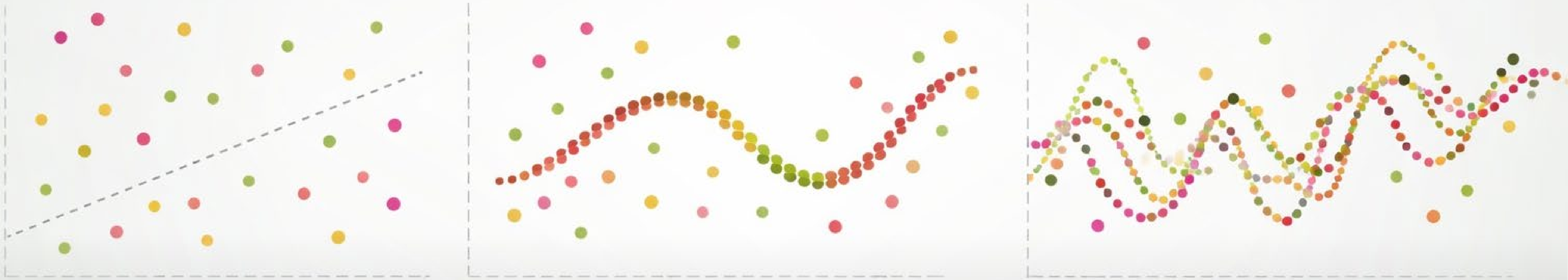


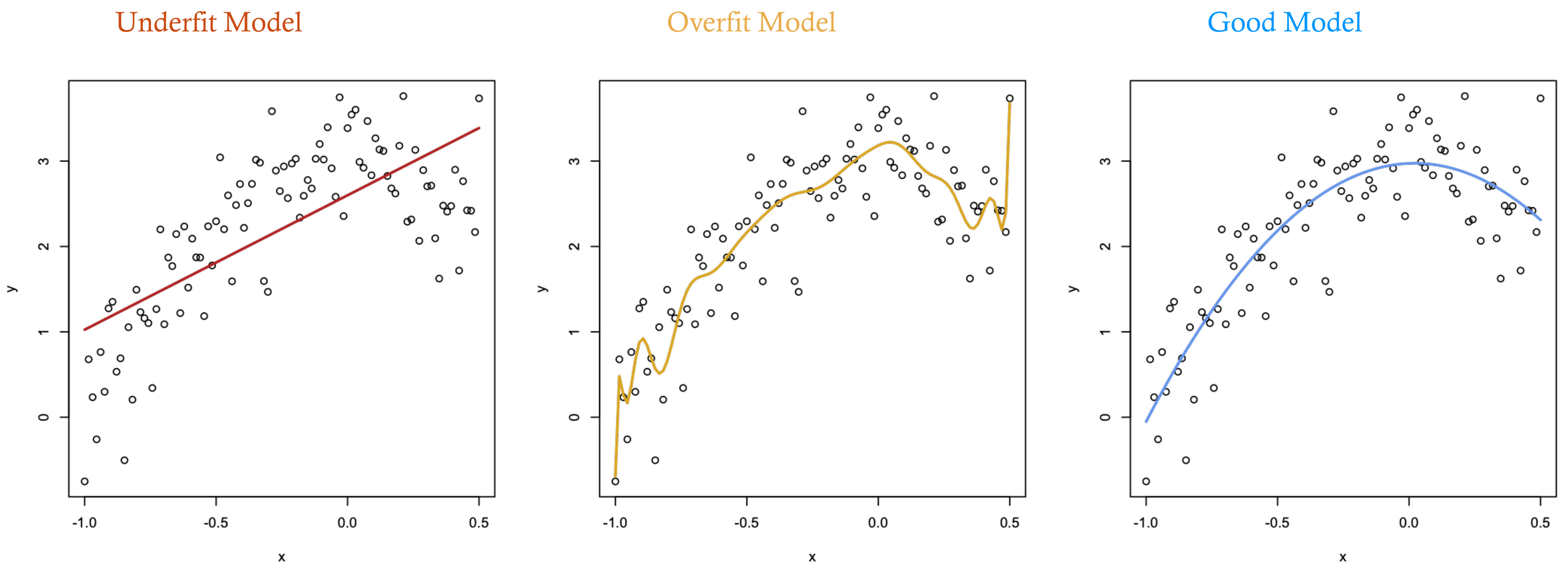
MODEL FITTING

OVER- AND UNDER-FITTING

OVERFITTING VS UNDERFITTING

- Underfitting: Model is too simple to learn the underlying structure of the data
- Overfitting: Model is too complex and learns the noise in the training data

OVERFITTING VS UNDERFITTING



UNDERFITTING

- Underfitting: Model is too simple to learn the underlying structure of the data
- Identifying: The model performs poorly on both training and test data
- Possible fixes:
  - Use a more complex model
  - Find better features
  - Reduce constraints on model (less regularization)
  - Tune hyperparameters

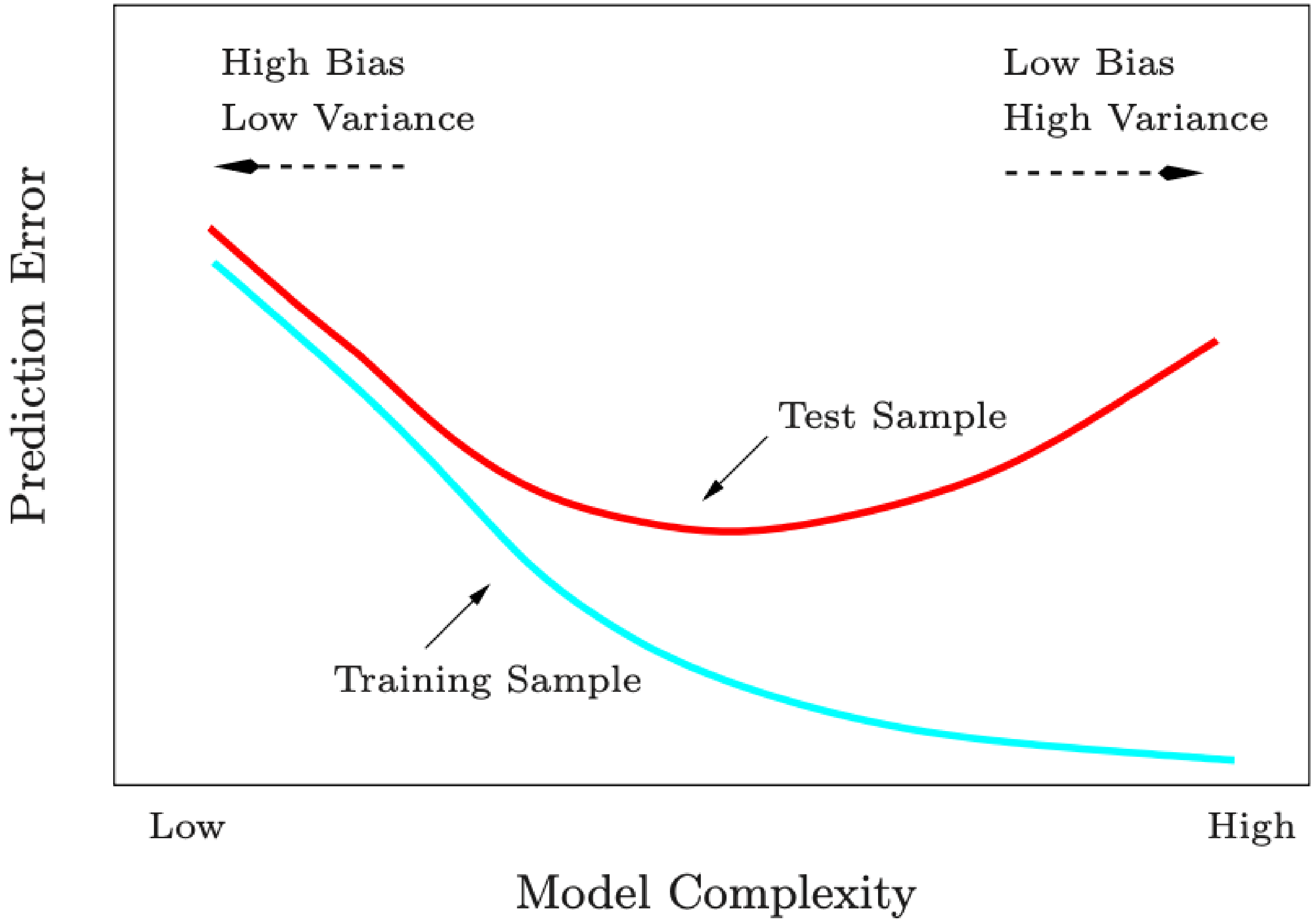
OVERFITTING

- Overfitting: Model is too complex and learns the noise in the training data
- Identifying: The model performs well on training data but poorly on test data
- Possible fixes:
    - Use a simpler model
    - Add constraints on model (more regularization)
    - Gather more data
    - Tune hyperparameters

BIAS-VARIANCE TRADEOFF

- The balance of finding a model that doesn’t underfit or overfit is related to the bias-variance tradeoff
    - **Bias** of refers to the error that is introduced by approximating a real-life problem with a simpler model
    - **Variance** refers to the amount that the prediction changes if a different training set is used (but from the same “population”)
- In general, as a model becomes more flexible and complex, variance increases and bias decreases
    - **Overfitting** is associated with high **variance**
    - **Underfitting** is associated with high **bias**

BIAS-VARIANCE TRADEOFF



From "Elements of Statistical Learning" page 38, by Hastie et al

BIAS-VARIANCE TRADEOFF

- $E\left[(y_0 - \hat{f}(x_0))^2\right] = \underbrace{Var(\hat{f}(x_0))}_{\text{Variance}} + \underbrace{[Bias(\hat{f}(x_0))]^2}_{\text{Bias (squared)}} + \underbrace{E[(y - f(x))^2]}_{\text{Irreducible error}}$

  - Variance of $\hat{f}$ refers to the amount that $\hat{f}$ changes if a different training set is used (but from the same "population")
  - Bias of $\hat{f}$ refers to the error that is introduced by approximating a real-life problem with a simpler model
  - Irreducible error: Random noise inherent in the data. The irreducible error is the best-case scenario given the noise in the data
- Tradeoff because usually reducing bias will increase variance and visa versa

HYPERPARAMETERS

- Also called tuning parameters
- Parameters of the learning algorithm and not the model
  - For example, k, in k-NN is a hyperparameter
- Hyperparameters are chosen by the modeler
  - Values that are actually *learned* from the data are usually called parameters

K-NN RECALL

- Recall what we learned about KNN classification and regression
  - Will a **low** value of *k* be more likely to overfit or underfit?
  - Will a **high** value of *k* be more likely to overfit or underfit?

Testing and Validating

HOW WELL DOES MY MODEL WORK?

- In the prediction setting:
  - **Model evaluation:** how well does our machine learning model generalize to **new data?**
  - Don’t get caught in the weeds of the possible validation methods
  - Find an **honest metric to quantify future model performance**

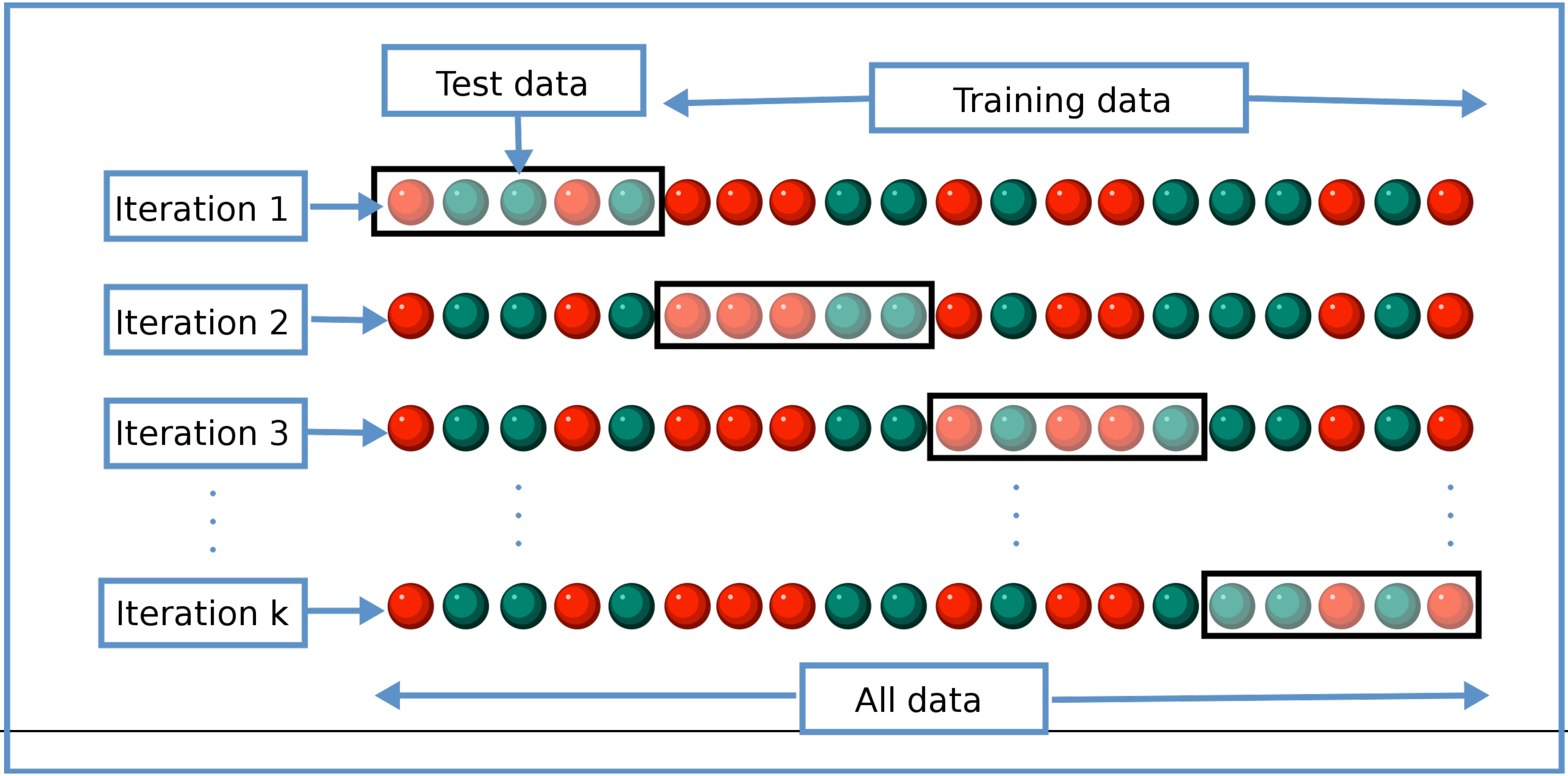
HOW WELL DOES MY MODEL WORK?

- The only way to evaluate how well a model predicts is to **apply it to new data**
- Comparing test metrics to training metrics can also help identify overfitting
- What if new data (with labels) isn't readily available
(and it usually isn't)

MODEL EVALUATION STRATEGIES

- Train/Test split
- Train/Validation/Test split
- Cross-validation

K-FOLD CROSS-VALIDATION



CHOOSING K (THE NUMBER OF FOLDS)

- The data scientist has yet another choice to make: k
  - k=n is leave-one-out cross-validation (LOOCV), this is deterministic
  - k=5 or k=10 are other popular choices
- Bias–variance tradeoff in *k*k-fold CV:
  - Small k → higher bias but lower variance (larger test sets, more stable estimates).
  - Large k → lower bias but higher variance (larger training sets, noisier fold estimates).
  - When CV is used for tuning, bias is less important than just finding the minimum error
- Computational cost:
  - Cost scales with k: small k is faster, while large k (especially LOOCV) is much more expensive.

A TYPICAL APPROACH

- Split data into training and test sets
- Train the model several times using different values of the hyperparameter
  - Choose the hyperparameter value that performs best on the training set *as measured with k-fold cross-validation*
- Use the test set to compute the generalization error
- Remember that the goal is to get a good estimate of the **generalization error**

A TEMPTING APPROACH…

- Split data into a training and a test set
- Train the model for many hyperparameter values
- Choose the hyperparameter value that performs best on test set
- **Beware of data leakage!!**