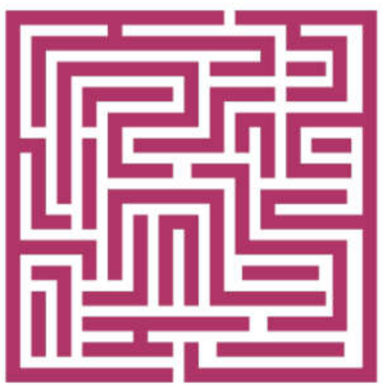
Overview

- Role of data in machine learning
- Features and labels
- The machine learning workflow
- Feature engineering to convert data to features
- Training, test, and validation data

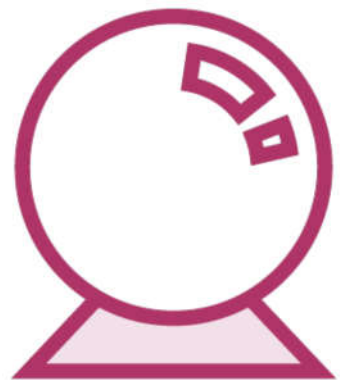
Features and Labels in Machine Learning

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

Machine Learning



Work with a huge maze of data



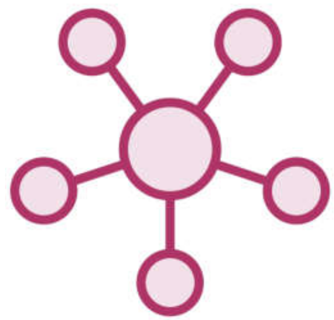
Find patterns

Make intelligent decisions

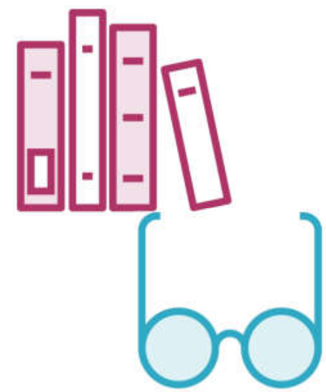
Types of Machine Learning Problems

Classification

Regression



Clustering



Dimensionality Reduction

Types of Machine Learning Problems

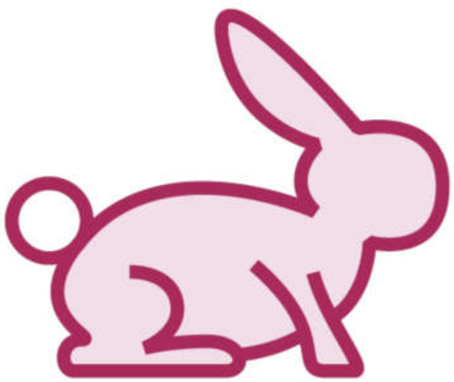
Classification

Regression

Clustering

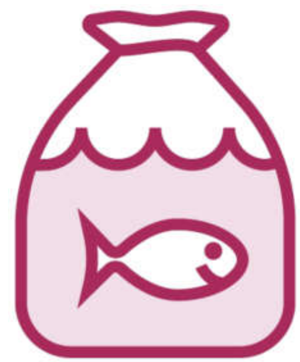
Dimensionality Reduction

Whales: Fish or Mammals?



Mammals

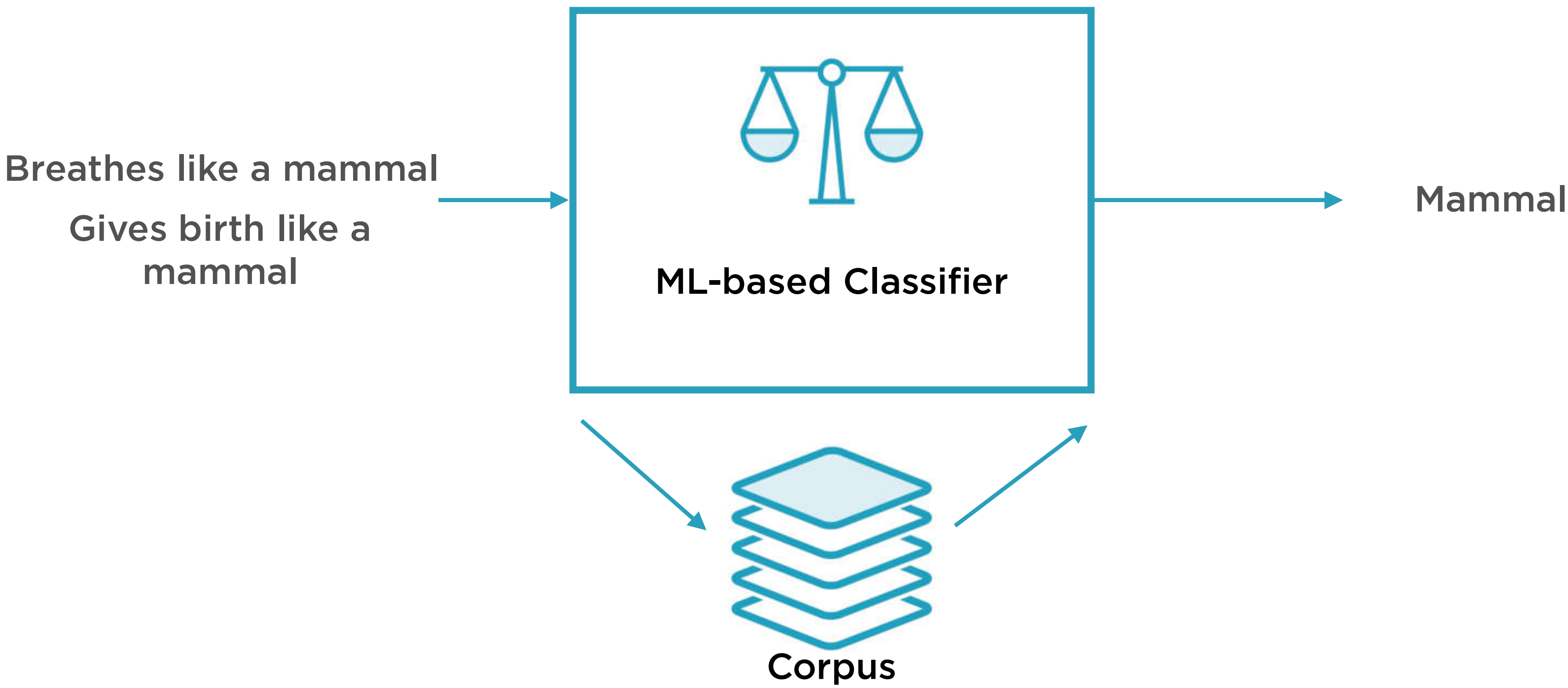
Members of the infraorder *Cetacea*



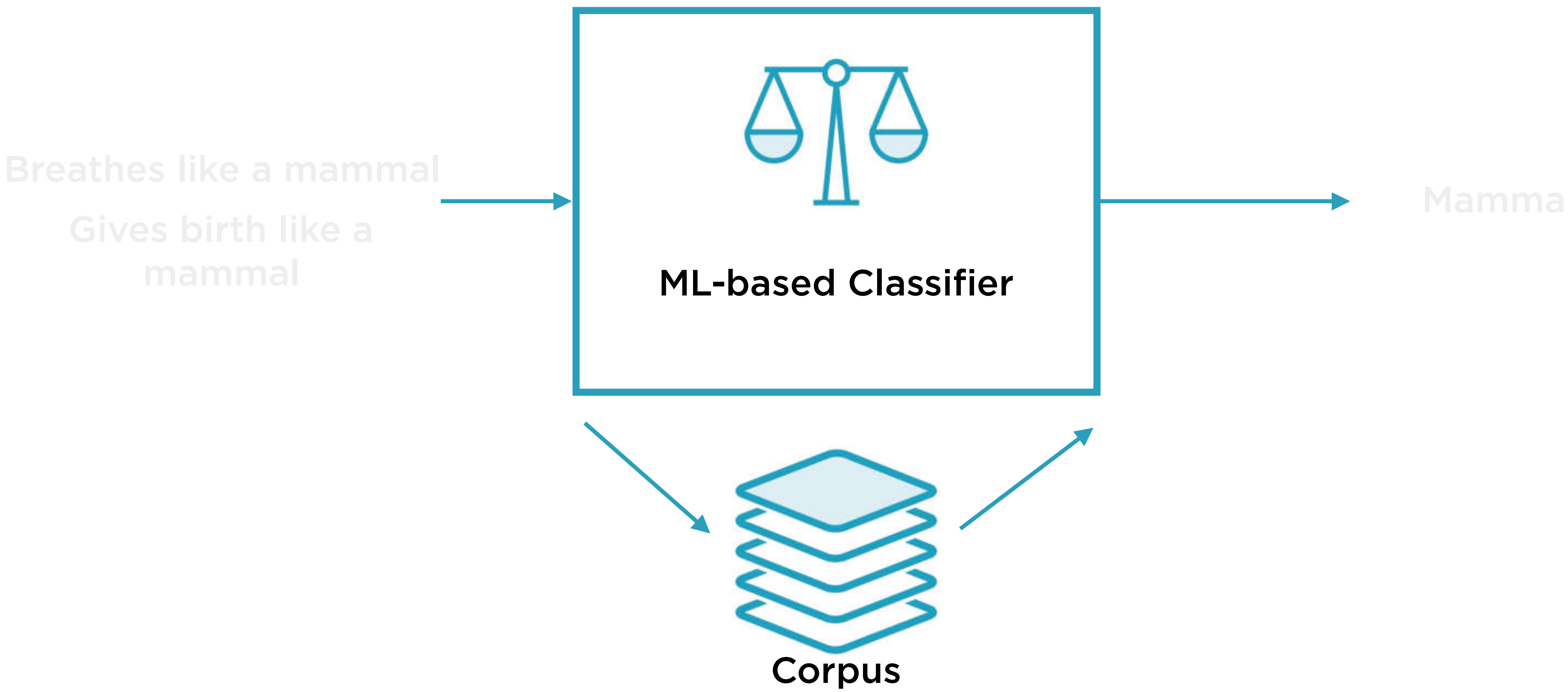
Fish

Look like fish, swim like fish, move with fish

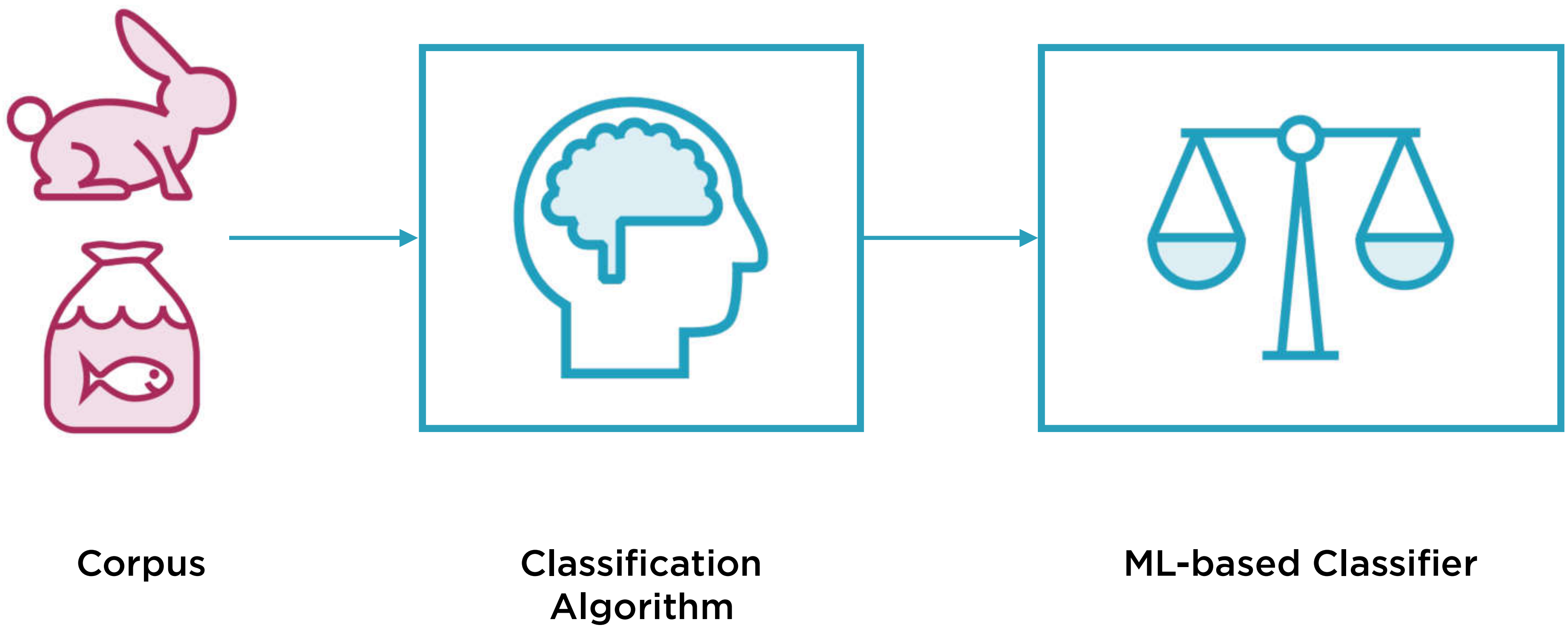
ML-based Binary Classifier



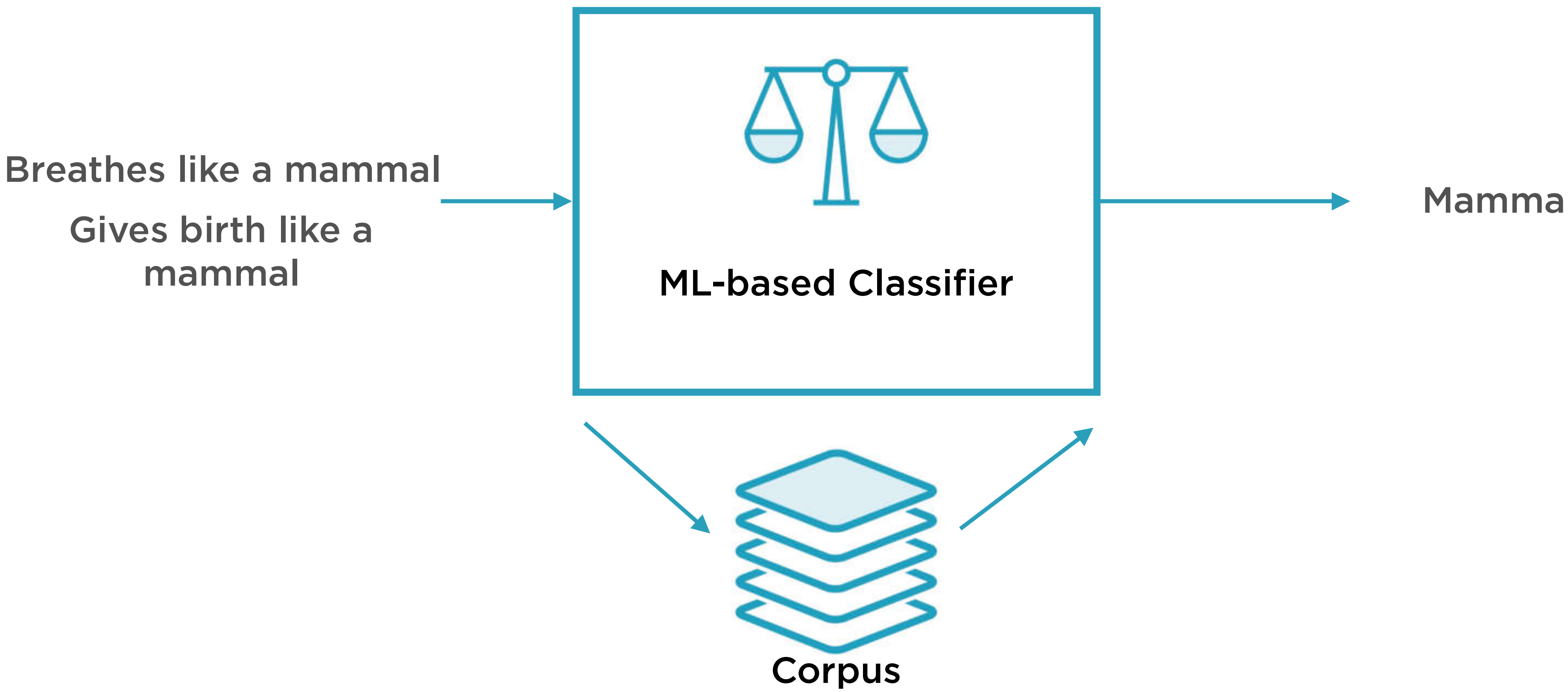
ML-based Binary Classifier



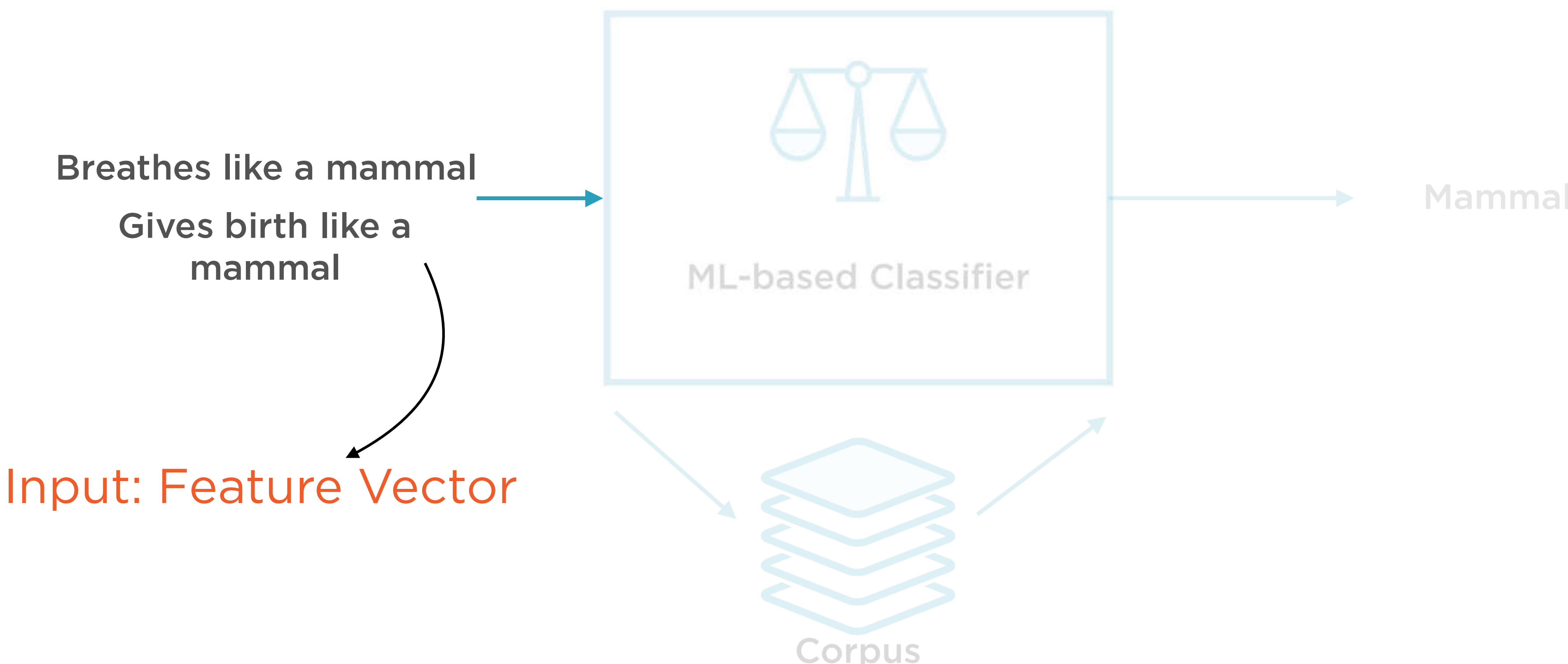
ML-based Binary Classifier



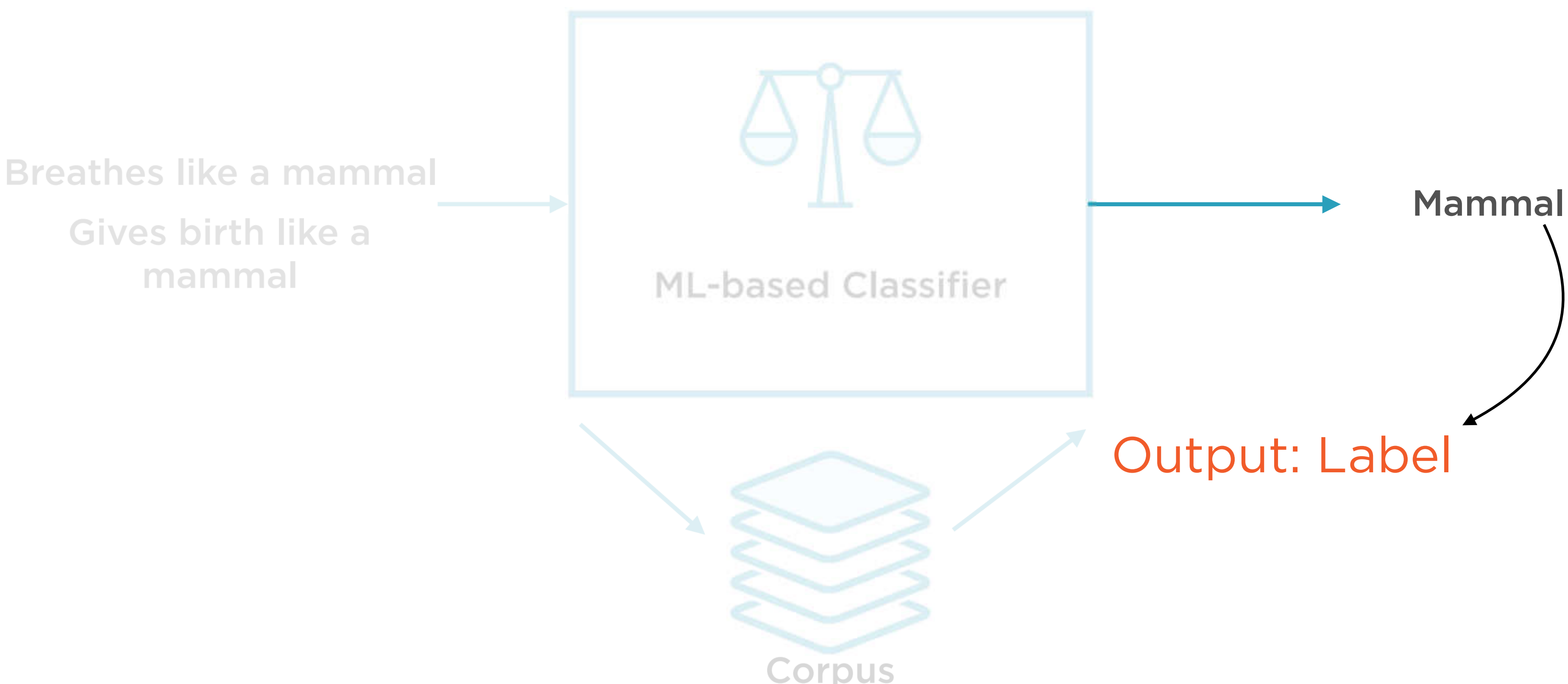
ML-based Binary Classifier



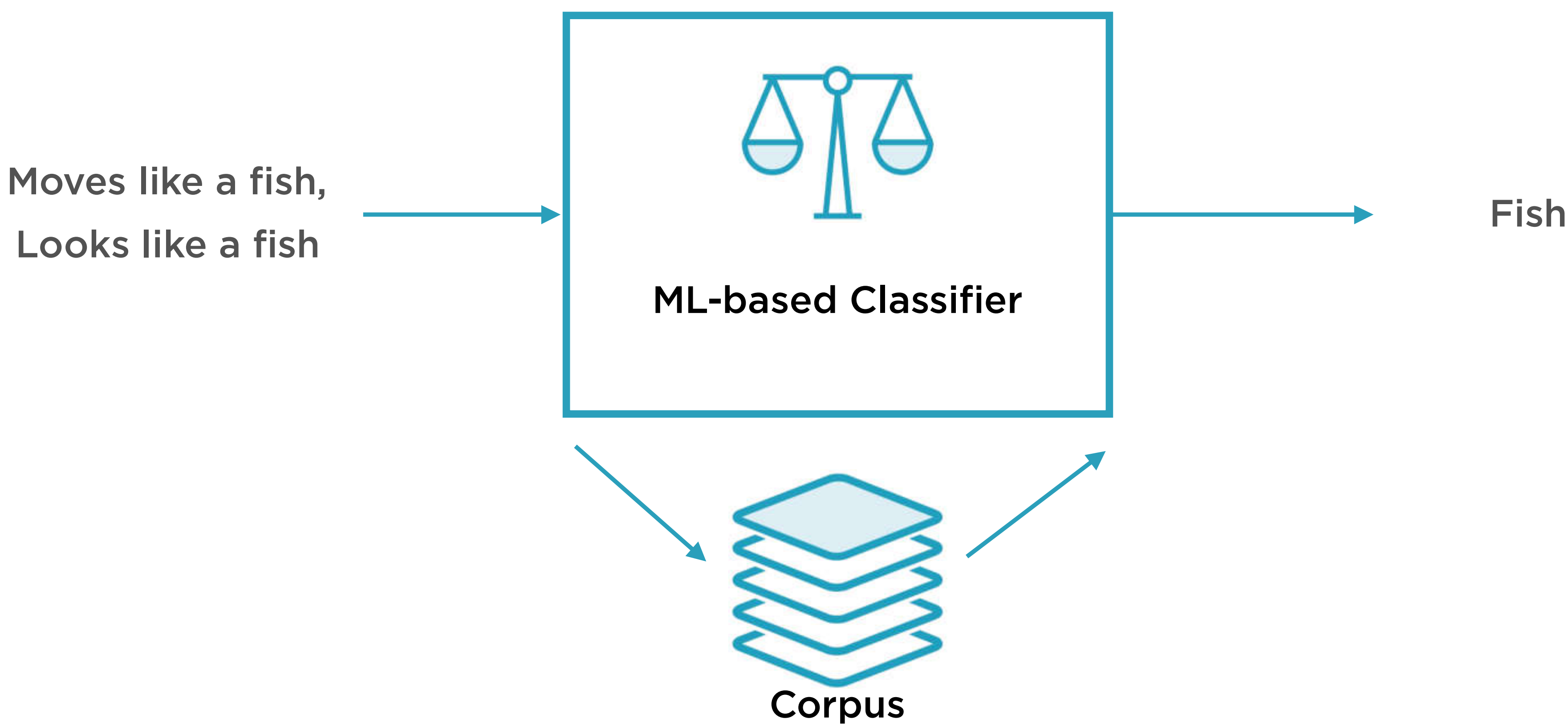
ML-based Binary Classifier



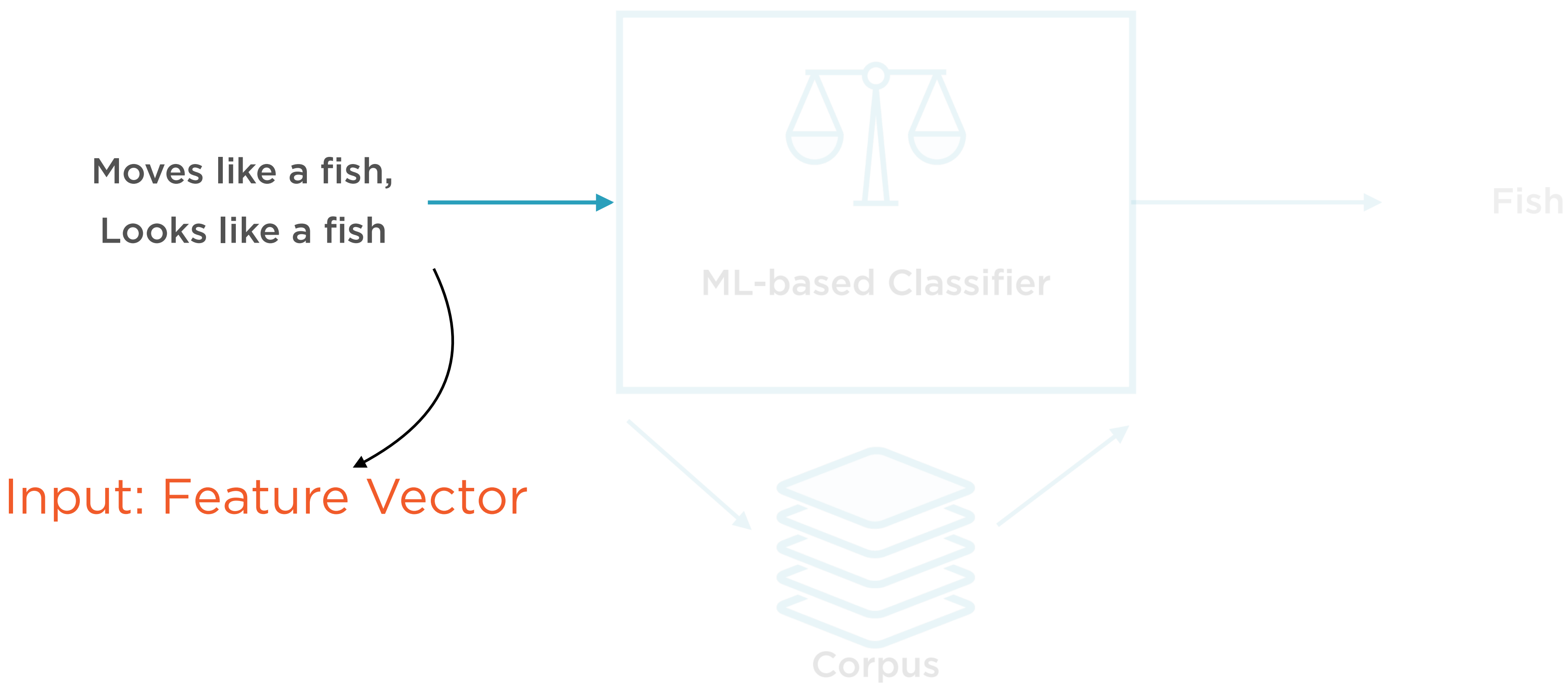
ML-based Binary Classifier



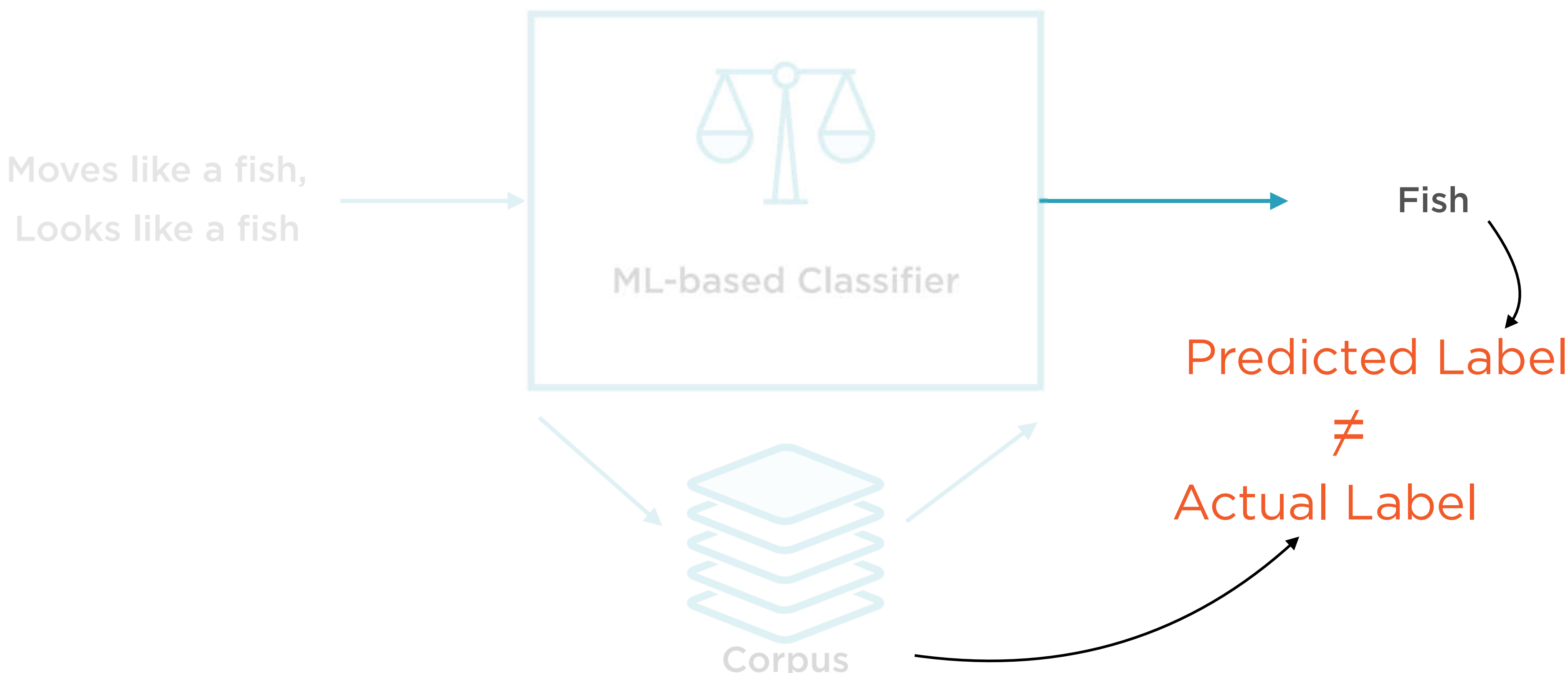
ML-based Binary Classifier



ML-based Binary Classifier



ML-based Binary Classifier



x Variables

The attributes that the ML algorithm focuses on are called features

Each data point is a list - or vector - of such features

Thus, the input into an ML algorithm is a feature vector

Feature vectors are usually called the x variables

y Variables

The attributes that the ML algorithm tries to predict are called labels

Labels are usually called the y variables

Types of labels

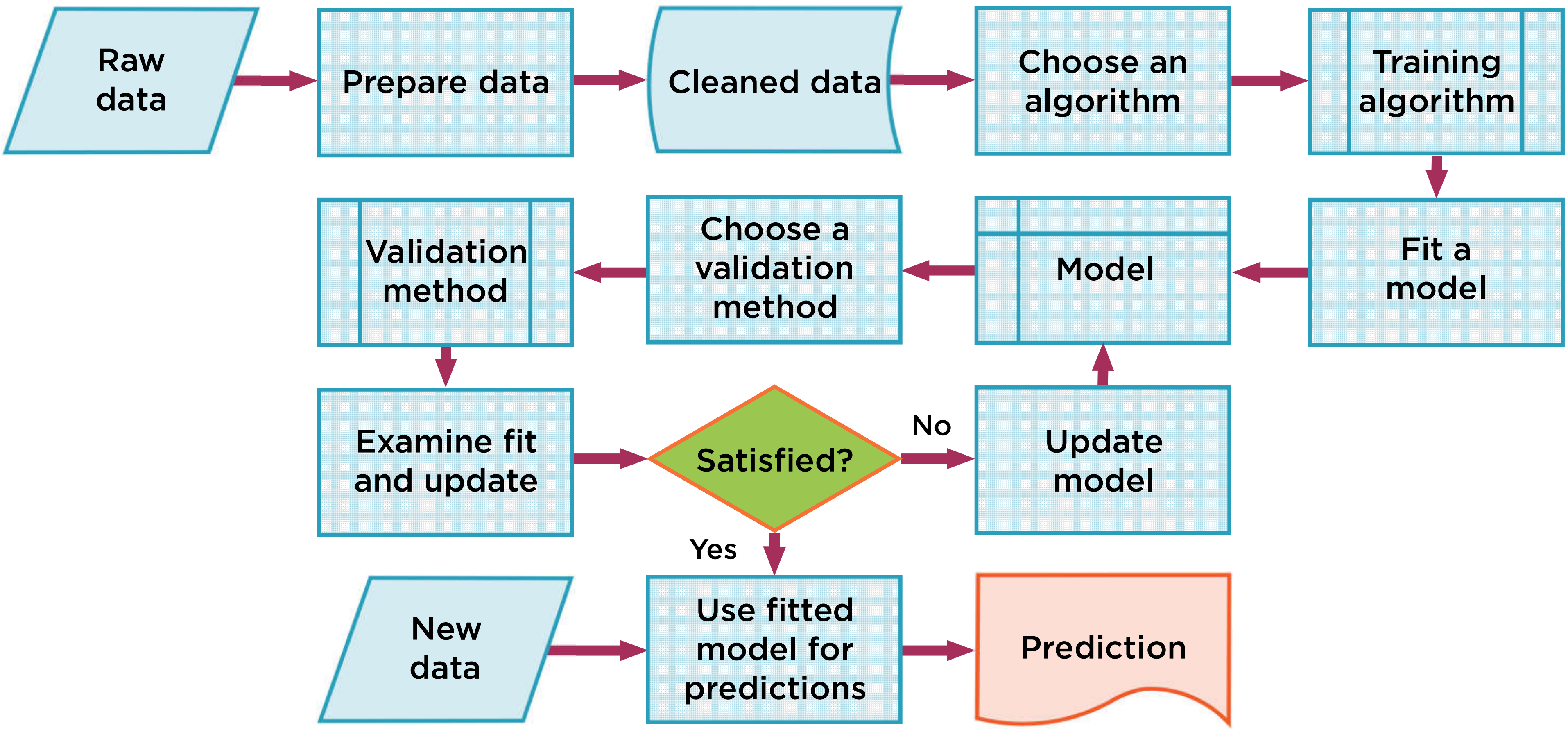
- categorical (classification)
- continuous (regression)

Garbage In, Garbage Out

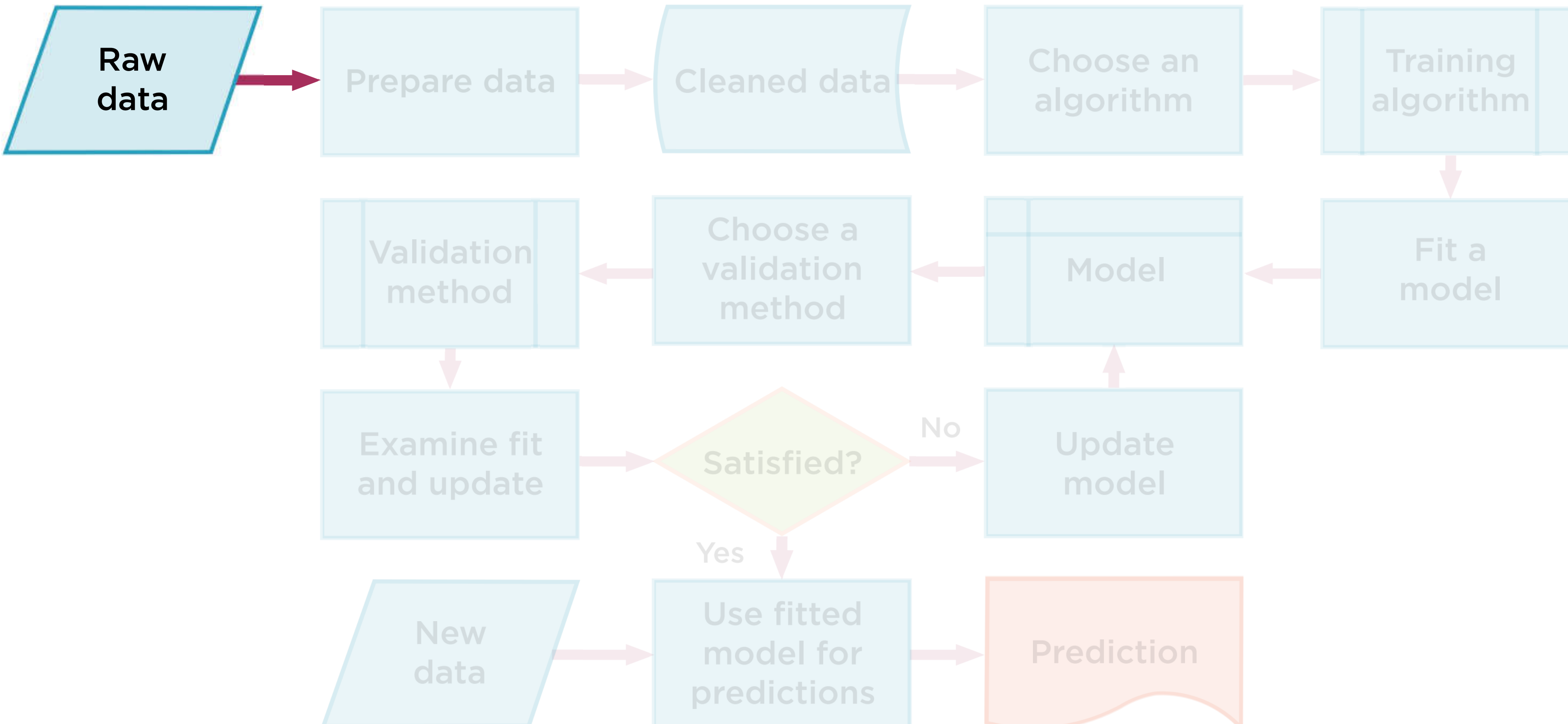
If data fed into an ML model is of poor quality, the model will be of poor quality

The Machine Learning Workflow

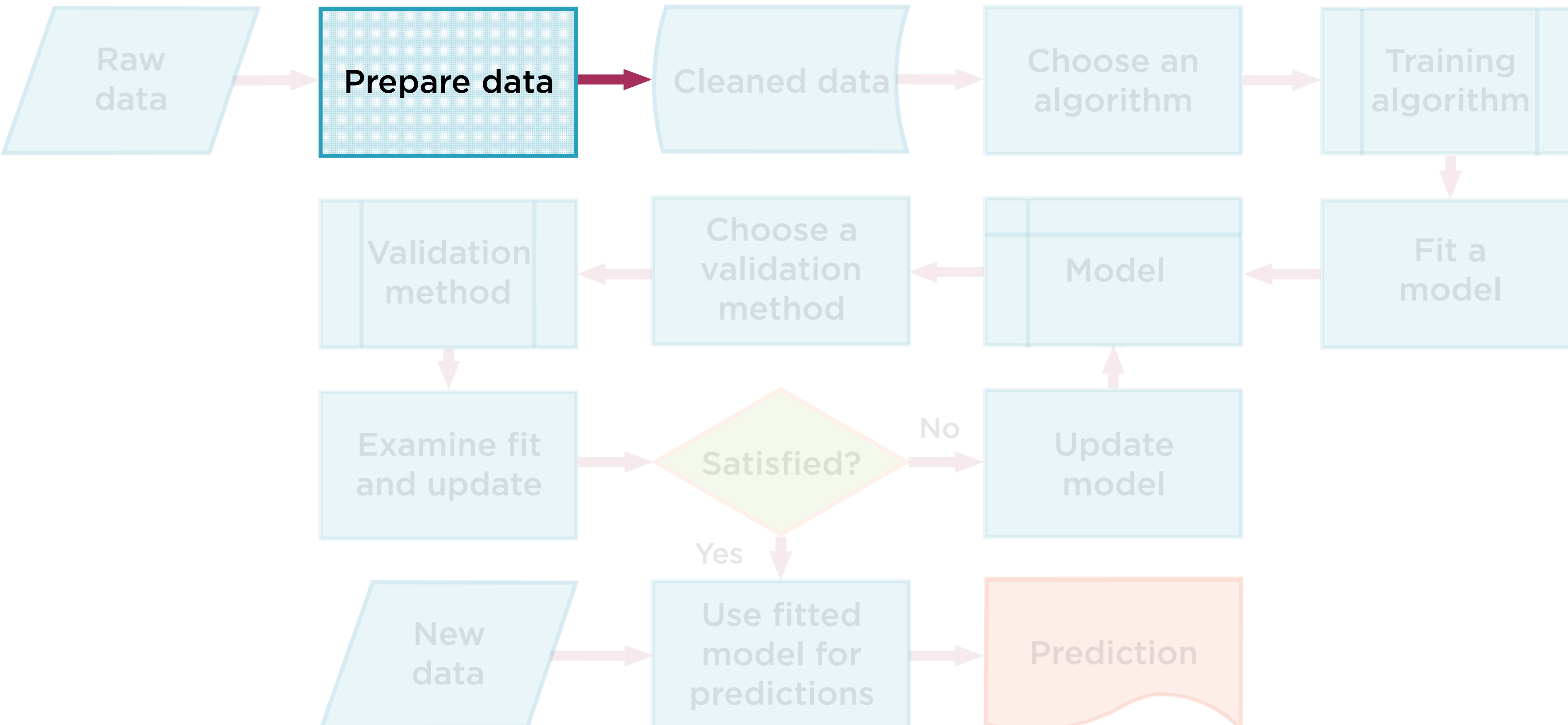
Basic Machine Learning Workflow



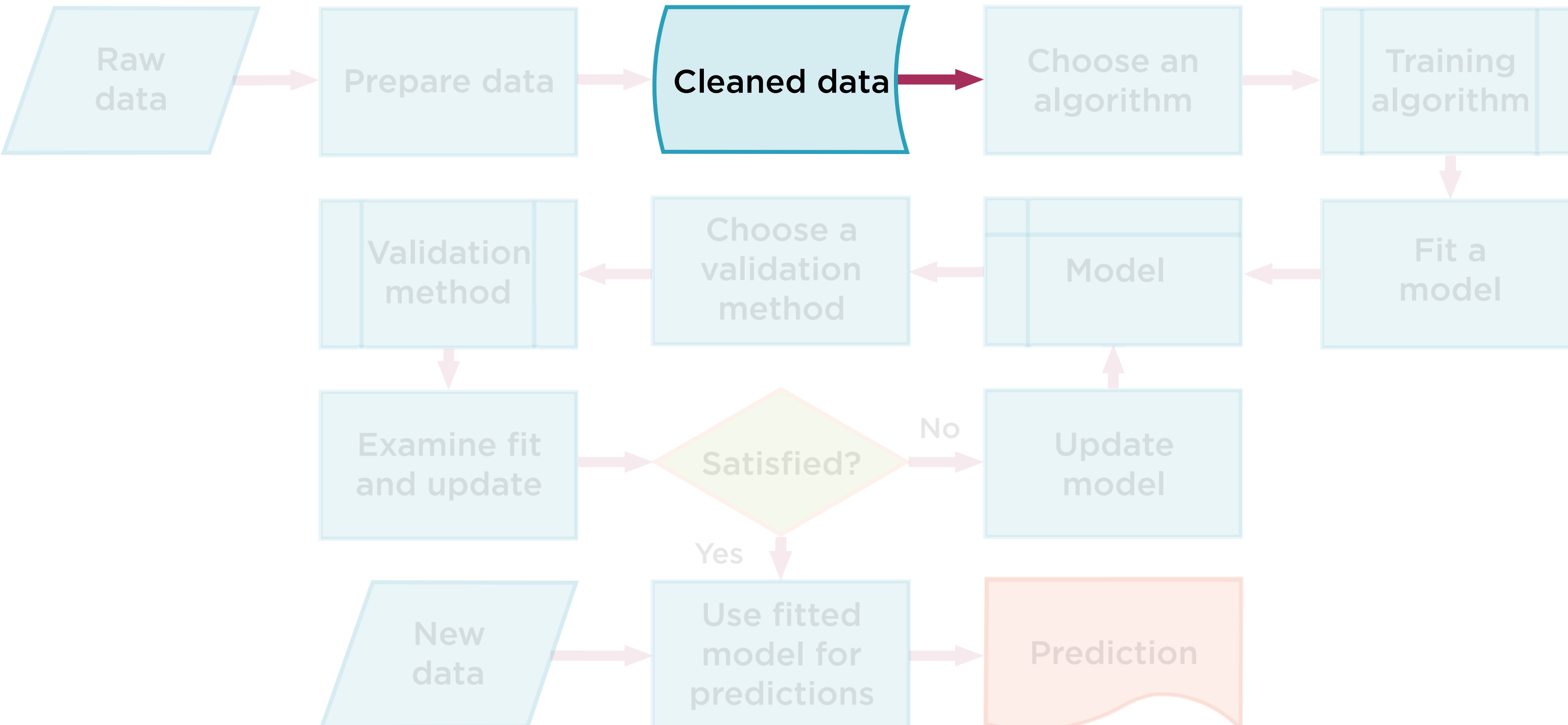
What Data Do You Have to Work With?



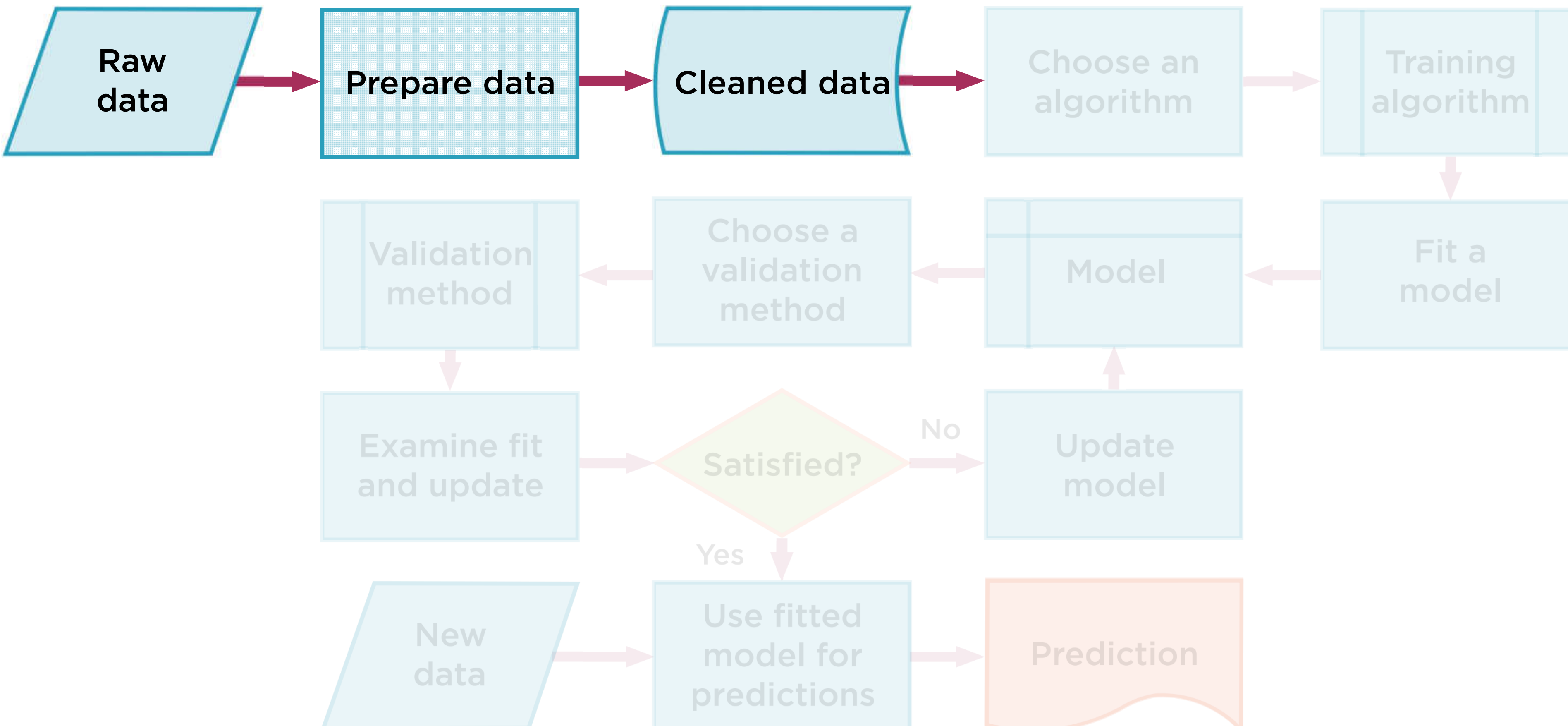
Load and Store Data



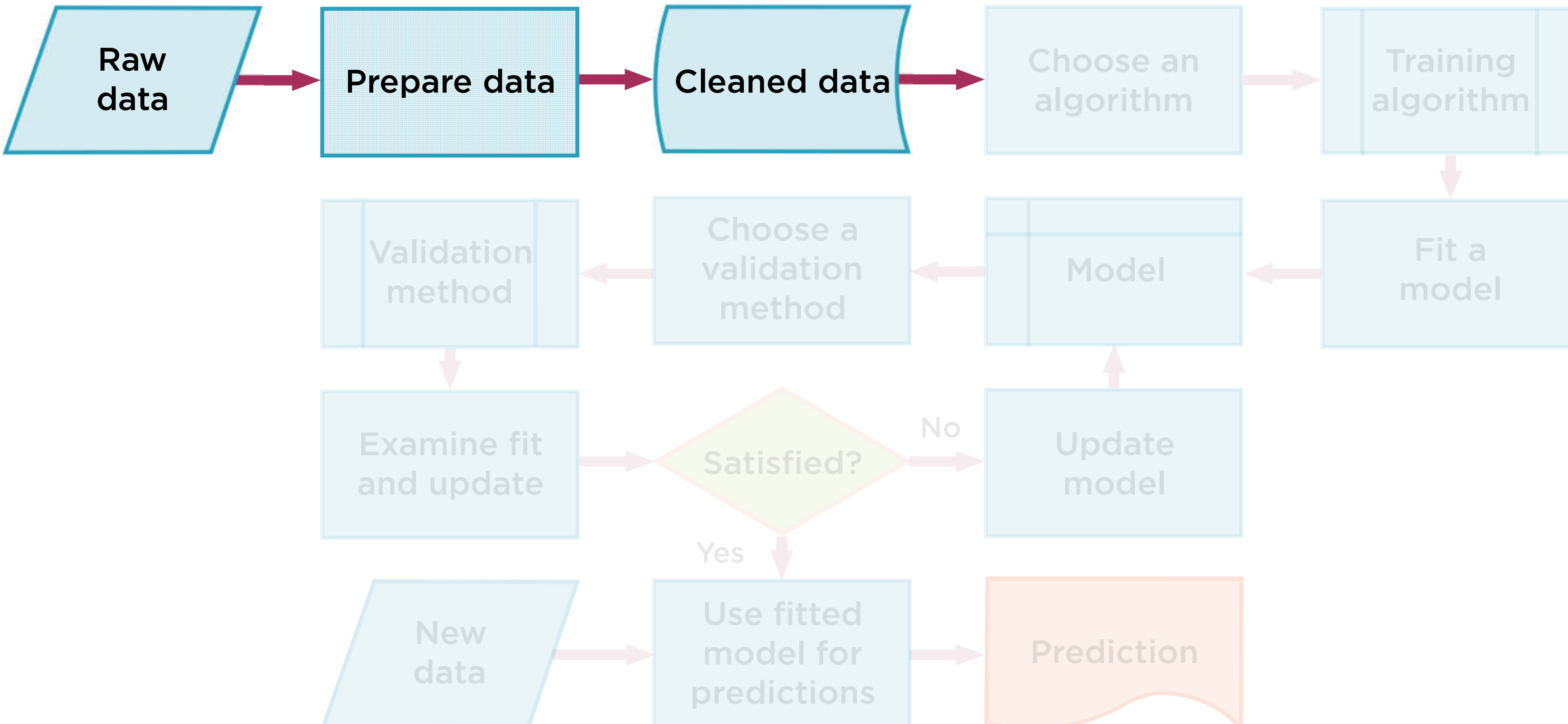
Data Preprocessing



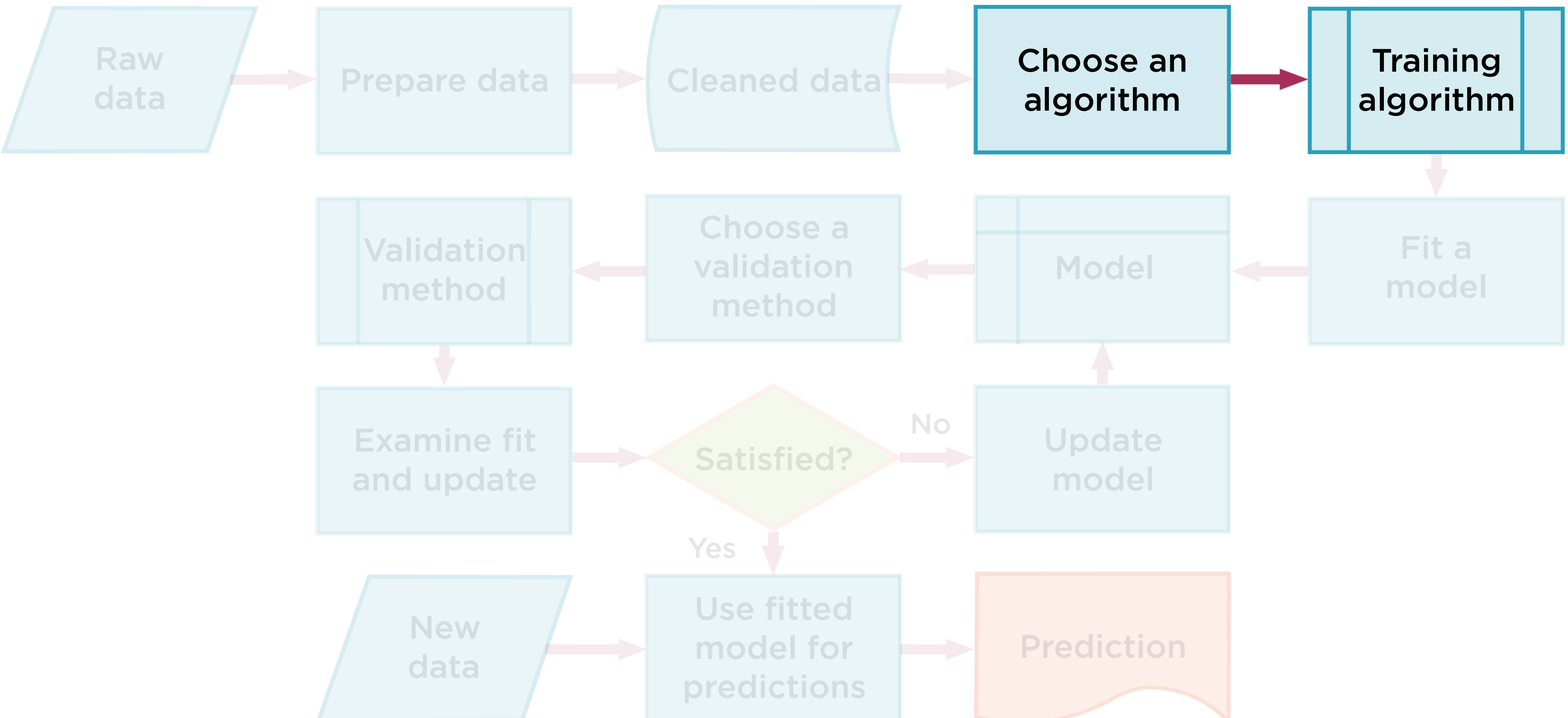
Selecting and Extracting Features



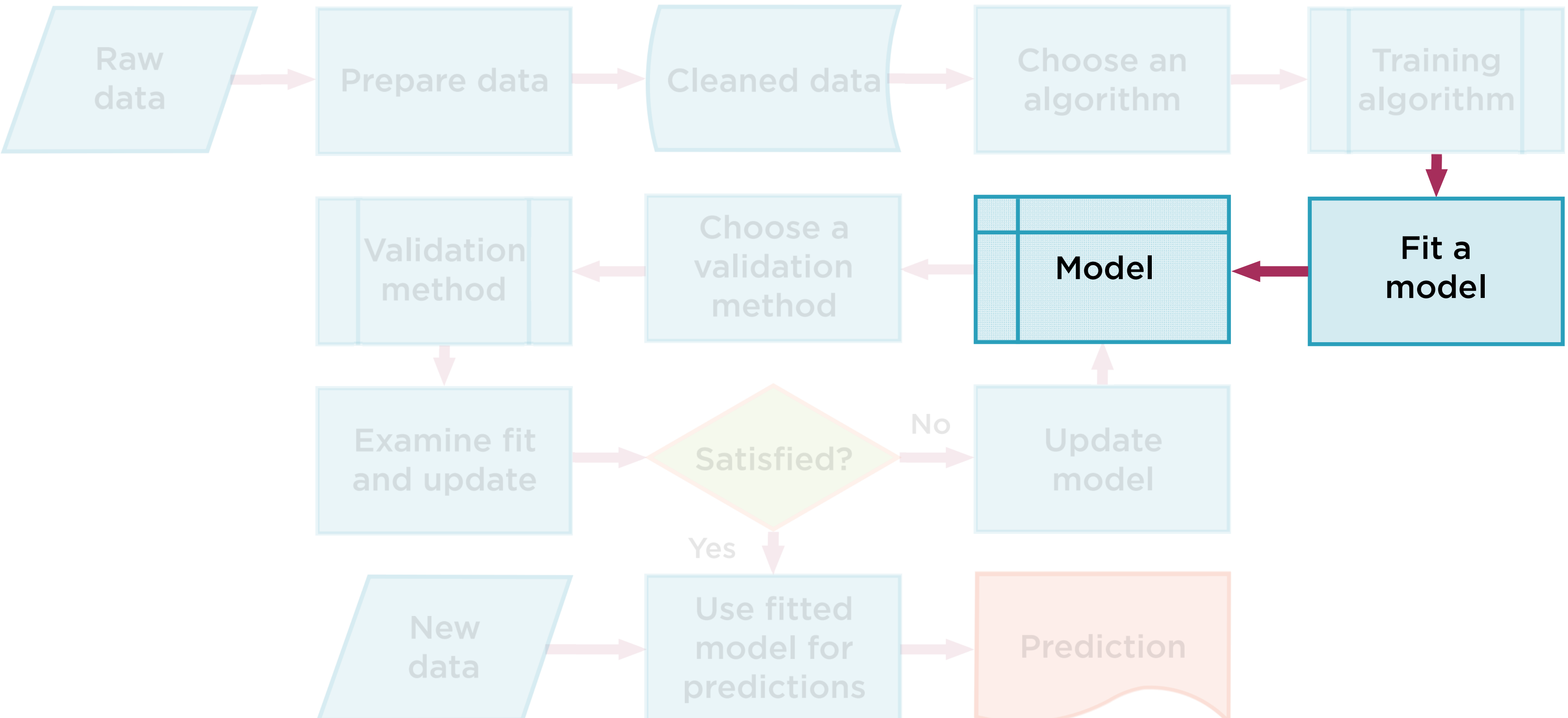
Critical and Time-consuming Steps



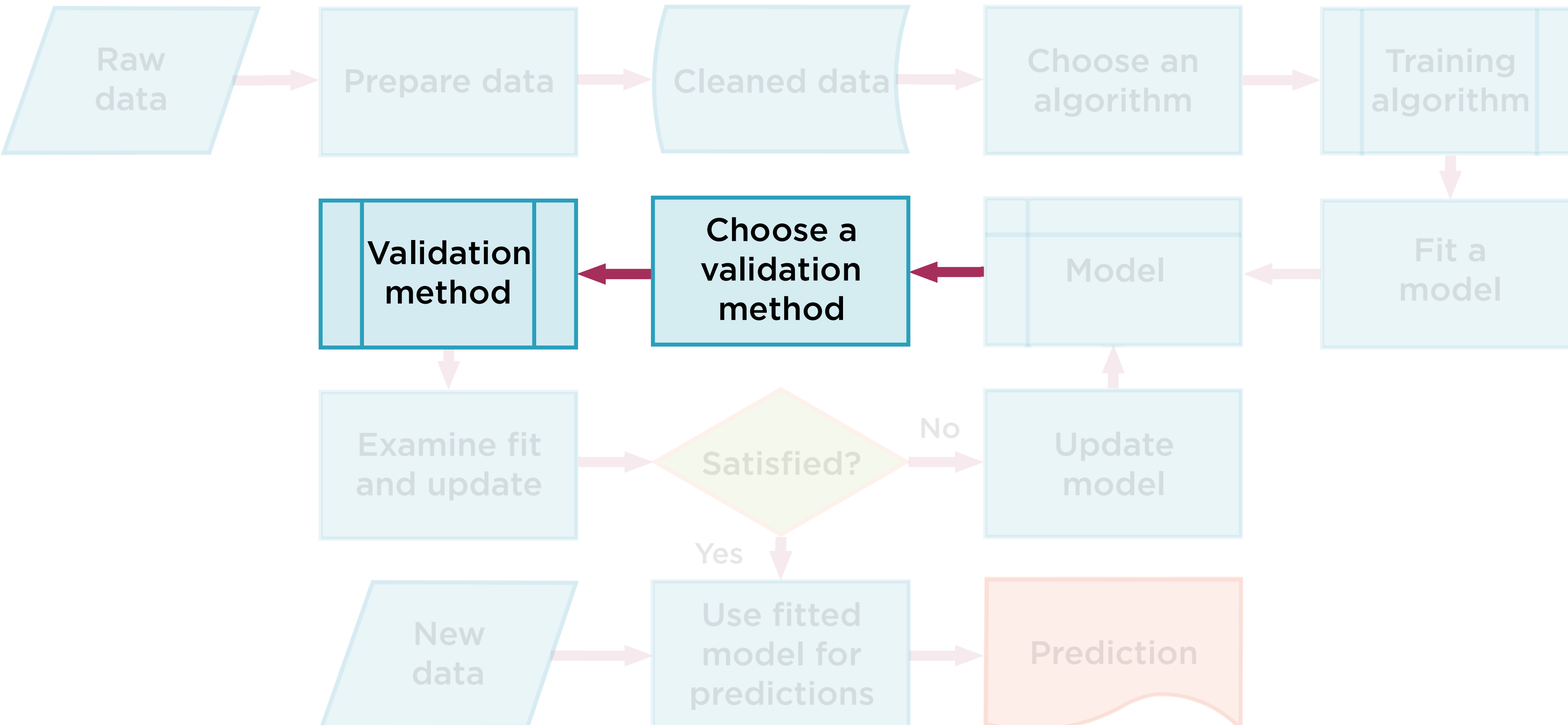
Decision Trees, Support Vector Machines?



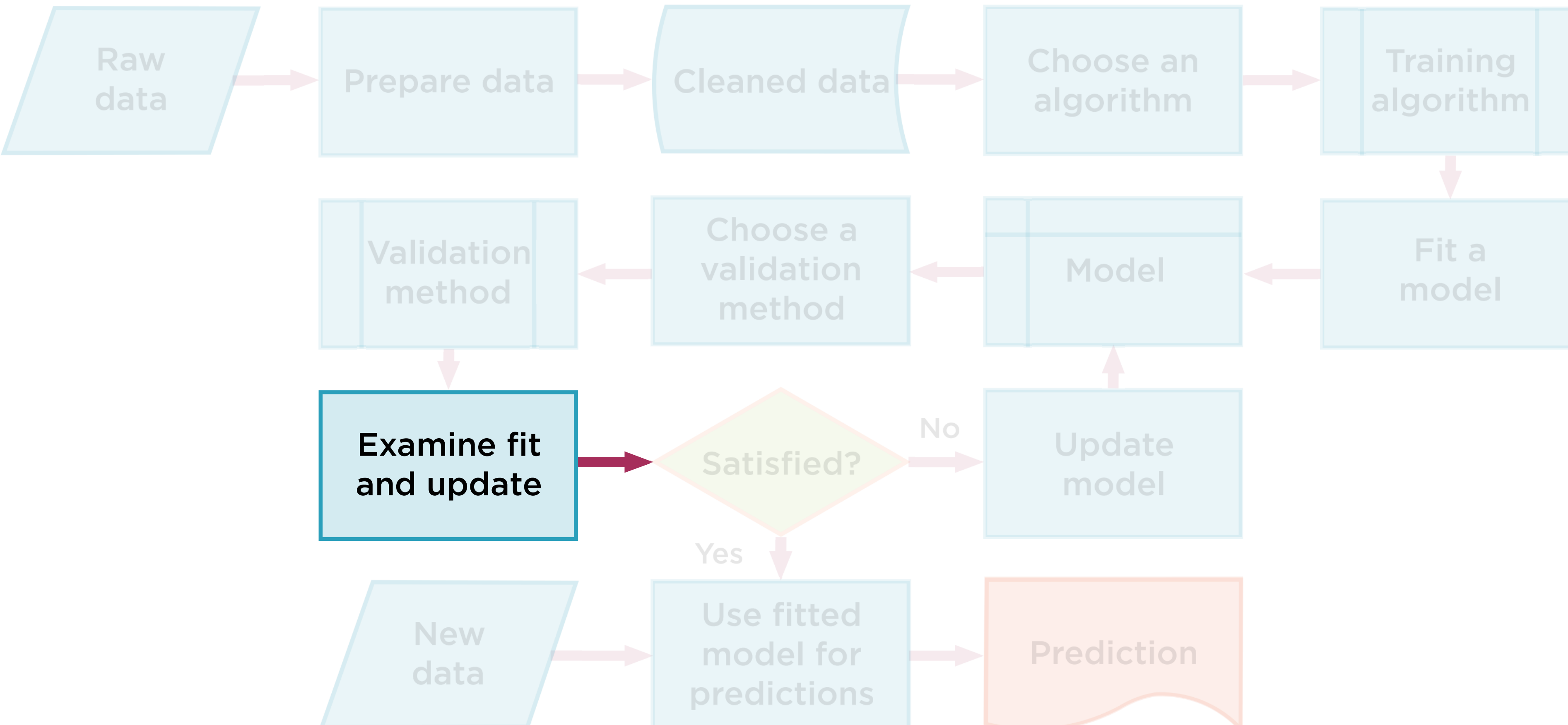
Training to Find Model Parameters



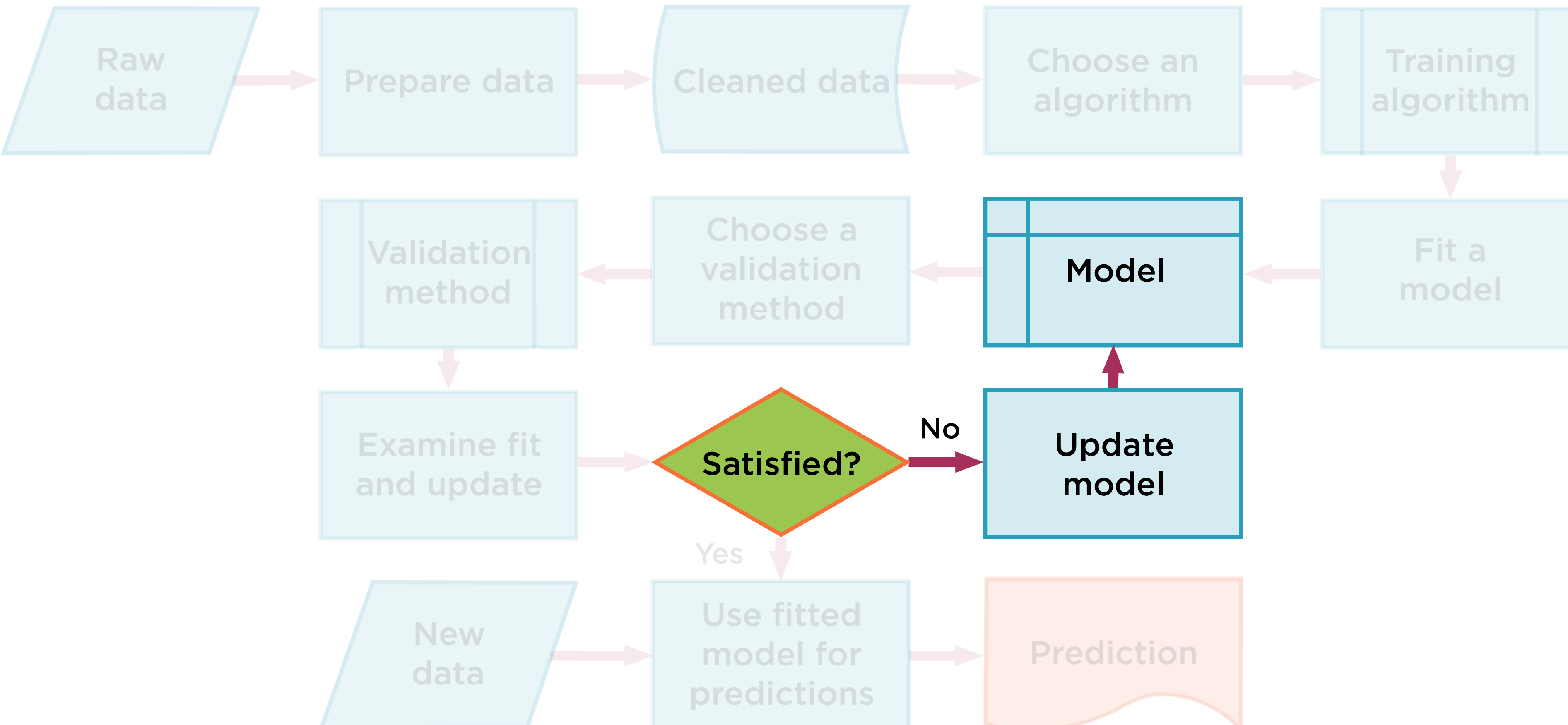
Evaluate the Model



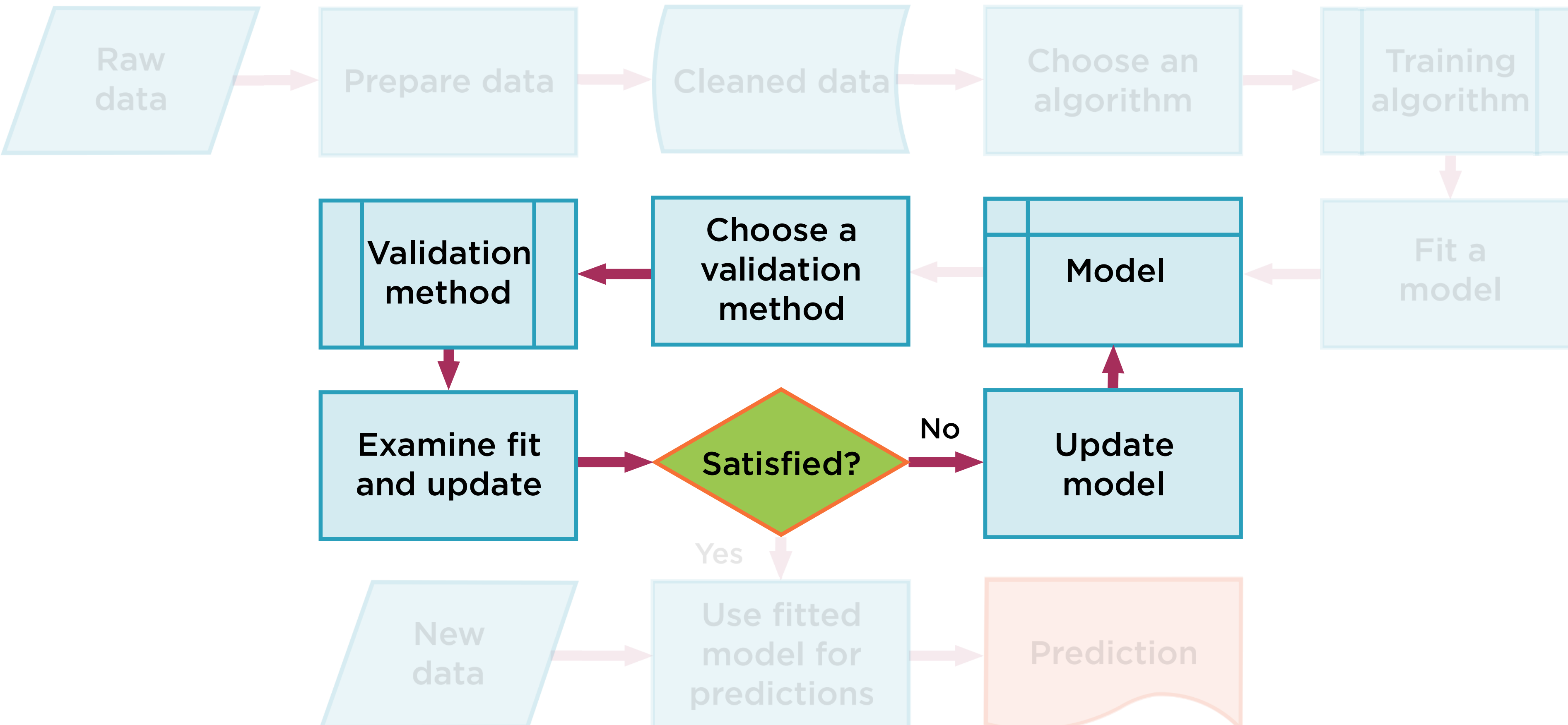
Score the Model



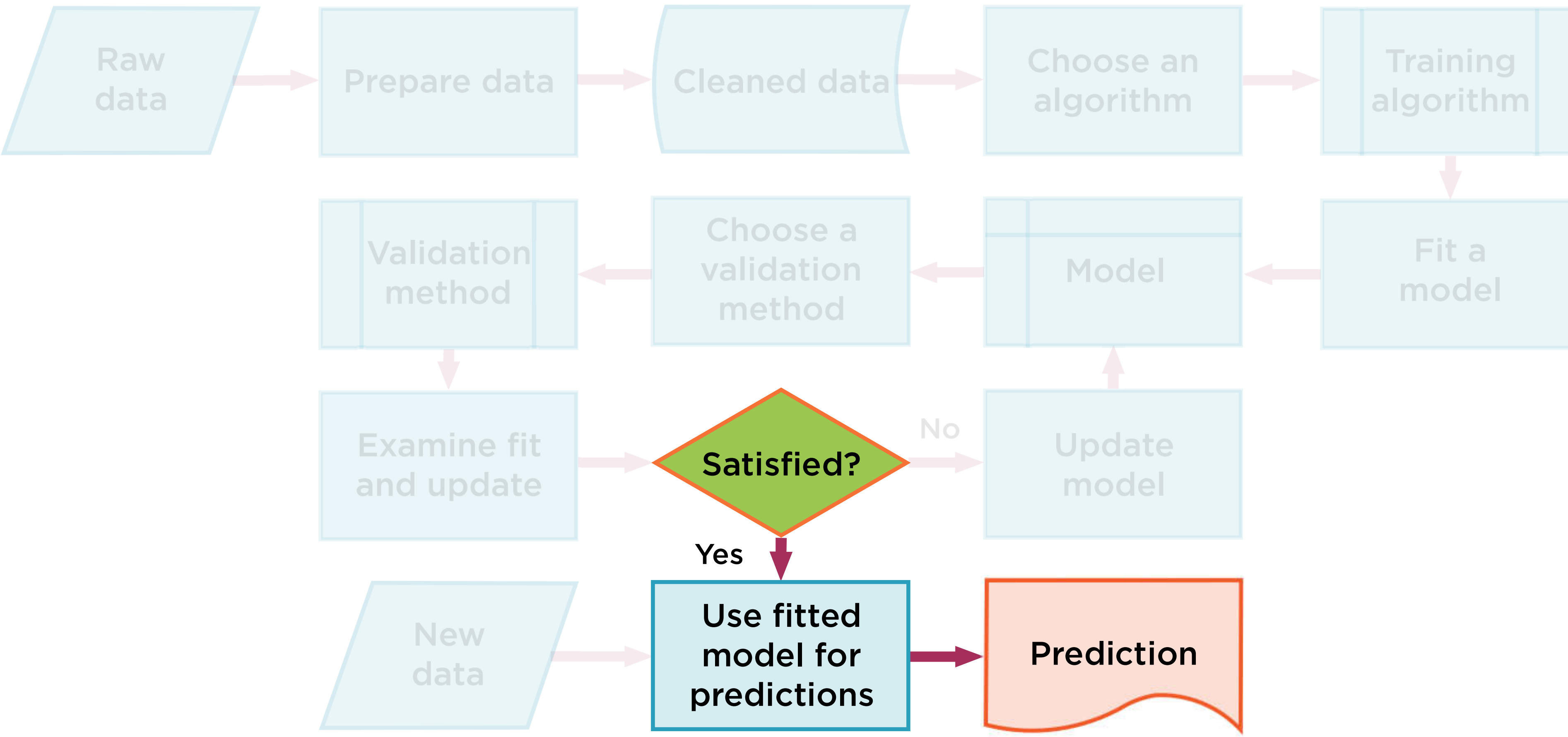
Different Algorithm, More Data, More Training?



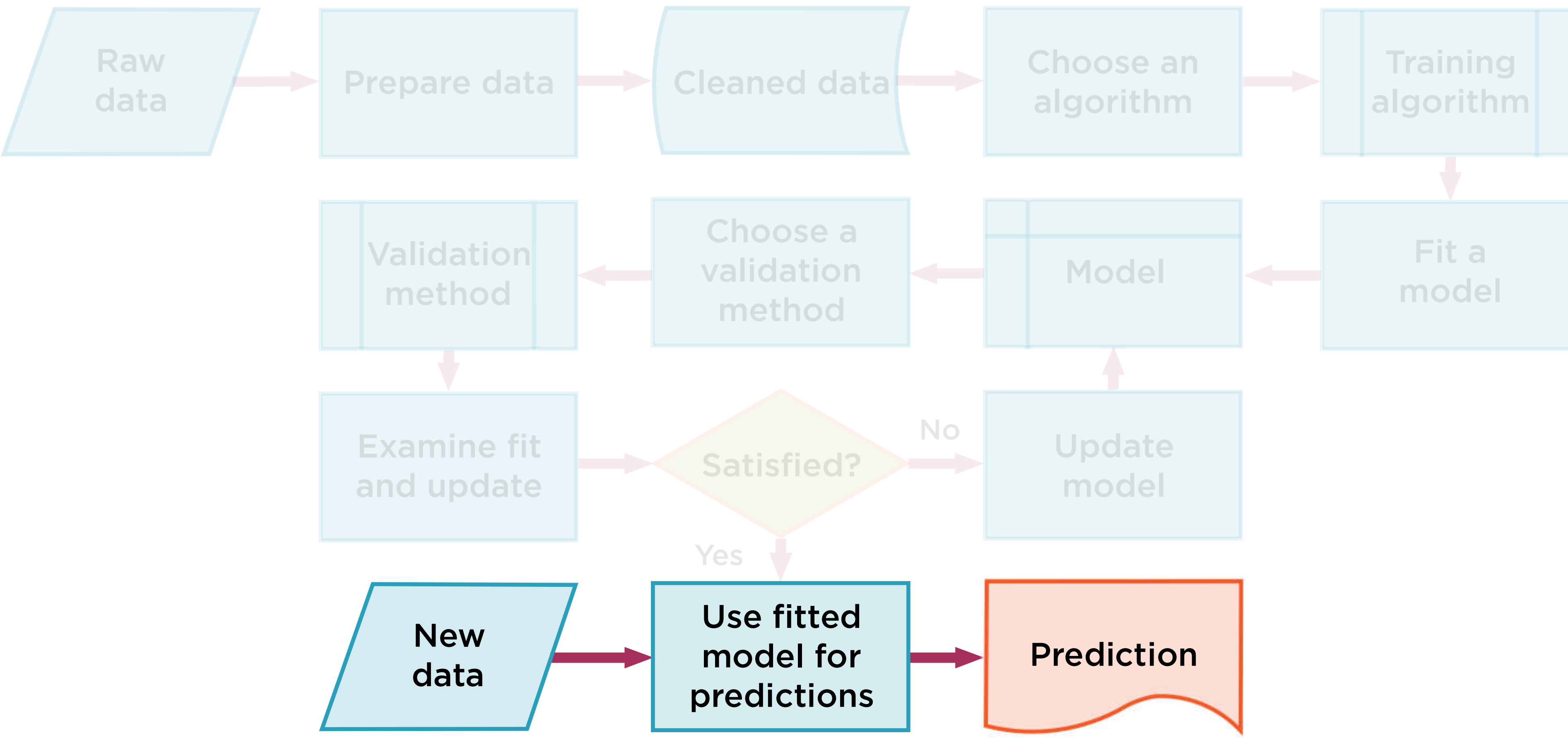
Iterate Till Model Finalized



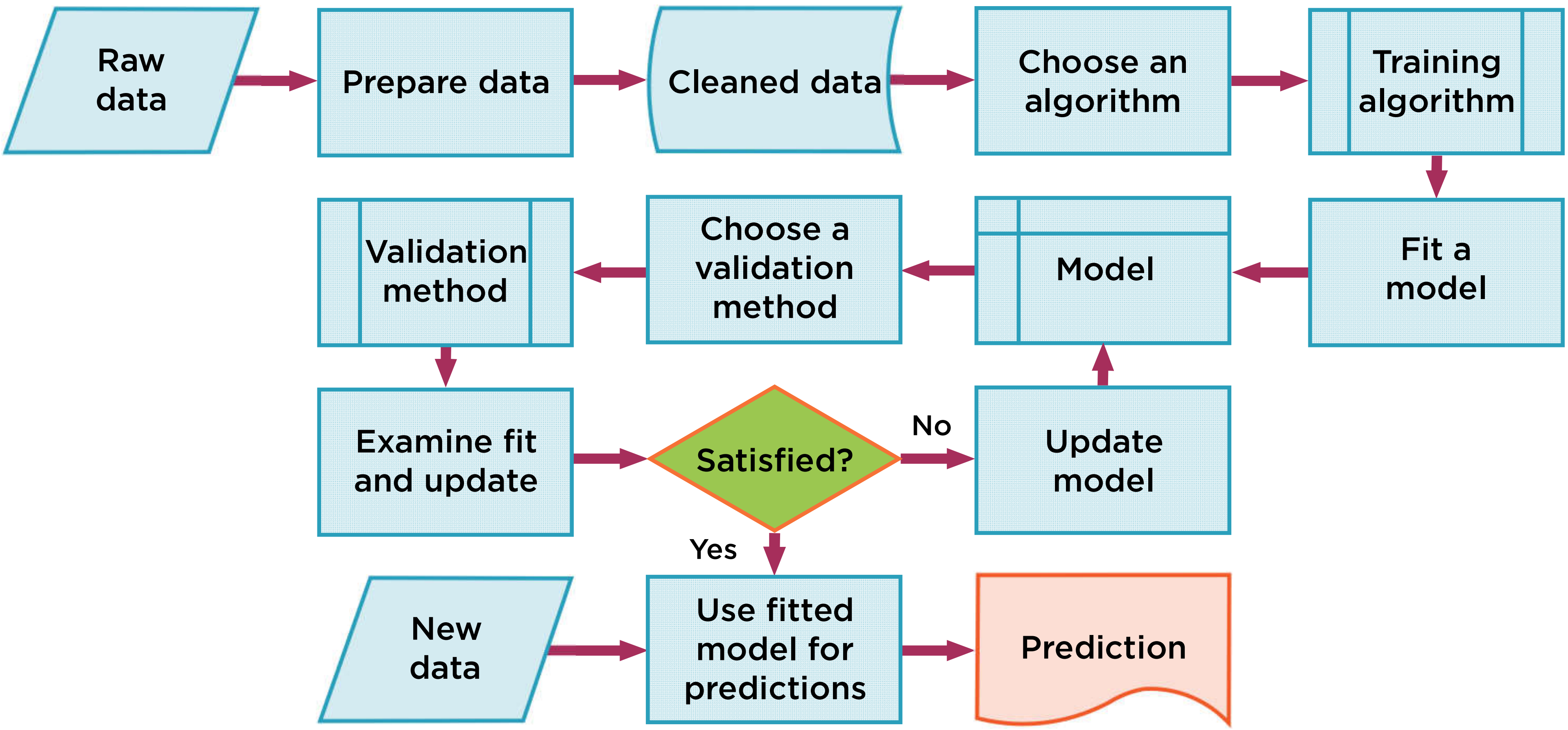
Model Used for Predictions



Retrained Using New Data

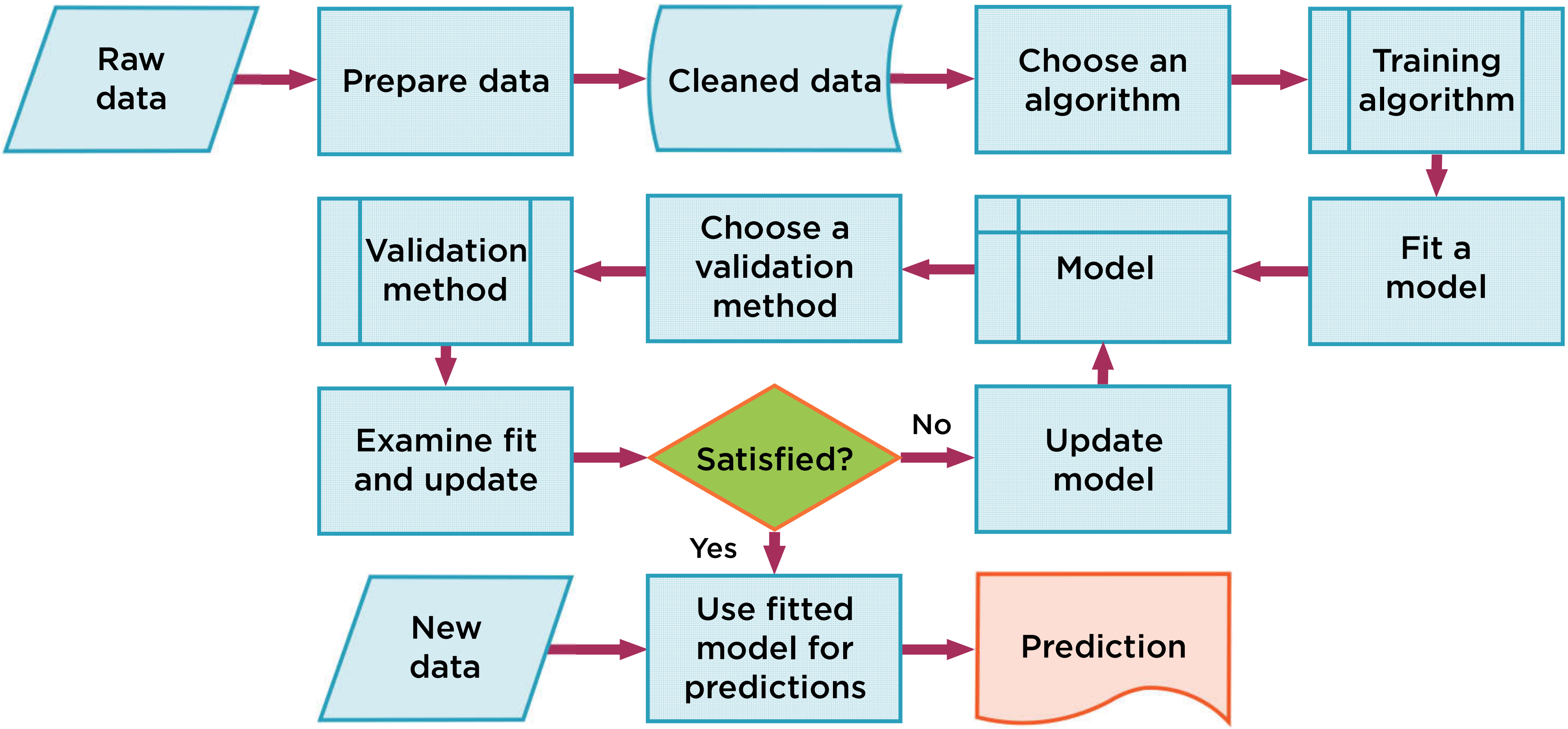


Basic Machine Learning Workflow

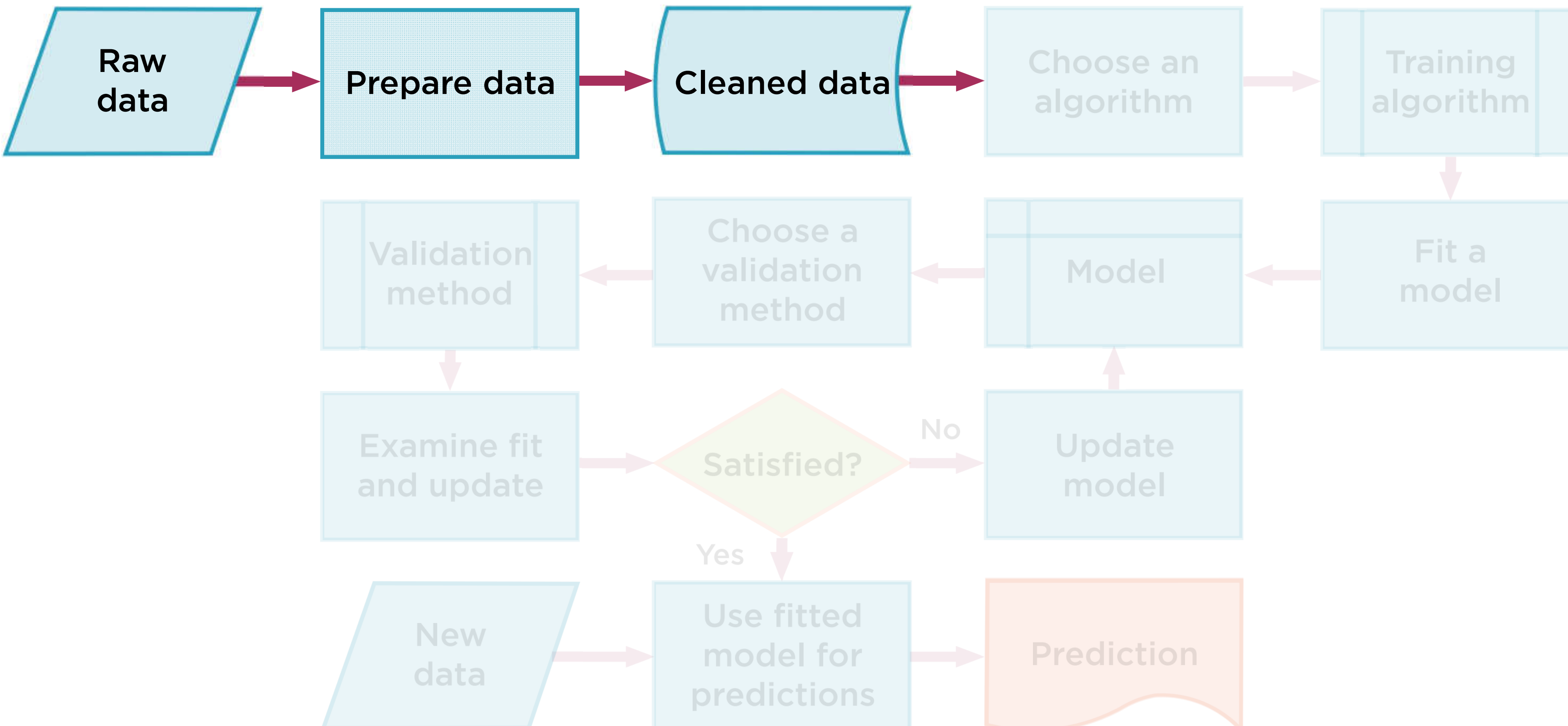


Feature Engineering

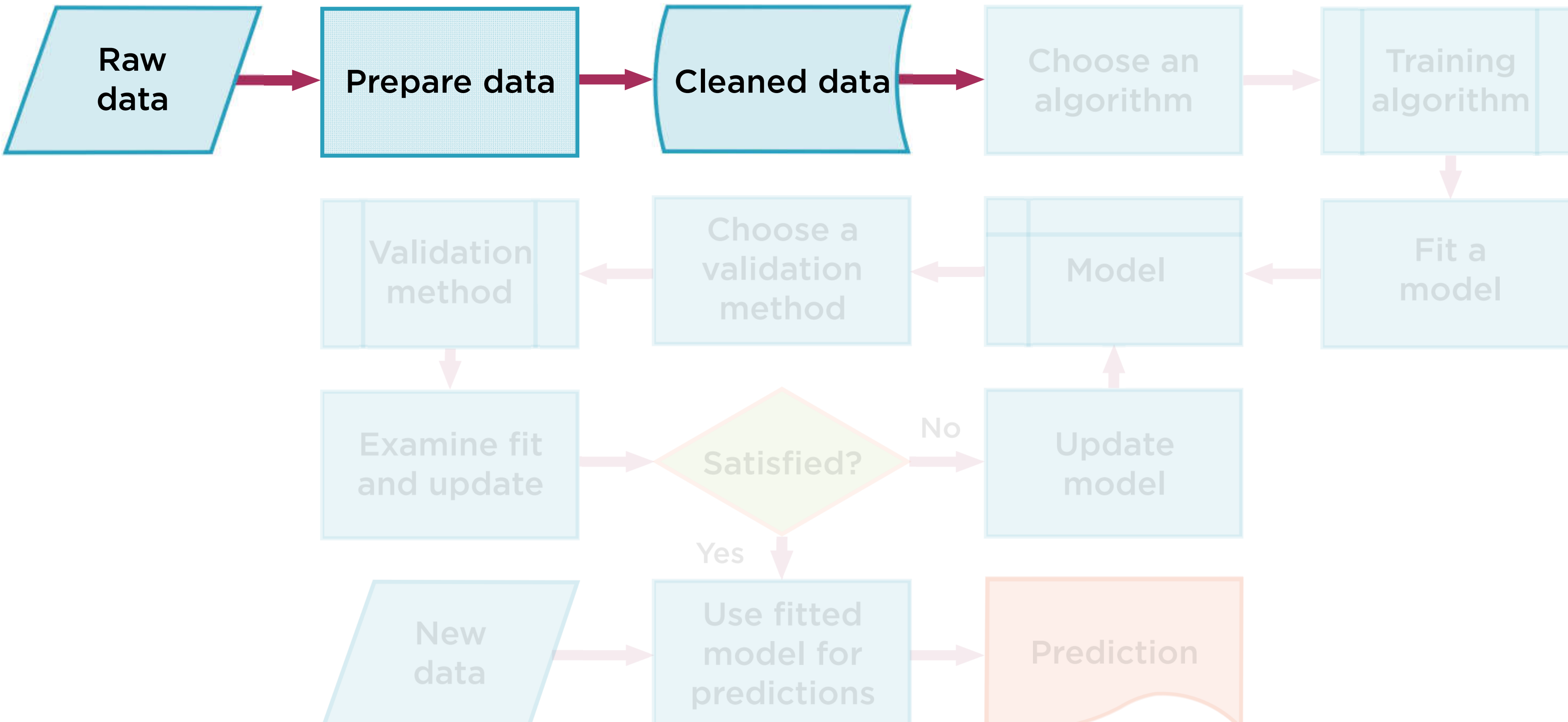
Basic Machine Learning Workflow



Selecting and Extracting Features



Feature Engineering



Feature Engineering

Engineering your features so that you get the best out of your ML model.

Feature Engineering

Block and tackle work

Bespoke - specific to:

- Problem
- Data

Not quite art, not quite science…

…More just engineering

Scope of Feature Engineering

- Feature selection
- Feature learning
- Feature extraction
- Feature combination
- Dimensionality reduction

Scope of Feature Engineering

Feature selection

Feature learning

Feature extraction

Feature combination

Dimensionality reduction

Feature Selection

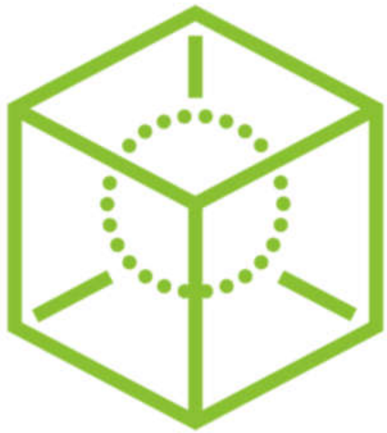
Choosing the best subset from within an existing set of features (x-variables), without substantially transforming them.

Choosing Feature Selection

Use Case	Possible Solution
Many X-variables	
Most of which contain little information	
Some of which are very meaningful	Feature selection
Meaningful variables are independent of each other	

Feature Selection Techniques

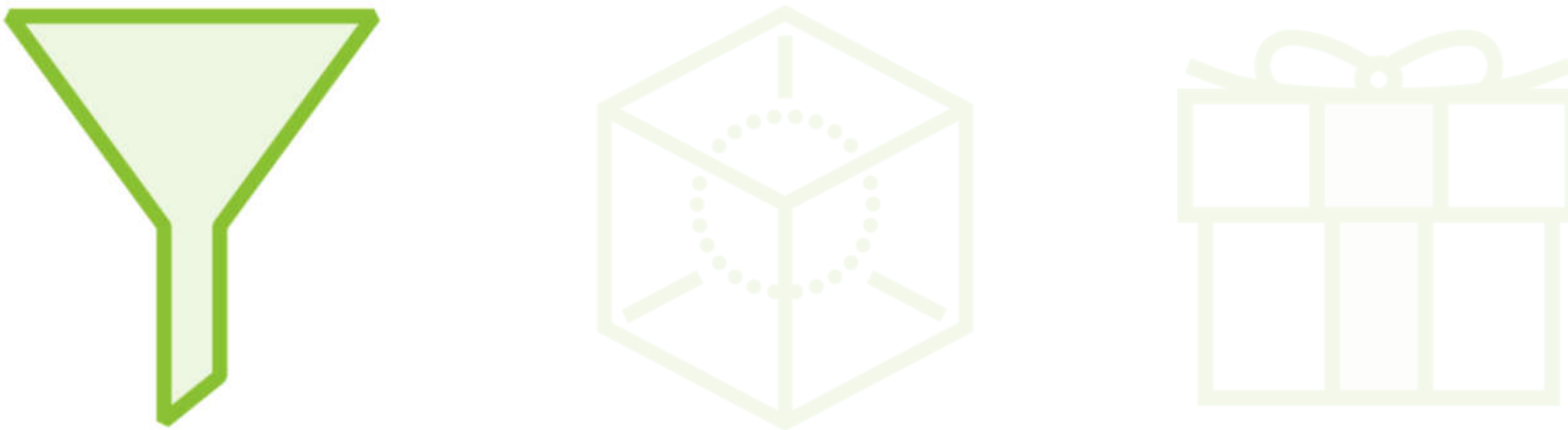
Filter methods



Embedded methods

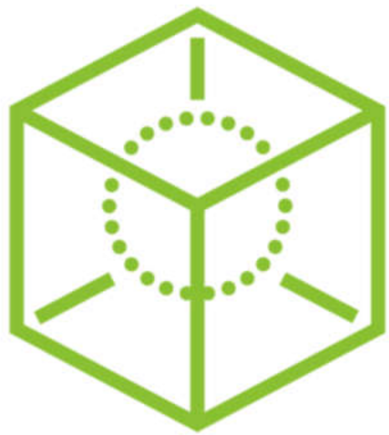
Wrapper methods

Filter Methods



Applying statistical techniques to select the most relevant features

Embedded Methods

Relevant features selected by training a machine learning model i.e. Lasso regression, decision trees

Wrapper Methods

Build candidate models by selecting feature subsets - choose the subset which gives the best model

Scope of Feature Engineering

- Feature selection
- **Feature learning**
- Feature extraction
- Feature combination
- Dimensionality reduction

Feature Learning

Rely on ML algorithms rather than human experts to "learn" the best representations of complex data such as images, videos.

(Also known as Representation Learning)

Supervised Feature Learning

Features are learnt using labeled data

Neural networks are classic example

Greatly reduce need for expert judgment

“Traditional” ML-based systems rely on experts to decide what features to pay attention to

“Representation” ML-based systems figure out by themselves what features to pay attention to

Neural networks are examples of such systems

Unsupervised Feature Learning

Features need to be learned in absence of labeled corpus

- Clustering
- Dictionary learning
- Autoencoders

Scope of Feature Engineering

Feature selection	Feature learning	**Feature extraction**
Feature combination	Dimensionality reduction	

Feature Extraction

Differs from feature selection in that input features are fundamentally transformed into derived features, which are often unrecognizable and hard to interpret.

Feature Extraction

Image descriptors for images

Principal components for matrices

Tf-Idf for documents

Feature Extraction

Feature extraction usually also leads to dimensionality reduction

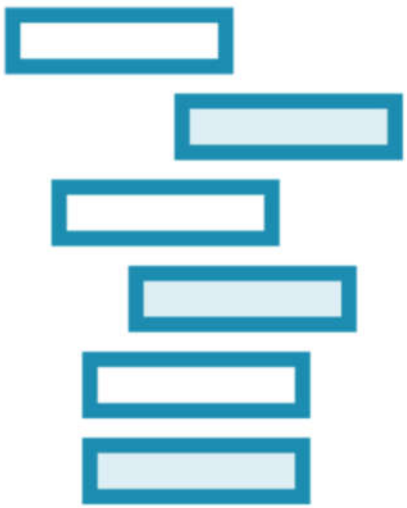
However explicit objective is to re-express feature in a “better” form

Not to reduce number of X columns

Scope of Feature Engineering

Feature selection	Feature learning	Feature extraction
Feature combination	Dimensionality reduction	

Feature Combination

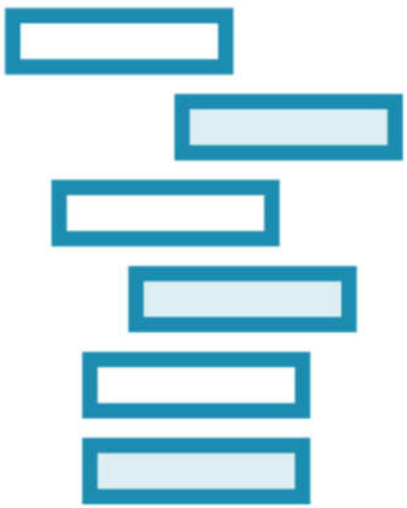


Some features naturally work better when considered together

Original feature might be raw or too granular

Improve the predictive power of features

Feature Combination



Feature cross in predicting traffic

- Day-of-week
- Time-of-day

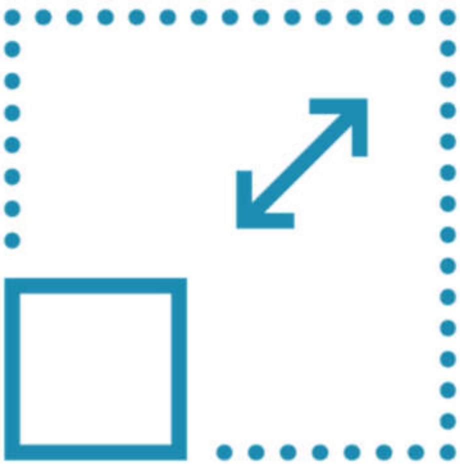
Feature cross in predicting temperature

- Season
- Time-of-day

Scope of Feature Engineering

- Feature selection
- Feature learning
- Feature extraction
- Feature combination
- Dimensionality reduction

Dimensionality Reduction



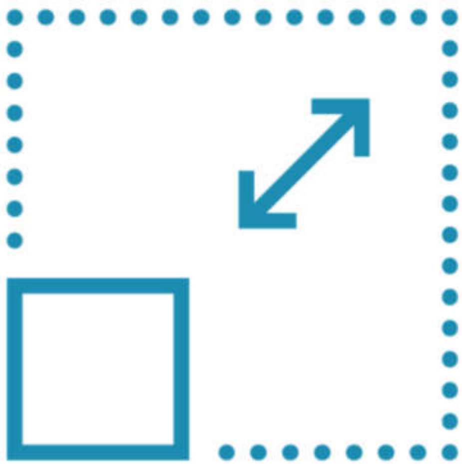
Apply pre-processing algorithms to reduce complexity of raw features

Specifically aim to reduce number of input features

Excessive number of features leads to severe problems

- Curse of Dimensionality

Dimensionality Reduction

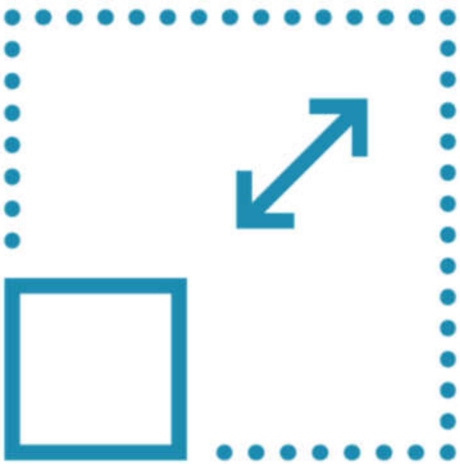


Dimensionality reduction explicitly aim to solve Curse of Dimensionality

While also preserving as much information as possible

Form of unsupervised learning

Dimensionality Reduction



Principle Components Analysis (PCA)

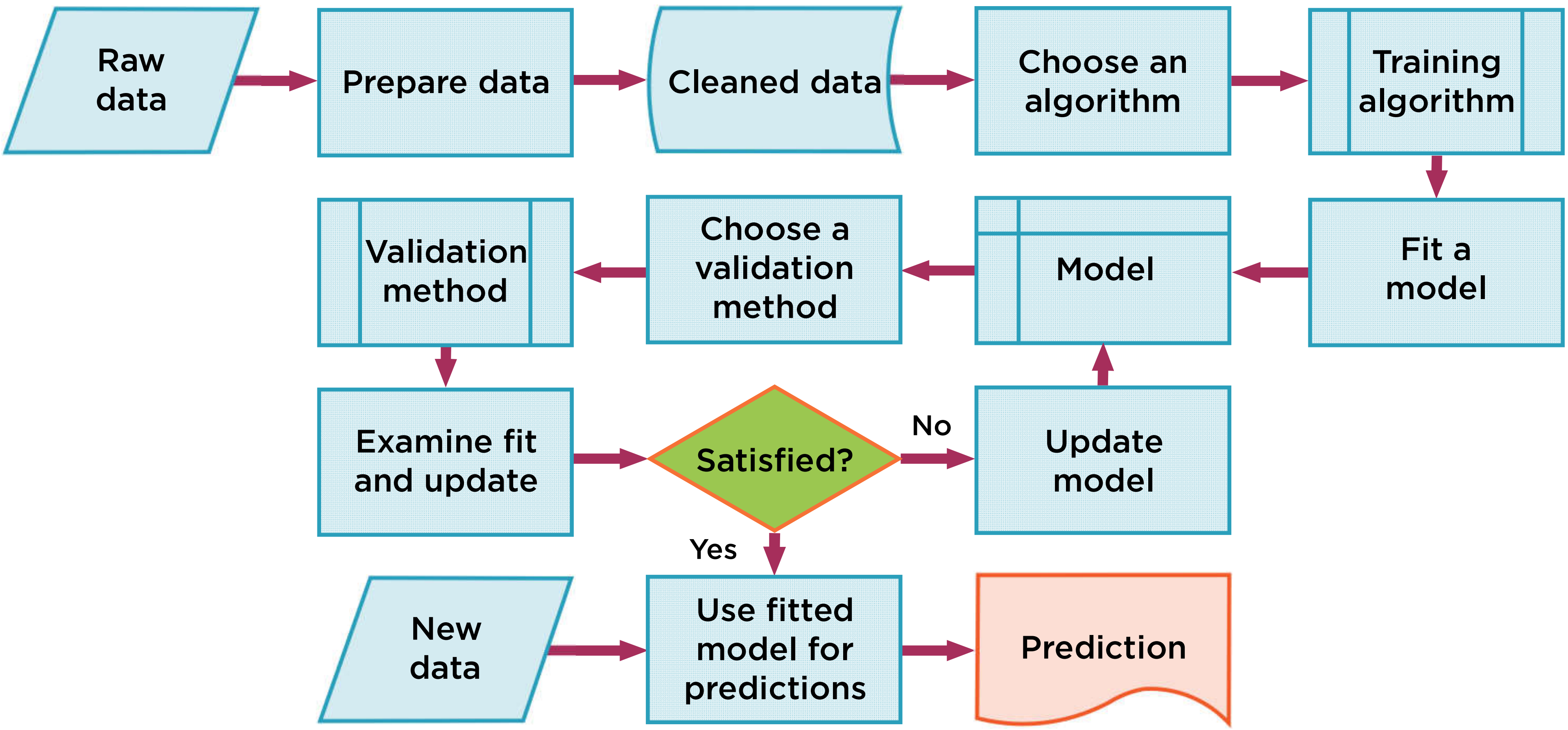
Manifold Learning

Latent Semantic Analysis

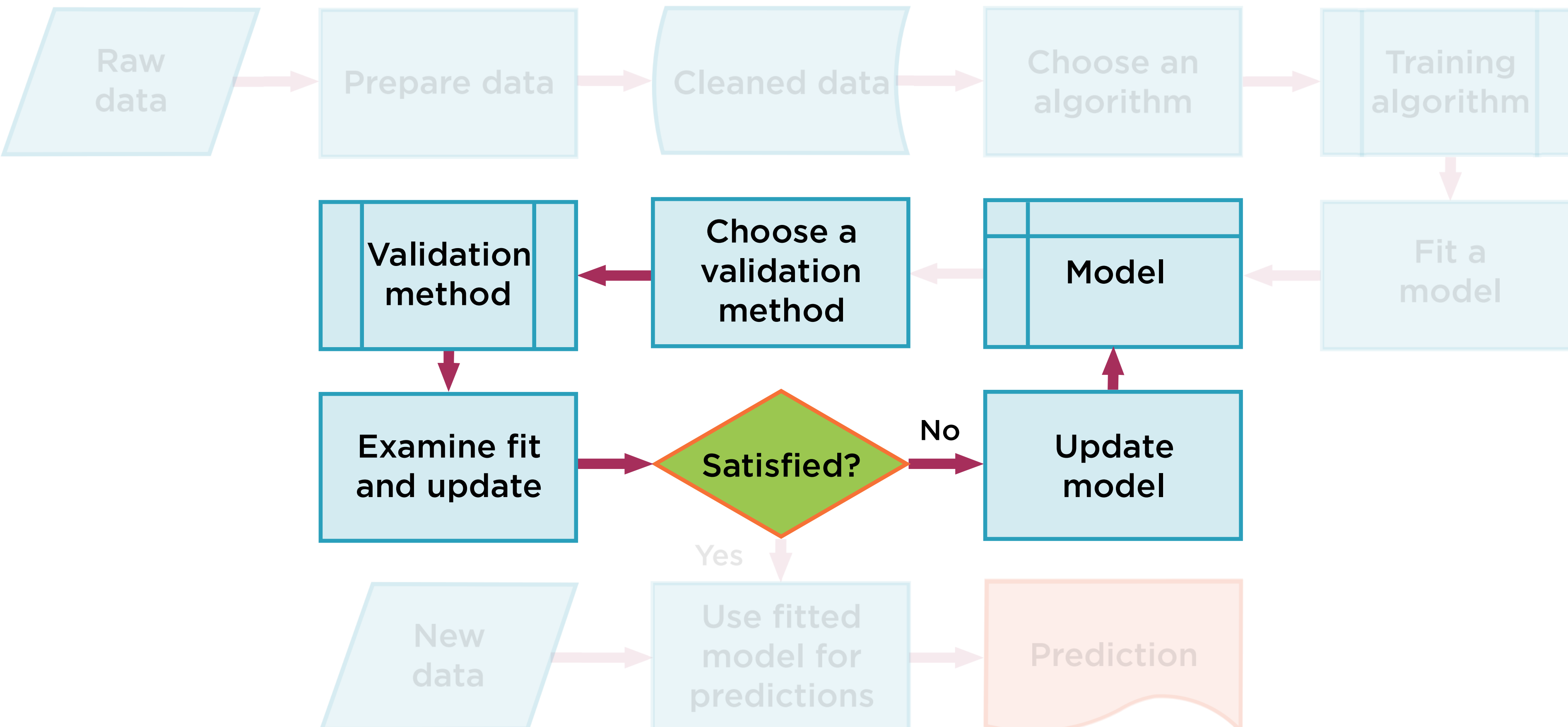
Autoencoding

Training, Test and Validation Data

Basic Machine Learning Workflow



Validate and Iterate Till Model Finalized



Data

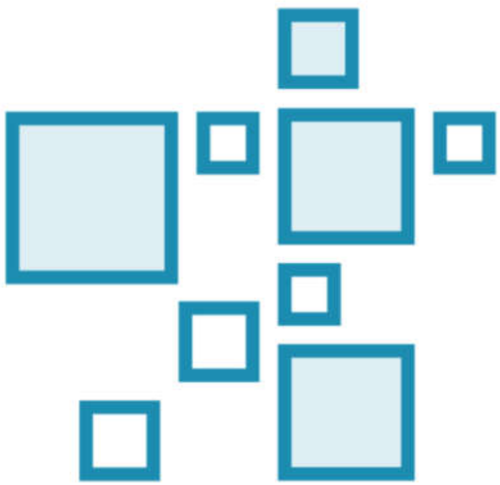
All data

All the data available

Training Data

Use all data to train your model

Training Data



Data used to train a model cannot be used to evaluate a model

Model may have memorized training instances

Model robustness cannot be measured on instances it has seen before

Training Data, Test Data

Typically 80% of the data used to train the model

Training Data, Test Data

20% set aside to sanity-check or measure model performance

Training Data, Test Data

One training process to generate one candidate model

Training Data, Test Data

For N candidate models, run N training and N test processes

Training Data, Test Data

Test set can be used to choose the best candidate model

Model evaluation on instances the model has not seen during training

Evaluation can become biased

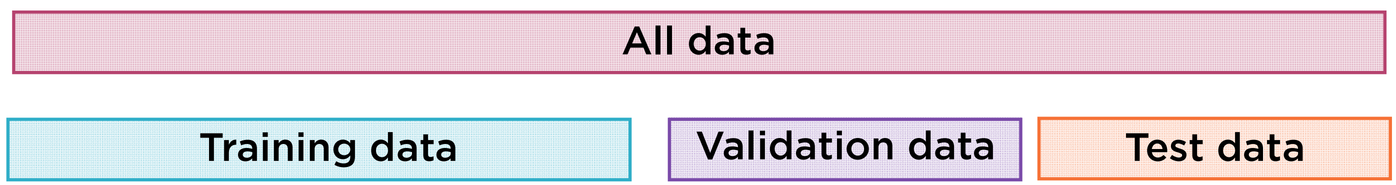
Overfitting on Test Set

Choosing best candidate model on the Test Set leads to this form of overfitting. Occurs when data is split into just two sets: Training and test.

Cross-validation

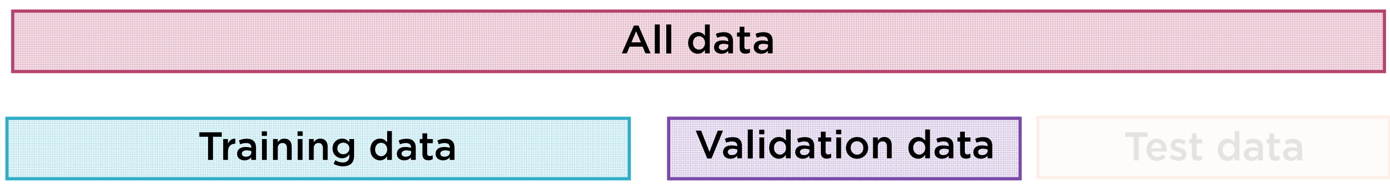
Carve out a separate validation set of data points; use this to evaluate different candidate models. Data now split into three sets: Training, validation and test.

Training, Test, Validation Data



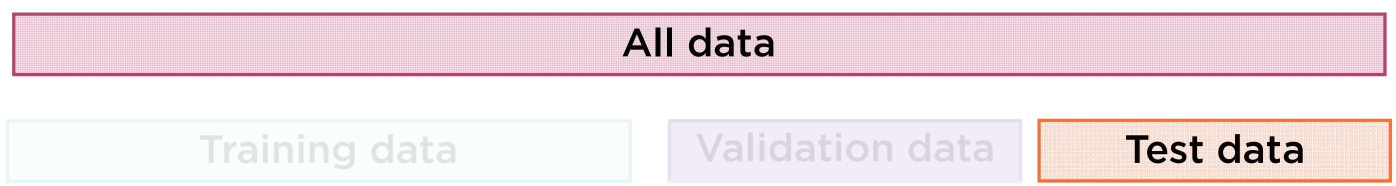
Hold out 2 subsets of the original data, validation data and test data

Training, Test, Validation Data



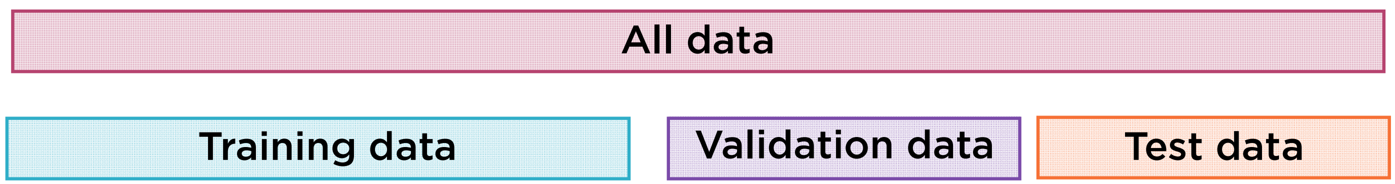
Training data to produce candidate models - validation data to evaluate models

Training, Test, Validation Data



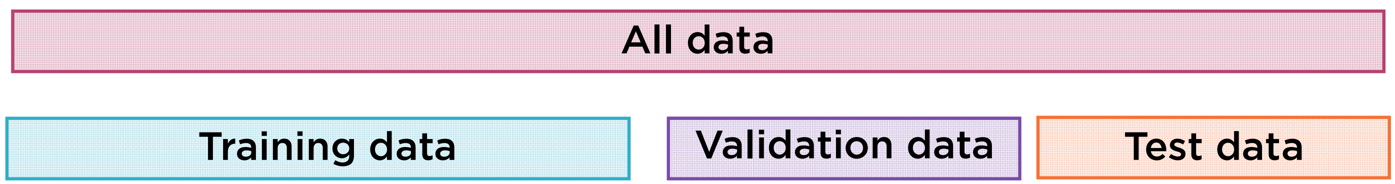
Test data applied to the selected model to provide an unbiased evaluation of the final model

Training, Test, Validation Data



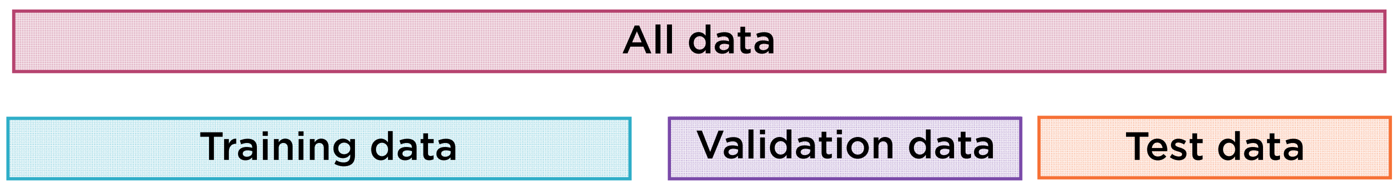
Now can have multiple candidate models, and select the best one - Hyperparameter Tuning

Training, Test, Validation Data

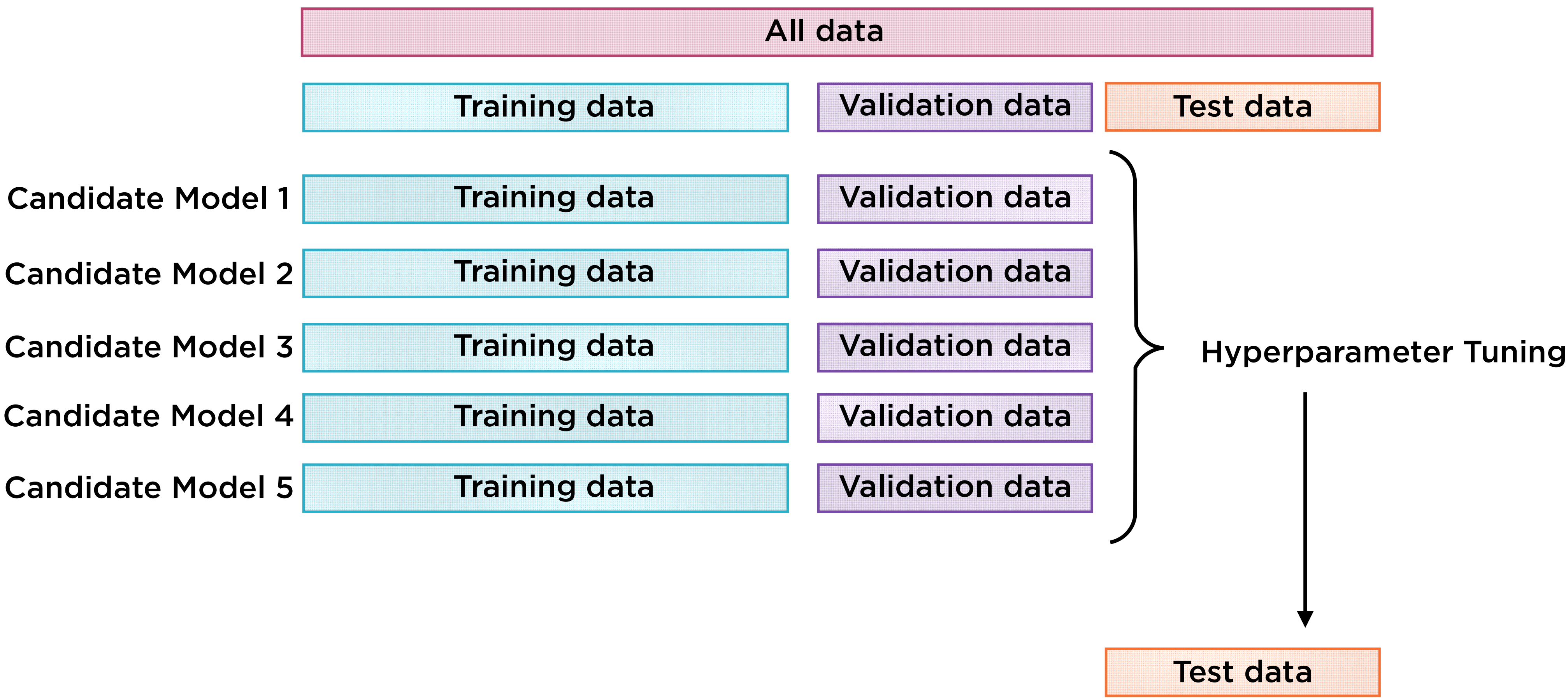


For N candidate models, run N training and N validation processes but just 1 test process

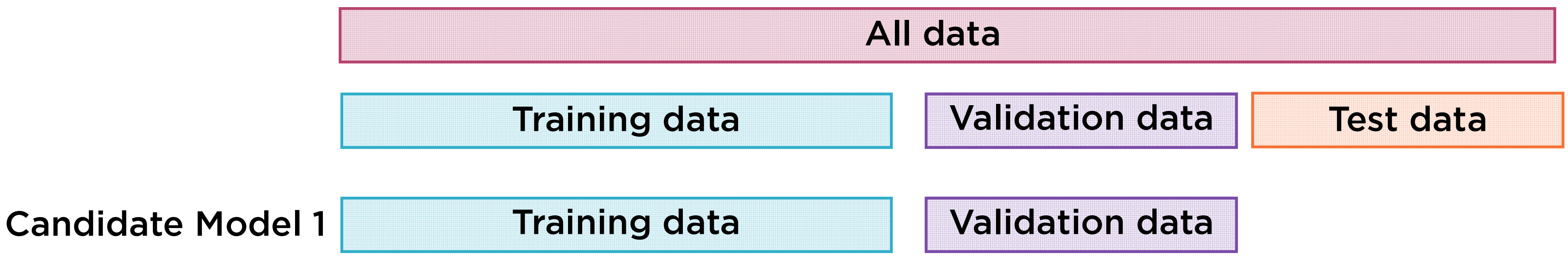
Singular Cross-validation



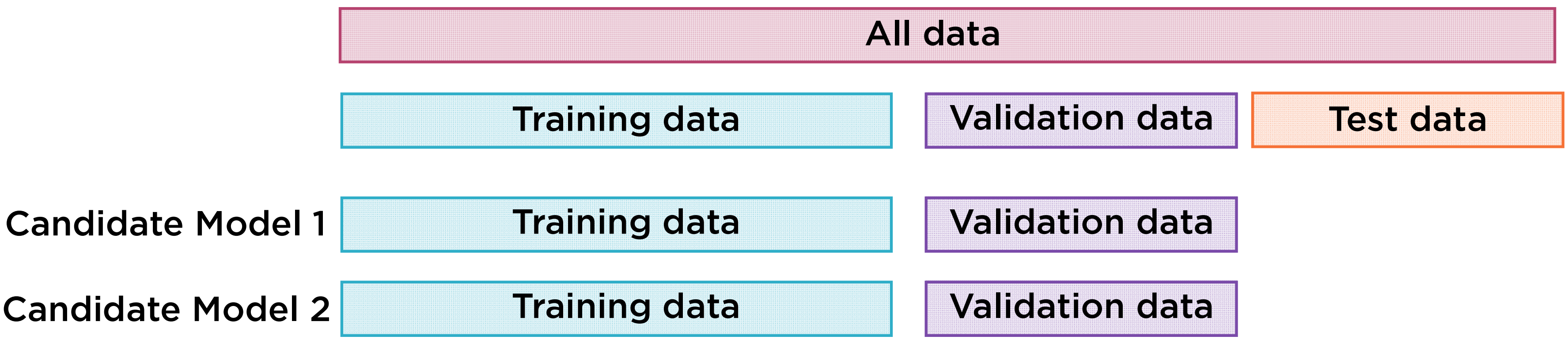
Singular Cross-validation



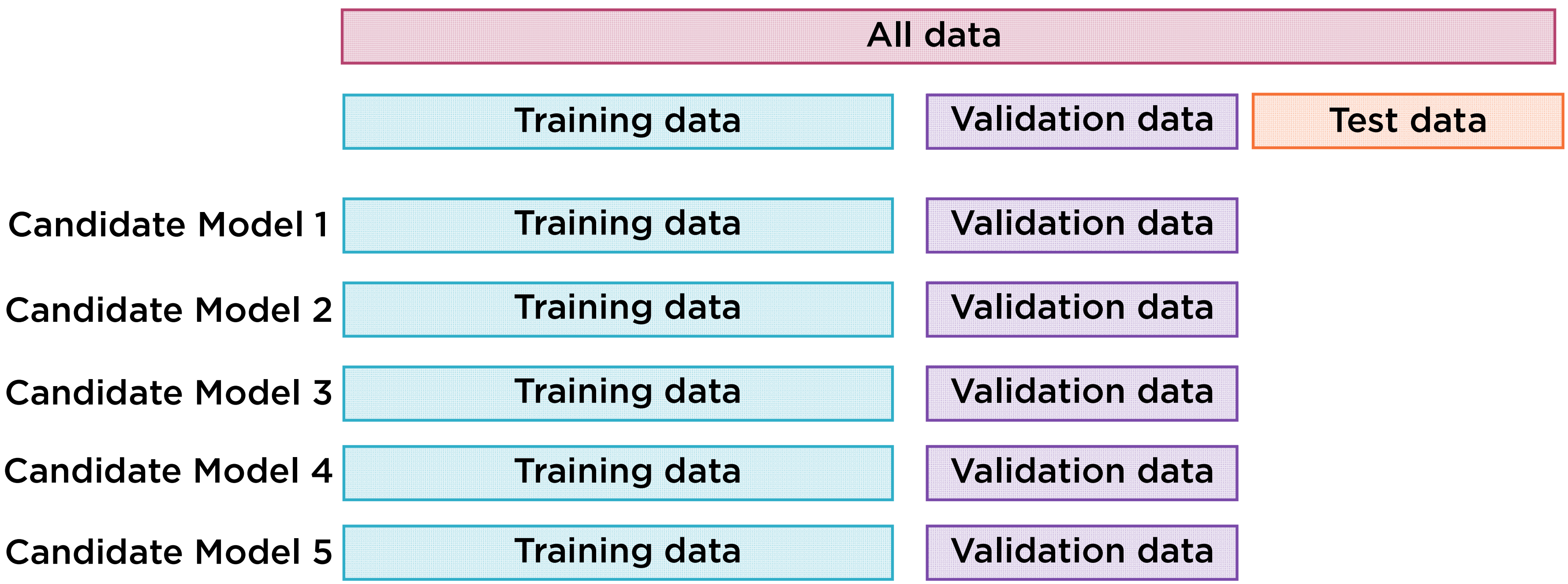
Singular Cross-validation



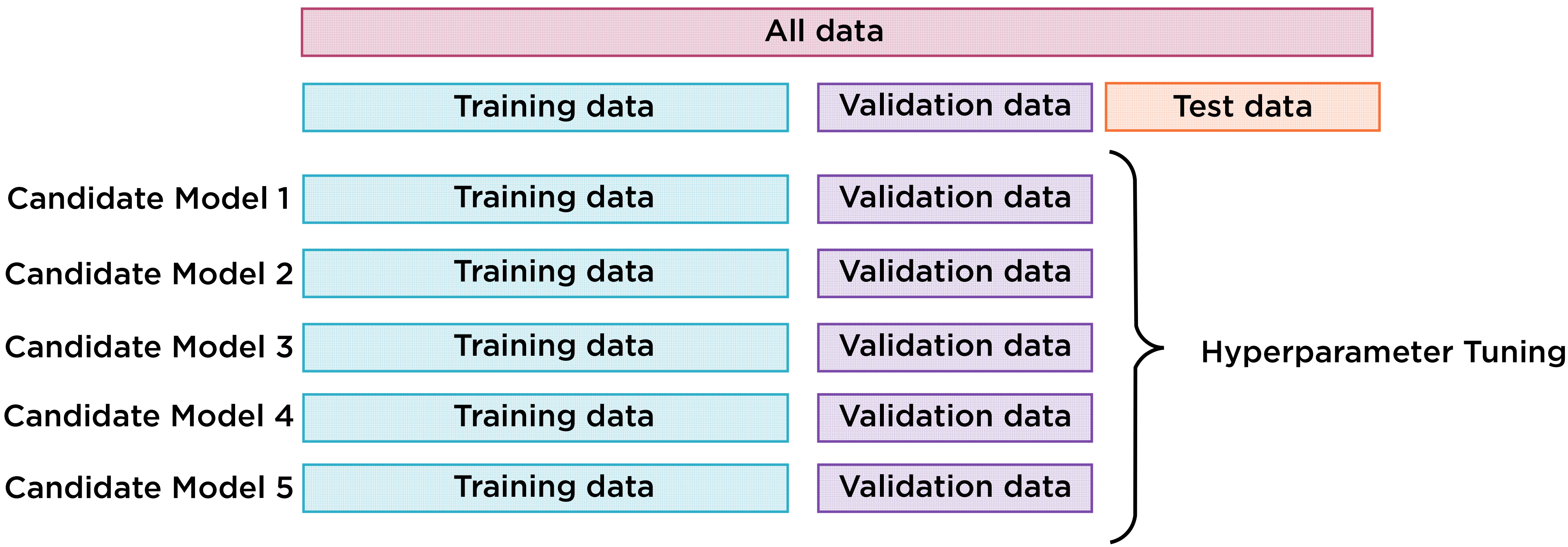
Singular Cross-validation



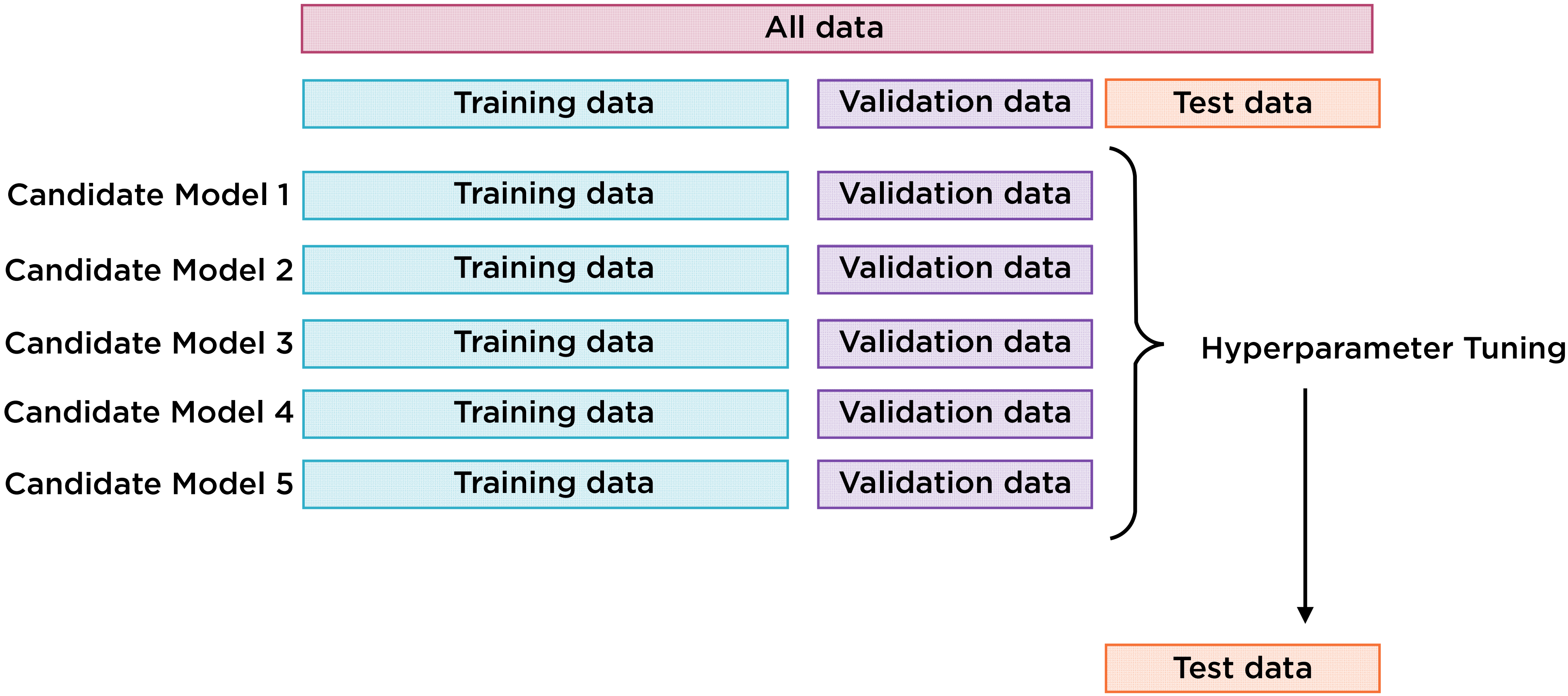
Singular Cross-validation



Singular Cross-validation



Singular Cross-validation



The model's performance on the validation set is incorporated into the model itself - this may introduce bias

K-fold Cross-validation

For each candidate model, repeatedly train, and validate using different subsets of training data. Much more computationally intensive, but very robust - does not “waste” data.

K-fold Cross-validation

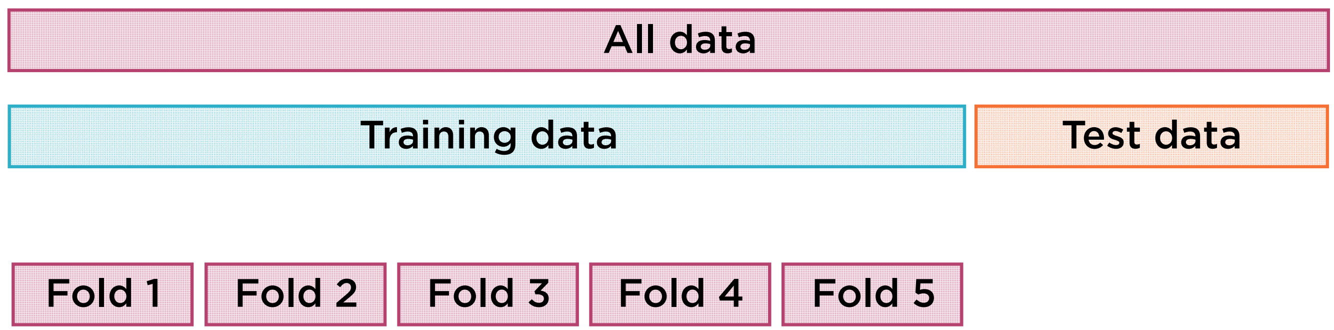
For each candidate model, repeatedly train, and validate using different subsets of training data. Much more computationally intensive, but very robust - does not “waste” data.

K-fold Cross-validation

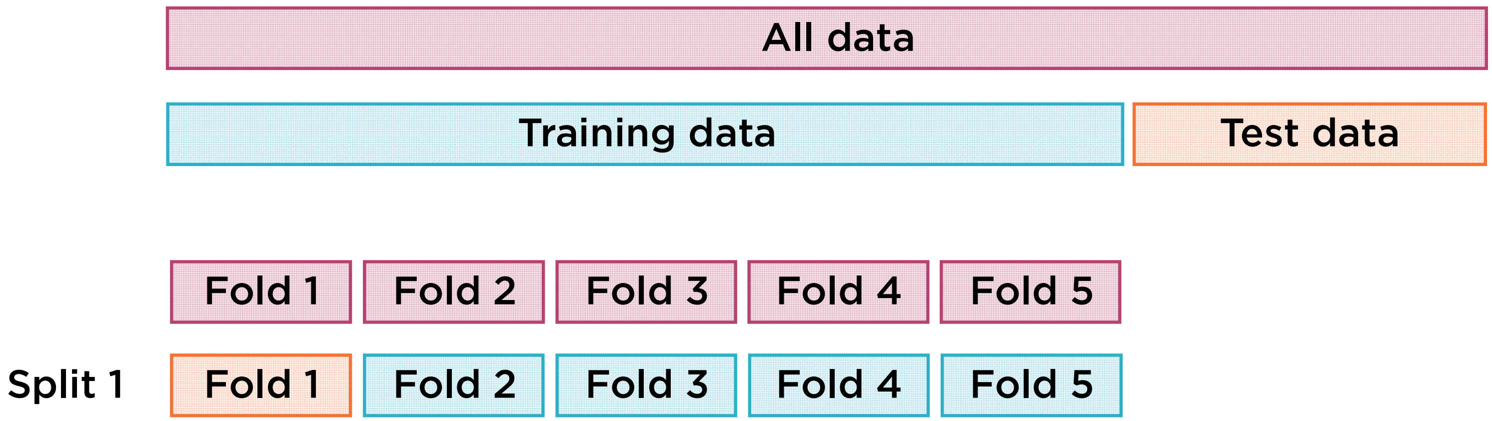
For each candidate model, repeatedly train, and validate using different subsets of training data. Much more computationally intensive, but very robust - does not “waste” data.

K-fold Cross-validation

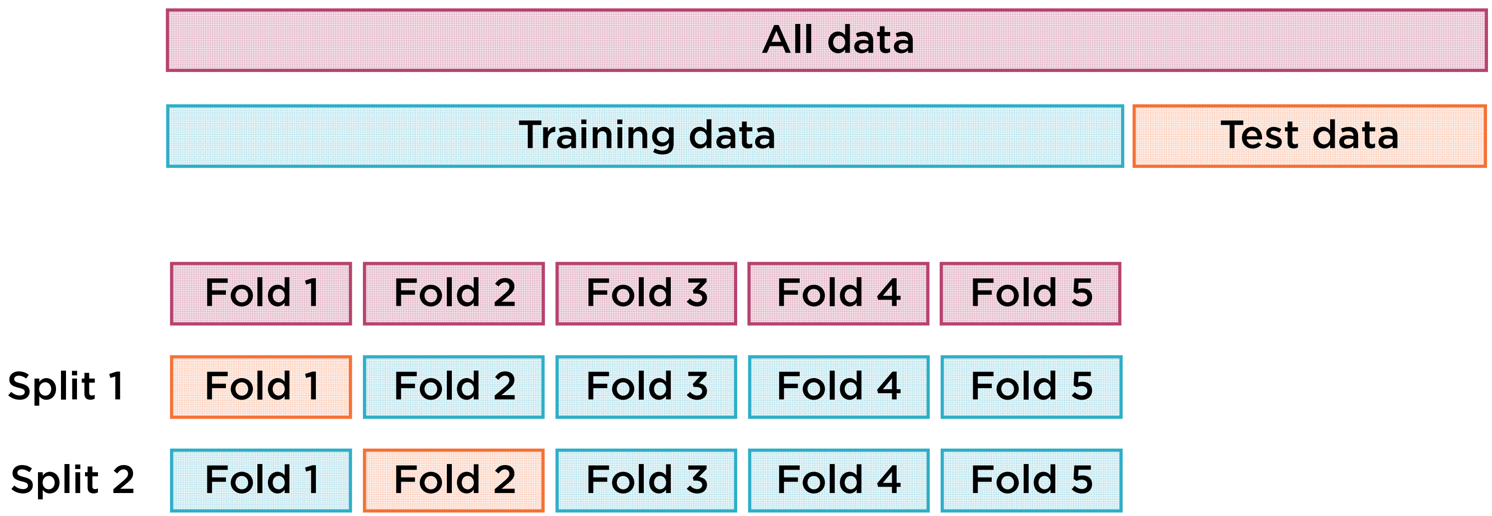
K-fold Cross-validation



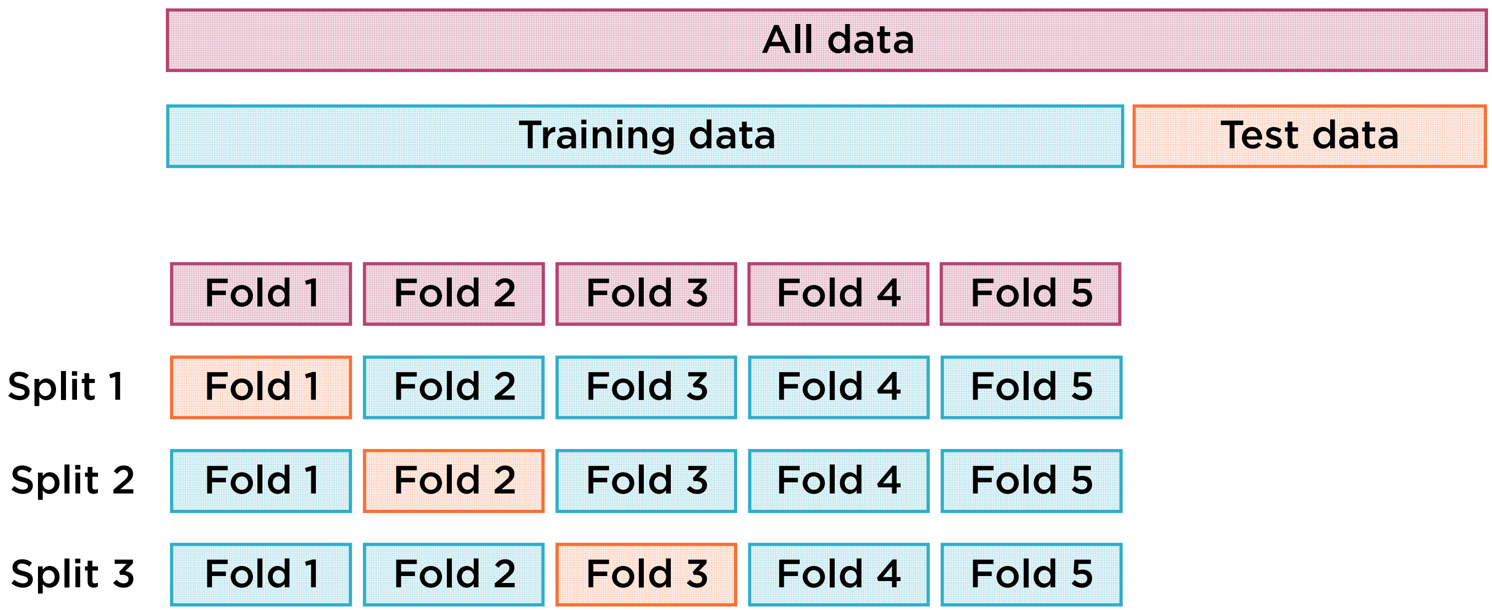
K-fold Cross-validation



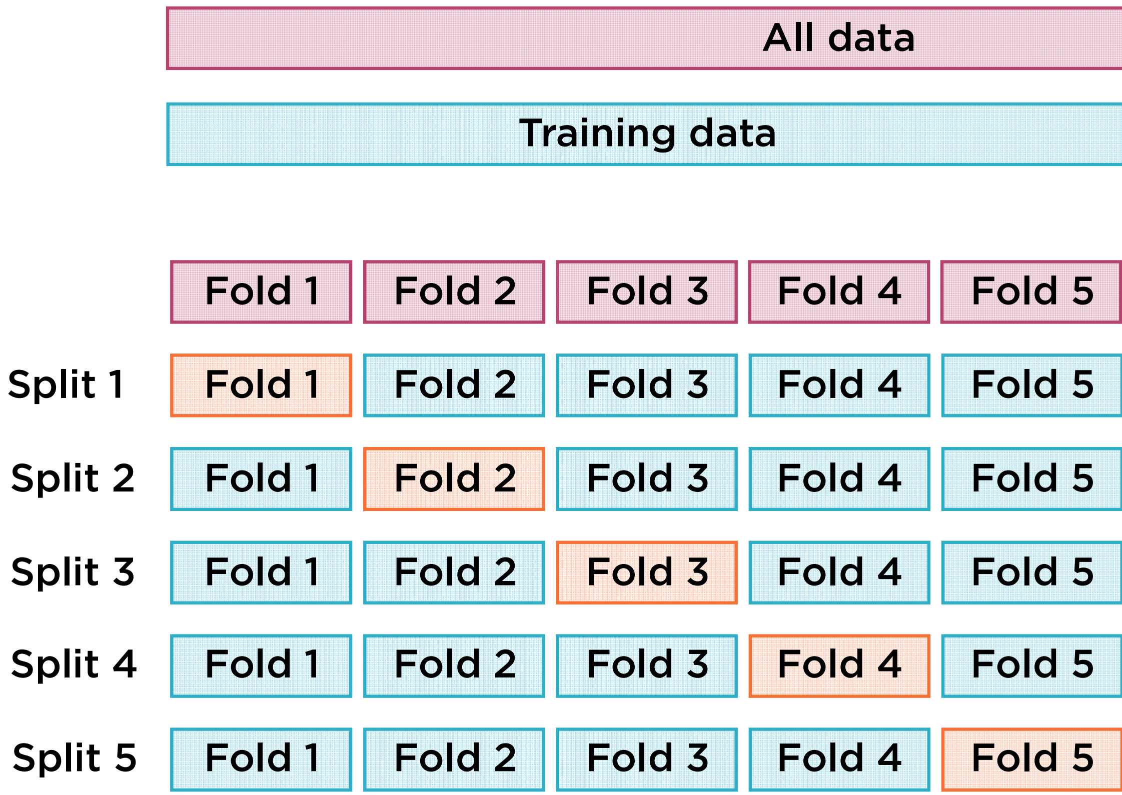
K-fold Cross-validation



K-fold Cross-validation



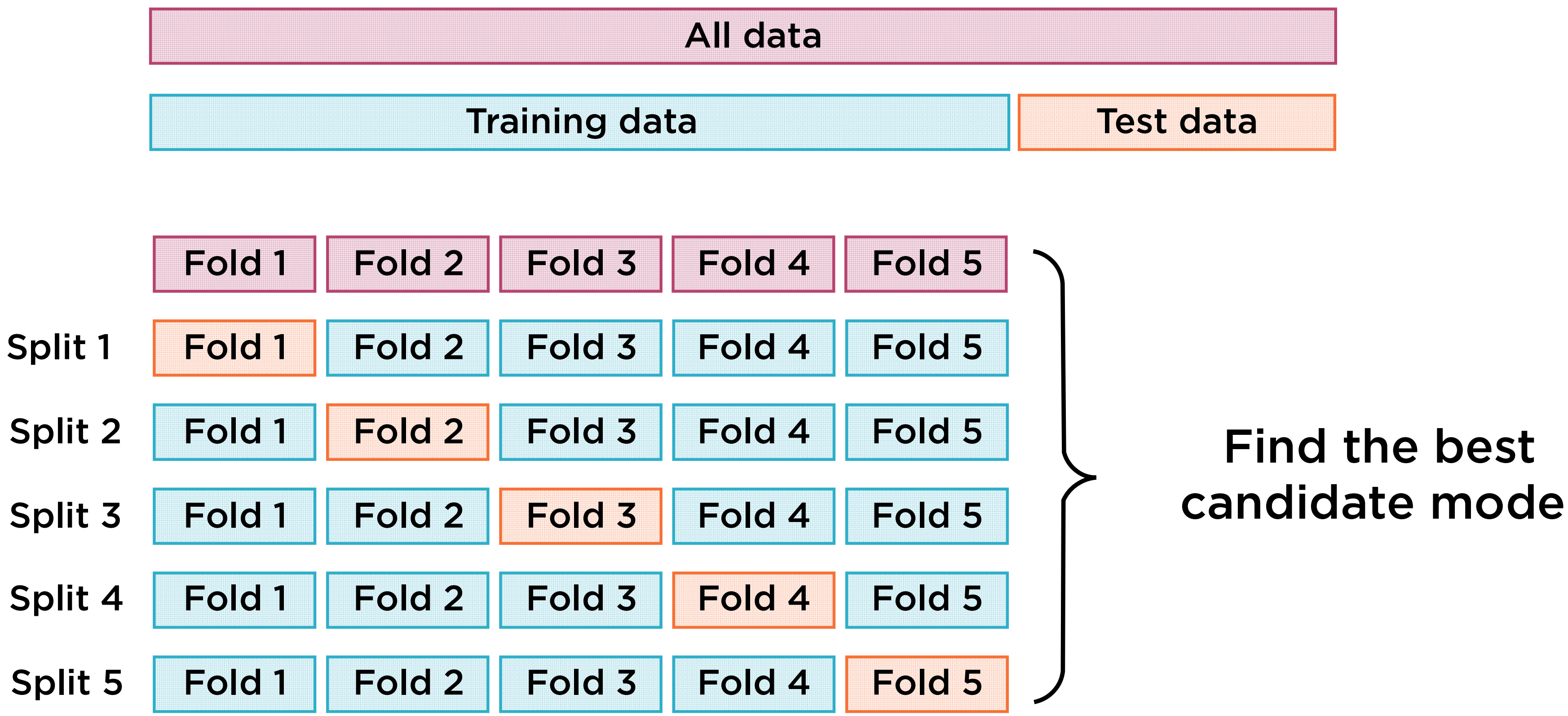
K-fold Cross-validation



Run training and validation for each fold for each candidate model

Average performance across all folds

K-fold Cross-validation



K-fold Cross-validation