Employing Ensemble Methods with scikit-learn

UNDERSTANDING ENSEMBLE LEARNING TECHNIQUES

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

Overview

Ensemble learning to improve robustness and reduce overfitting

Different kinds of ensemble learning techniques

Averaging, boosting, voting, stacking

Built-in support for ensemble learning in scikit-learn

Implementing hard and soft voting in scikit-learn

Prerequisites and Course Outline

Prerequisites

Comfortable with Python programming

Prior ML exposure recommended

Building simple classification and regression models using scikit-learn

Prerequisite Courses

Building Your First scikit-learn Solution

Building Classification Models with scikit-learn

Building Regression Models with scikit-learn

Course Outline

Introducing ensemble learning

Ensemble learning using averaging - bagging and pasting models

Ensemble learning using boosting - adaptive and gradient boosting

Ensemble learning using stacking

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.

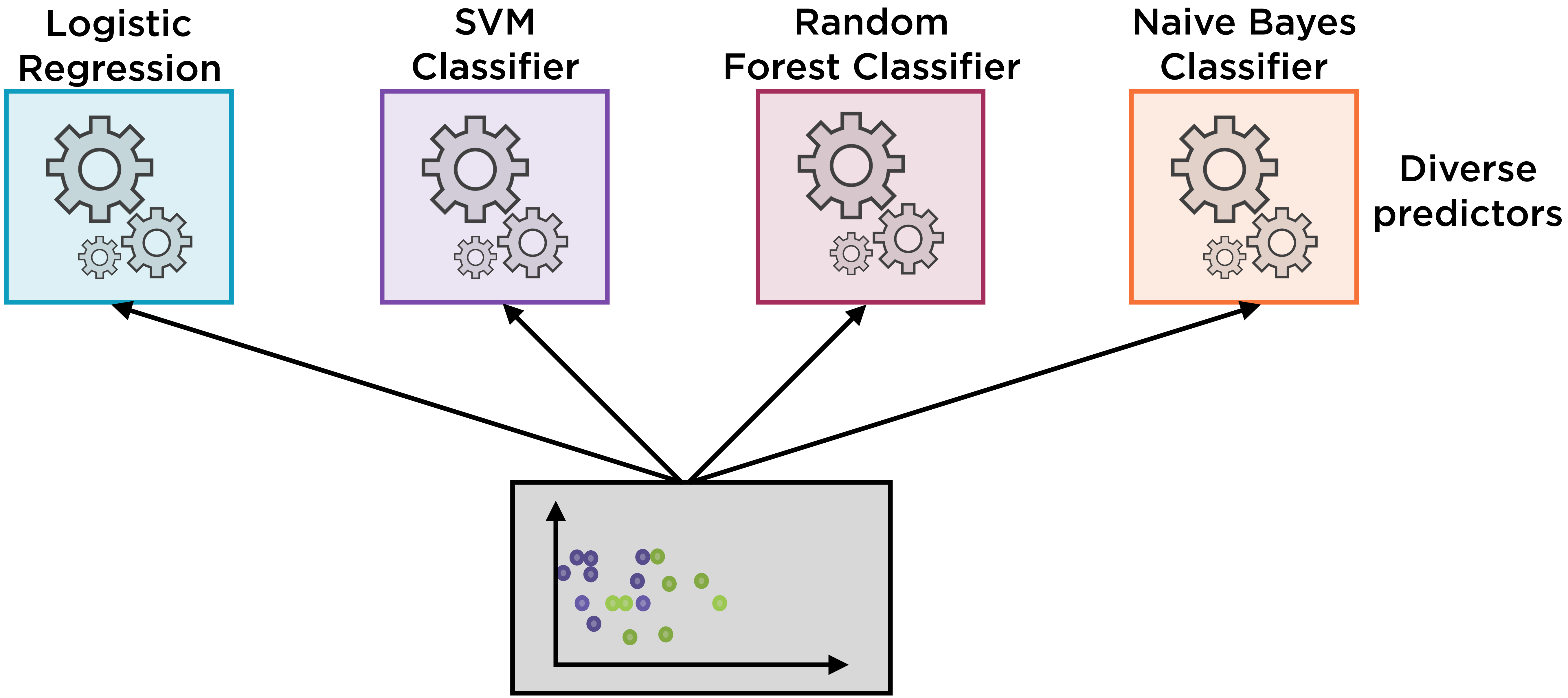
Ensemble Learning

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

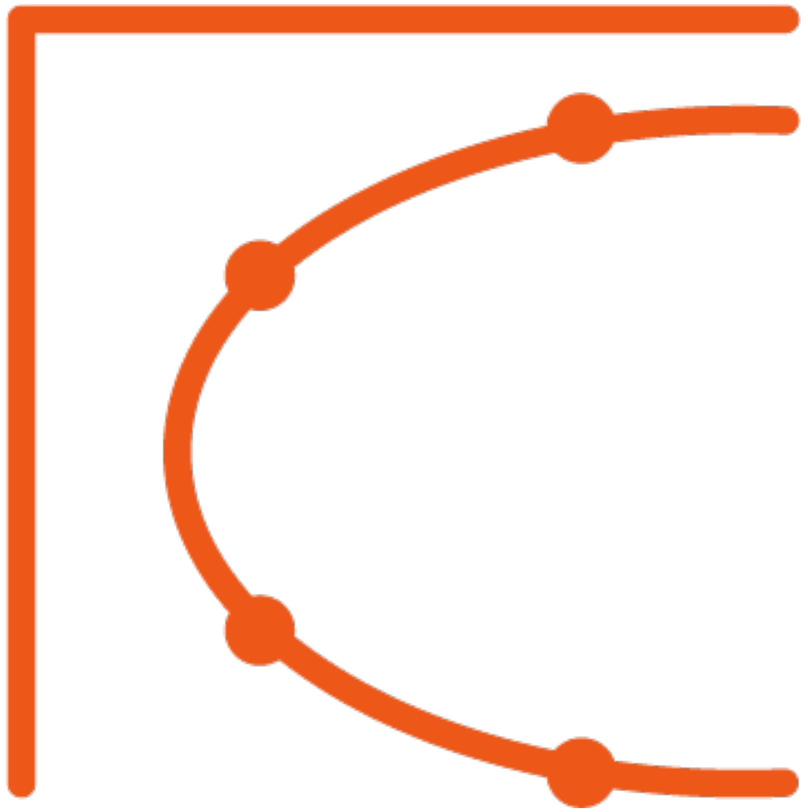
Important Questions in Ensemble Learning

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

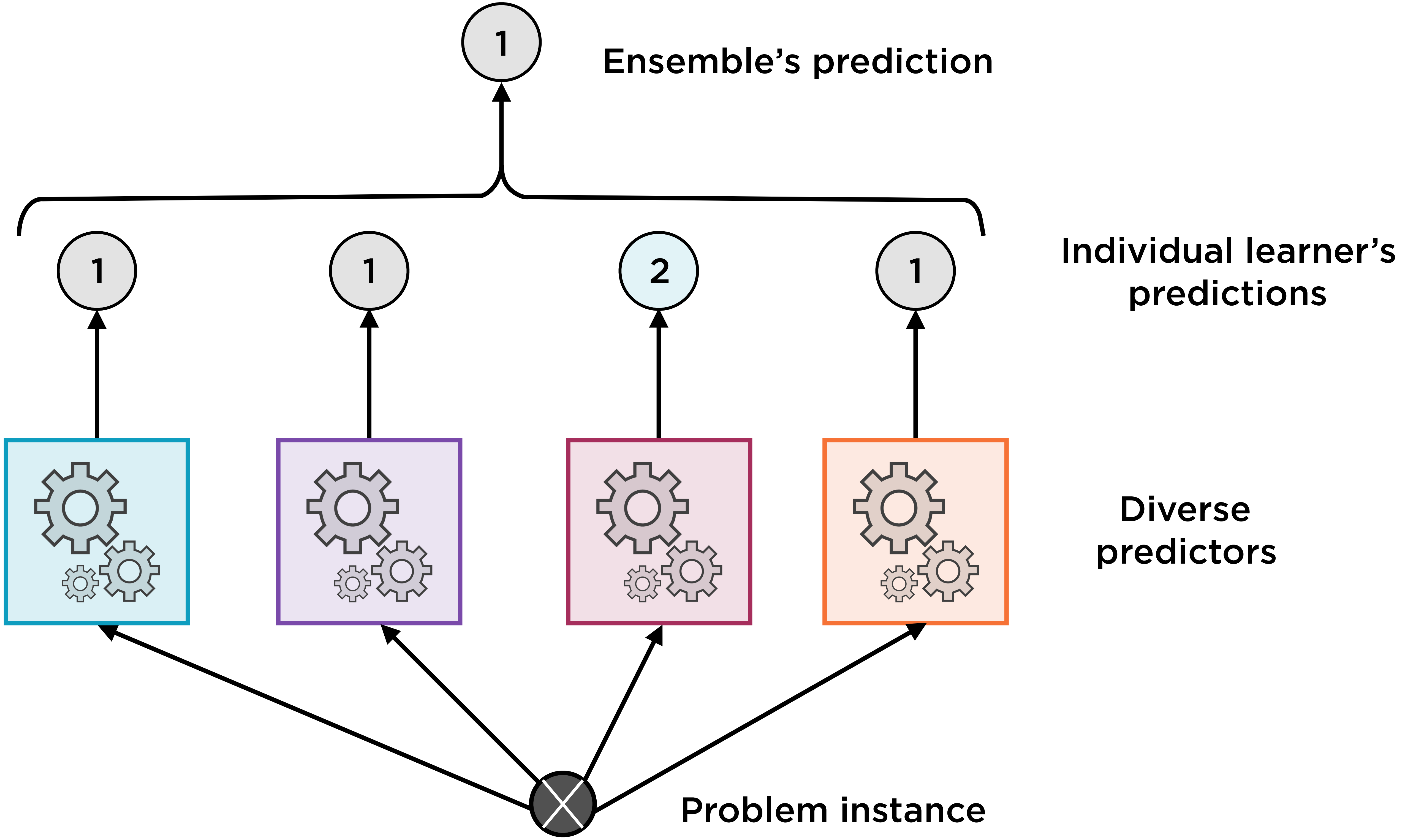
Important Questions in Ensemble Learning

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

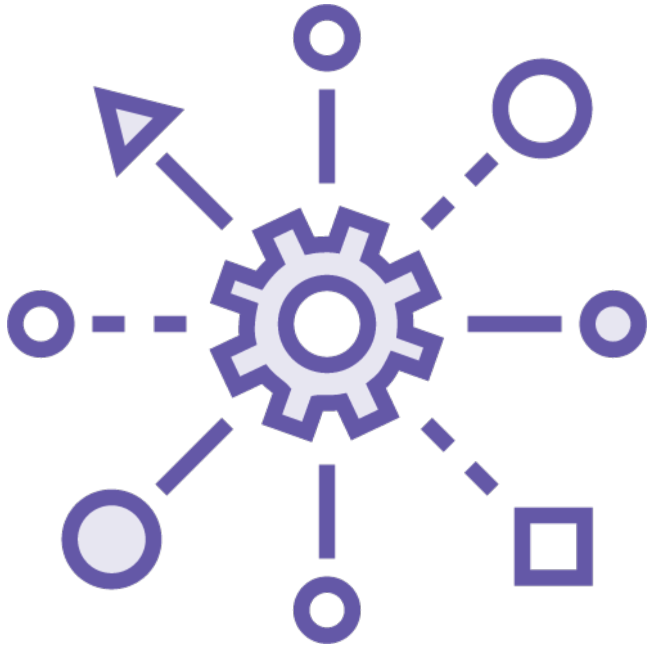
Hard voting: Majority vote of individual learners (classification)

Soft voting: Probability-weighted average

Stacking: Train additional model to combine predictions from individual learners

Ensemble Learning Techniques

Averaging and Boosting



Averaging

Train predictors in parallel and average scores of individual predictors

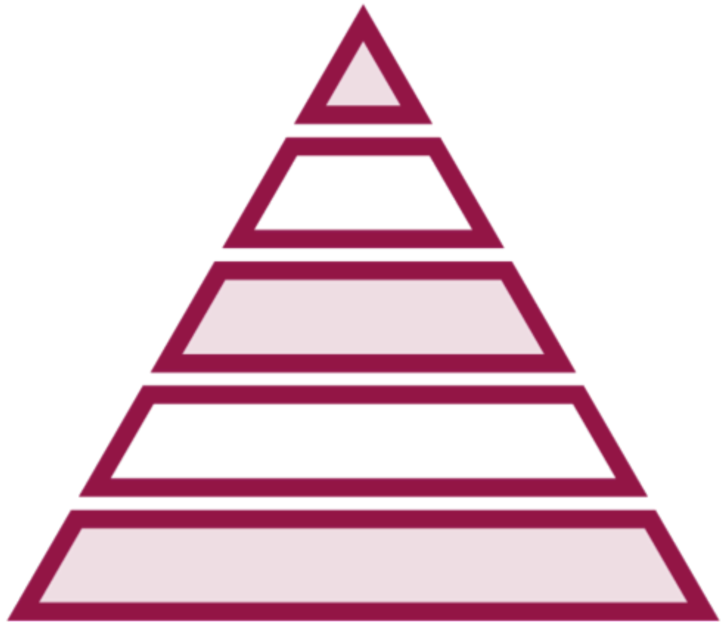
Boosting

Train predictors in sequence where each predictor learns from earlier mistakes

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

Voting

Get individual predictions from each learner

Each learner uses a different training algorithm

The different algorithms add diversity to the predictions

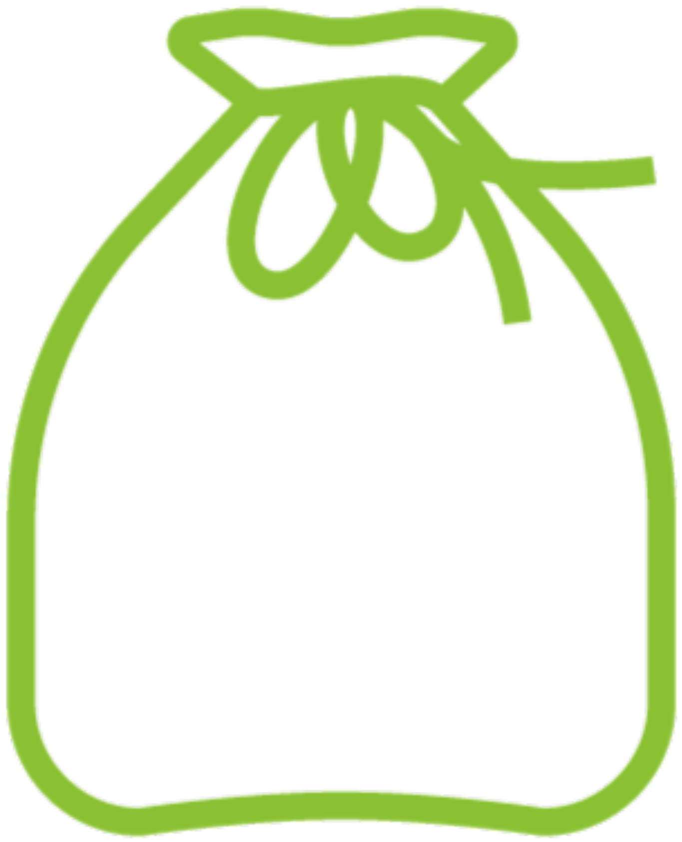
Voting

Hard Voting: **Final output of the ensemble is the majority vote**

Soft Voting: **Final output of the ensemble is the category with the highest probability score**

- Need to be able to aggregate probability scores for each output category

Averaging



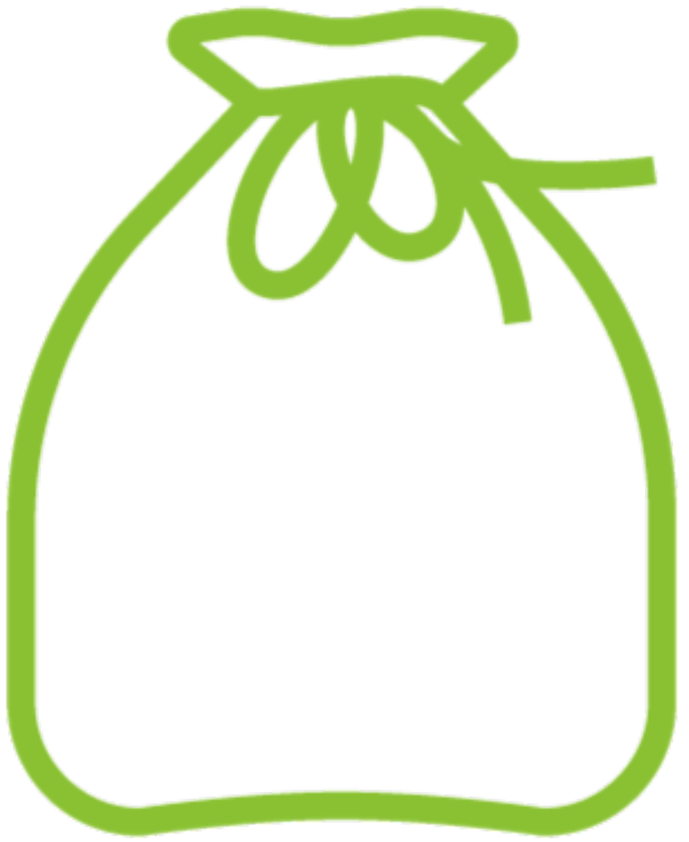
Train multiple learners in parallel

Get individual predictions from each learner

Final prediction of the ensemble is an average of individual predictions

Voting can be considered an averaging technique

Averaging



Usually use decision trees or random forests to build different models

Train model on different samples of training data

- Bagging: Sample data with replacement
- Pasting: Sample data without replacement

Boosting

Train multiple learners **sequentially**

Each model learns from the mistakes made by previous models

Can tweak the learning rate or contribution of each model

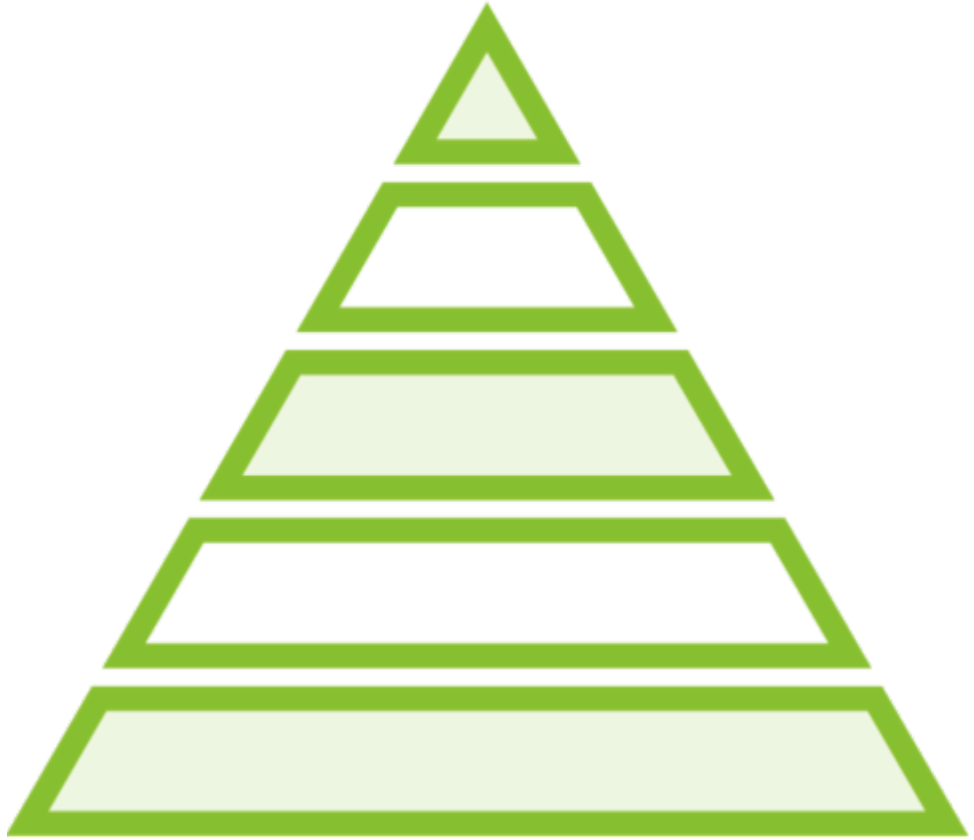
Addition of a learner boosts the accuracy of the model

Boosting

Adaptive Boosting: each model pays more attention to training instances the previous model got wrong

Gradient Boosting: each model in sequence fits on residual errors of the previous model

Stacking



Train diverse individual learners

Get predictions from individual predictors

Fit a model to make the final predictions of the ensemble

“Blender model” or “Meta-learner”

Decision Trees in 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?

Choice of Individual Learners

Individual learners (models) could be of absolutely any type

Could combine:

- Neural networks
- Support Vector Machines
- Naive Bayes Classifiers
- Decision Trees
- Random Forests (group of decision trees)

Ensemble Learning

Individual learners should be as different as possible

For most techniques, hard to generate large number of very different models

To the rescue: Decision trees and random forests

Decision Trees are the most common building blocks for Ensemble Learning

Decision Trees

ML models that construct trees based on threshold values of x-variables. Differ from rule-based trees because thresholds are determined by training.

Random Forest

An ensemble (collection) of decision trees, in which individual trees are trained on different random subsets of training data.

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 Models

If individual learners are not decision trees

- Then each has independent, full training process

Training Random Forests

Most common form of ensemble learning uses random forests

Individual learners are decision trees

Each tree is iteratively trained on randomly sampled subset

Training Random Forests

Will return to this in a later module

Decision Trees

Jockey or Basketball Player?

Jockeys

Tend to be light to meet horse carrying limits

Basketball Players

Tend to be tall, strong and heavy

Jockey or Basketball Player?

Intuitively know

Jockeys tend to be light

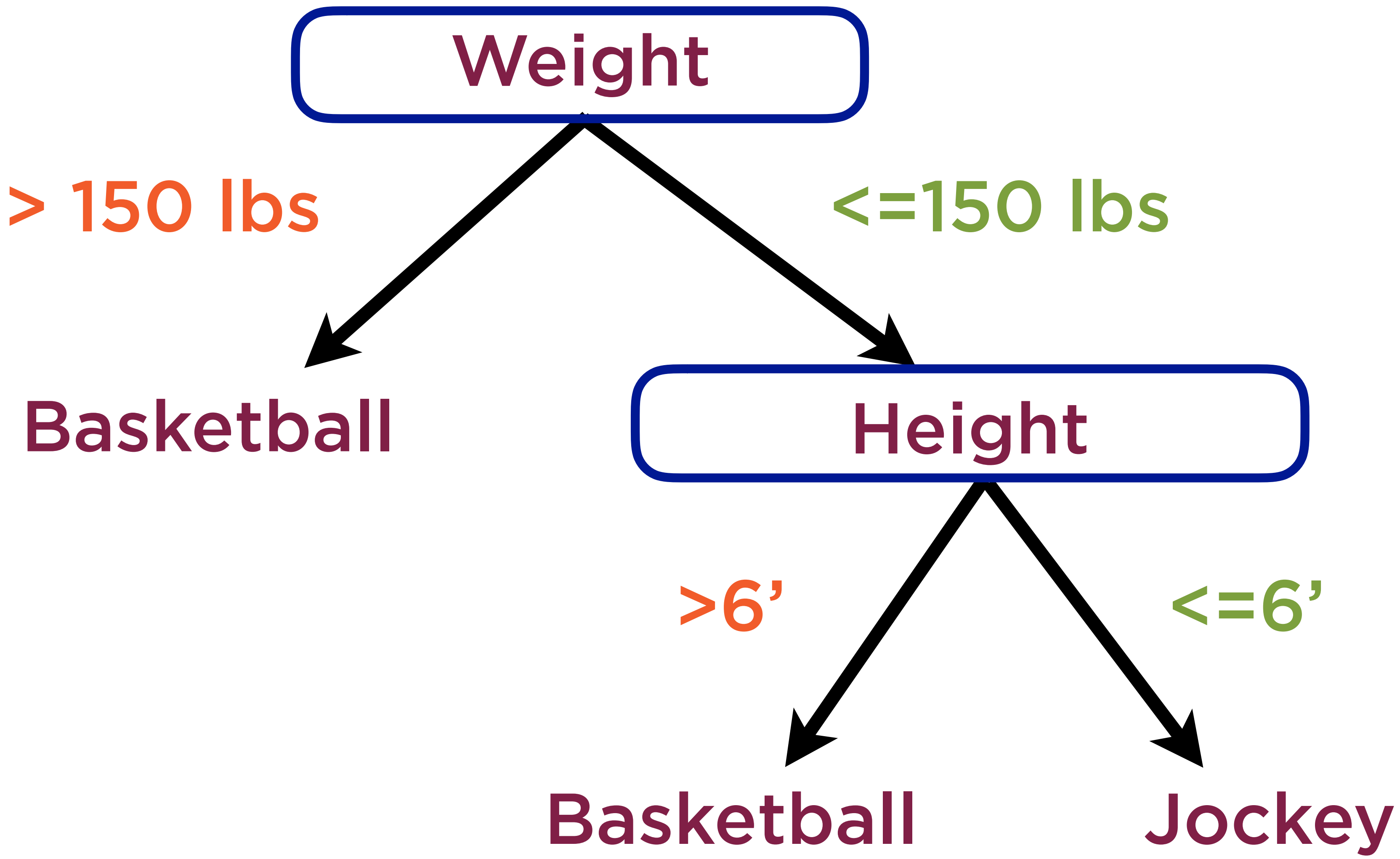
And not very tall

Basketball players tend to be tall

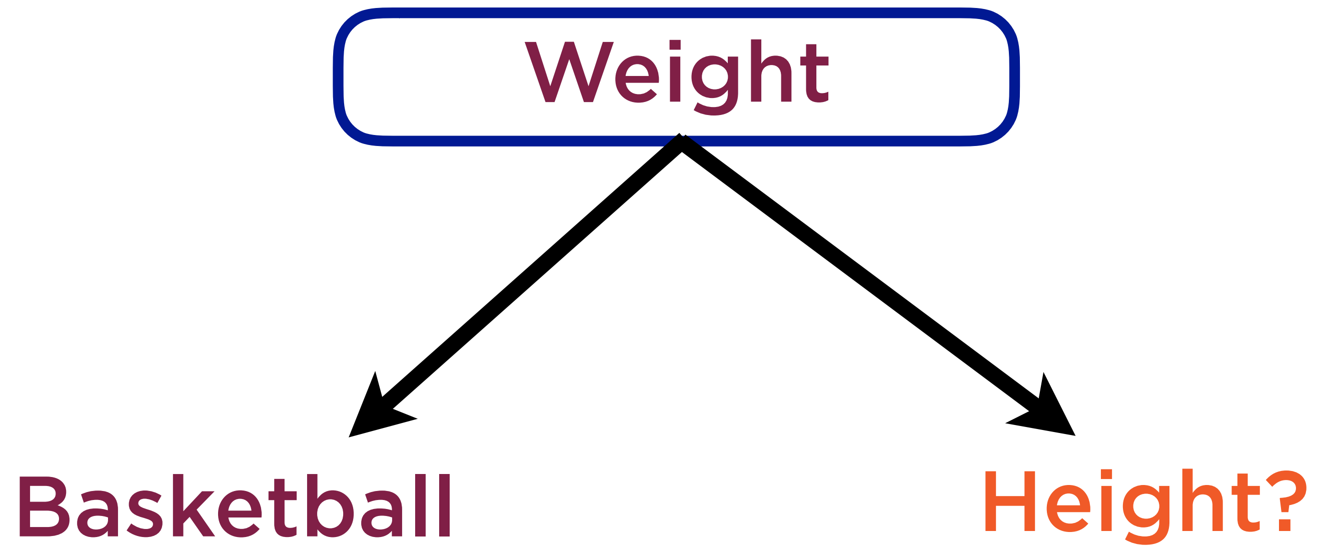
And also quite heavy

Decision trees set up a tree structure on training data which helps make **decisions** based on **rules**

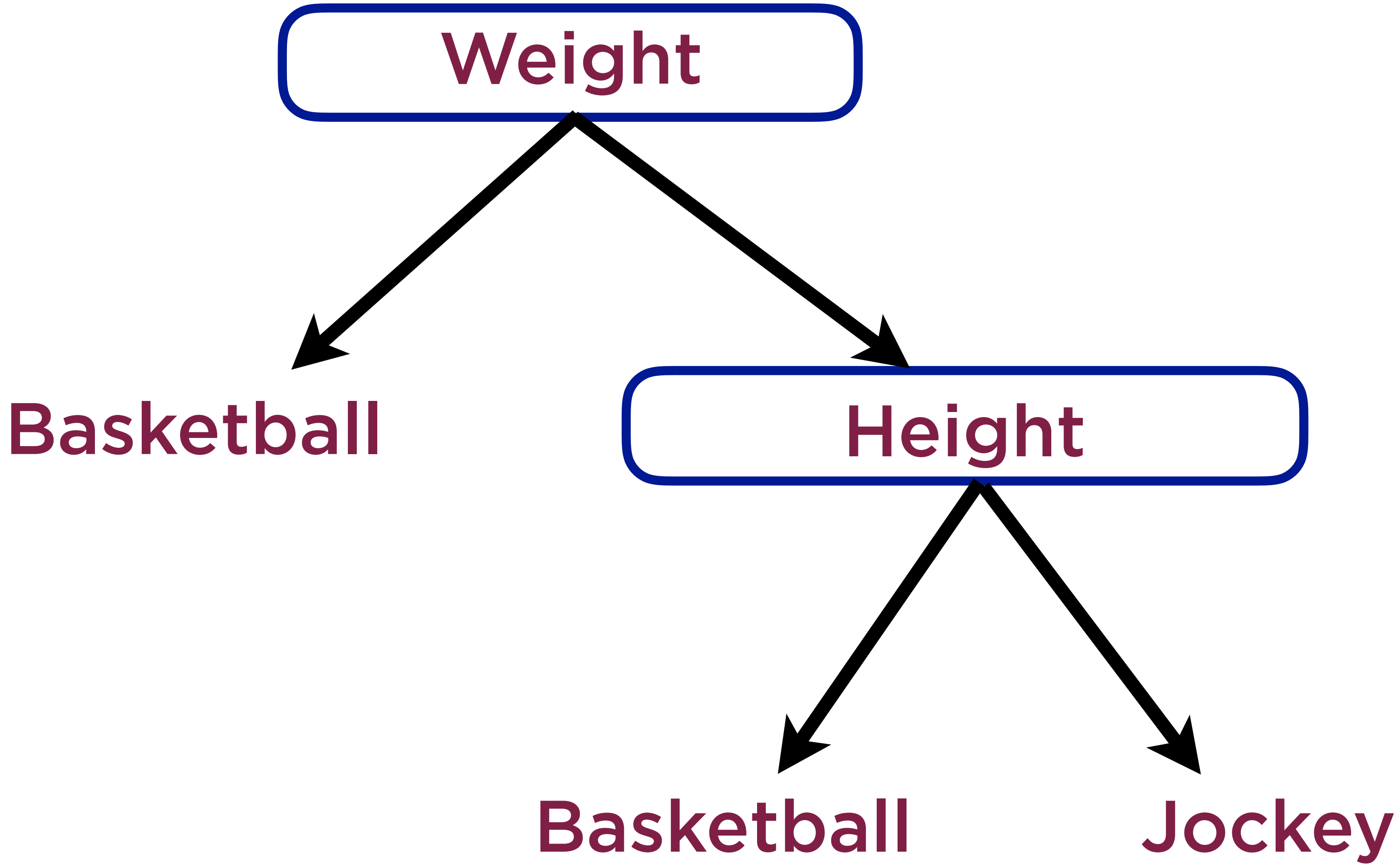
Fit Knowledge into Rules



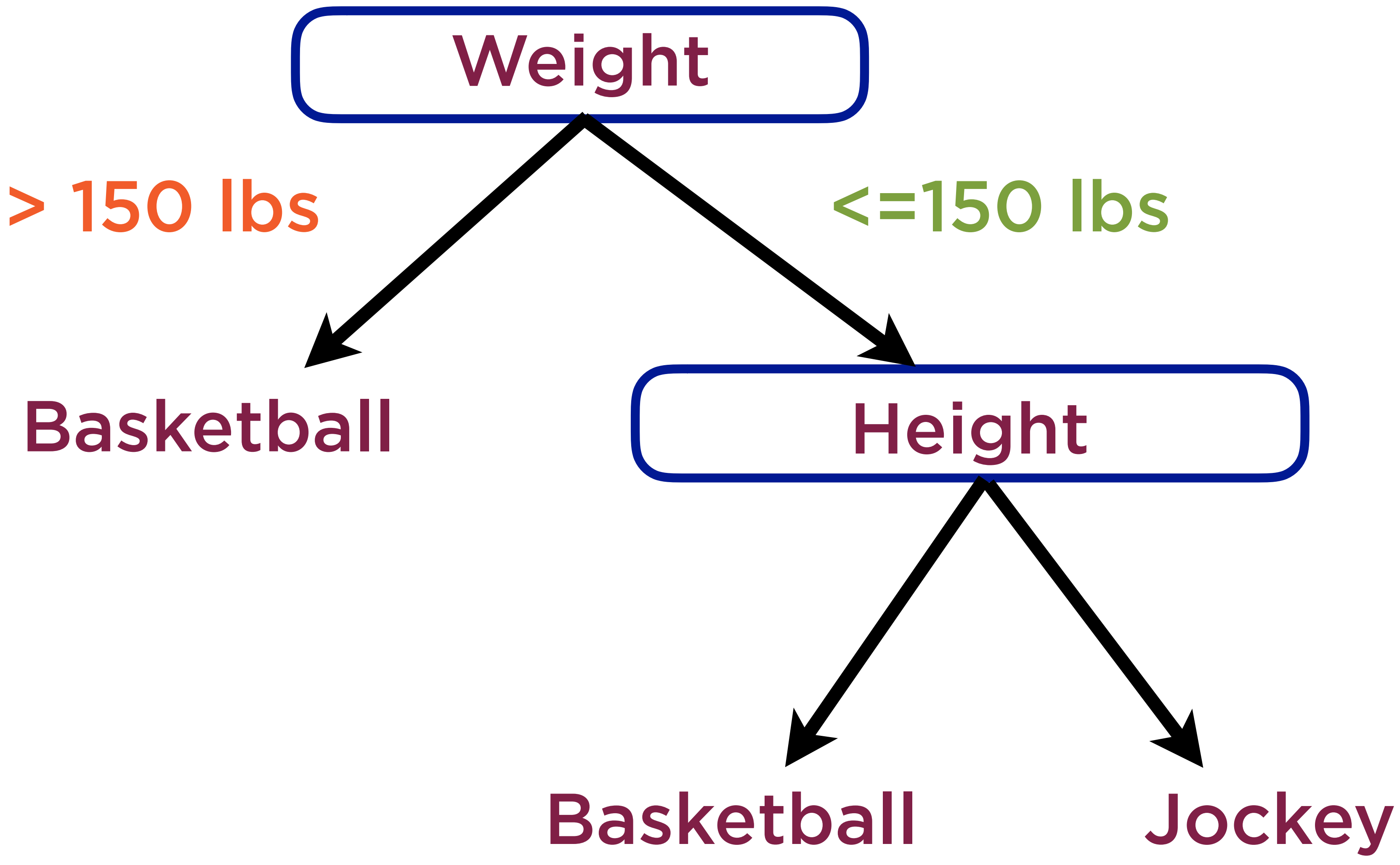
Decision Based on Weight



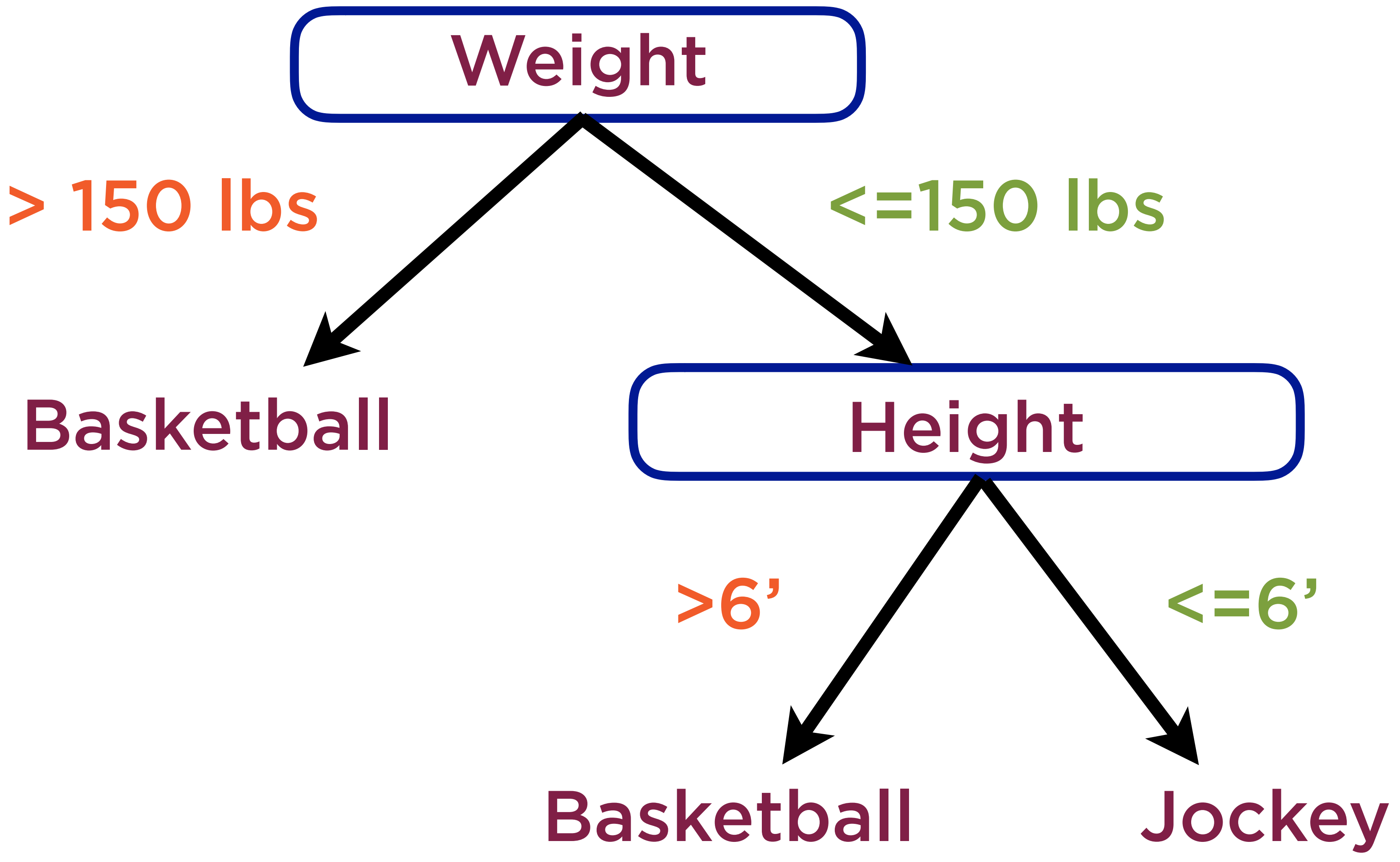
Decision Based on Weight



Fit Knowledge into Rules



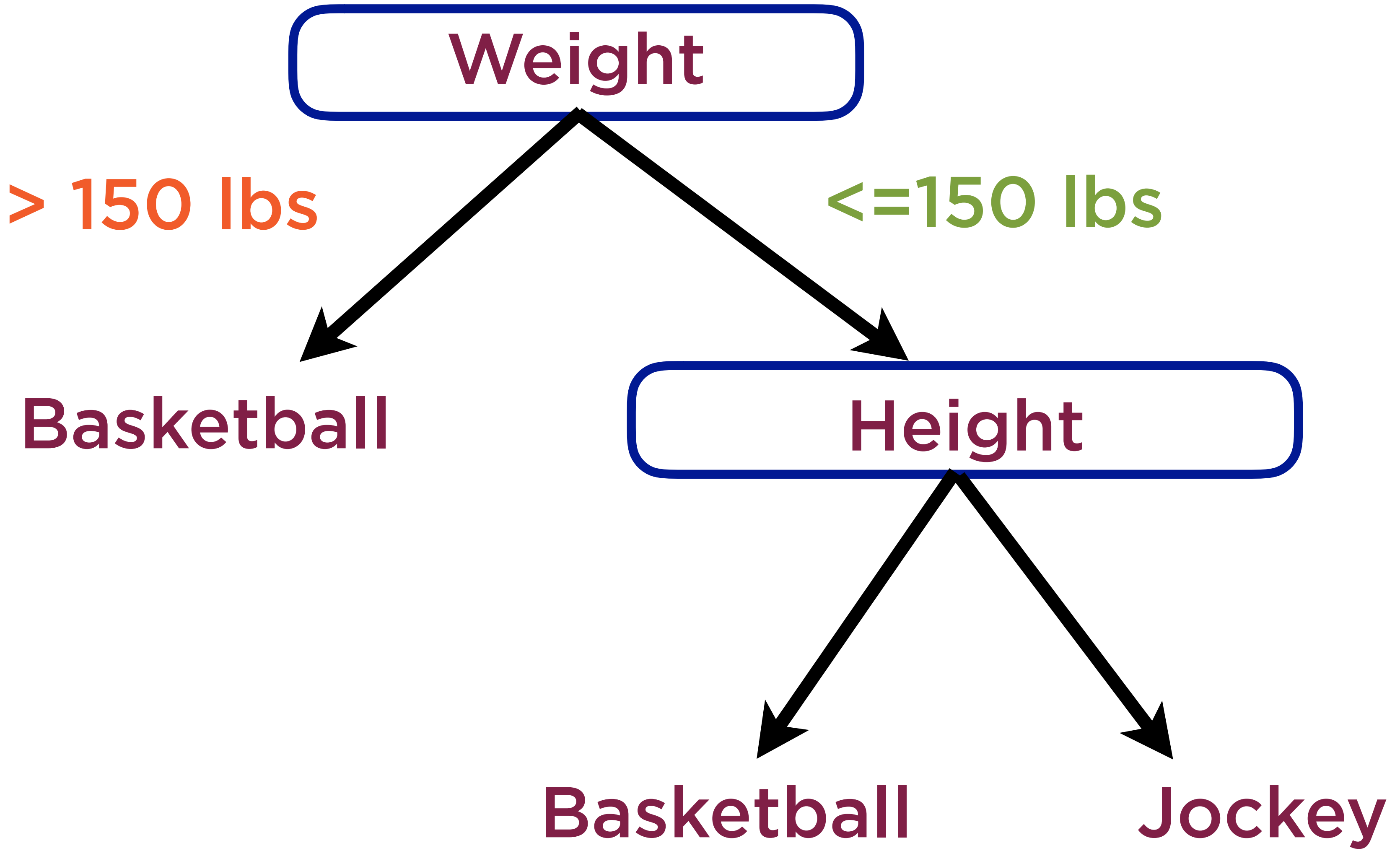
Fit Knowledge into Rules



Decision Tree

Fit knowledge into rules

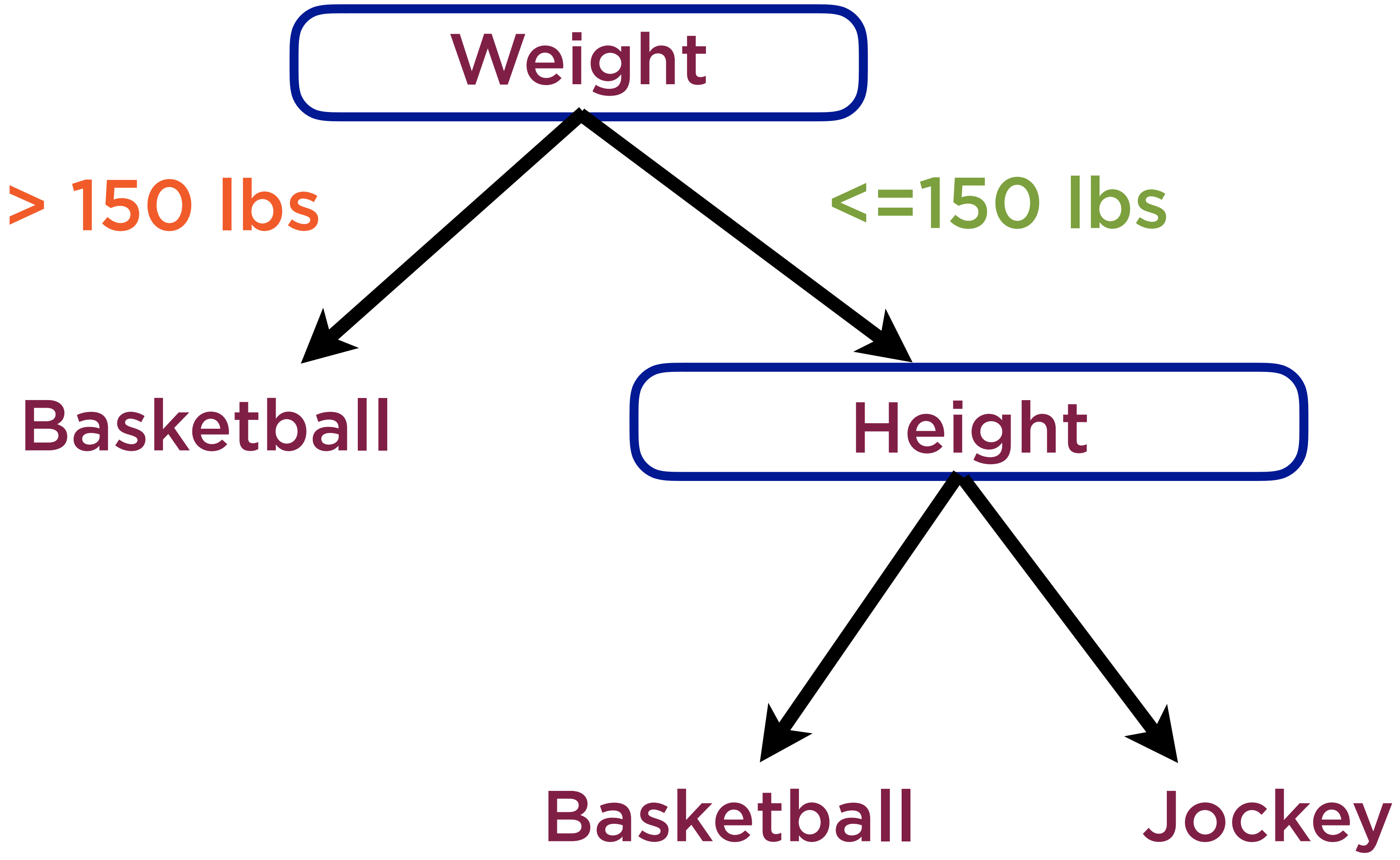
Each rule involves a threshold



Decision Tree

Order of decision variables matters

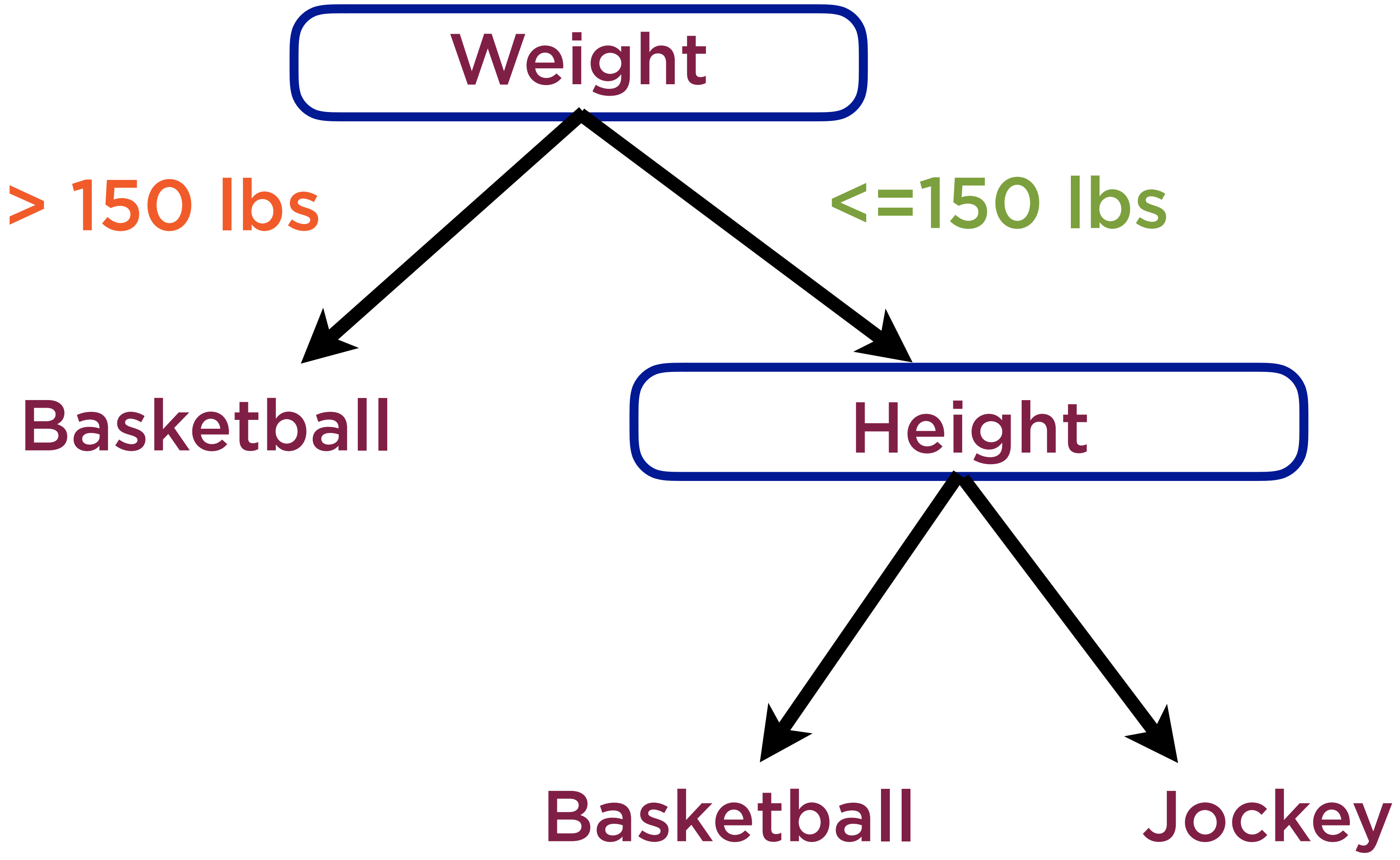
Rules and order found using ML



Splitting a Decision Tree

Tree selects the **best** feature

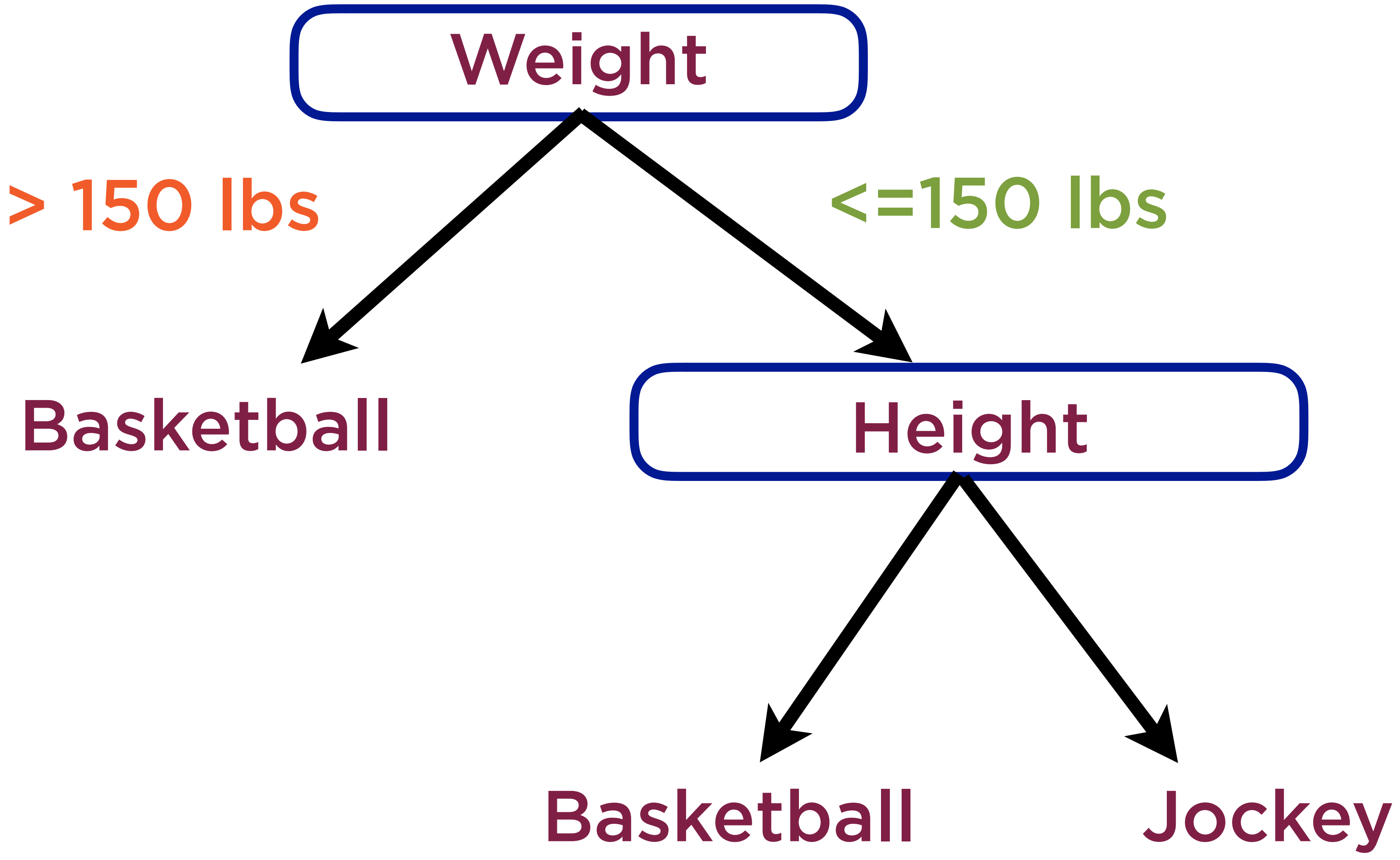
And finds the best **threshold** for the feature



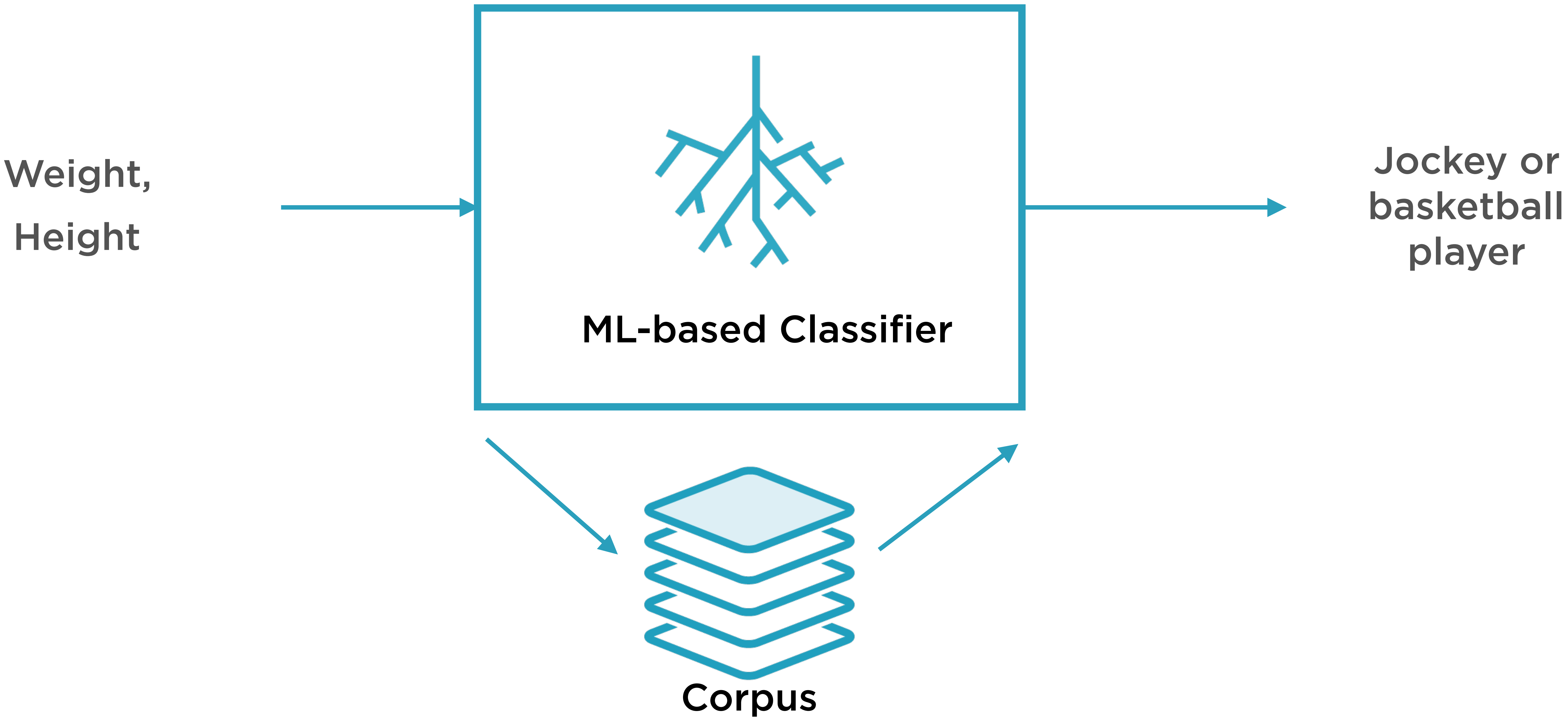
Decision Tree

"CART"

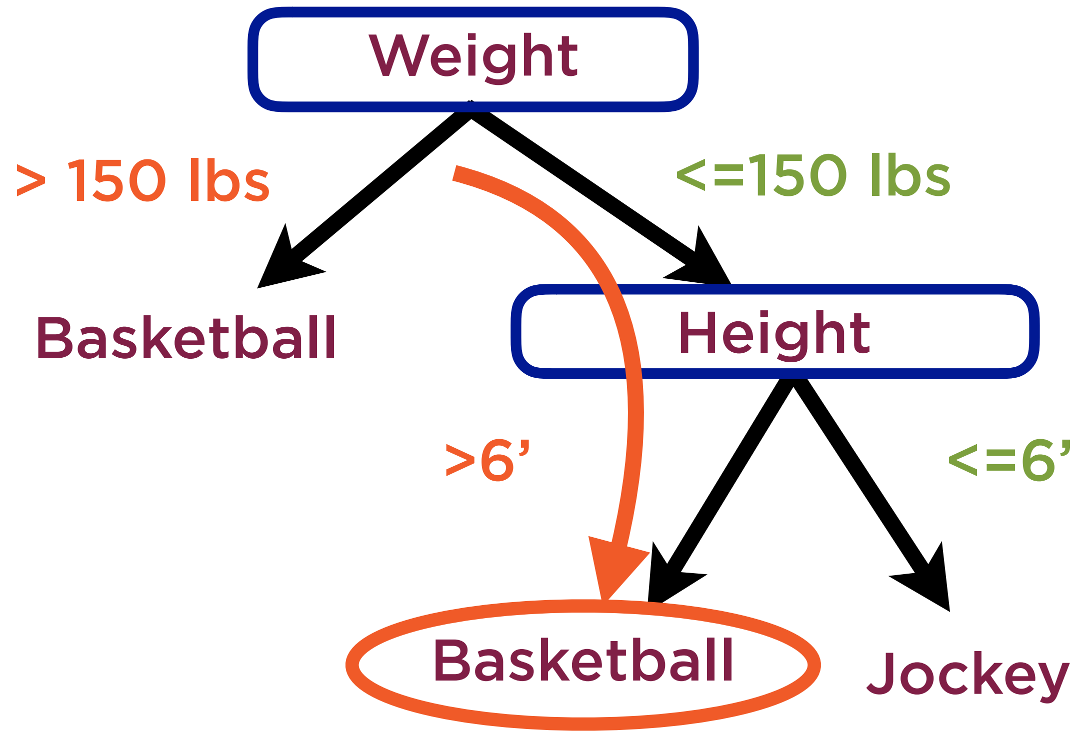
Classification And Regression Tree



Decision Trees for Classification



Decision Trees for Classification

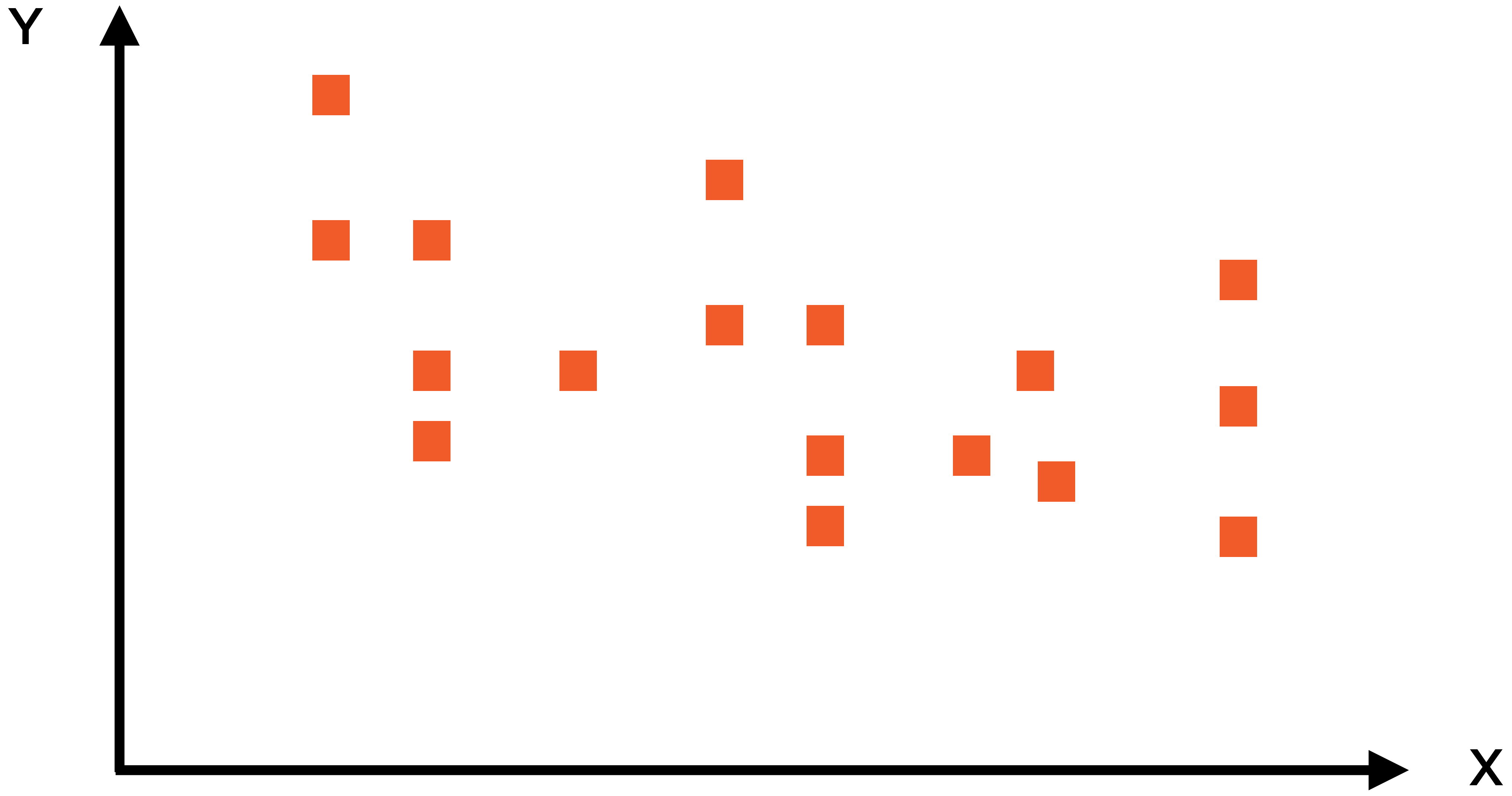


Traverse tree to find right node

Return most frequent label of all training data points in that node

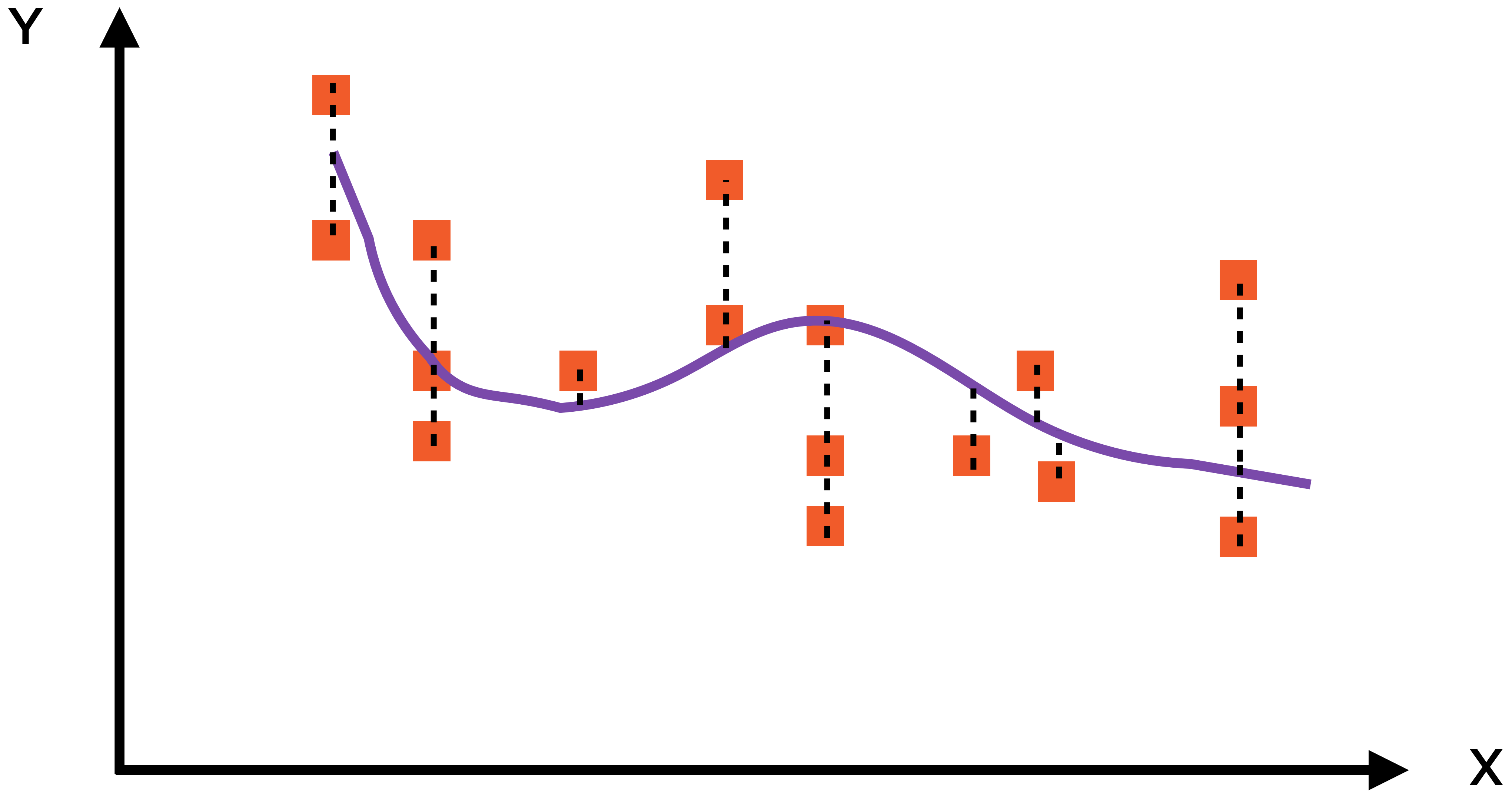
Ensemble Learning to Mitigate Overfitting

Connecting the Dots



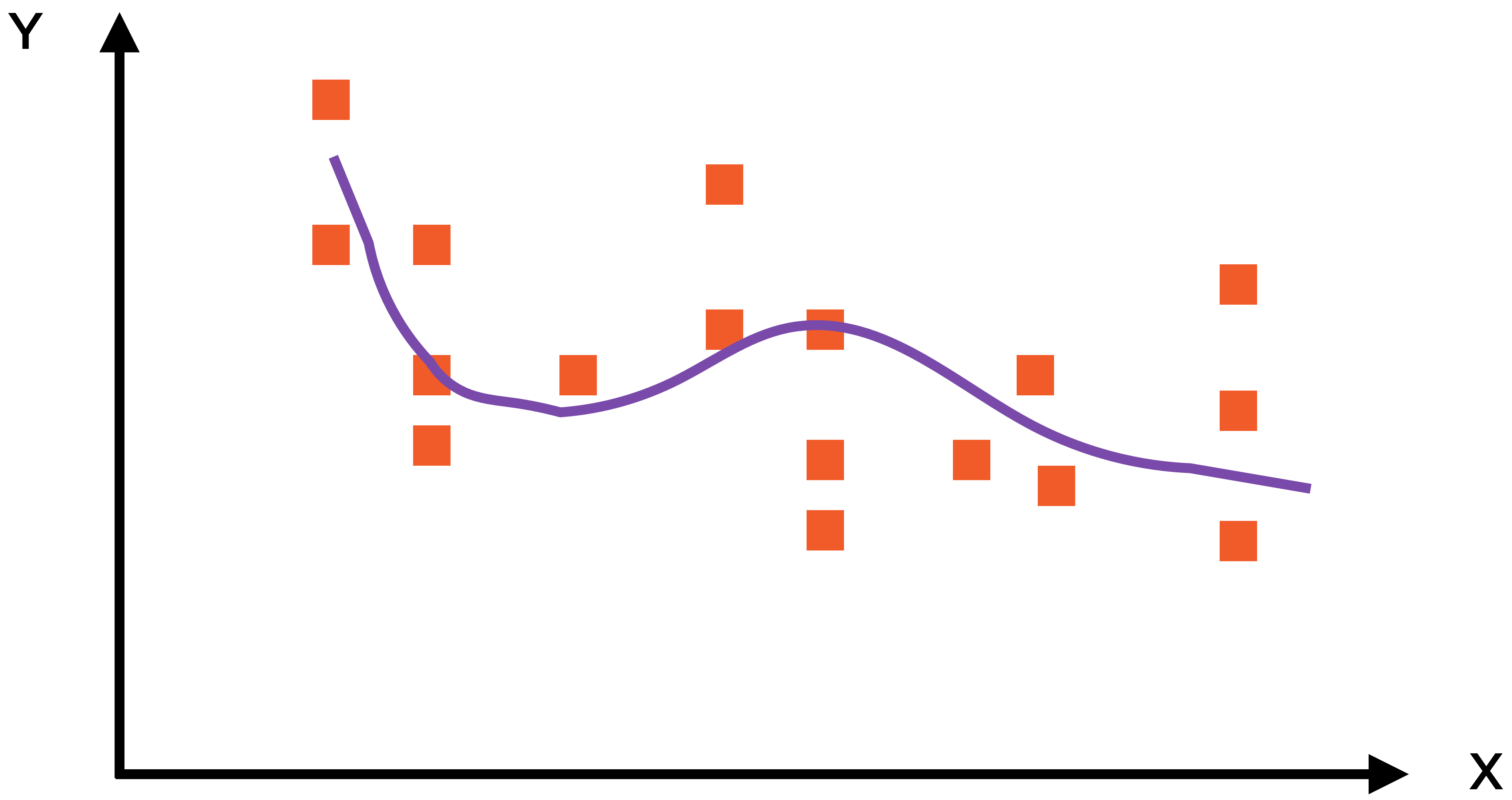
Challenge: Fit the "best" curve through these points

Good Fit?



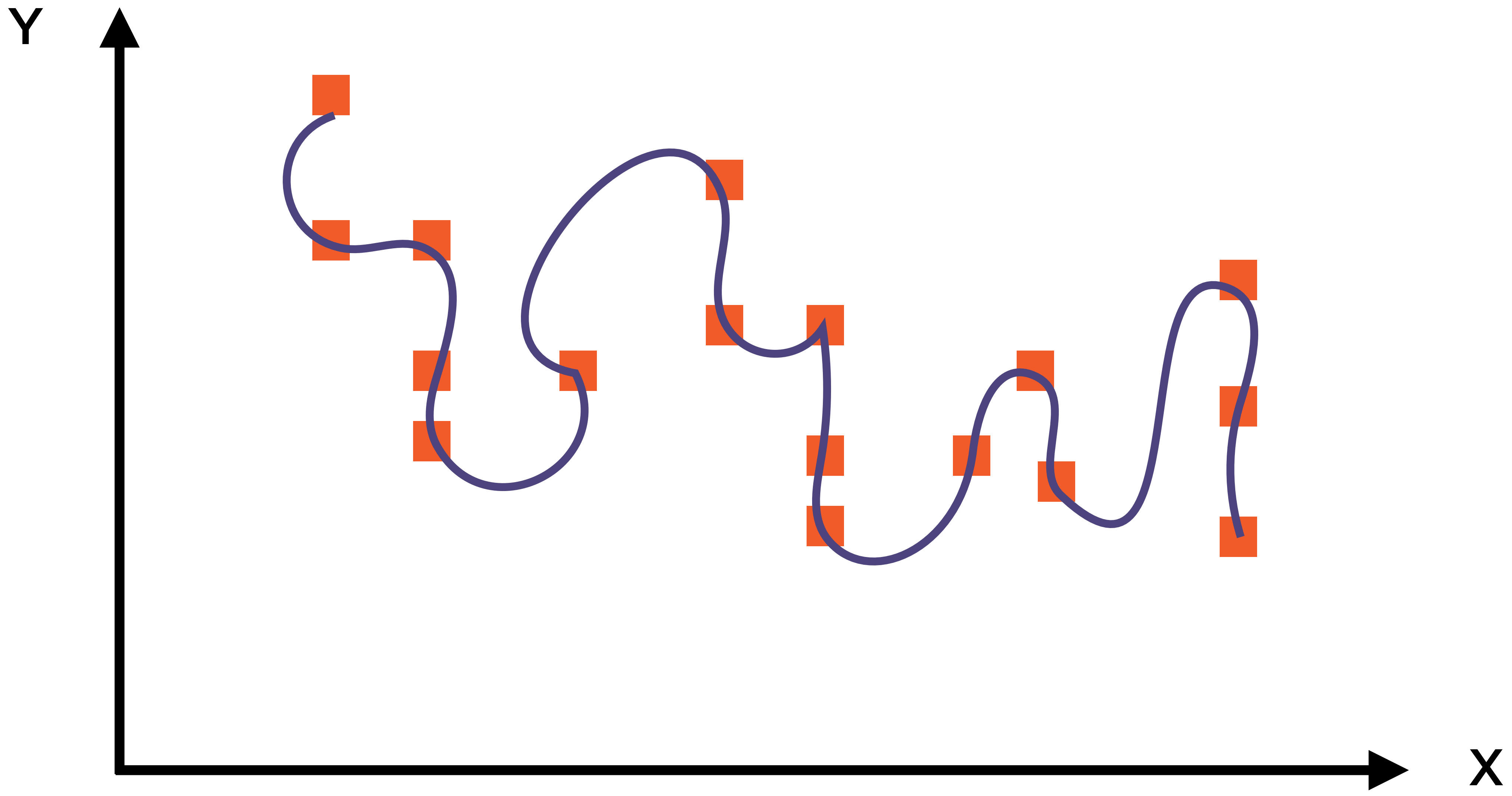
A curve has a "good fit" if the distances of points from the curve are small

Connecting the Dots



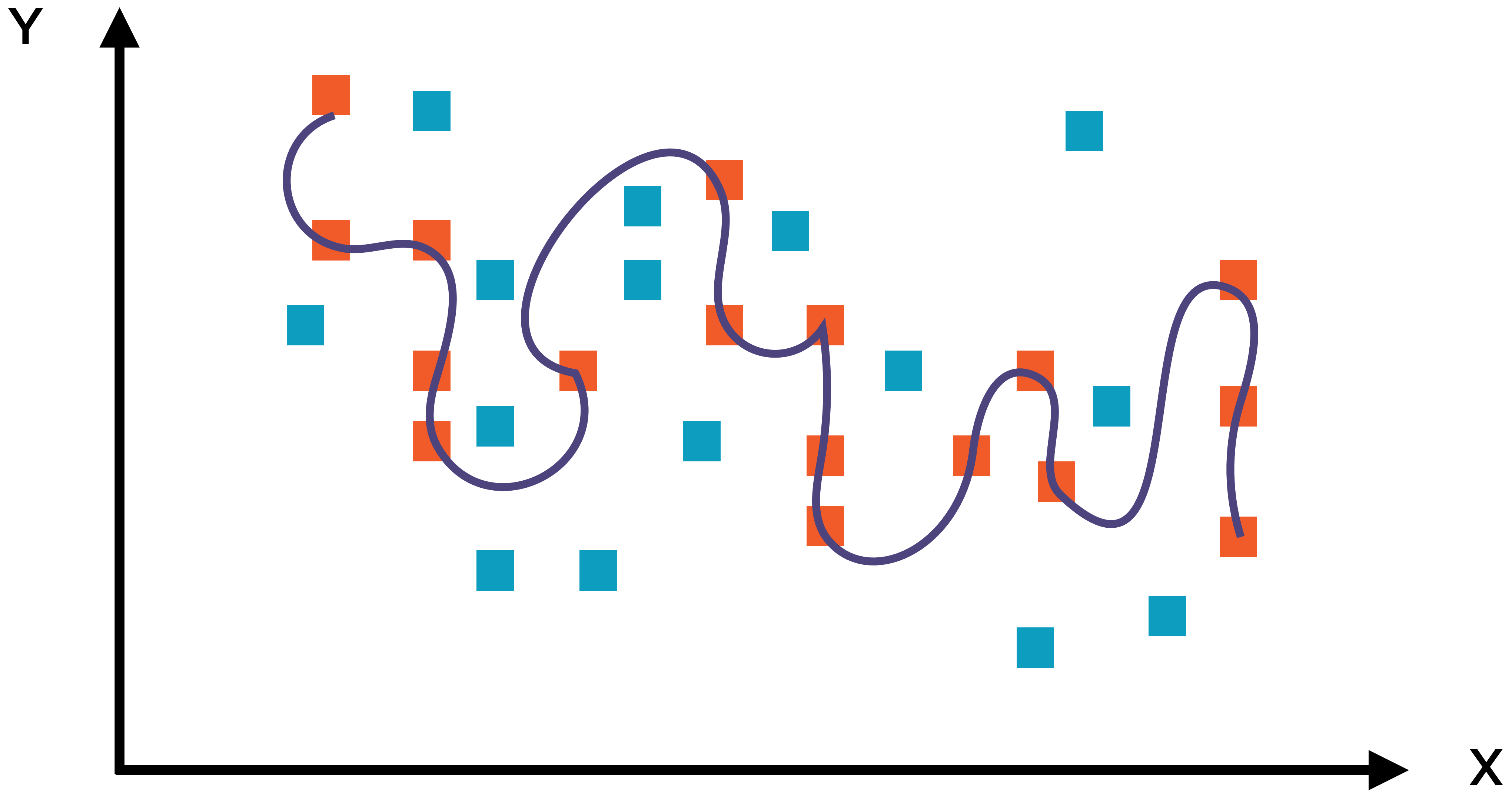
We could draw a pretty complex curve

Connecting the Dots



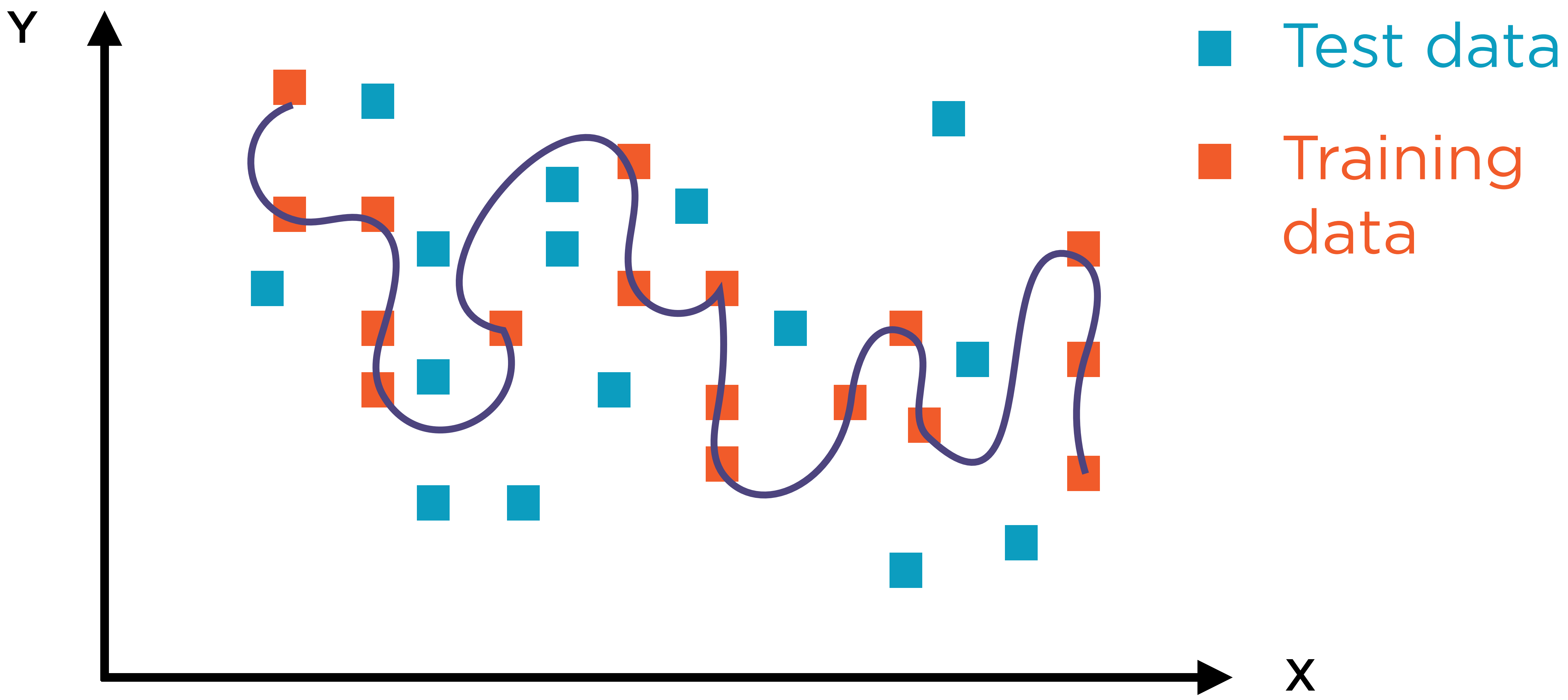
We can even make it pass through every single point

Connecting the Dots



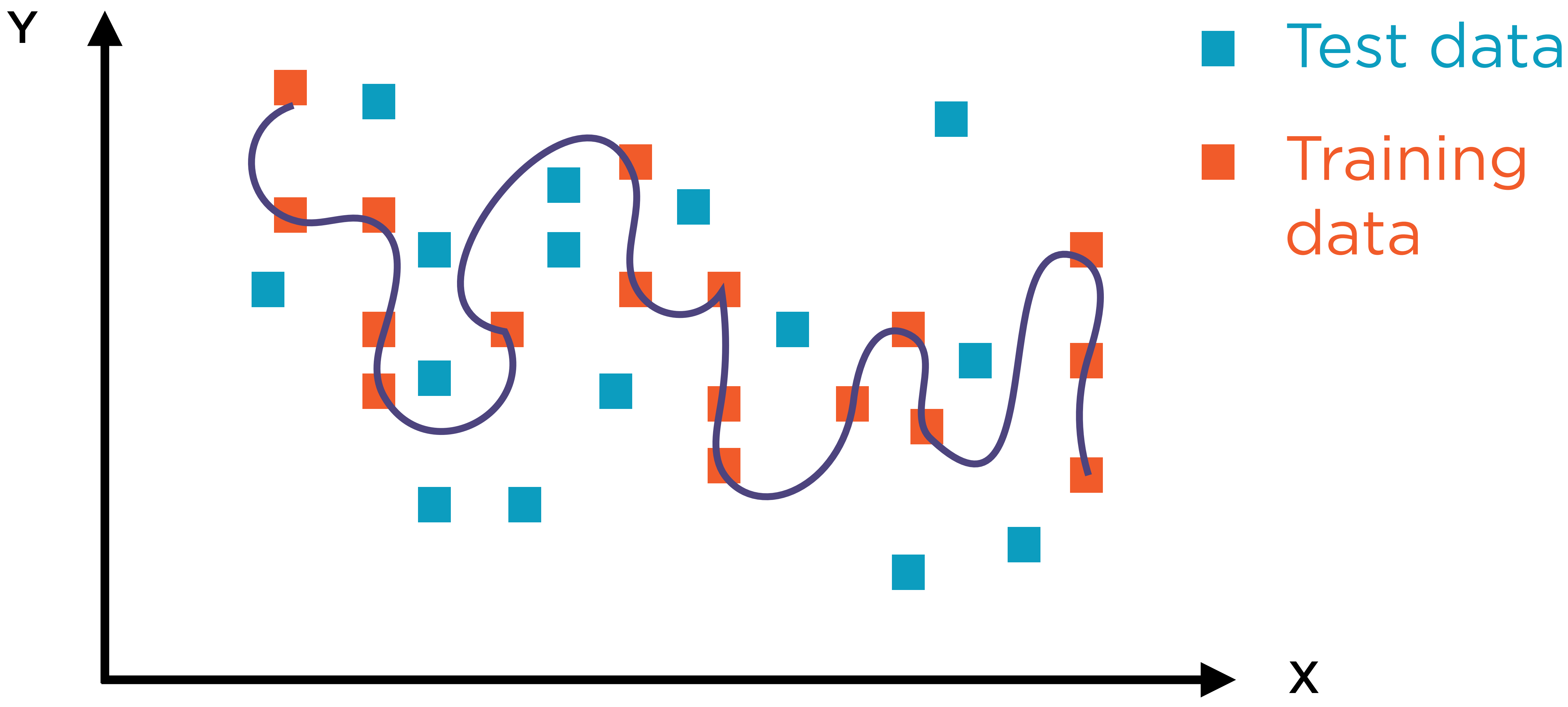
But given a new set of points, this curve might perform quite poorly

Connecting the Dots



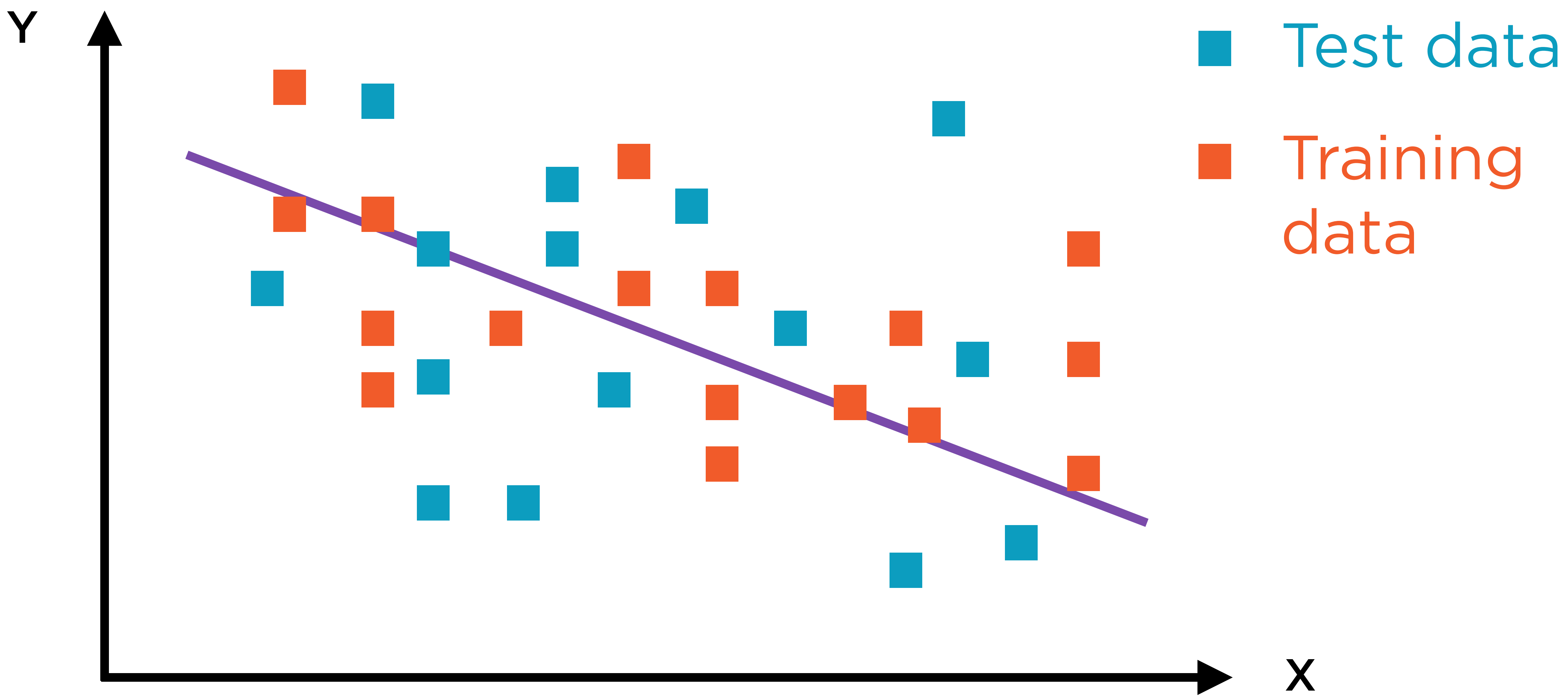
The original points were "training data", the new points are "test data"

Overfitting



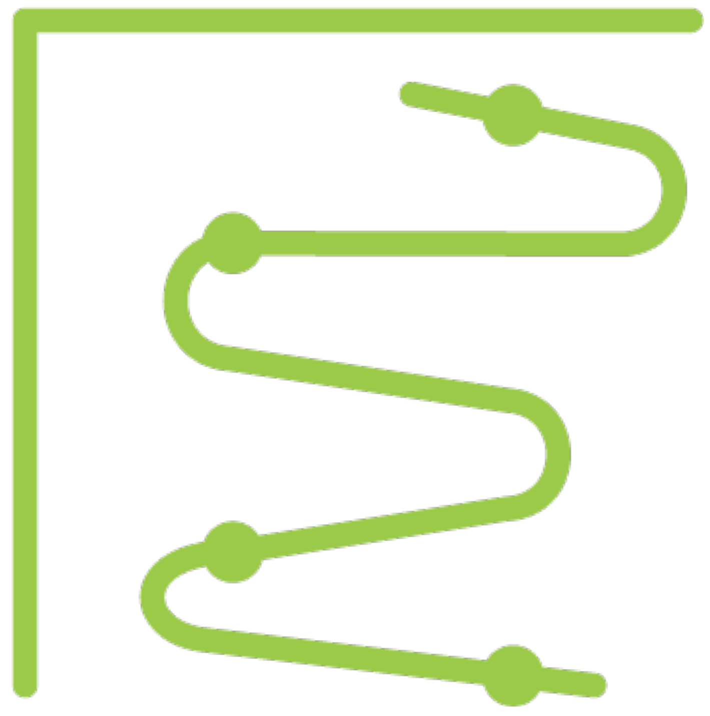
Great performance in training, poor performance in real usage

Connecting the Dots



A simple straight line performs worse in training, but better with test data

Overfitting



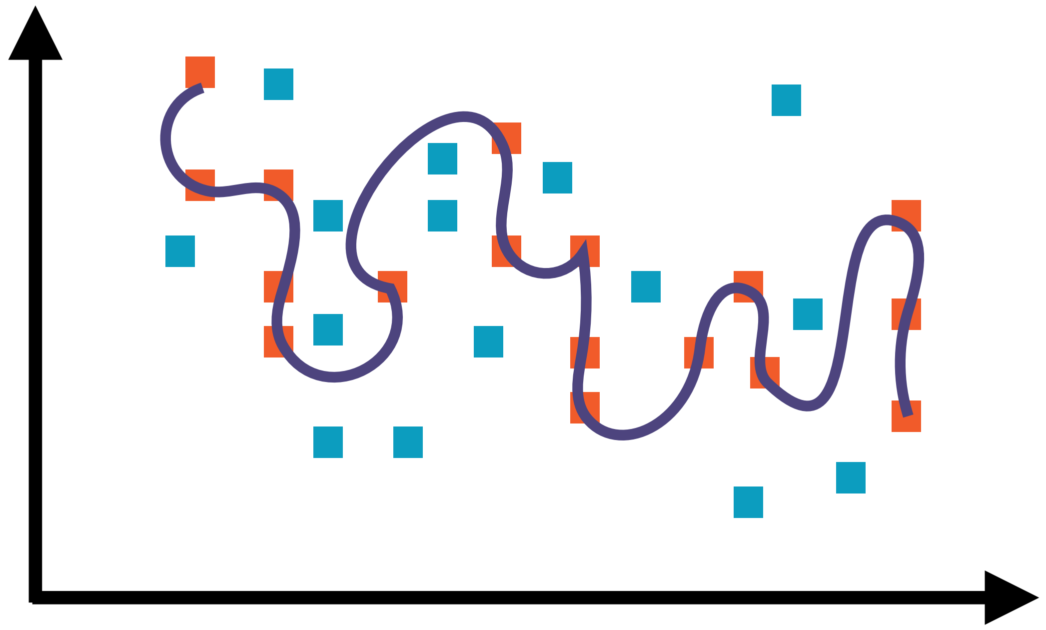
Model has memorized the training data

Low training error

Does not work well in the real world

High test error

Cause of Overfitting

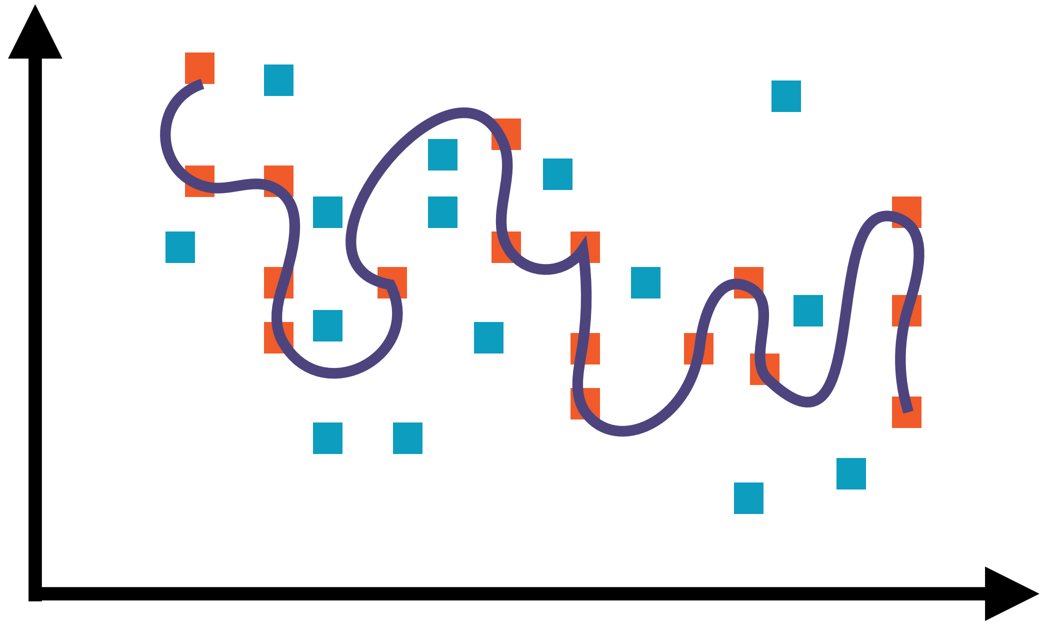


Sub-optimal choice in the bias-variance trade-off

An overfitted model has:

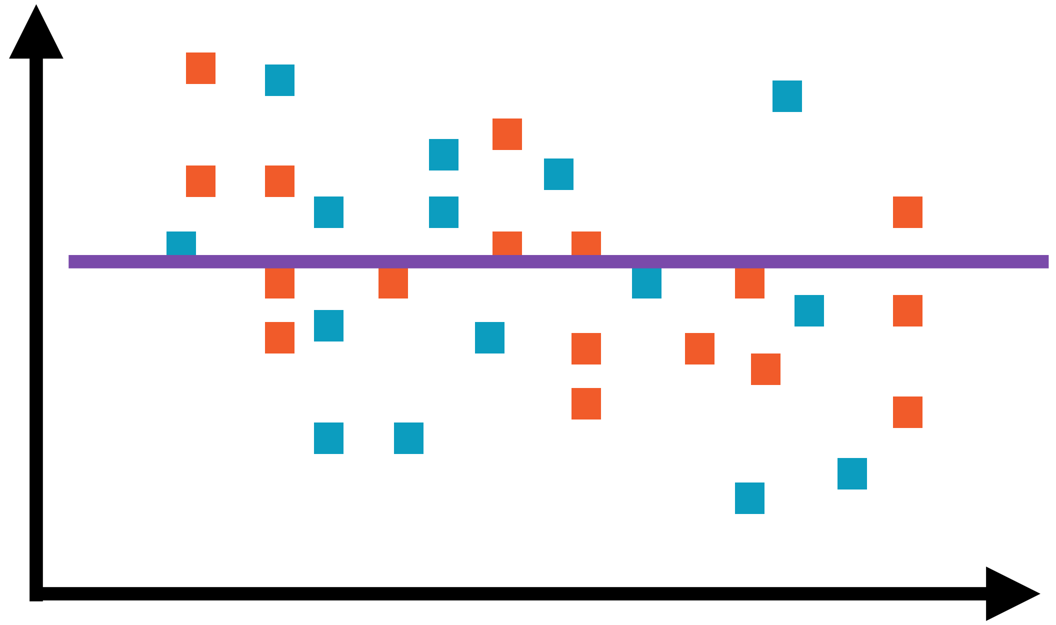
- high variance error
- low bias error

Bias



Low bias

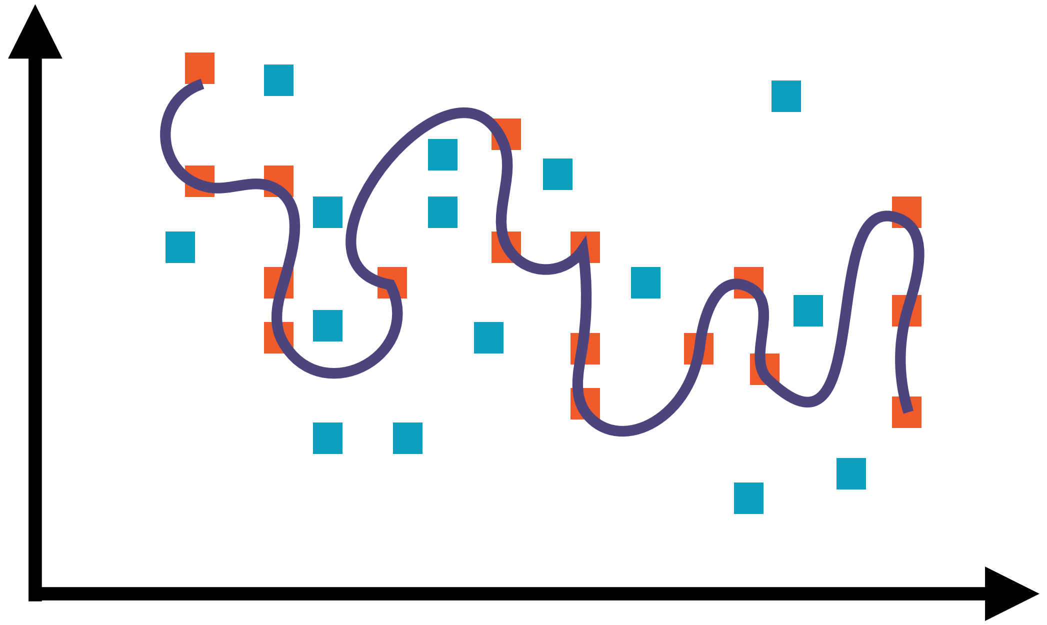
Few assumptions about the underlying data



High bias

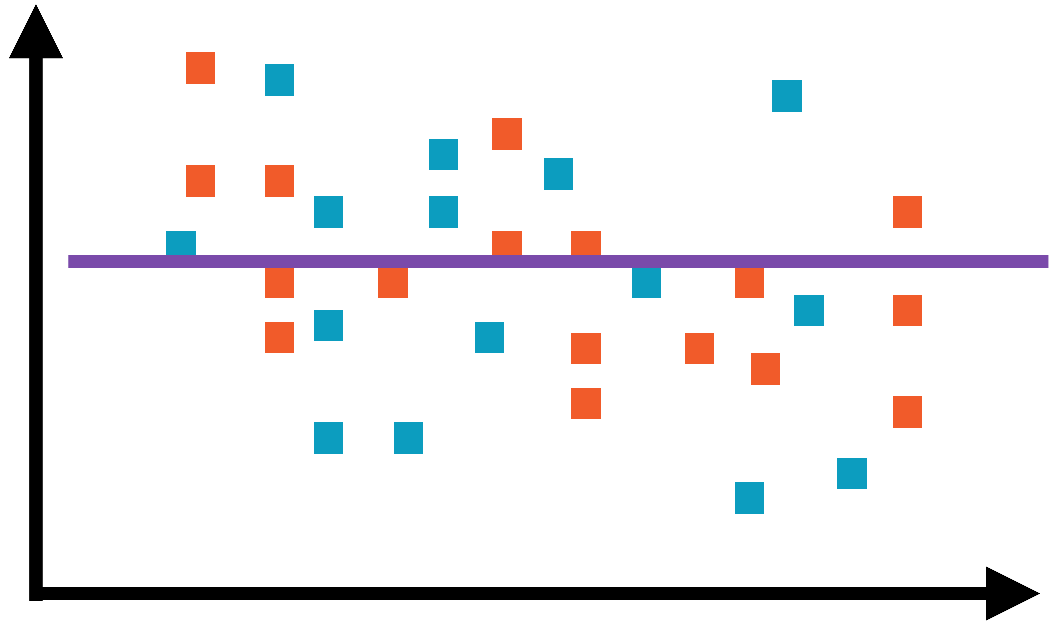
More assumptions about the underlying data

Bias



Model too complex

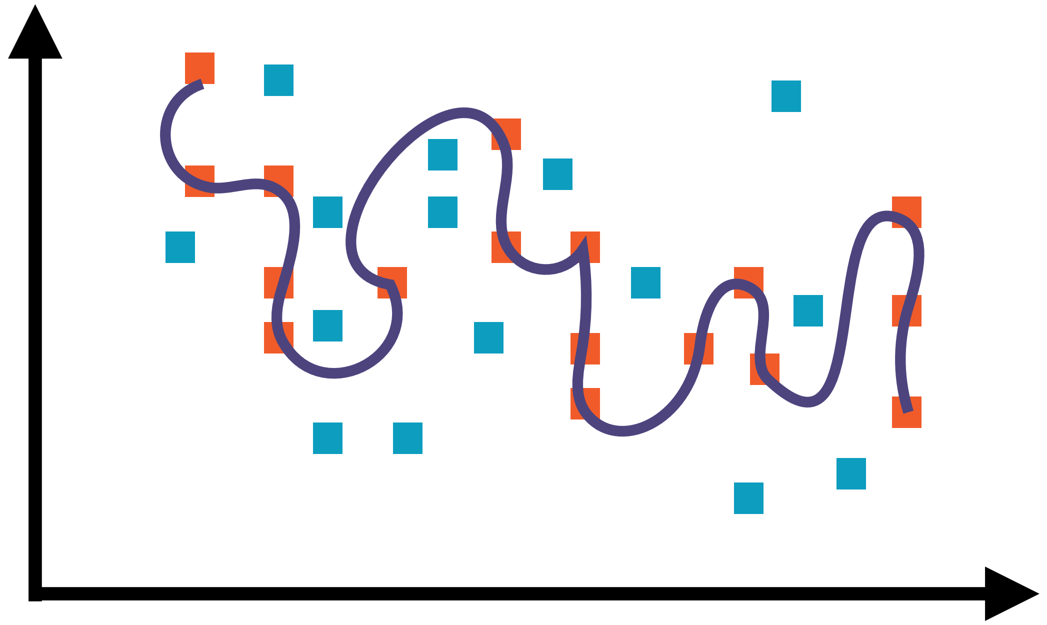
Training data all-important, model parameter counts for little



Model too simple

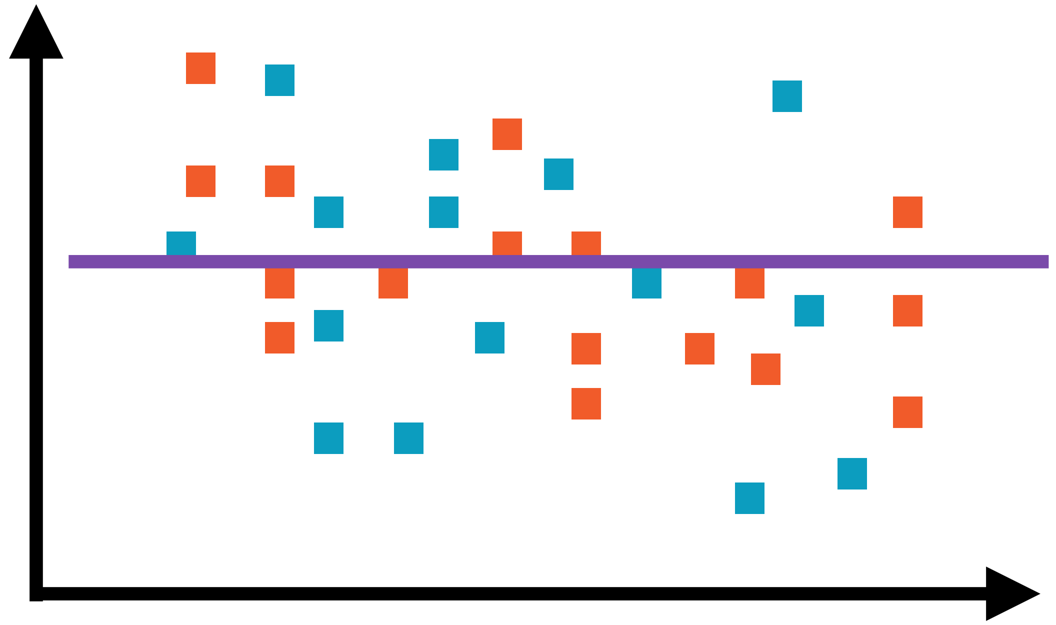
Model parameter all-important, training data counts for little

Variance



High variance

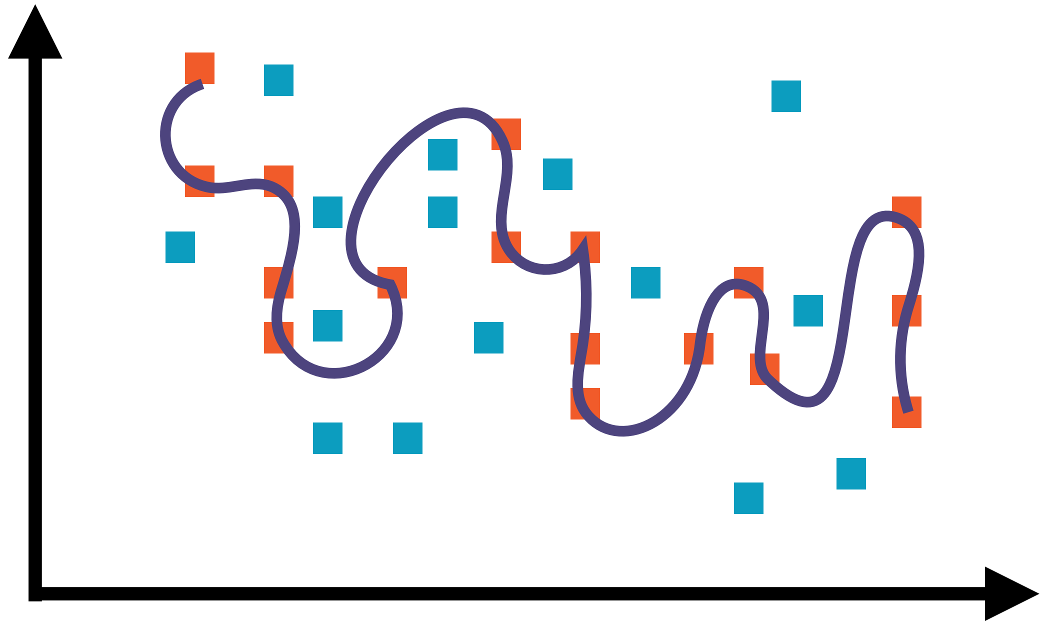
The model changes significantly when training data changes



Low variance

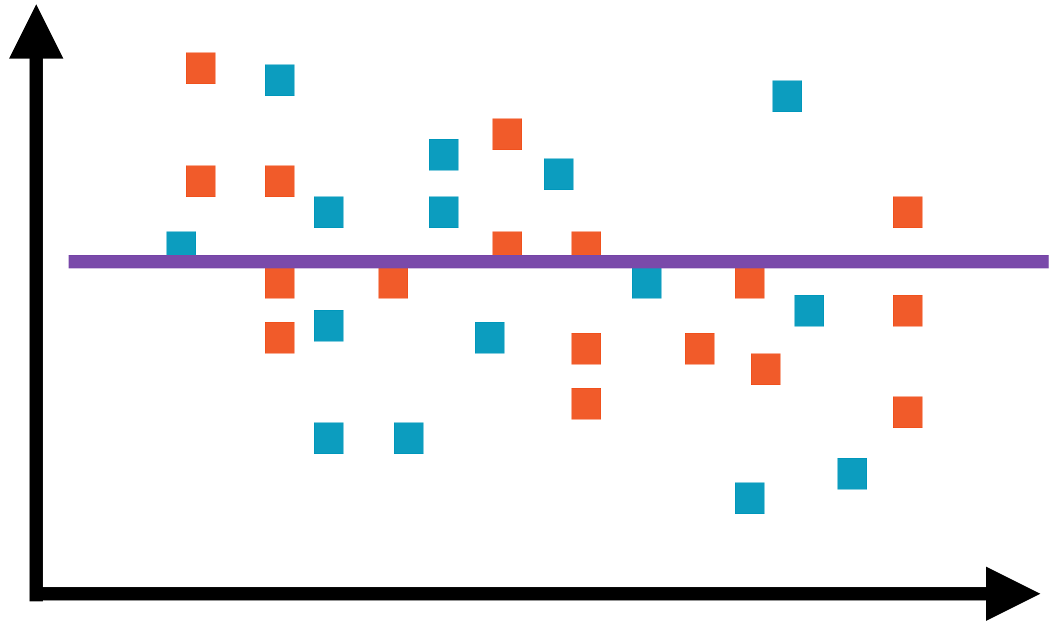
The model doesn't change much when the training data changes

Variance



Model too complex

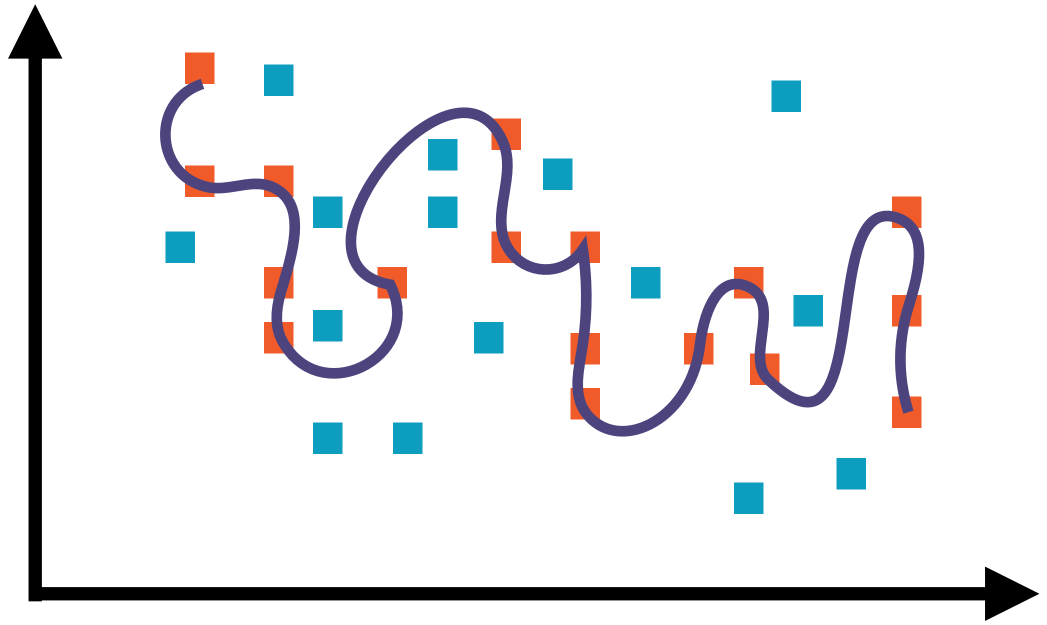
Model varies too much with changing training data



Model too simple

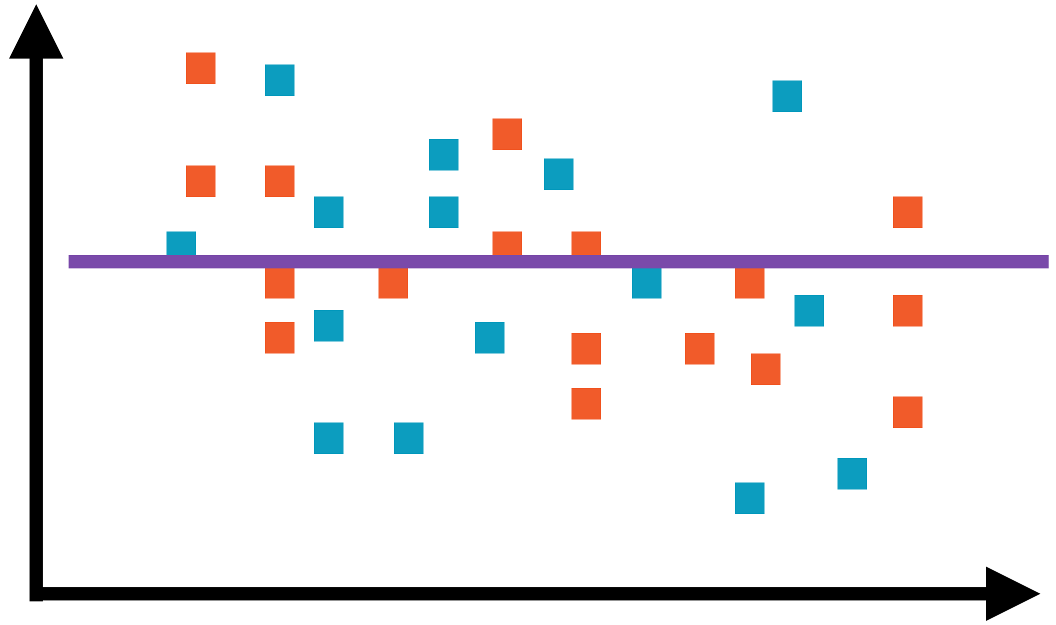
Model not very sensitive to training data

Bias-variance Trade-off



Model too complex

High variance error



Model too simple

High bias error

Preventing Overfitting

Regularization - Penalize complex models

Cross-validation - Distinct training and validation phases

Dropout (NNs only) - Intentionally turn off some neurons during training

Ensemble Learning an important technique to mitigate overfitting

Demo

Exploring the environment and tools

Demo

Performing hard and soft voting using the VotingClassifier

Summary

Ensemble learning to improve robustness and reduce overfitting

Different kinds of ensemble learning techniques

Averaging, boosting, voting, stacking

Built-in support for ensemble learning in scikit-learn

Implementing hard and soft voting in scikit-learn