k-Nearest Neighbors (k-NN) and Bias Variance

Quách Đình Hoàng

k-Nearest Neighbors (k-NN)

- k-NN regression function for a data point x is

$$f(x) = \frac{1}{k} \sum_{x_i \in N_k(x,D)} y_i$$

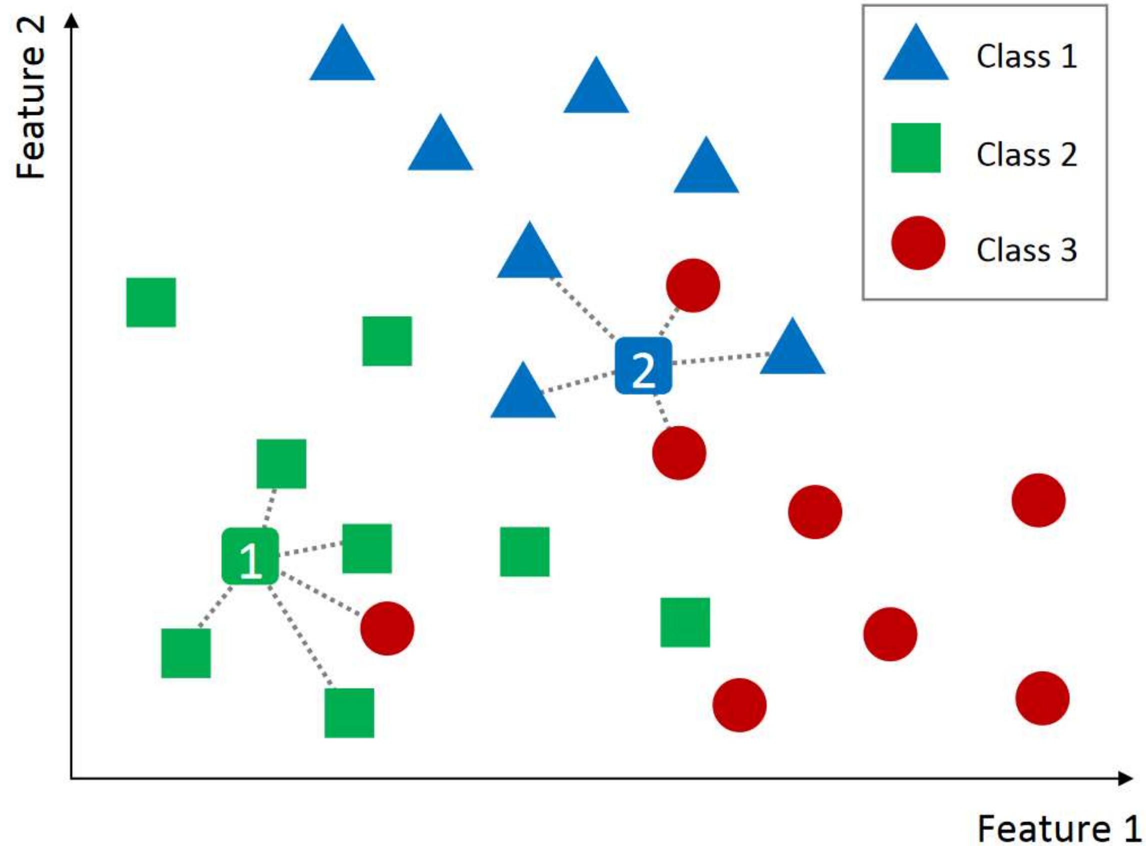
- k-NN classification function for a data point x is

$$f(x) = \underset{c \in \{1,2,\cdots,m\}}{\operatorname{argmax}} \sum_{x_i \in N_k(x,D)} \delta(y_i, c)$$

- In there:
  - $D = \{(x_i, y_i)\}_{i=1:n}$ is training set,
  - $N_k(x, D)$ is k-nearest neighbours of x in training set D,
  - $\delta(y_i, c) = 1$ if $y_i = c$ và $\delta(y_i, c) = 0$ if $y_i \neq c$

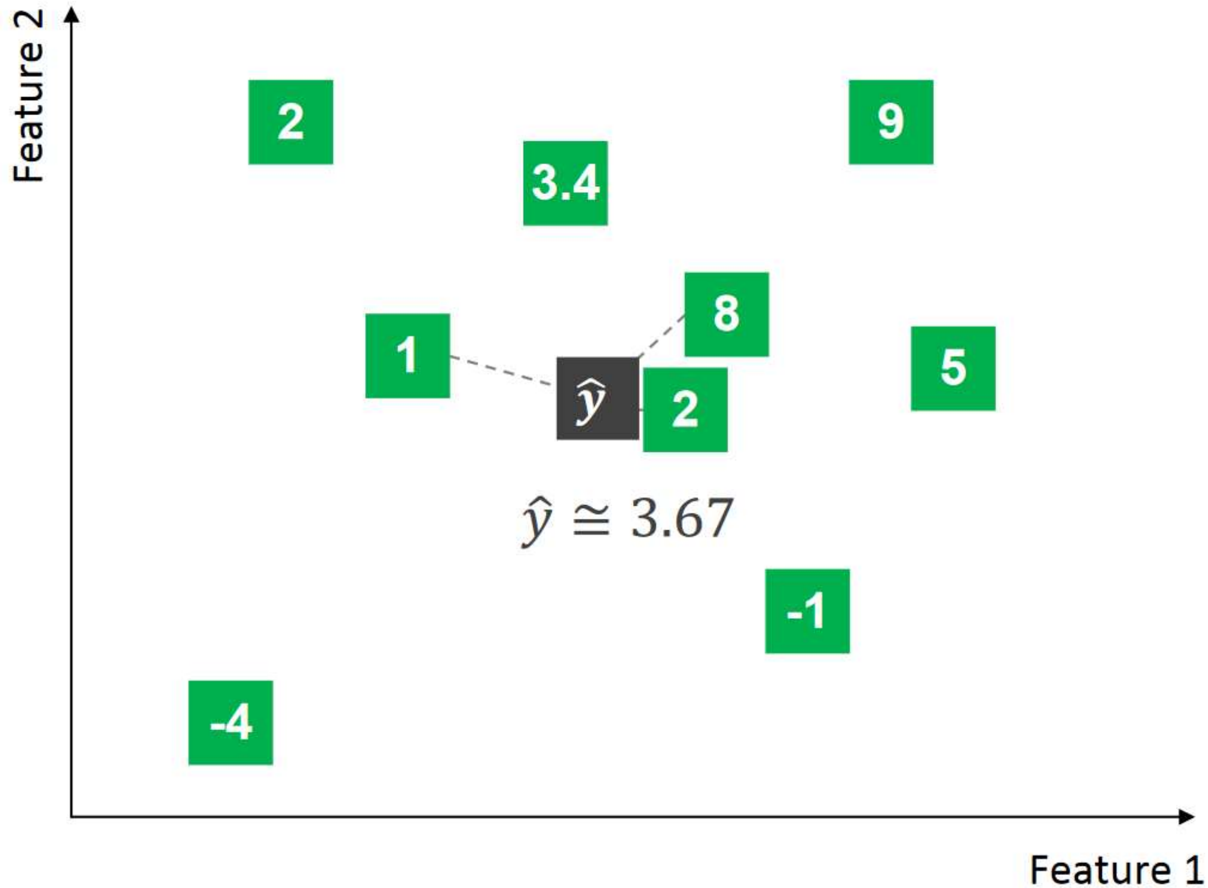
k-Nearest Neighbors for Classification

$$f(x) = \underset{c \in \{1,2,\cdots,m\}}{\operatorname{argmax}} \sum_{x_i \in N_k(x,D)} \delta(y_i, c)$$



k-Nearest Neighbors for Regression

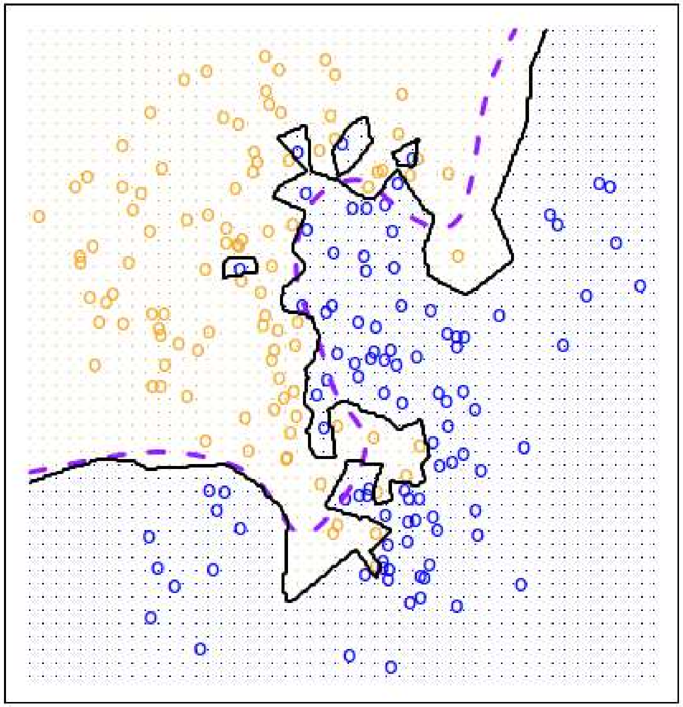
$$f(x) = \frac{1}{k} \sum_{x_i \in N_k(x,D)} y_i$$



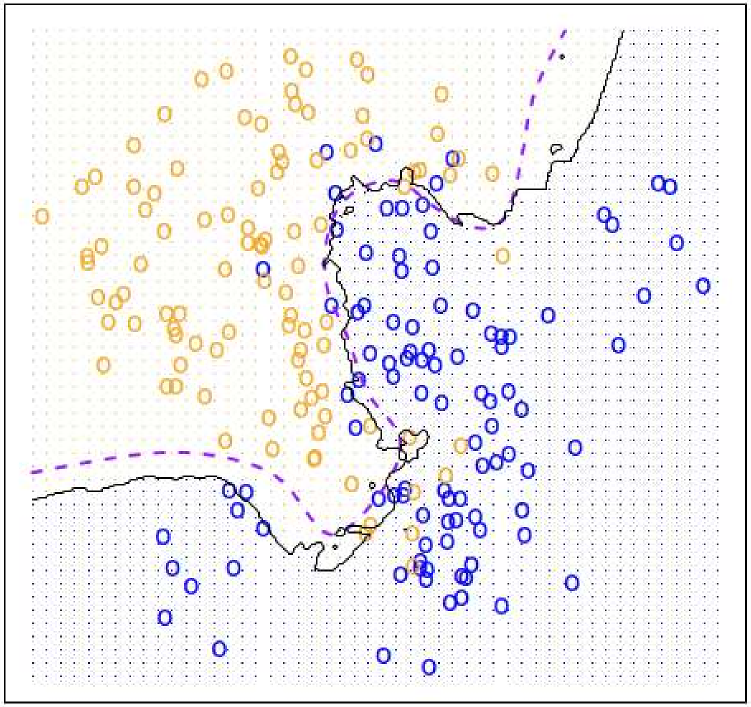
Quiz 1: k-NN for Classification

Given k in {1, 10}

k = ?



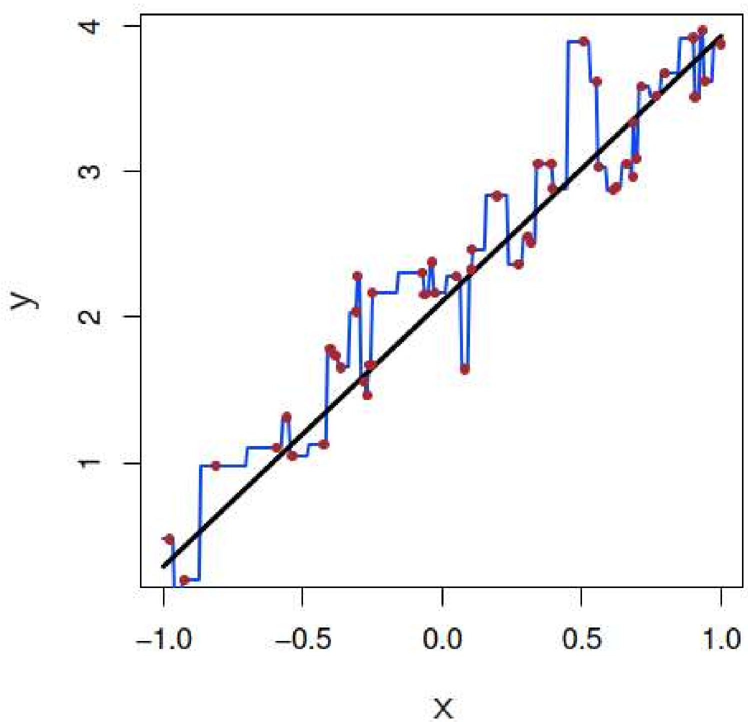
k = ?



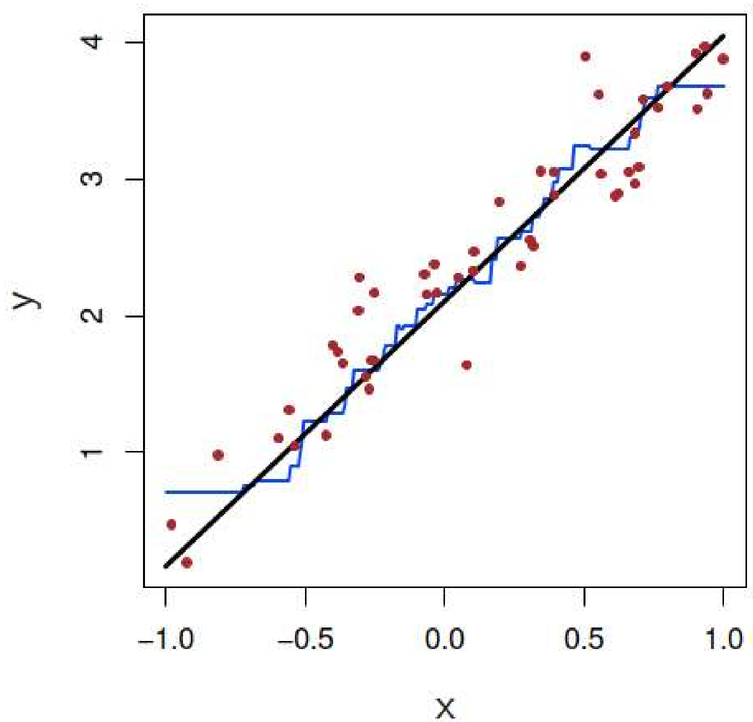
Quiz 2: k-NN for Regression

Given k in {1, 9}

k = ?



k = ?



Some distance measures for continuous features

- Miskowski (L_r norm)

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

  - Euclidean: r = 2 (L_2 norm)
  - Manhatan: r = 1 (L_1 norm)
  - Supremum: r = ∞ (L_{max} norm, L_∞ norm)
- Mahalanobis

$$\mathrm{d}(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y}))^{-0.5}$$

Σ: the covariance matrix

- Cosine

$$\mathbf{d}(\mathbf{x}, \mathbf{y}) = \langle\mathbf{x}, \mathbf{y}\rangle / ||\mathbf{x}|| \, ||\mathbf{y}||$$

$\langle\mathbf{x}, \mathbf{y}\rangle$: inner product (dot product) of vectors, **x** and **y**

Distance measures for discrete features

- Hamming
- Jarcard
- Dice
- …

k-NN hyperparameters

- Value of k
- Feature nomalization
- Distance measure
- Weighted feature

k-NN pros and cons

- Pros
  - Simple to implement and interpret
  - No training (although it is necessary to organize the training data to find k nearest neighbors efficiently)
  - Can handle multi-class classification problem naturally
- Cons
  - Expensive computational cost to find nearest neighbors
  - Requires all training data to be stored in the model
  - Performance can be affected by unbalanced data
  - Performance can be affected by multidimensional data

Improving k-NN Efficiency

- Use special data structures like KD-Tree, Ball-Tree, ...
  - Helps to quickly find the k nearest neighbors
- Reduce dimensionlity with feature selection/extraction
  - Helps reduce the impact of the problem curse of dimensionality
- Use prototype selection methods
  - Help reduce the amount of data point → reduce computational volume

k-NN extentions - Distance-weighted k-NN

- k-NN regression function for a data point x is

$$f(x) = \frac{\sum_{x_i \in N_k(x,D)} w_i y_i}{\sum_{i=1}^{k} w_i}$$

- k-NN classification function for a data point x is

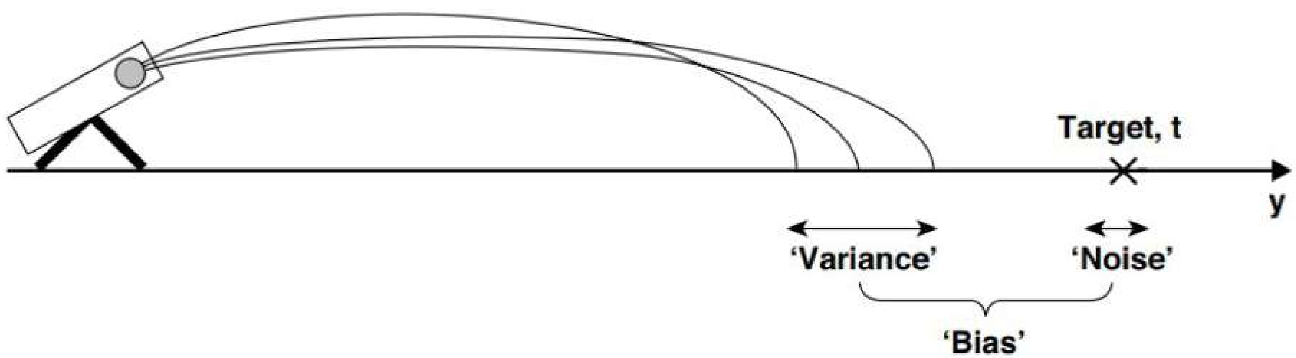
$$f(x) = \underset{c \in \{1,2,\cdots,m\}}{\operatorname{argmax}} \sum_{x_i \in N_k(x,D)} w_i \delta(y_i, c)$$

- Trong đó:
  - $w_i = d(x, x_i)^{-2}$ is the inverse of the squared distance between x and x_i.
  - $D = \{(x_i, y_i)\}_{i=1:n}$ is training set,
  - $N_k(x, D)$ is k-nearest neighbours of x in training set D,
  - $\delta(y_i, c) = 1$ if $y_i = c$ and $\delta(y_i, c) = 0$ if $y_i \neq c$

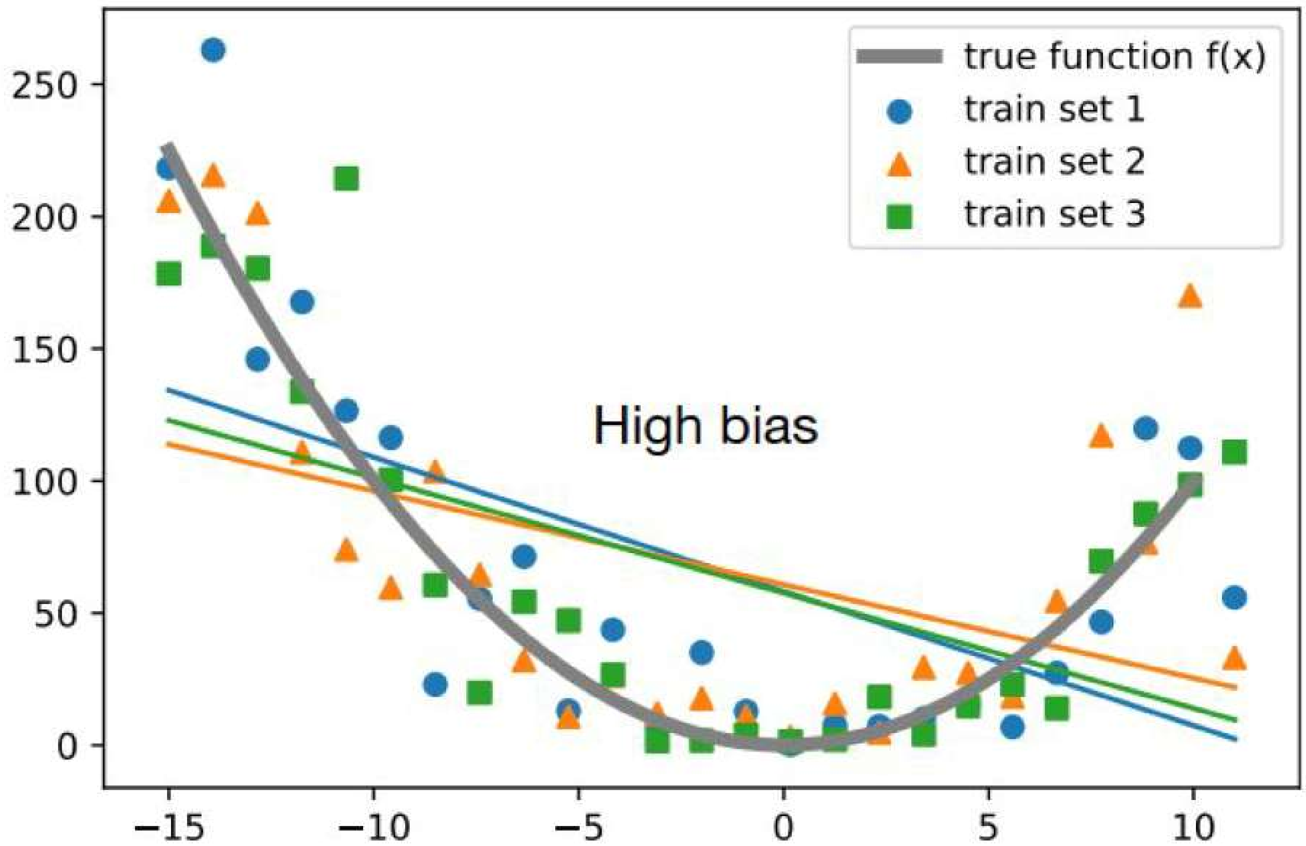
Bias, Variance, and Noise

- Bias: mean deviation (on different data sets) from the target.
- Variance: the degree of variation between the predictions of the function f if a different training data set is used.
- Noise: component that describes the change in the target's position
  - Because of measurement errors or input variables in are not enough to predict

$$Error = Bias^2 + Variance + Noise$$

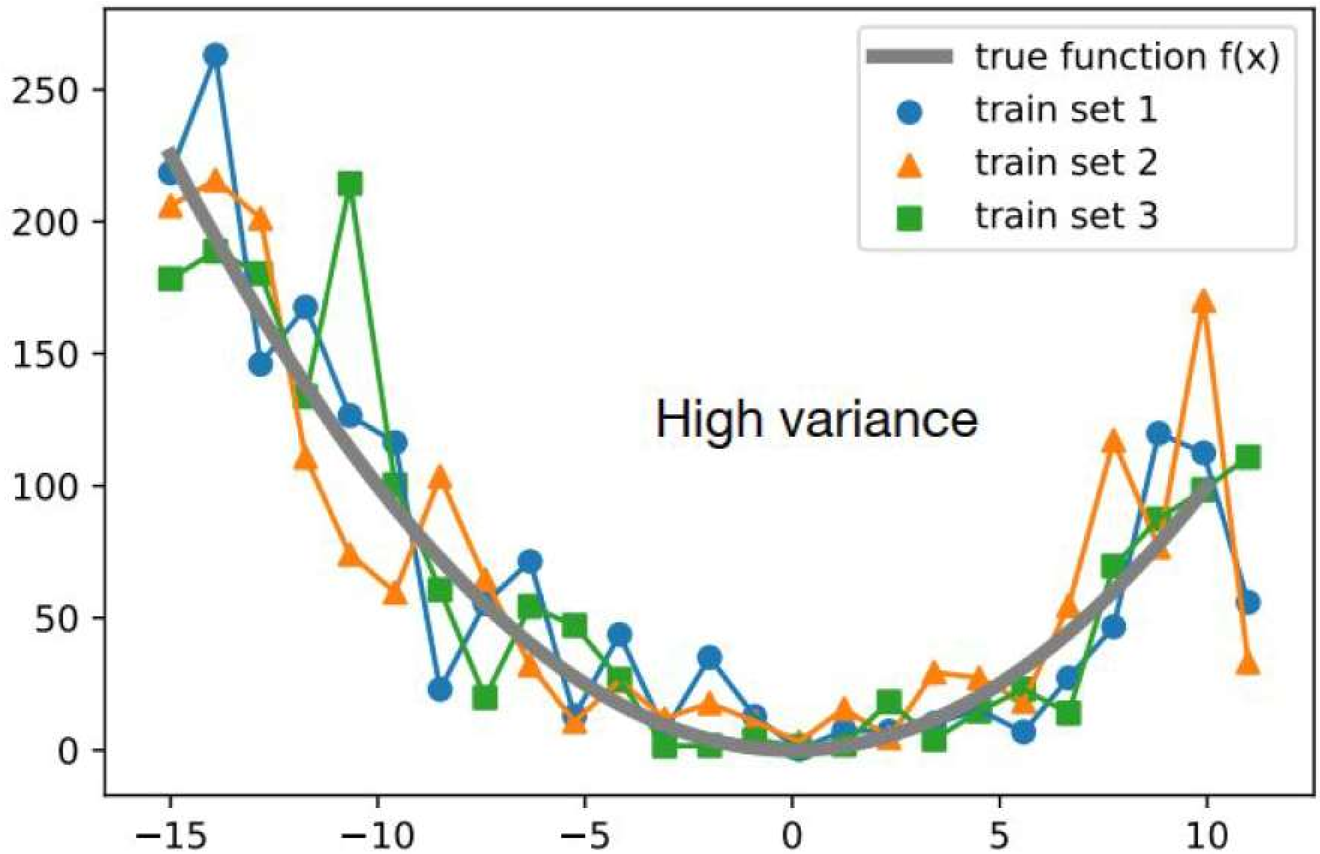


High bias

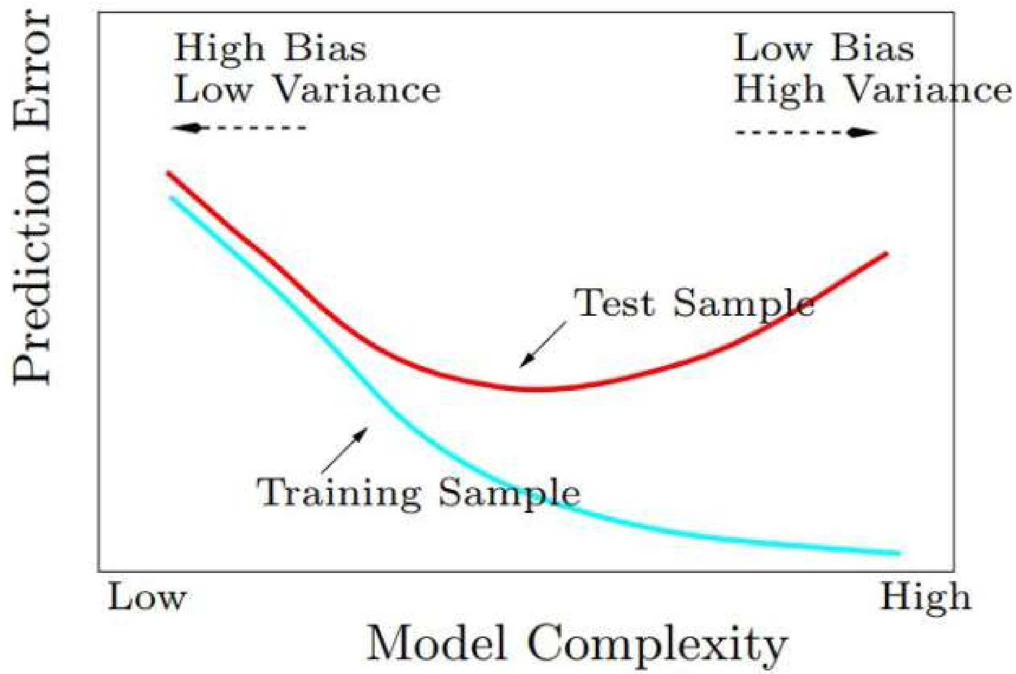


true function f(x)
train set 1
train set 2
train set 3
High bias
250
200
150
100
50
0
−15
−10
−5
0
5
10

High variance

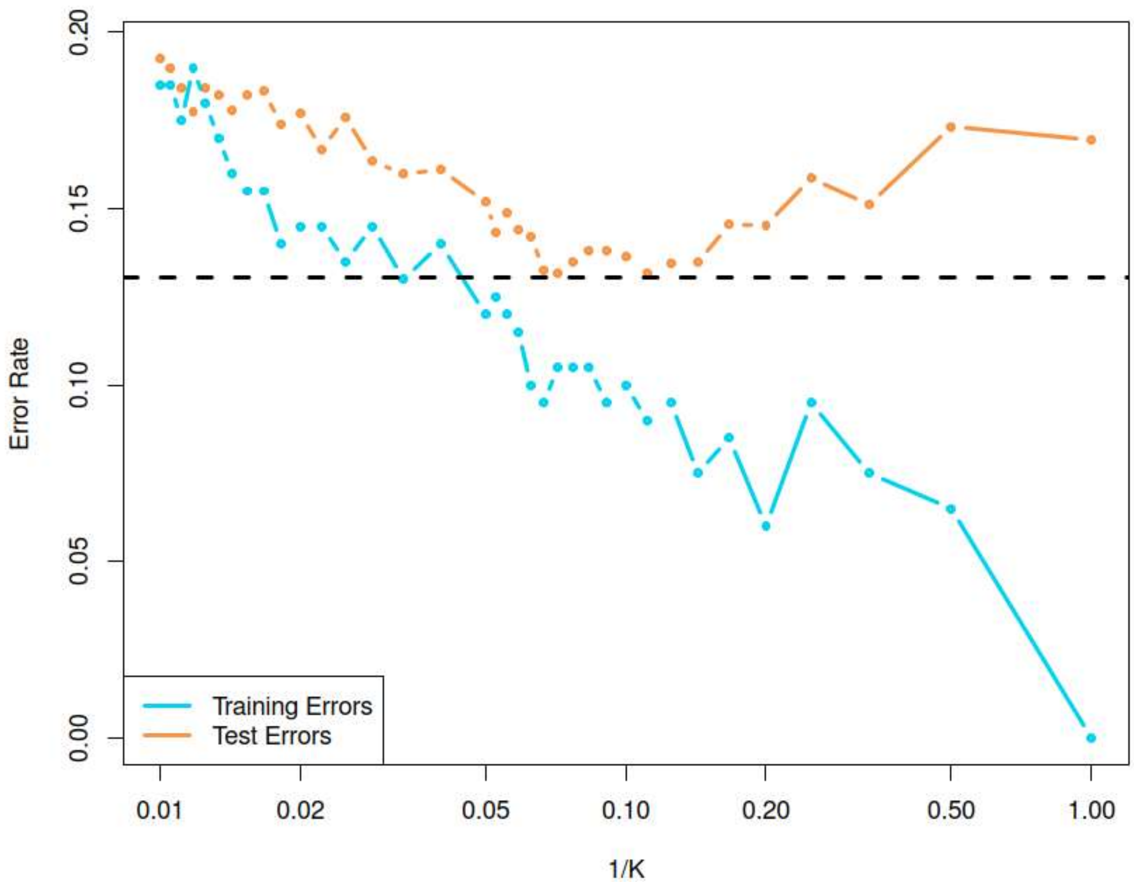


Bias-variance trade-off

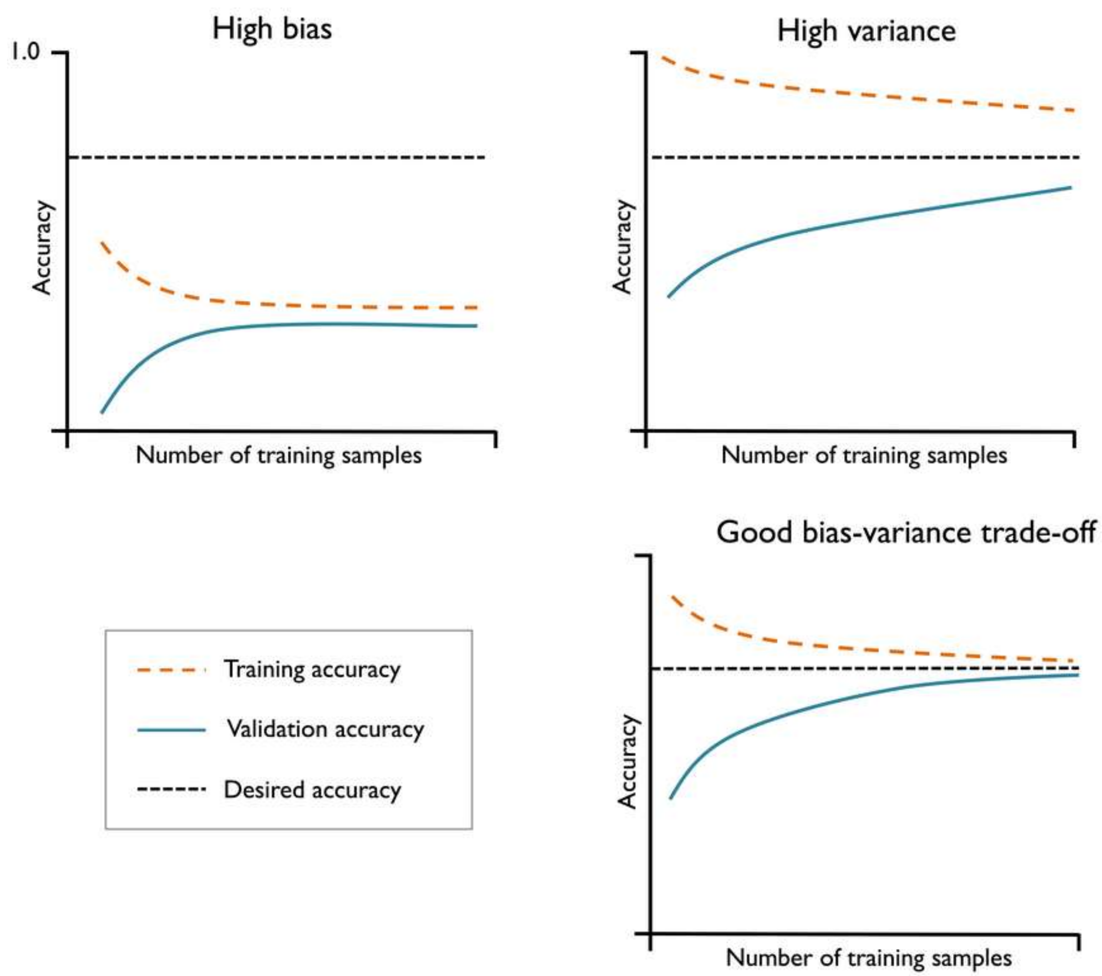


- In general, complex models often have small bias and large variance.
- Similarly, simple models often have large bias and small variance.
- This relationship is called bias-variance trade-off.

K-NN and Bias-Variance trade-off



Diagnostic bias/variance with learning curves



Fixing bias and variance

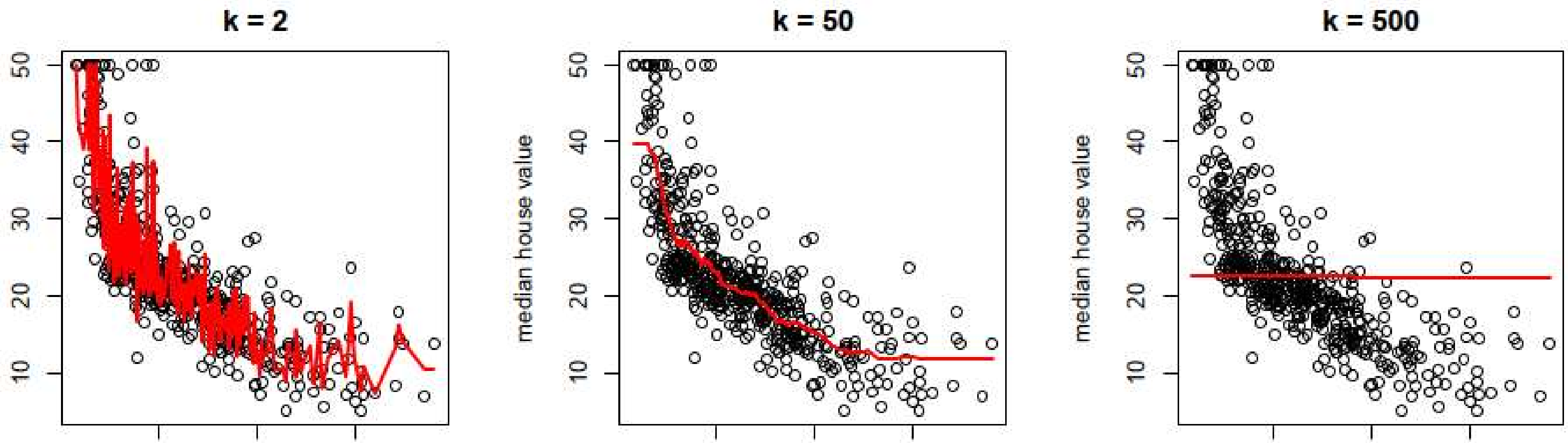
- Increase the size (complexity) of the model → Fixing bias
- Collect more input features → Fixing bias
- Reduce or remove regularization → Fixing bias
- Collect more training data → Fixing variance
- Use regularization → Fixing variance
- Reduce the number of features (select/extract) → Fixing variance
- Using ensemble models → Fixing variance

Summary

- Use bias-variance tradeoff to choose $f(X)$ (corresponds to choose k):

  Not too complicated but not too simple either

- Such $f(.)$ functions is easier to understand, interpret the results, and often make better predictions on new data
  - If $f(.)$ too simple, it will underfitting
  - If $f(.)$ too complex, it will overfitting



References

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, ***An Introduction to Statistical Learning with Applications in R, Second Edition***, Springer, 2021.