MACHINE LEARNING

MACHINE LEARNING

- Machine learning is the study and development of algorithms that can learn from and make predictions on data
- Machine learning has quite a bit of overlap with statistical modeling, optimization and data mining

MACHINE LEARNING

- Two broad classes
  - Unsupervised learning
  - Supervised learning

MACHINE LEARNING

- Unsupervised learning is:
  - having the computer learn general patterns in the data
  - without focusing on predicting a particular variable
- The data is unlabeled
- Typically some form of cluster analysis or partitioning method
- Meant to identify homogeneous subgroups

MACHINE LEARNING

- Examples of unsupervised learning
  - k-means clustering
  - hierarchical clustering
  - Self-organizing maps
  - Mixture models

SUPERVISED LEARNING

- One variable is “labelled”, in that we know the truth
  - Class labels (high risk/low risk, tumor/normal)
  - Continuous labels (income, age)
- Objective is to try and predict the label from other predictors (called “features” in this literature)

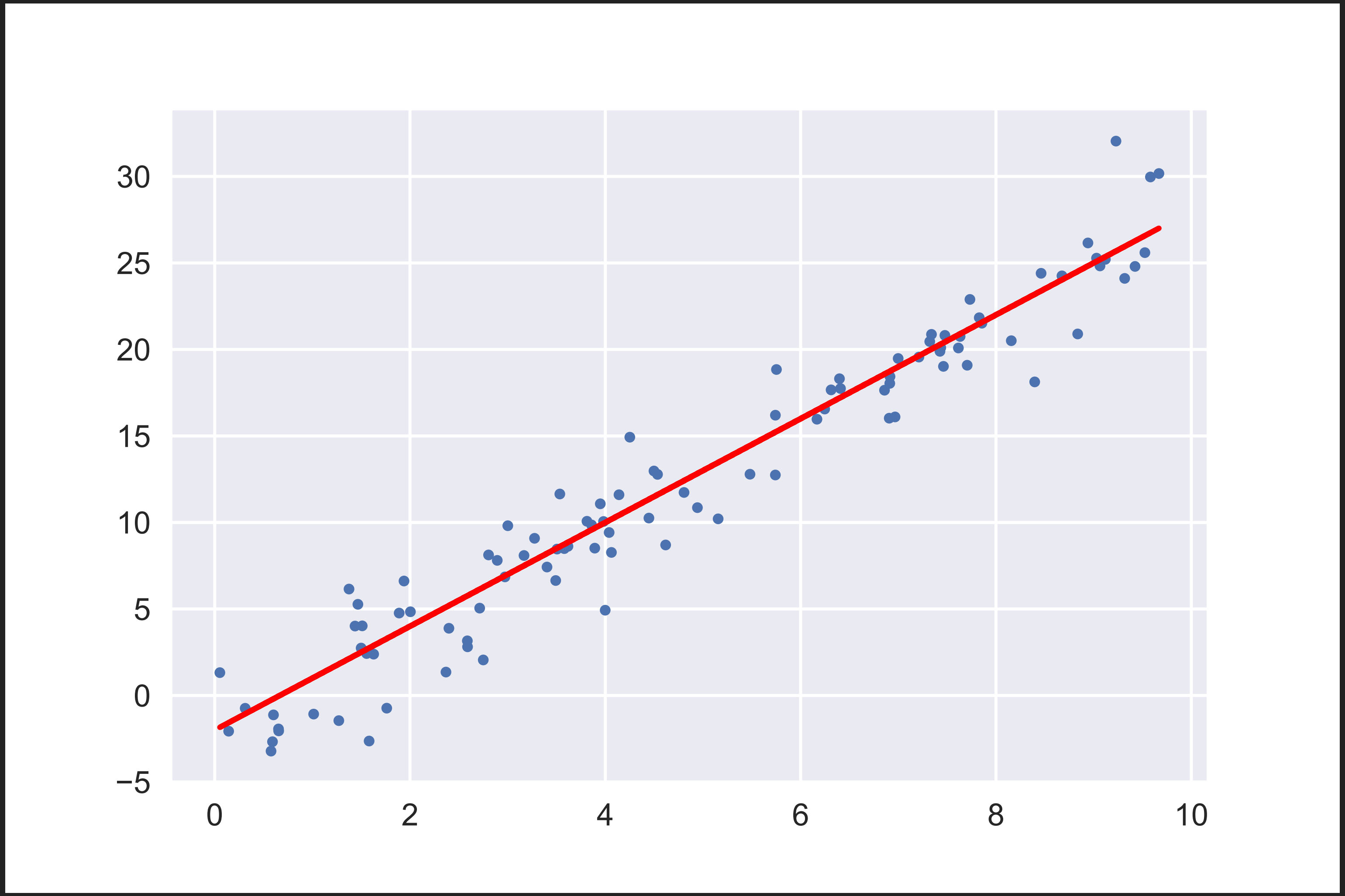
SUPERVISED LEARNING

- If labels are discrete -> Classification
- If labels are continuous -> Regression

- This distinction is somewhat arbitrary

SUPERVISED LEARNING

- We already know one example of supervised learning
    - Linear regression



STATISTICS VS MACHINE LEARNING

- How do we learn to fit a linear regression?
- We optimize the sum of squared deviations

$$\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

- This is a global solution, i.e. we want that line that fits best on average

STATISTICS VS MACHINE LEARNING

- Machine learning takes a different approach
- It wants a model that, once trained on a data set, will accurately predict on other data sets
- This is an important distinction

STATISTICS VS MACHINE LEARNING

- A model that is fit really well to the data (R^2 is large)
  - may actually overfit the data
  - may not work well on new data
- So our perspective has to change here.

STATISTICS VS MACHINE LEARNING

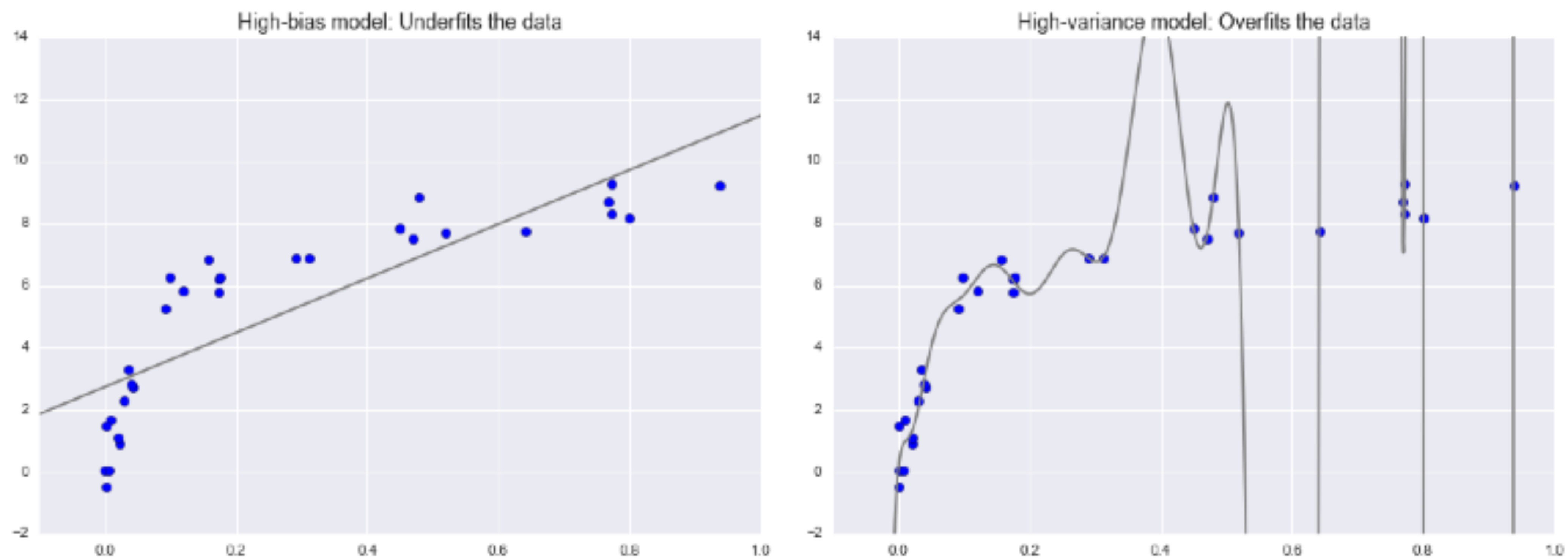
- The standard way to fit machine learning methods is:
  - Split your data set into a training set and a test set
  - Train your data on the training set
  - See how well it performs (predictively) on the test set
  - Take that model which does well on the test set

THE VARIANCE-BIAS TRADE-OFF

- Some models are too simple
  - They under-fit the data
  - They do equally well on training and test data
  - These are *high bias* models

THE VARIANCE-BIAS TRADEOFF

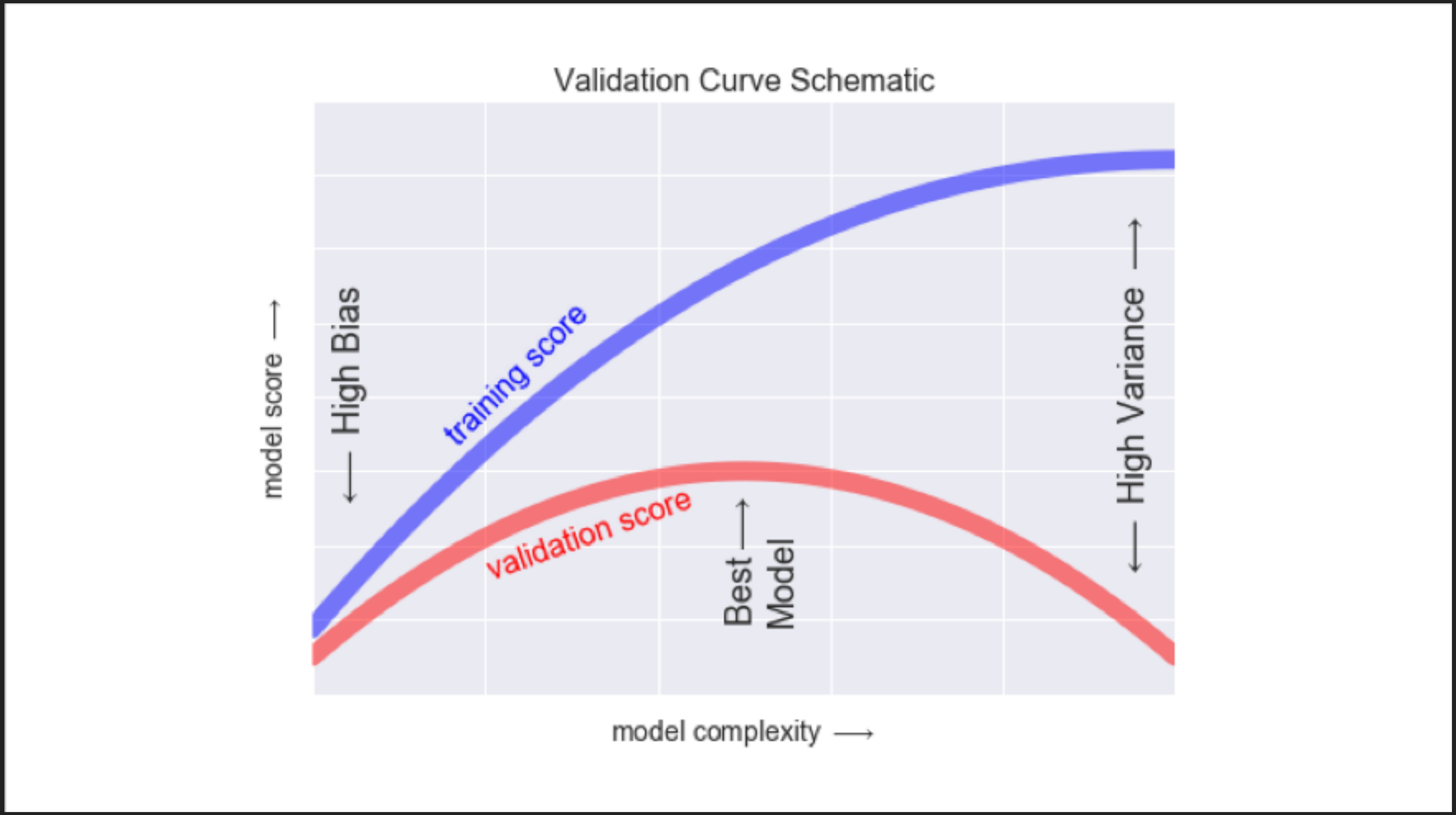
- Some models are too complex
  - They fit the data too well (overfitting)
  - Think interpolation
  - However they do very poorly on new data
  - These are termed *high variance* models



High-bias model: Underfits the data
High-variance model: Overfits the data
14
12
10
8
6
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
1.0

THE VARIANCE-BIAS TRADE-OFF

- We want a model that is low-variance, low-bias



MODEL PERFORMANCE

- Need a metric to see how well a model works
- Can’t rely on only one test data

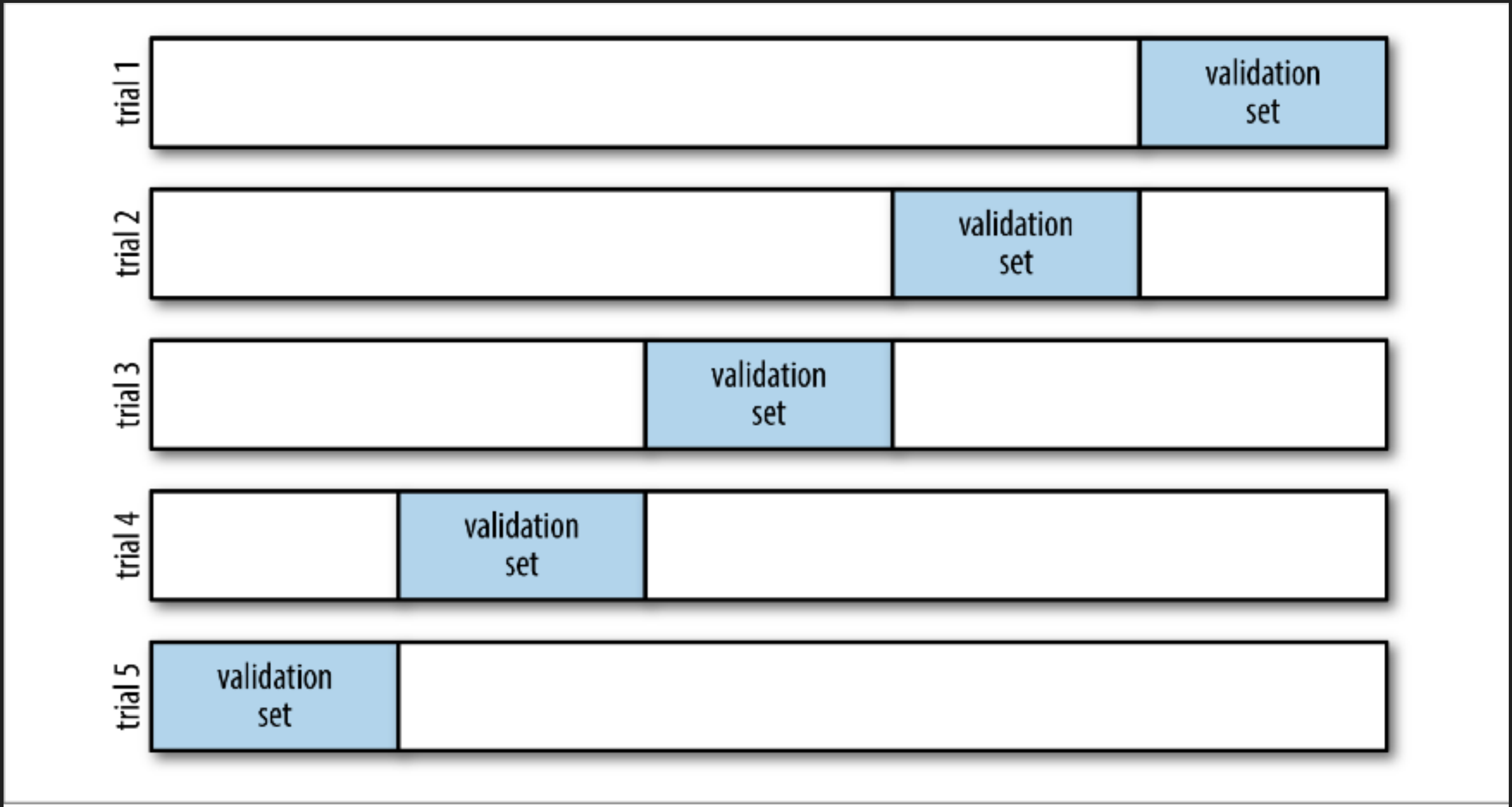
MODEL PERFORMANCE

- Classification
  - Accuracy (misclassification)
- Regression
  - Mean square error

MODEL PERFORMANCE

- Take multiple random splits to generate training and test data
- See overall performance across different splits
- Idea behind cross-validation

CROSS-VALIDATION



SUPERVISED LEARNING METHODS

- k-nearest neighbors
- Decision trees
- Ensemble methods
  - Random forests
  - Boosted trees
- Support Vector Machines

SUPERVISED LEARNING METHODS

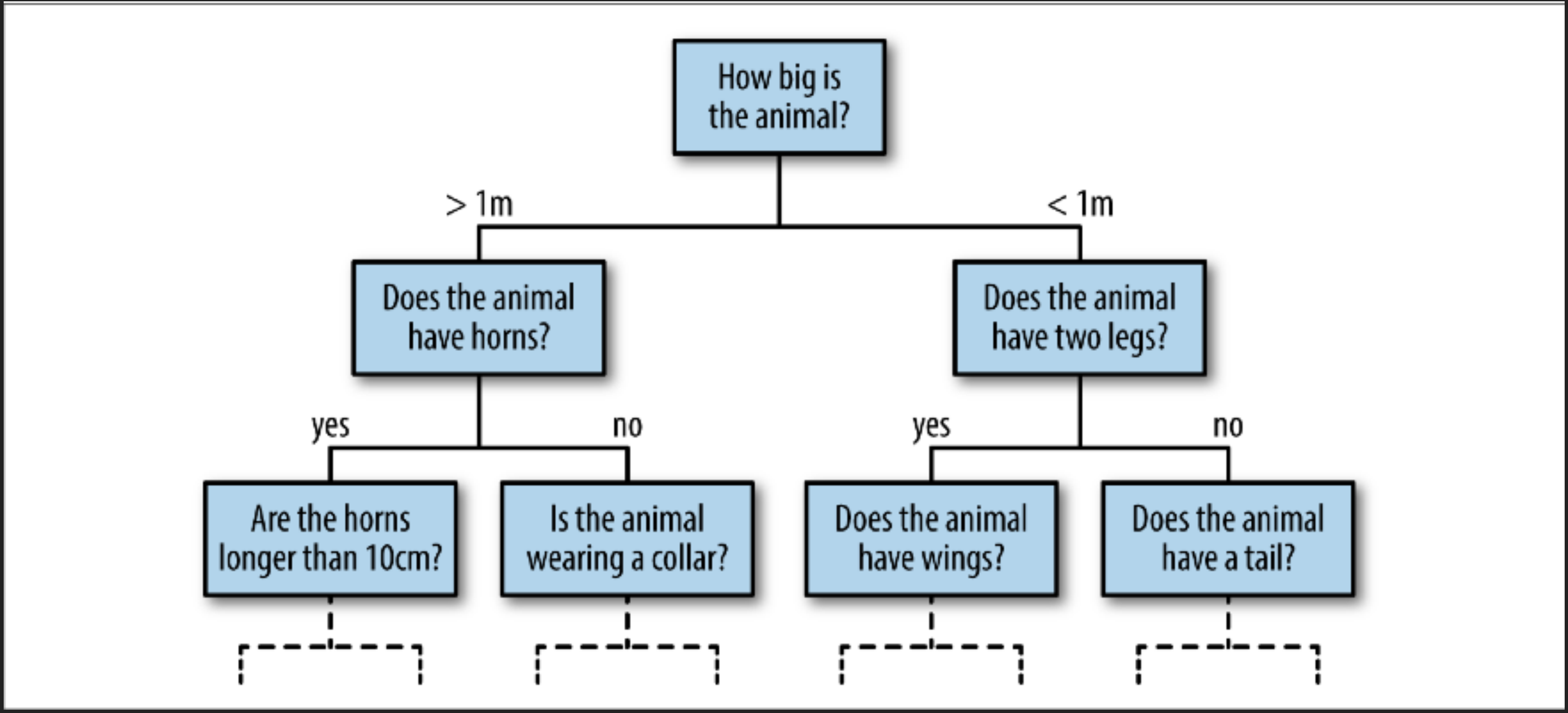
- These methods often look at **local** estimates rather than global estimates

K-NEAREST NEIGHBORS

- For each data point, find its k nearest neighbors in the predictor space
  - Decide on a distance metric
- The prediction at that data point is the
  - average of the labels (regression)
  - the most prevalent of the labels (classification)
  - observed in those neighbors

GOING FORWARD

- We'll be concentrating on decision trees



THE SCIKIT-LEARN PACKAGE

- This is the workhorse for machine learning
- We will use this extensively in fitting machine learning models to data
- General paradigm
  1. Import a ML algorithm from sklearn
  2. Define the characteristics of the model
  3. Fit training data to the model
  4. Predict on test data
  5. Compute performance metrics and cross-validation