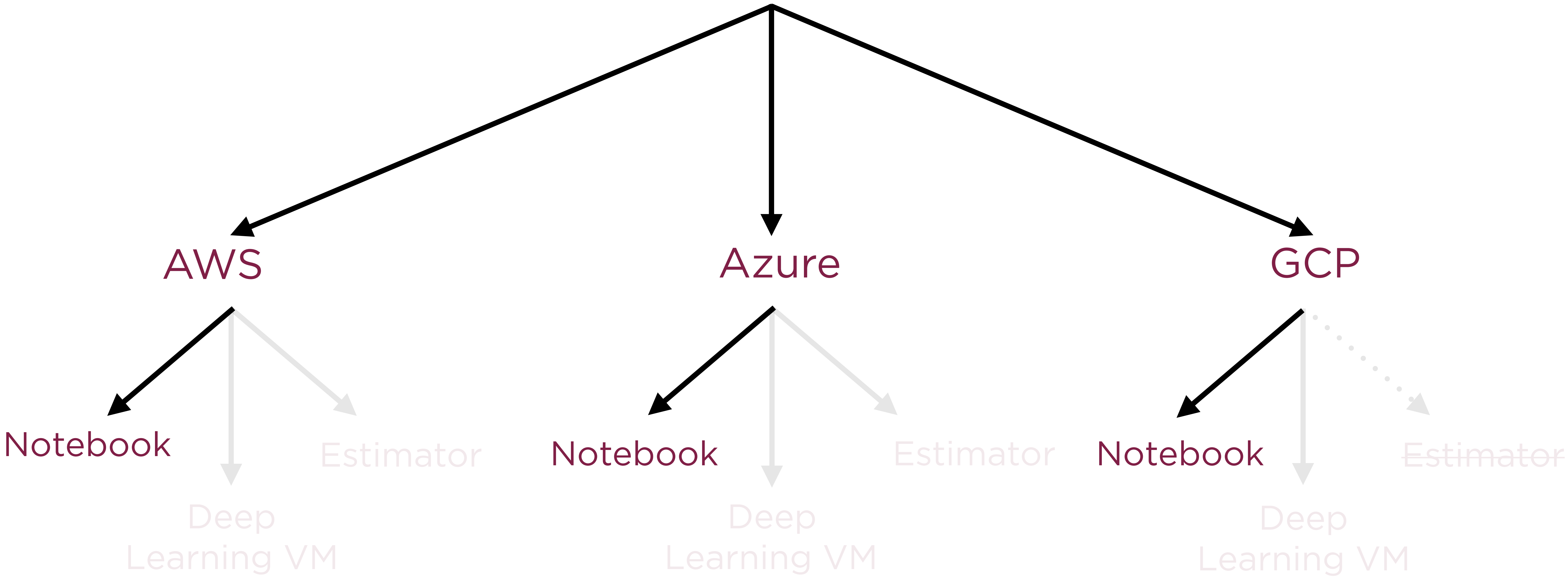
ML Design Choices

- Training
  - Local
  - Distributed
    - On-cloud
- Prediction
  - Batch
  - Online

PyTorch on the Cloud



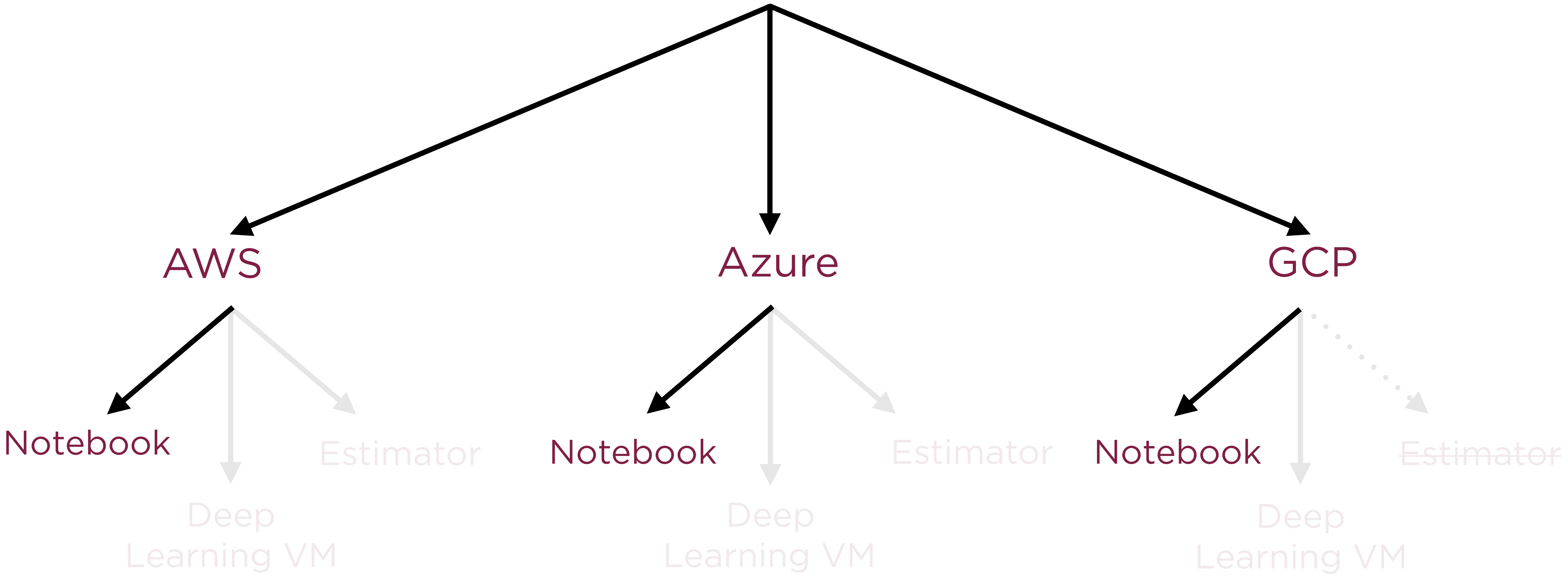
PyTorch on the Cloud



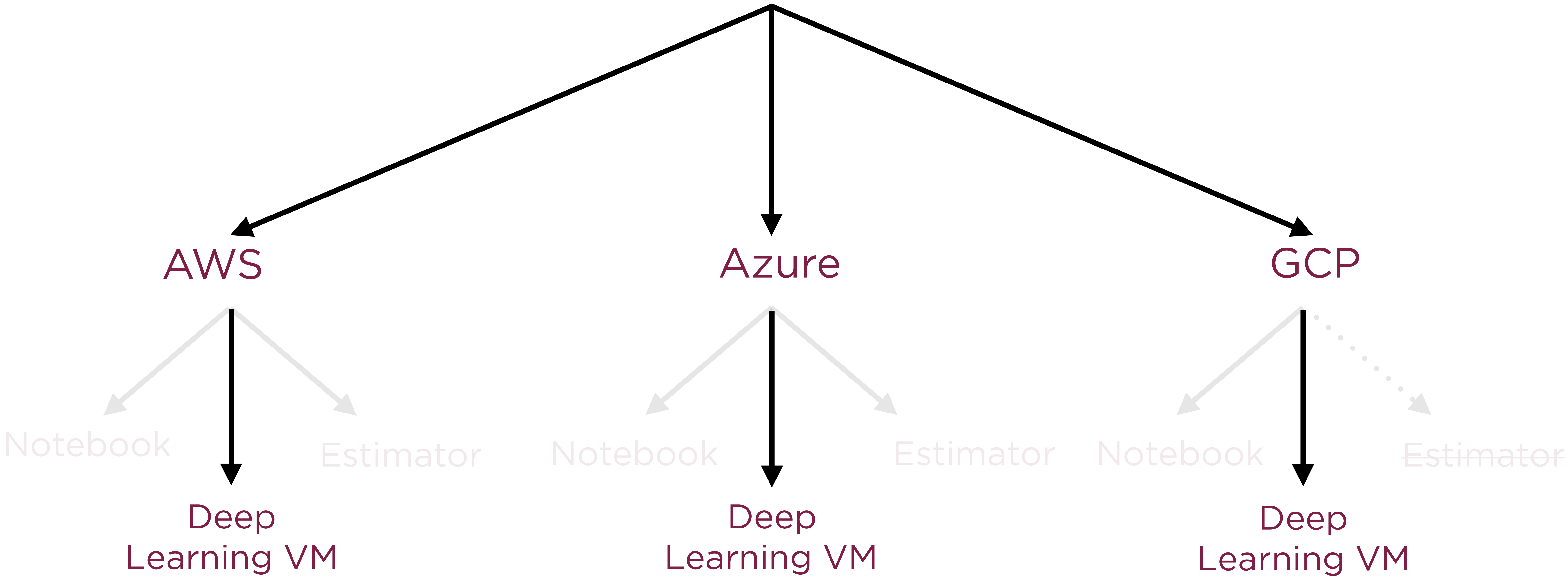
Notebook

Cloud-hosted Python notebook. Could be platform-agnostic (Jupyter) or platform-specific (e.g. Datalab on GCP)

PyTorch on the Cloud



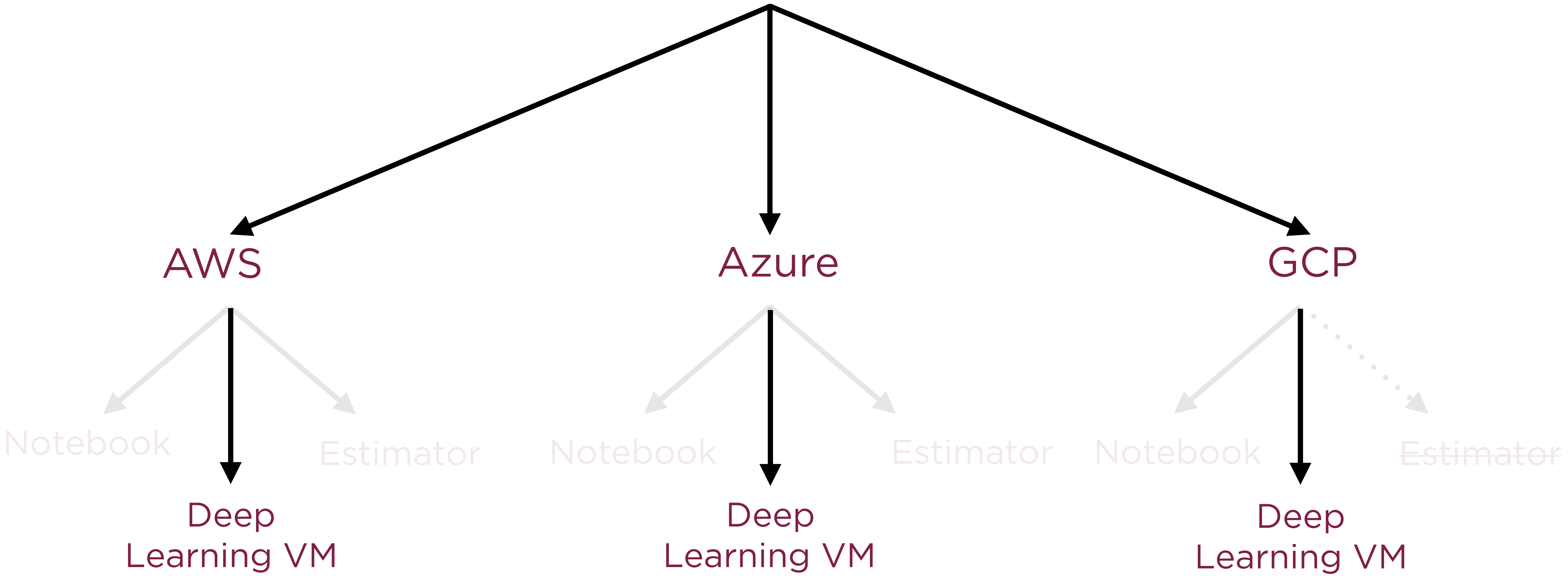
PyTorch on the Cloud



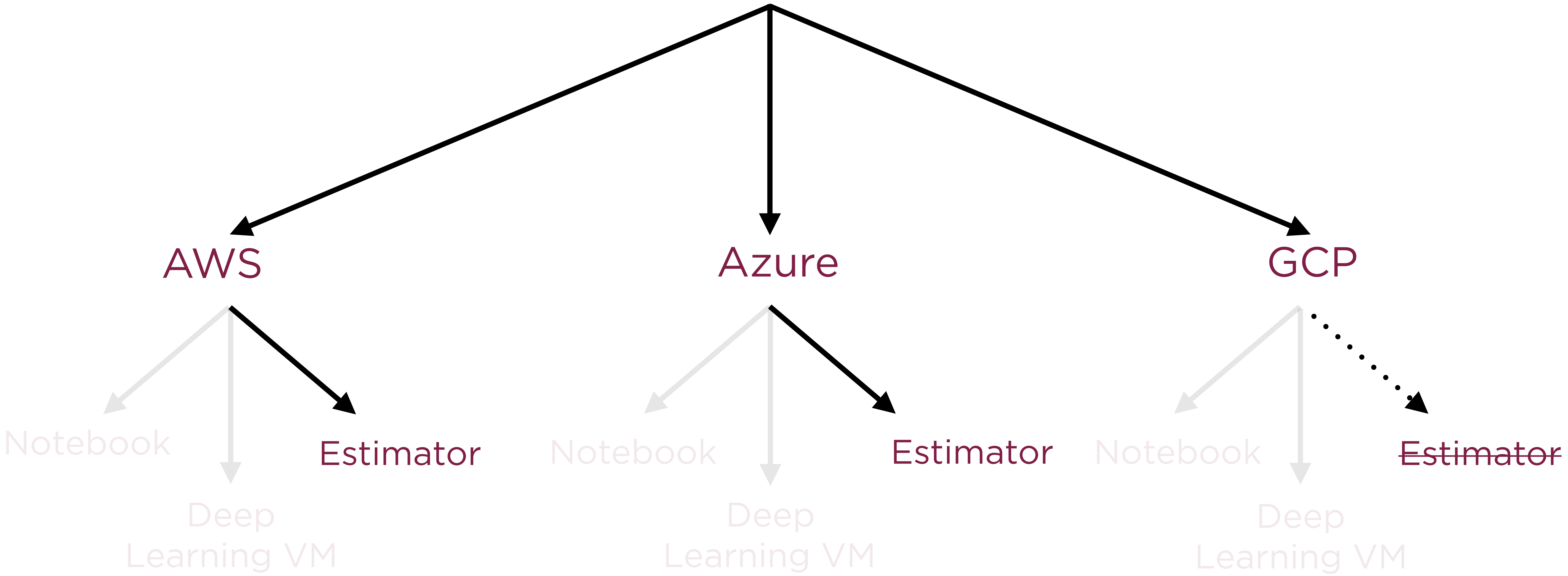
Deep Learning VM

Cloud-specific virtual machine instance (e.g. EC2 on AWS, GCE on GCP) equipped with GPUs for optimized PyTorch performance

PyTorch on the Cloud



PyTorch on the Cloud



Estimator

High-level API, specific to a cloud platform, that helps build, train, and deploy PyTorch models

Quick Overview of Ensemble Learning

Ensemble Learning

Machine learning technique in which several learners are combined to obtain a better performance than any of the learners individually.

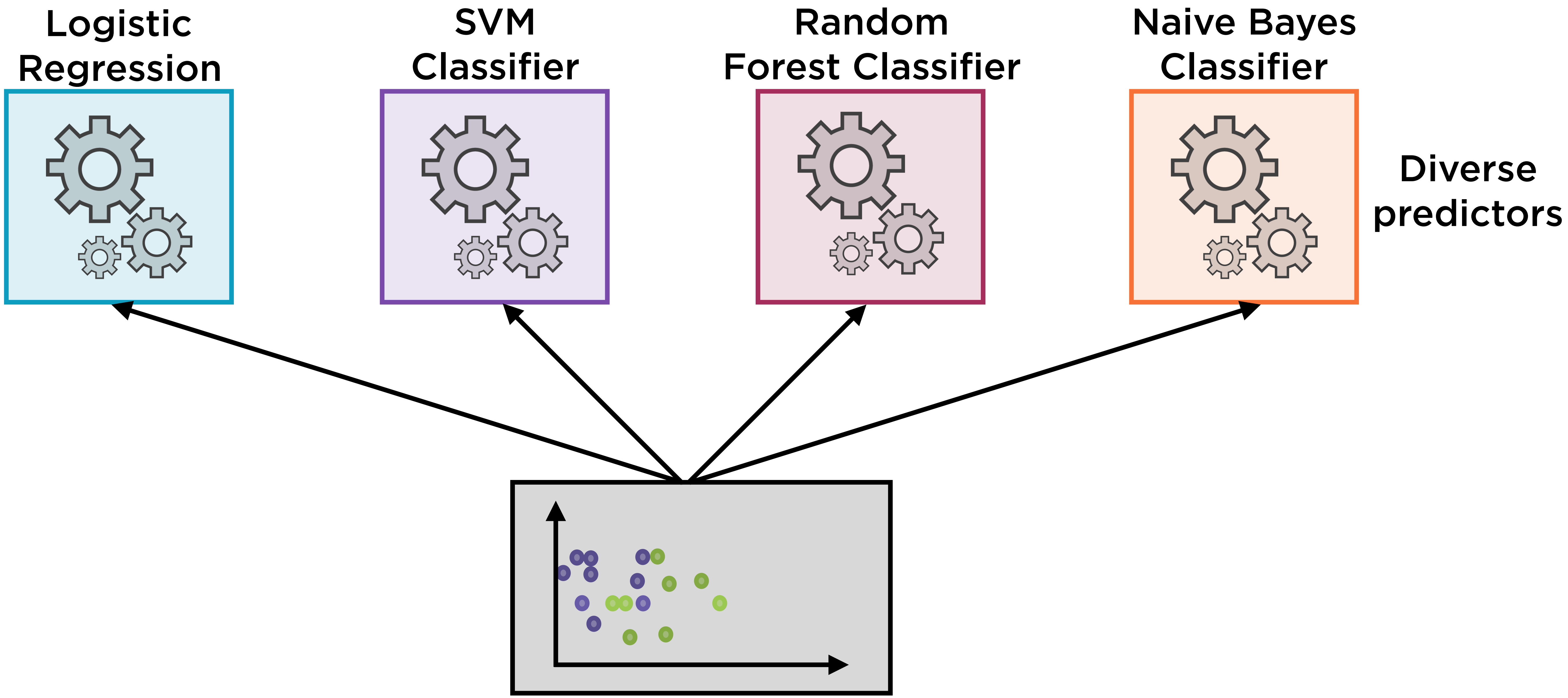
Ensemble Learning

Machine learning technique in which several learners are combined to obtain a better performance than any of the learners individually.

Ensemble Learning

Machine learning technique in which several learners are combined to obtain a better performance than any of the learners individually.

Ensemble Learning



Important Questions in Ensemble Learning

What kind of individual learners to use?

How should individual learners be trained?

How should individual learners be combined?

Important Questions in Ensemble Learning

What kind of individual learners to use?

How should individual learners be trained?

How should individual learners be combined?

Choice of Individual Learners

Individual learners (models) could be of absolutely any type

Each learner should be as different as possible from other learners

Choice of Individual Learners

Decision trees are most often used

An ensemble of decision trees is a Random Forest

Random forests make it easy to build uncorrelated learners

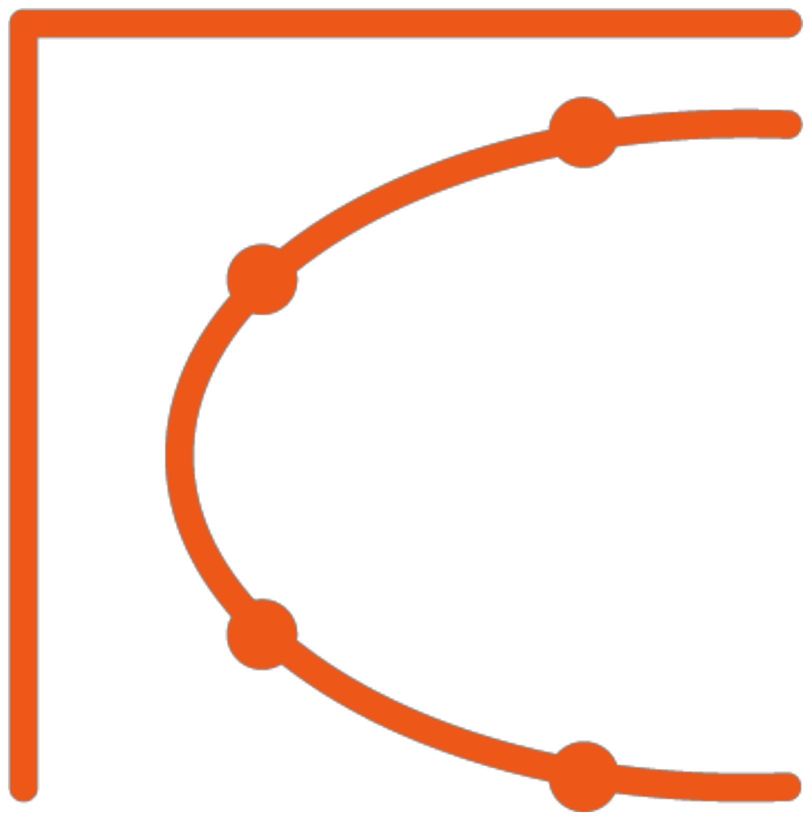
Important Questions in Ensemble Learning

What kind of individual learners to use?

How should individual learners be trained?

How should individual learners be combined?

Training Individual Learners



If learners are different, each learner can be trained on the entire dataset

For similar learners:

- Each model is trained on random samples of training data
- Can also use random set of features to train different models

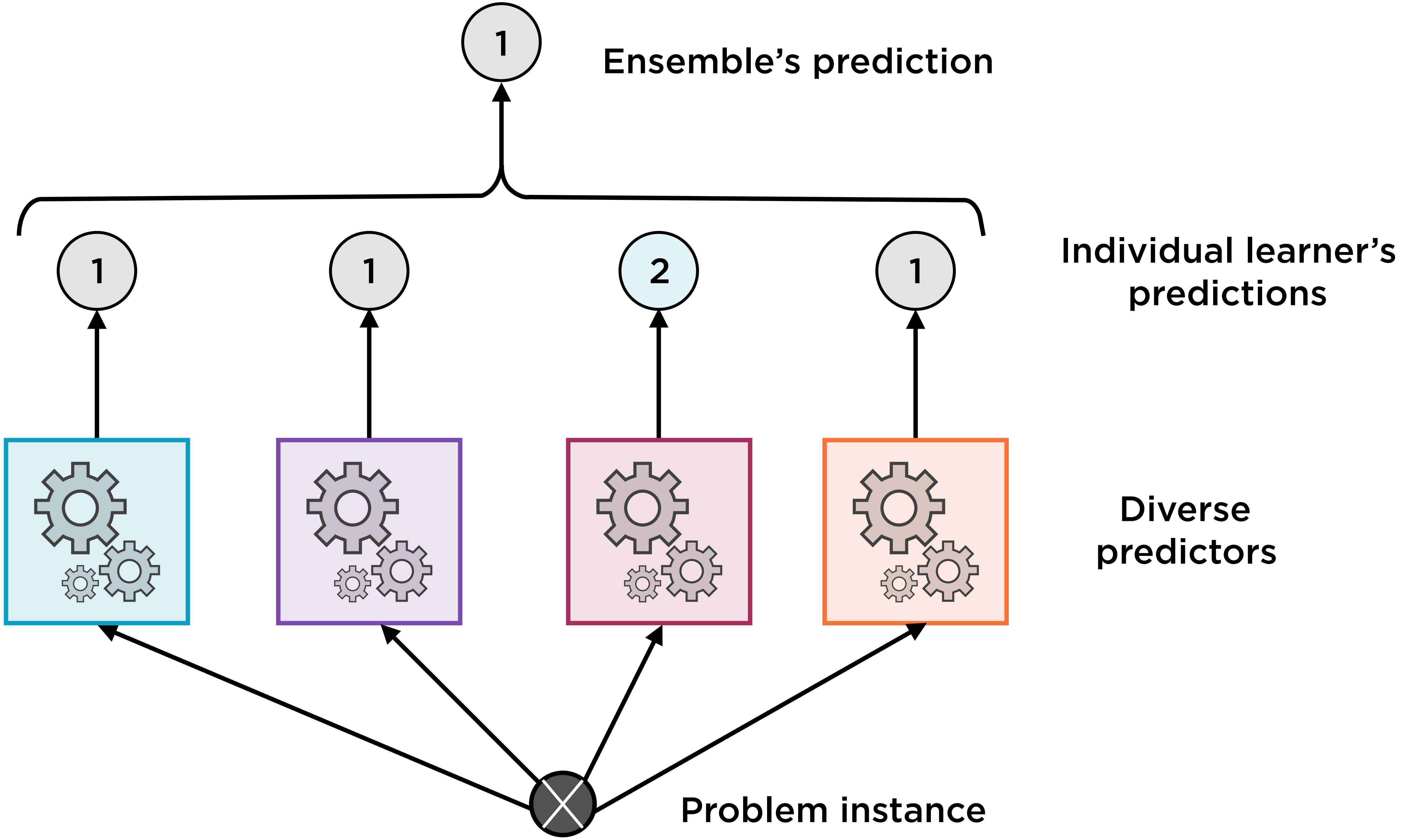
Important Questions in Ensemble Learning

What kind of individual learners to use?

How should individual learners be trained?

How should individual learners be combined?

Combining Classifier Predictions



Combining Individual Learners

Hard voting: Majority vote of individual learners (classification)

Soft voting: Probability-weighted average

Stacking: Train additional model to combine predictions from individual learners

Ensemble Learning Techniques

Important Questions in Ensemble Learning

What kind of individual learners to use?

How should individual learners be trained?

How should individual learners be combined?

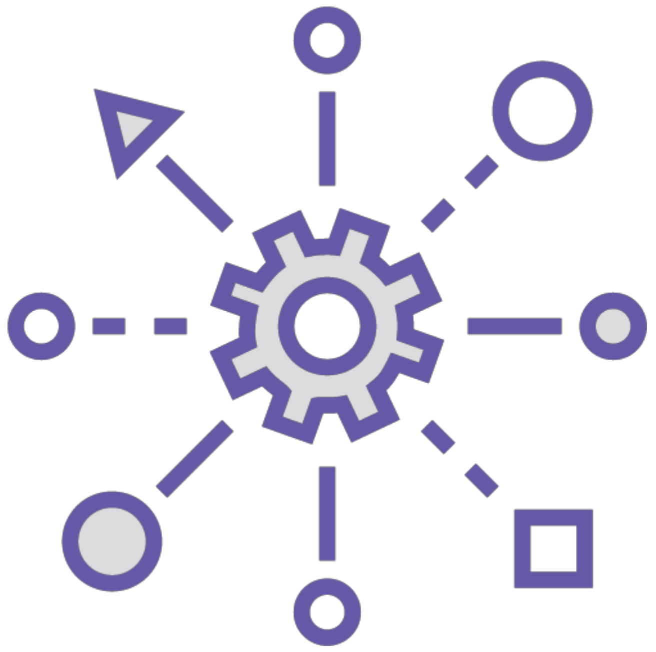
Important Questions in Ensemble Learning

What kind of individual learners to use?

How should individual learners be trained?

How should individual learners be combined?

Averaging and Boosting



Averaging

Train predictors in parallel and average scores of individual predictors

Boosting

Train predictors in sequence where each predictor learns from earlier mistakes

Averaging vs. Boosting

Averaging	Boosting
Individual learners are independent	Individual learners are linked to previous learners
Can build trees in parallel	Need to build tree sequentially
Learners do not learn from mistakes of other learners	Individual learners explicitly configured to learn from previous mistakes

Important Questions in Ensemble Learning

What kind of individual learners to use?

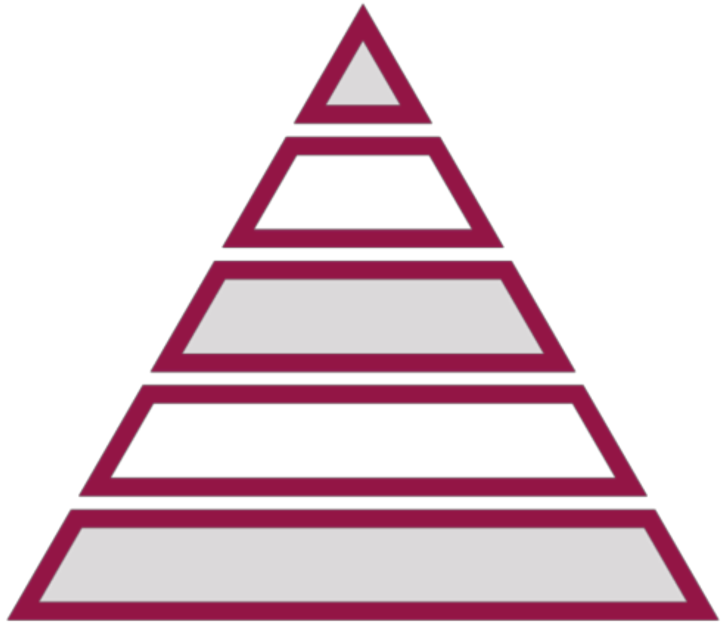
How should individual learners be trained?

How should individual learners be combined?

Voting and Stacking

Voting

Majority vote of the individual predictors is the final prediction of the ensemble

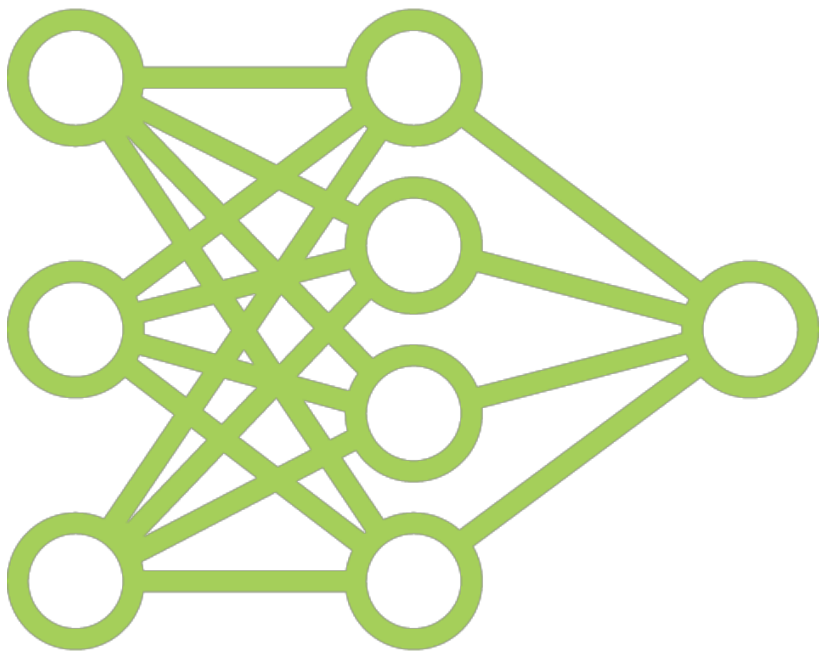


Stacking

Fit a model on the individual predictions to get the final prediction of the ensemble

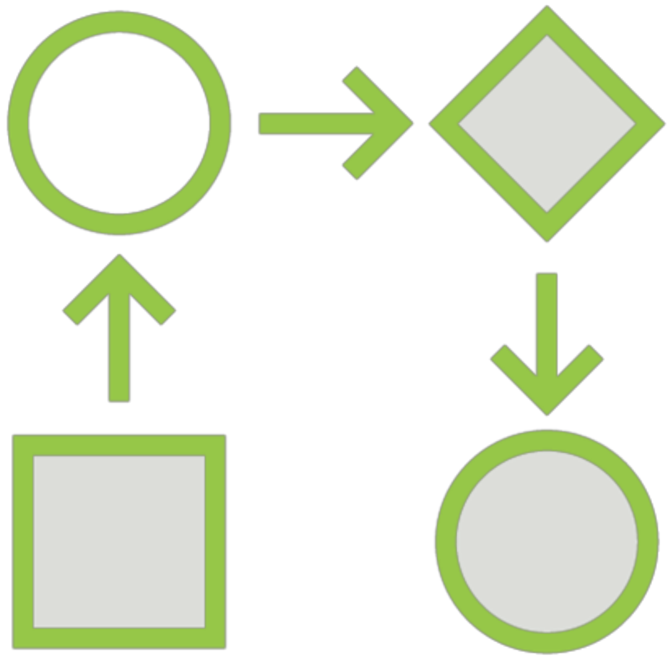
Neural Network Models

Deep Learning Models



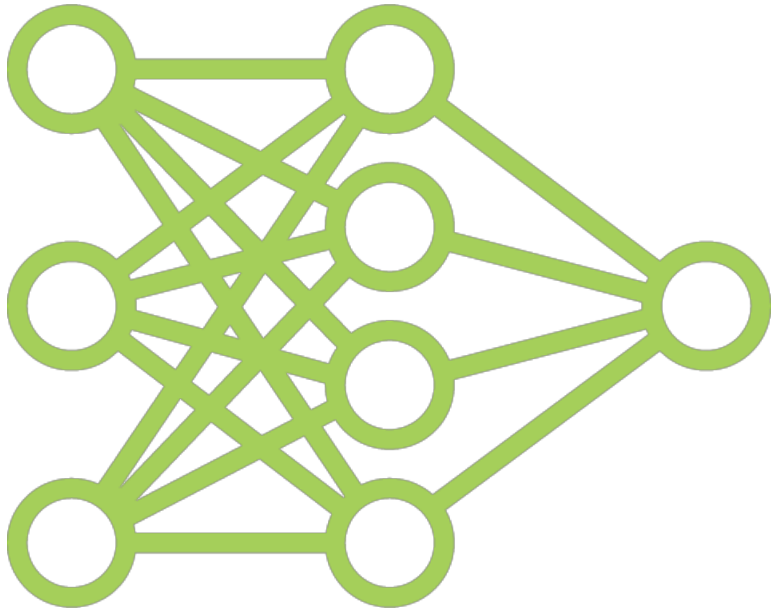
Fully-connected, dense neural networks

Convolutional neural networks



Recurrent neural networks

Dense Neural Networks



Work well with numeric features

Traditional classification, regression

Layers of interconnected neurons

All neurons in one layer connected to neurons in the previous and next layers

Convolutional Neural Networks

Specialize in working with image data

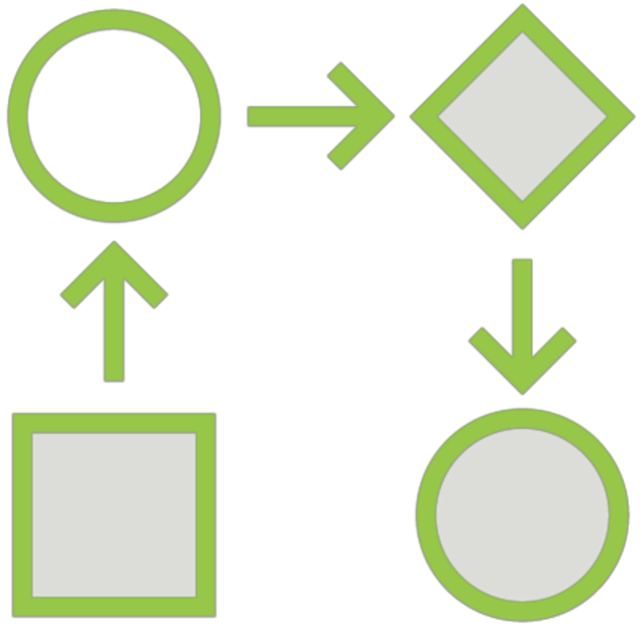
Designed to mimic the visual cortex of the brain

Sparse neural networks

Convolutional layers for feature detection

Pooling layers for subsampling of inputs

Recurrent Neural Networks



Specialize in sequential data such as text or time series data

Neurons have “memory” or state

Neural network layers represent instances in time

Summary

End-to-end machine learning workflows

Local, distributed, and cloud-based training and prediction

Understanding the need for ensemble techniques

Choosing the right neural network based on the problem