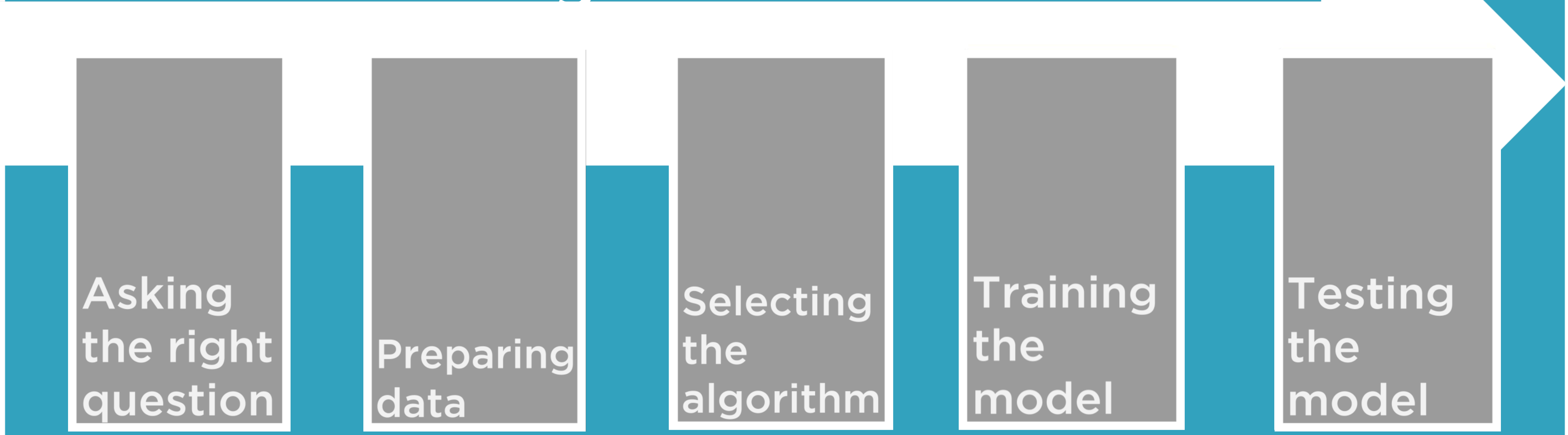
Machine Learning Workflow



Asking the right question

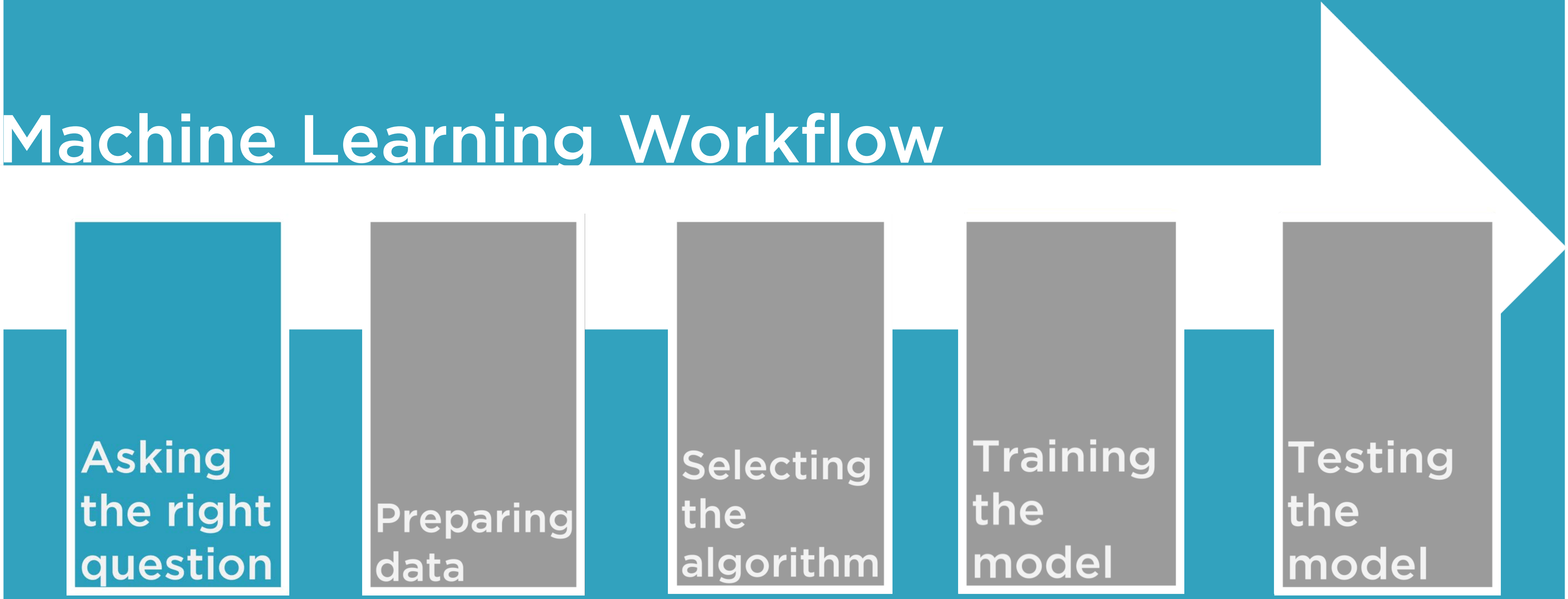
Preparing data

Selecting the algorithm

Training the model

Testing the model

Machine Learning Workflow



Asking the right question

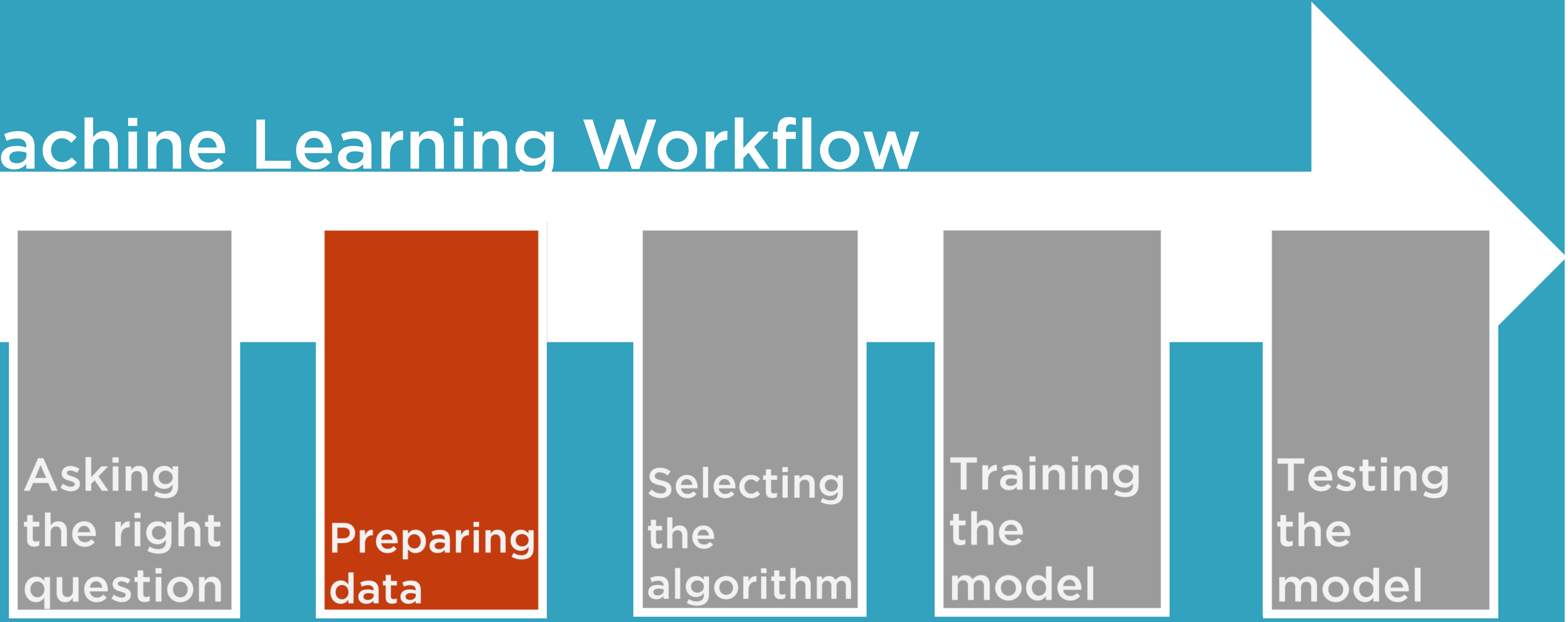
Preparing data

Selecting the algorithm

Training the model

Testing the model

Machine Learning Workflow



Asking the right question

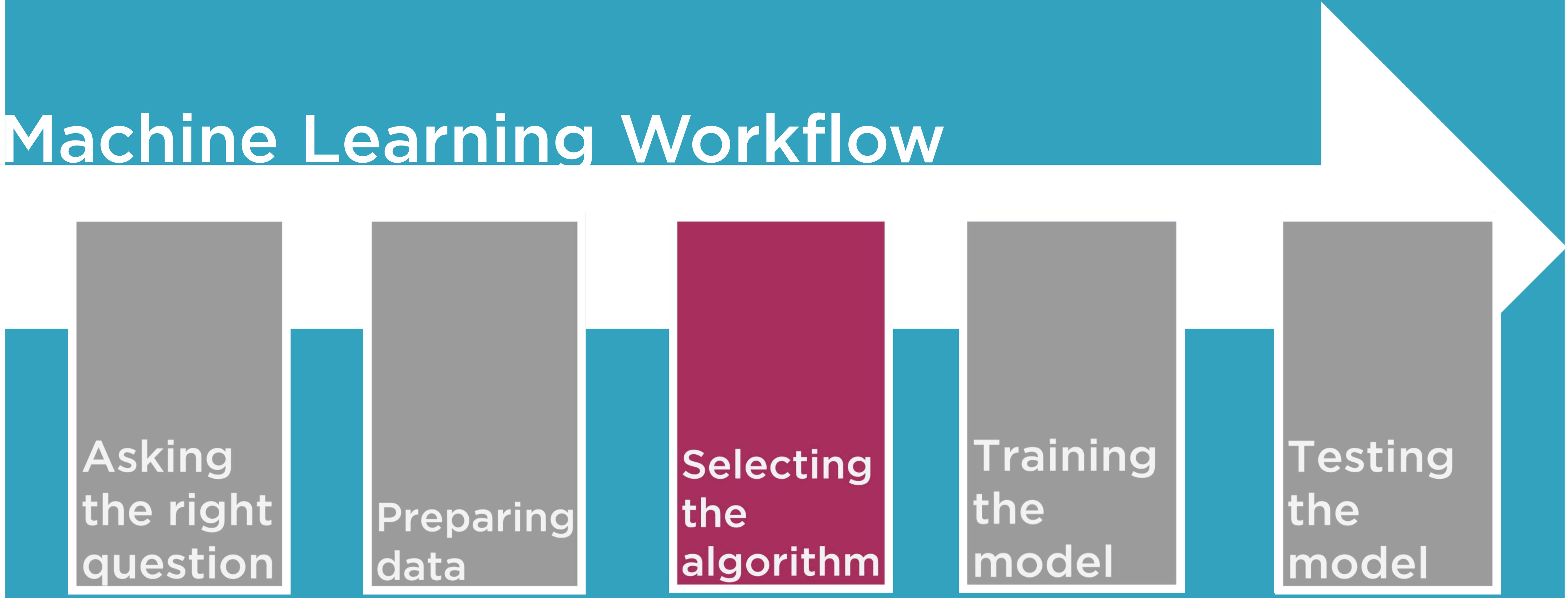
Preparing data

Selecting the algorithm

Training the model

Testing the model

Machine Learning Workflow



Asking the right question

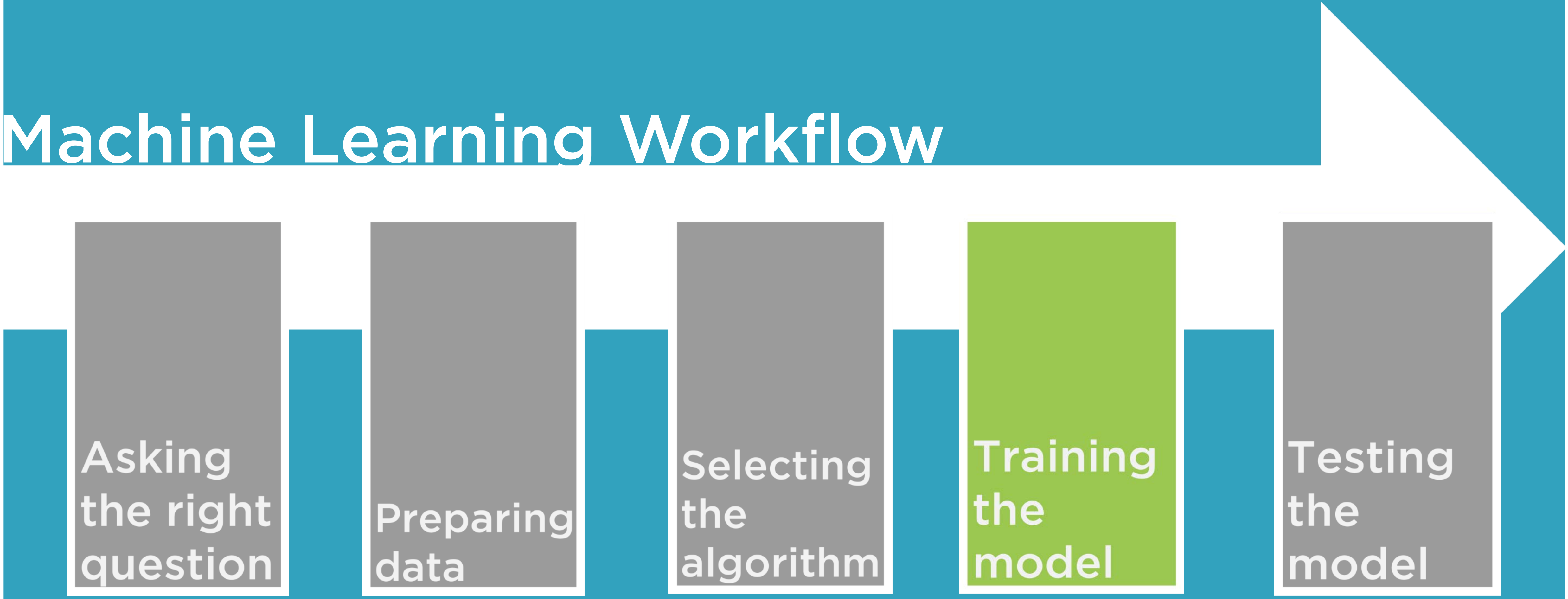
Preparing data

Selecting the algorithm

Training the model

Testing the model

Machine Learning Workflow



Asking the right question

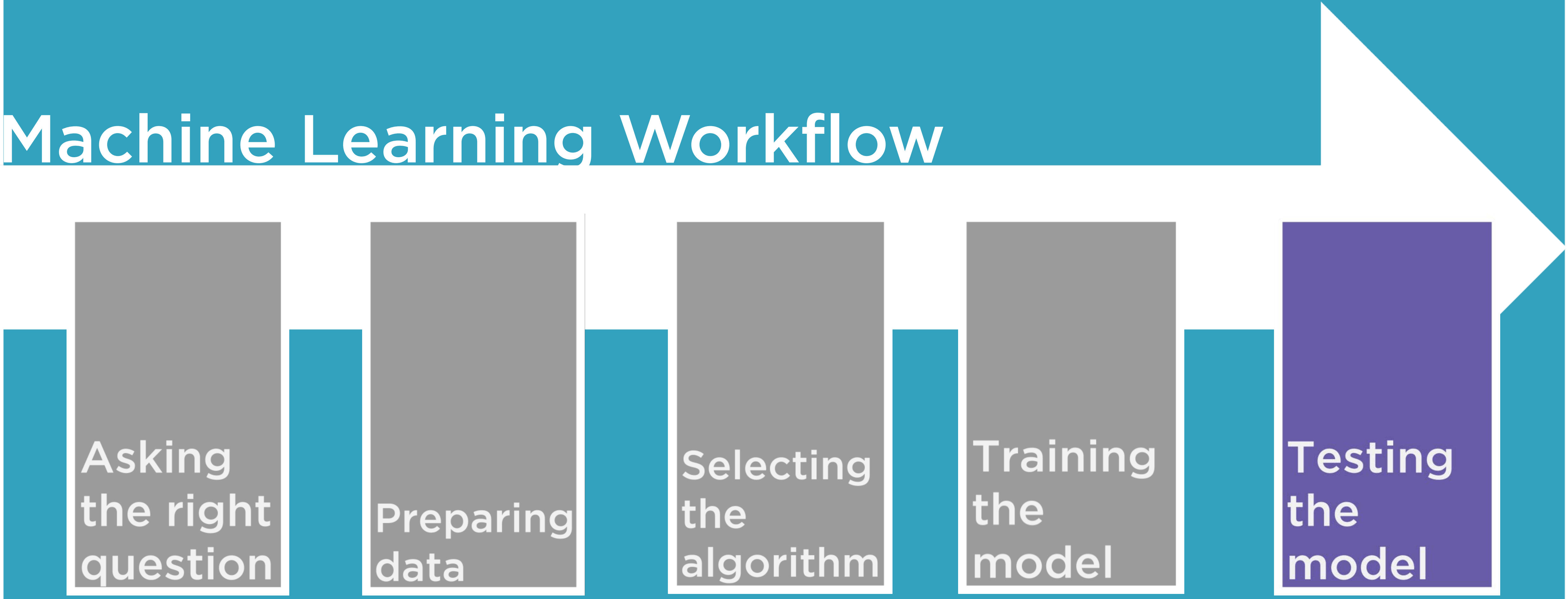
Preparing data

Selecting the algorithm

Training the model

Testing the model

Machine Learning Workflow



Asking the right question

Preparing data

Selecting the algorithm

Training the model

Testing the model

Overview

Evaluate the model against test data

Interpret results

Improve results

Statistics are only data.

We define what is good or bad.

Performance Improvement Options

Adjust current algorithm

Get more data or improve data

Improve training

Switch algorithms

Random Forest

Ensemble Algorithm

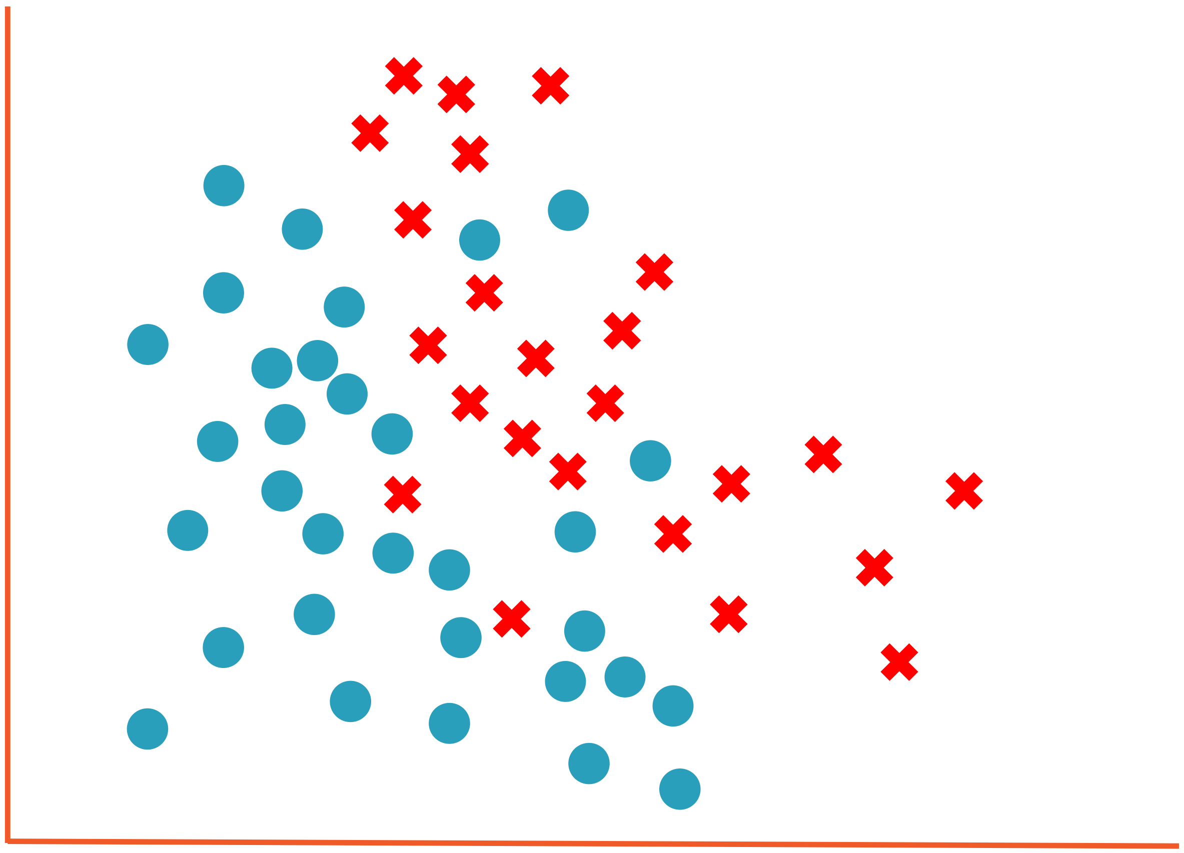
Fits multiple trees with subsets of data

Averages tree results to improve performance and control overfitting

Demo

Train and evaluate Random Forest

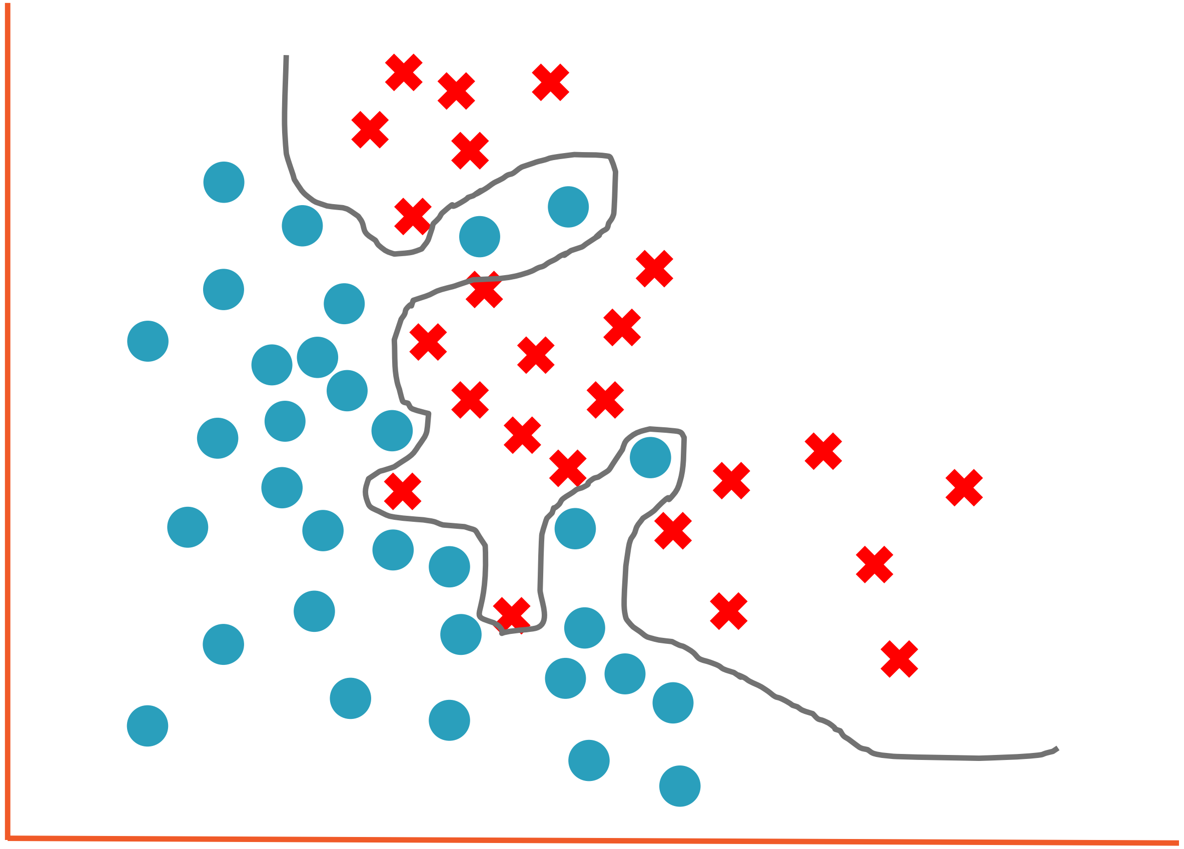
Training Data



Red “X” - Positive

Blue “Circles” - Negative

Fitting Training Data



Train with training data

$y = x_1 + w_2 x_2{}^3 + w_3 x_3{}^8$

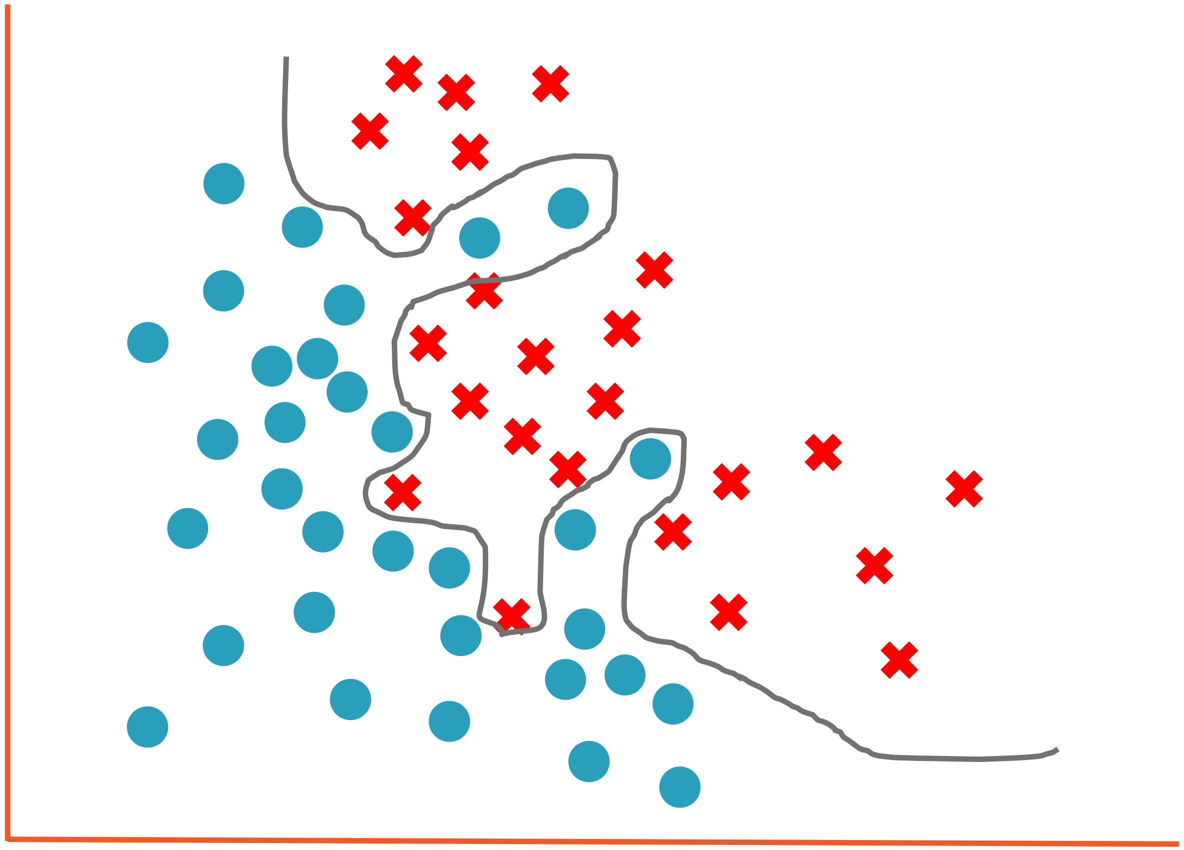
Complex decision boundary

Good fit of training data

Poor fit of test data

Overfitting

Fixing Overfitting



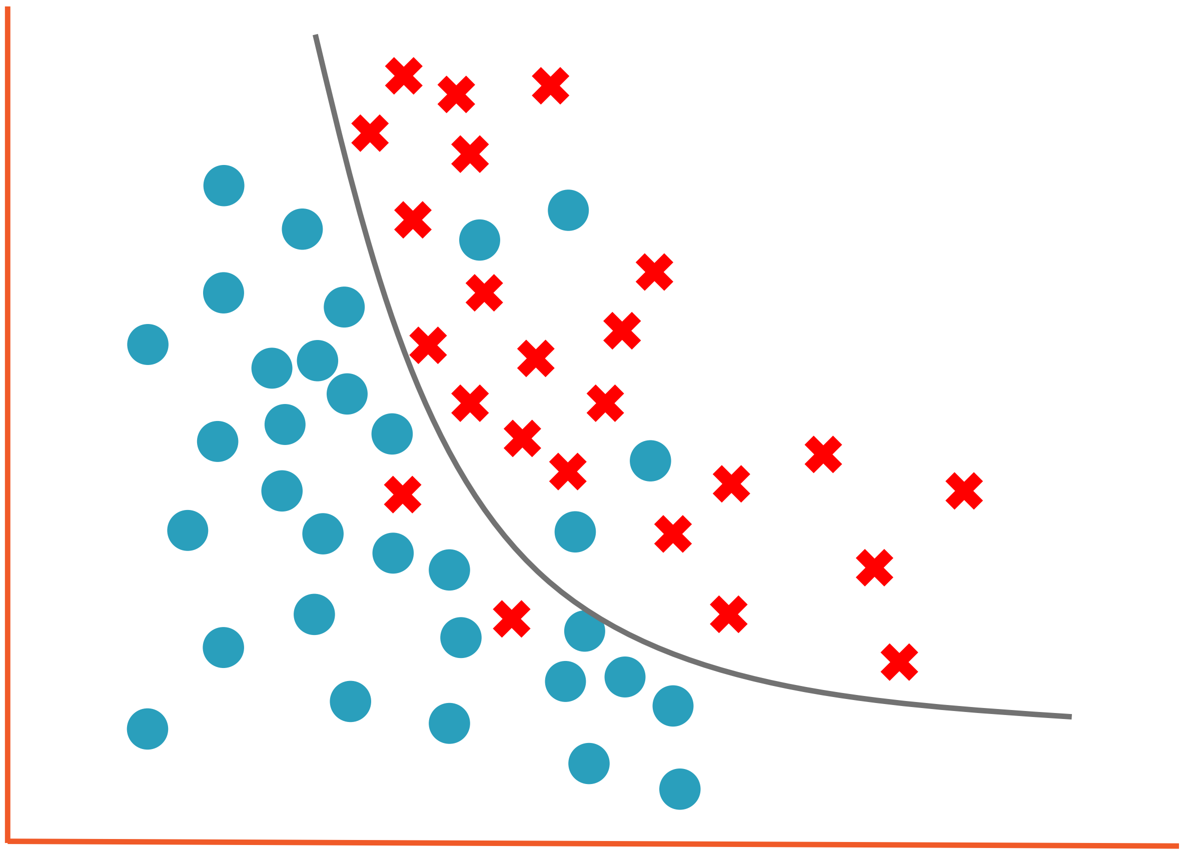
Regularization hyperparameter

$$y = x_1 + w_2{x_2}^3 + w_3{x_3}^8 - \frac{f(W)}{\lambda}$$

Cross validation

Bias – variance trade-off

Fixing Overfitting



Regularization hyperparameter

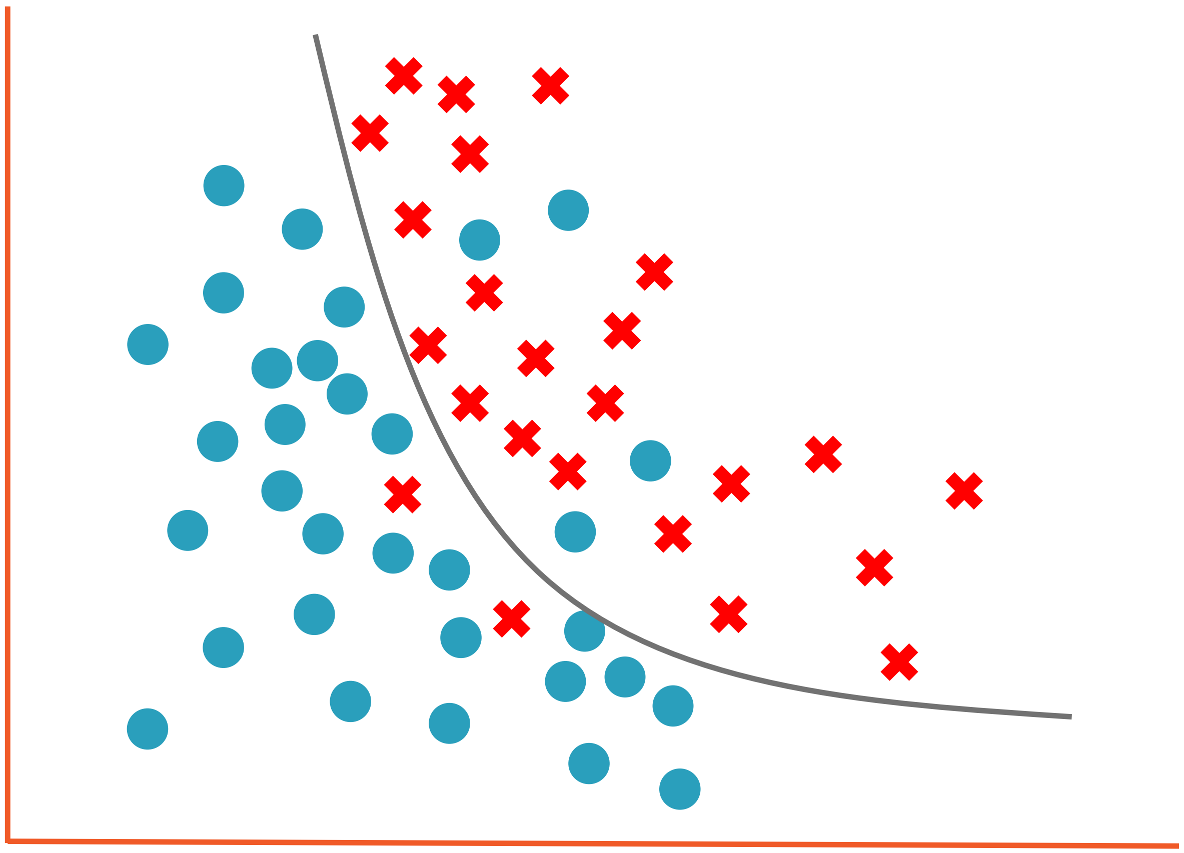
$$y = x_1 + w_2 {x_2}^3 + w_3 {x_3}^8 - \frac{f(W)}{\lambda}$$

Cross validation

Bias – variance trade-off

Sacrifice some perfection for better overall performance.

Fixing Overfitting



Sacrifice some perfection for better overall performance.

Performance Improvement Options, Take 2

Adjust current algorithm

Get more data or improve data

Improve training

Switch algorithms

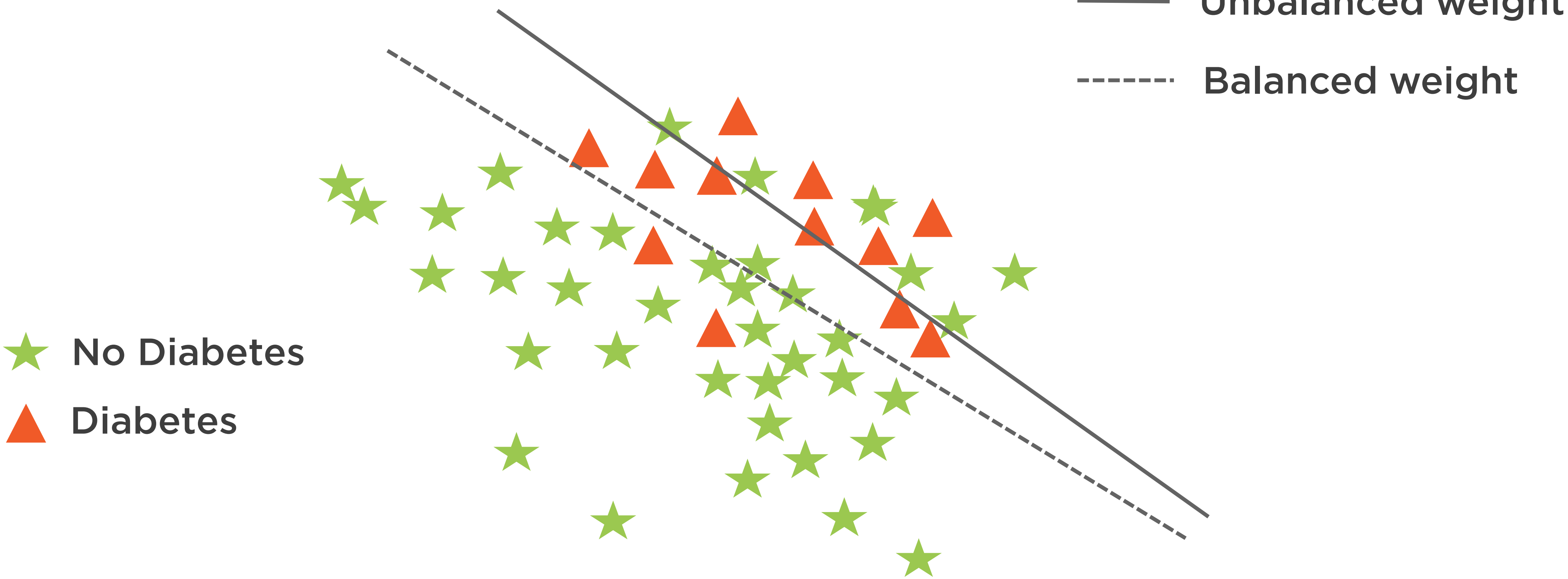
Unbalanced Classes

More of one class than the others

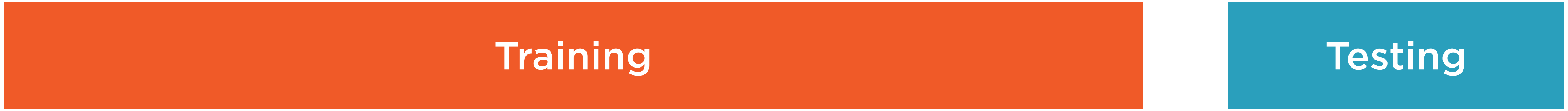
Our Data – 65% No Diabetes, 35% Diabetes

Can be causing biases estimation yielding poor prediction results.

Fixing Unbalanced Classes



Training – Test Split



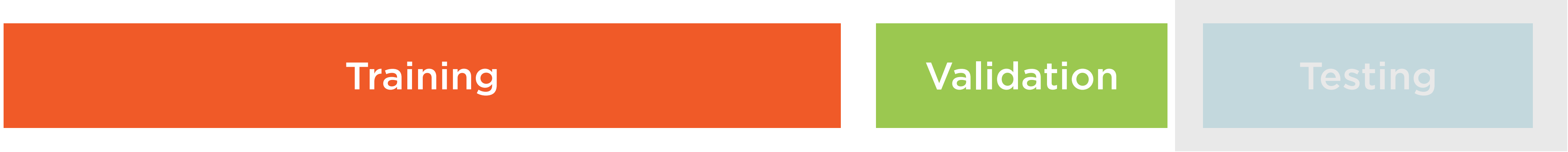
Training

Testing

Are we being influenced by results with test data?

How can we evaluate training without using Testing Data?

Training – Validation - Test Split



Training

Validation

Testing

How do we choose the validation data?

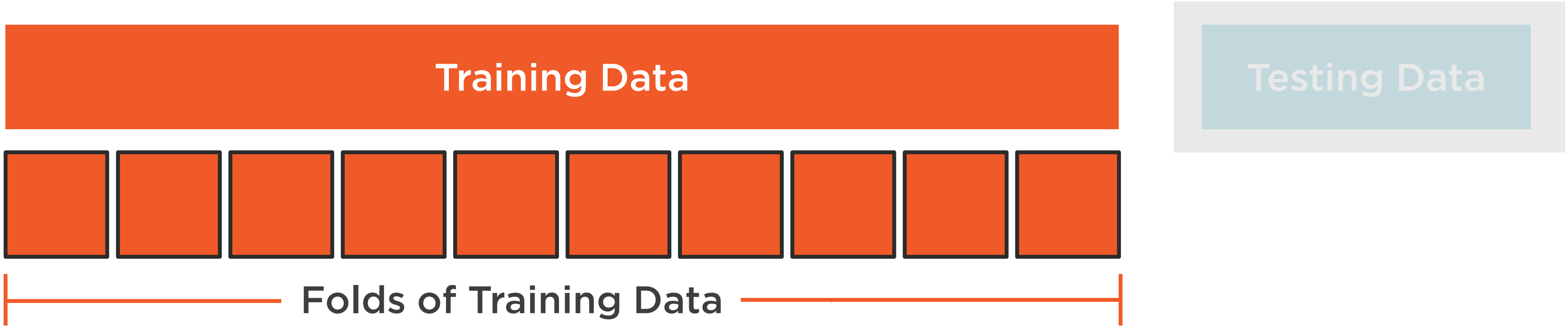
What if we don’t have enough data?

Does this approach mitigate overtraining?

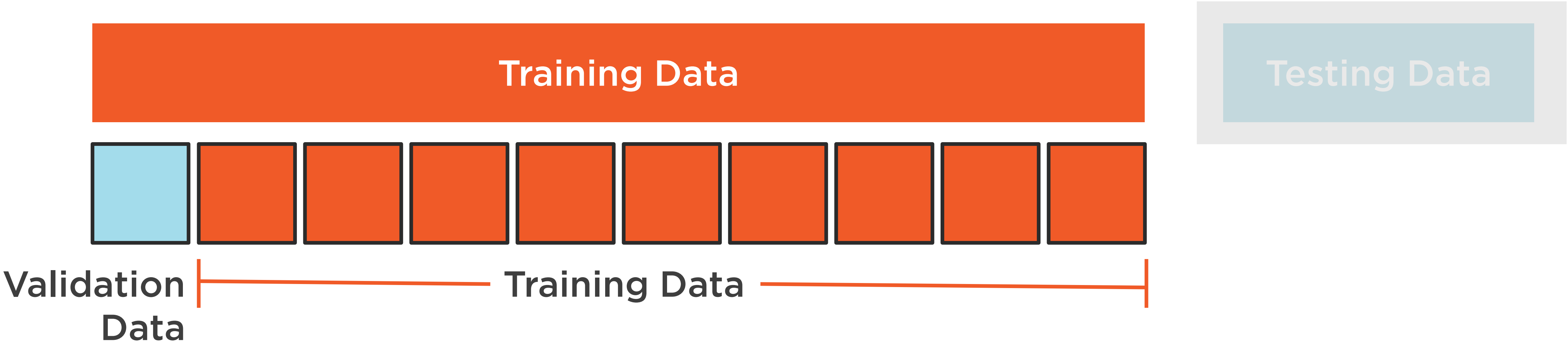
Cross Validation

Training Data
Testing Data

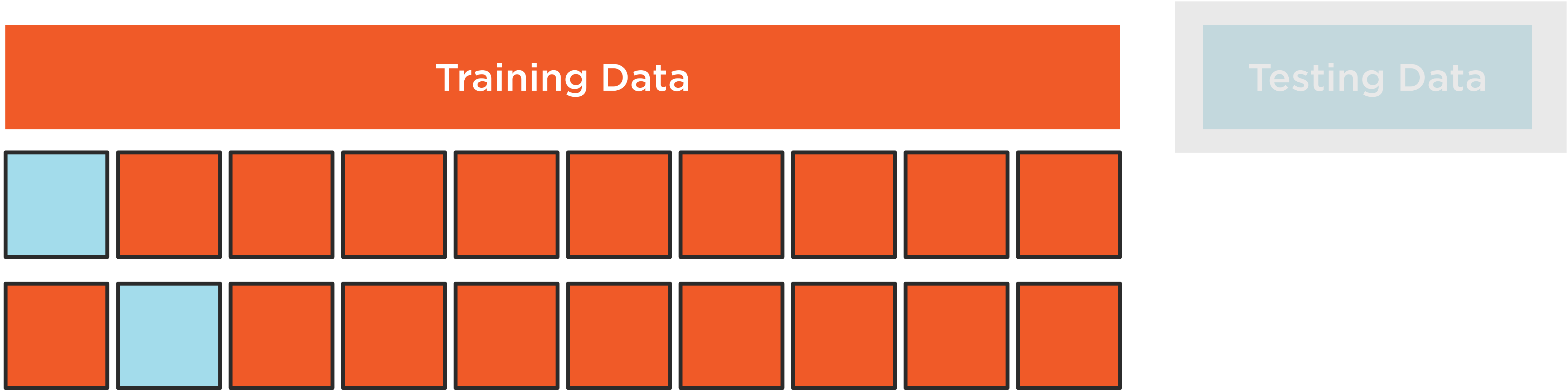
K-fold Cross Validation



K-fold Cross Validation



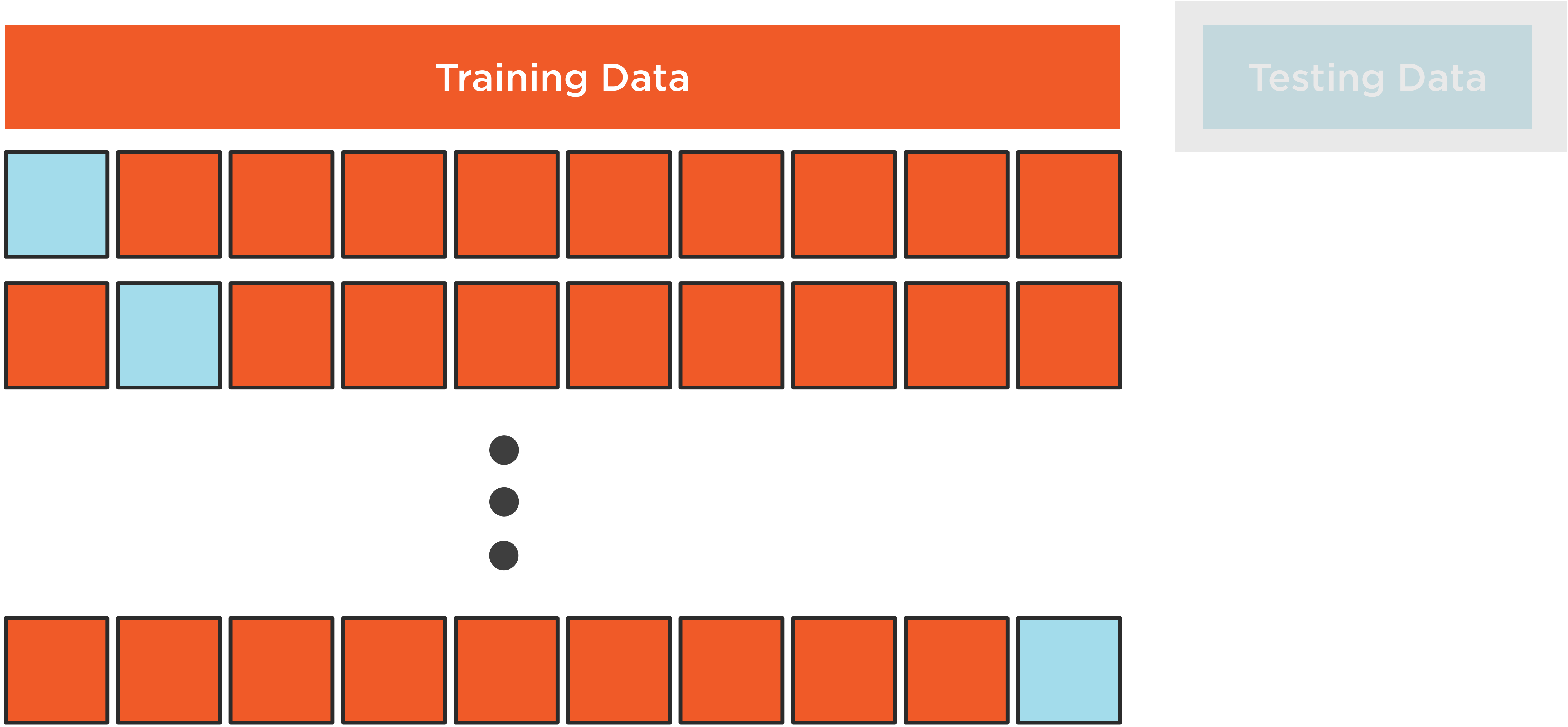
K-fold Cross Validation



Training Data

Testing Data

K-fold Cross Validation



Training Data

Testing Data

Tuning Hyperparameters with Cross Validation

```
For each fold
    Determine best hyperparameter value
Next

Set model hyperparameter value to average best
```

Algorithm CV Variants

Algorithm + Cross Validation = AlgorithmCV

Ends in “CV”

Exposes fit(), predict(), ...

Runs the algorithm K times

Can be used like normal algorithm

Performance Improvement Cycle

Change data, settings, algorithm or all of the above

Improve each cycle

The difficult part is knowing when to stop

> “Genius is one percent inspiration and ninety-nine percent perspiration.”

Thomas A. Edison

Summary

Evaluated Naïve Bayes model

- predict()
- confusionMatrix()

Tried using Random Forest algorithm

- Overfit

Improved performance with Logisitic Regression

- Regularization
- Achieved performance goal

Logistic Regression Cross Validation

- Slightly below 70% target
- Better performance on real world data