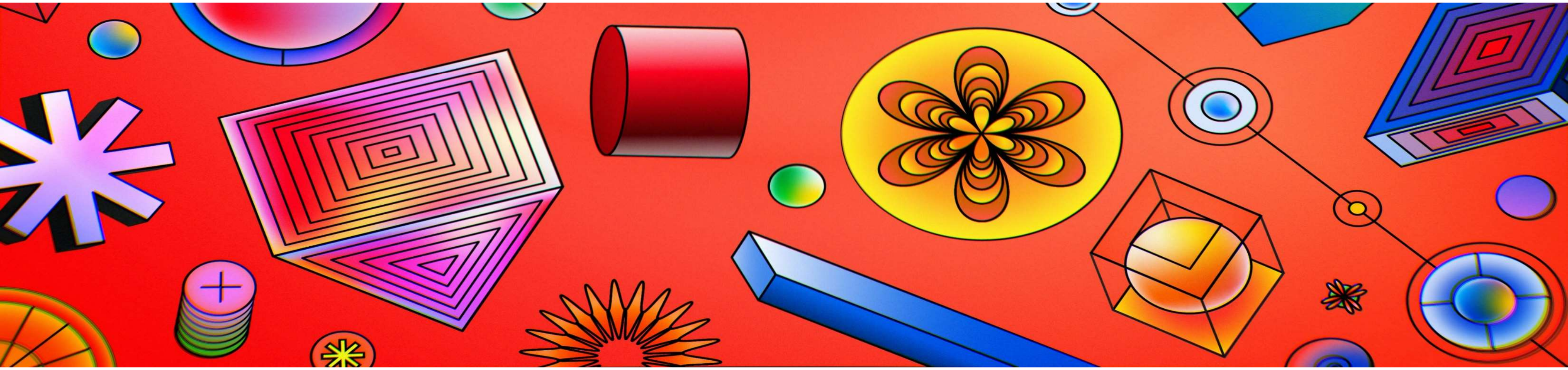
Fall 2023

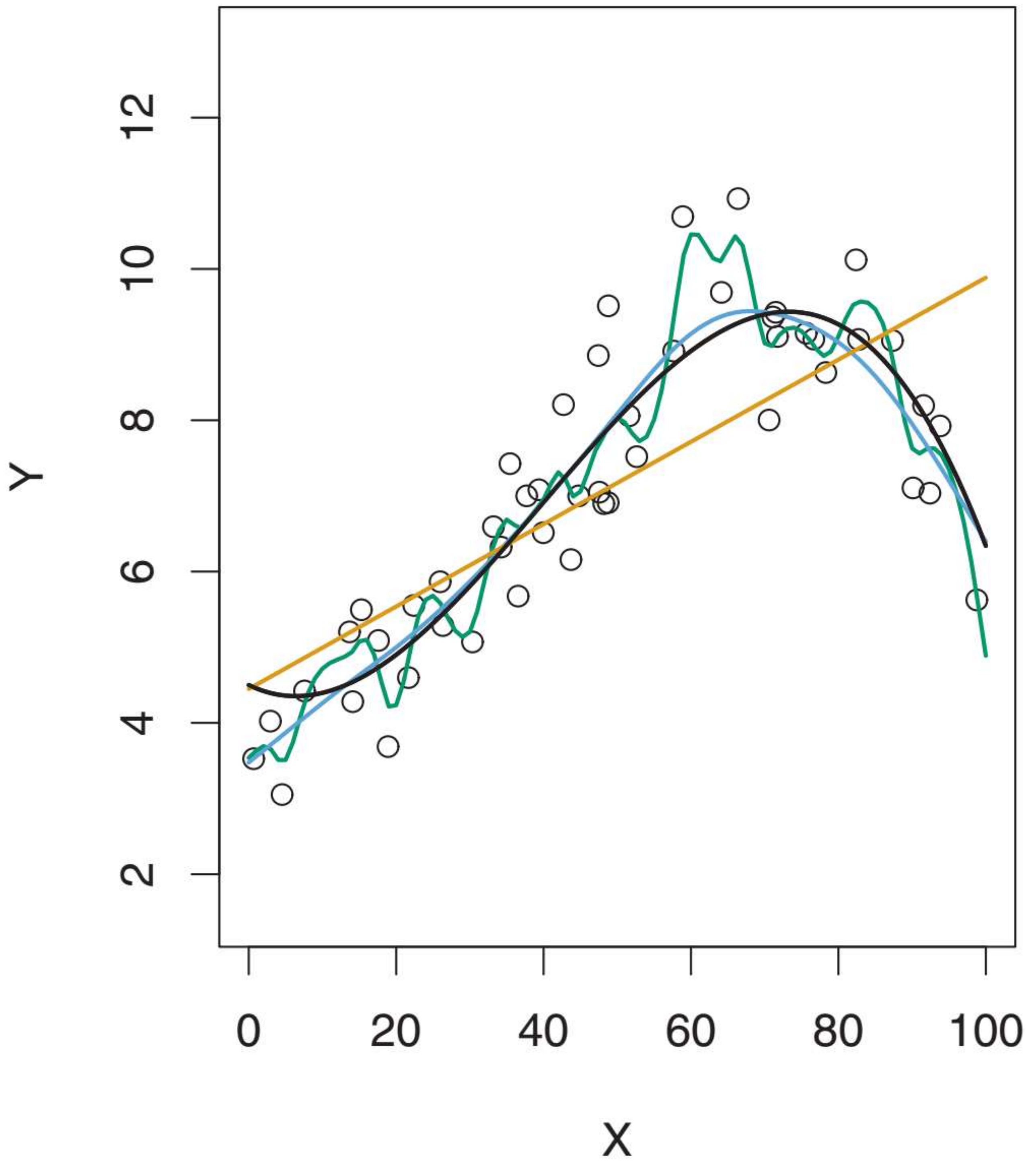
BIF524/CSC463 Data Mining

Statistical Learning

Eileen Marie Hanna, *PhD*

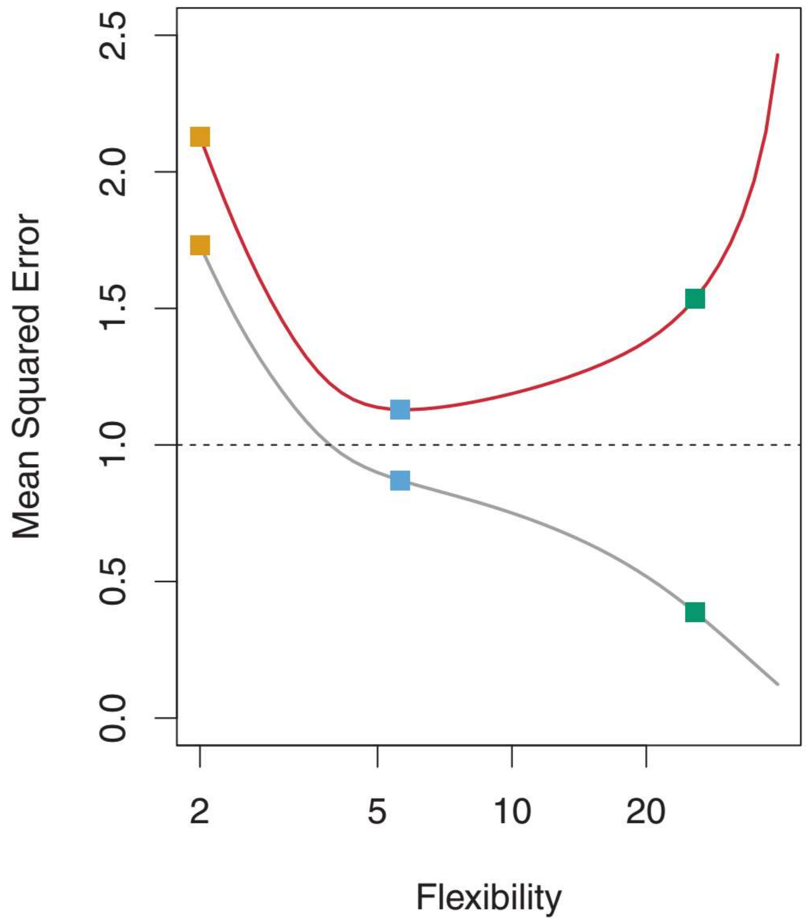
12/09/2023





Y
X
2
4
6
8
10
12
0
20
40
60
80
100

Mathematically



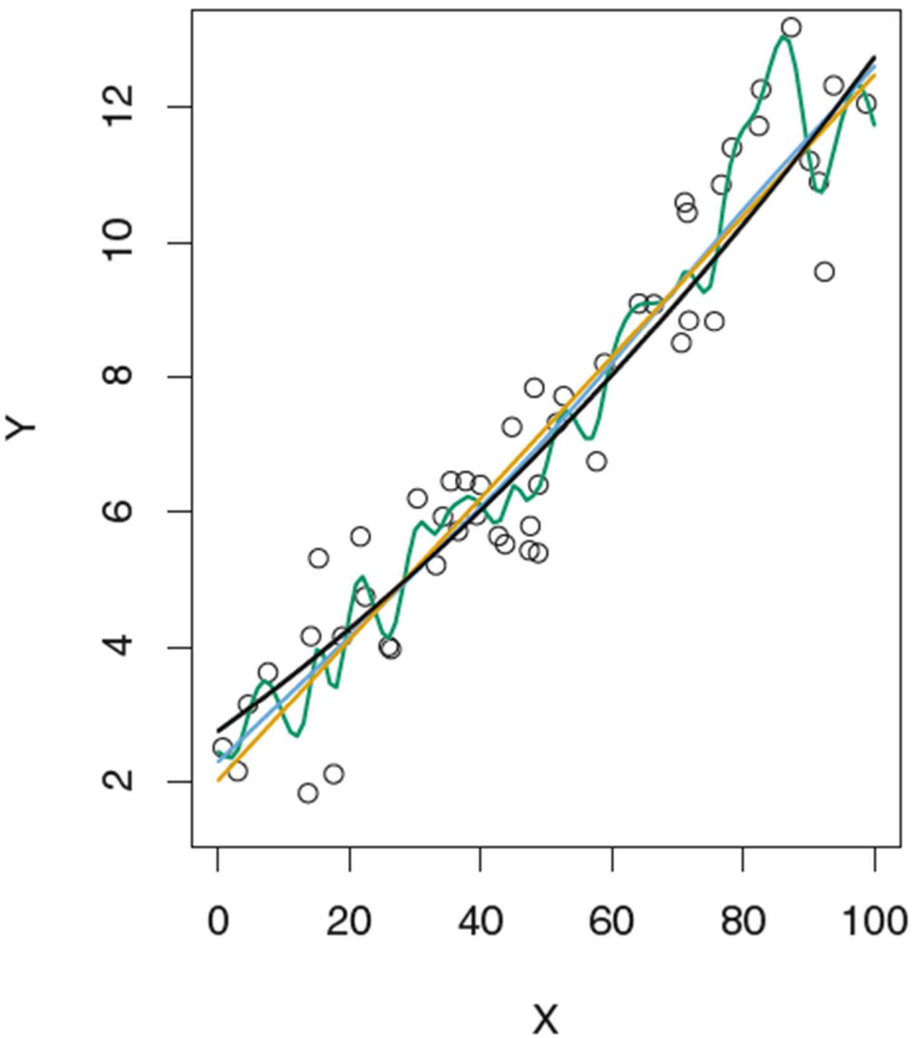
- In general, this MSE behavior (decrease and U-shape) holds regardless of the dataset and of the statistical method.

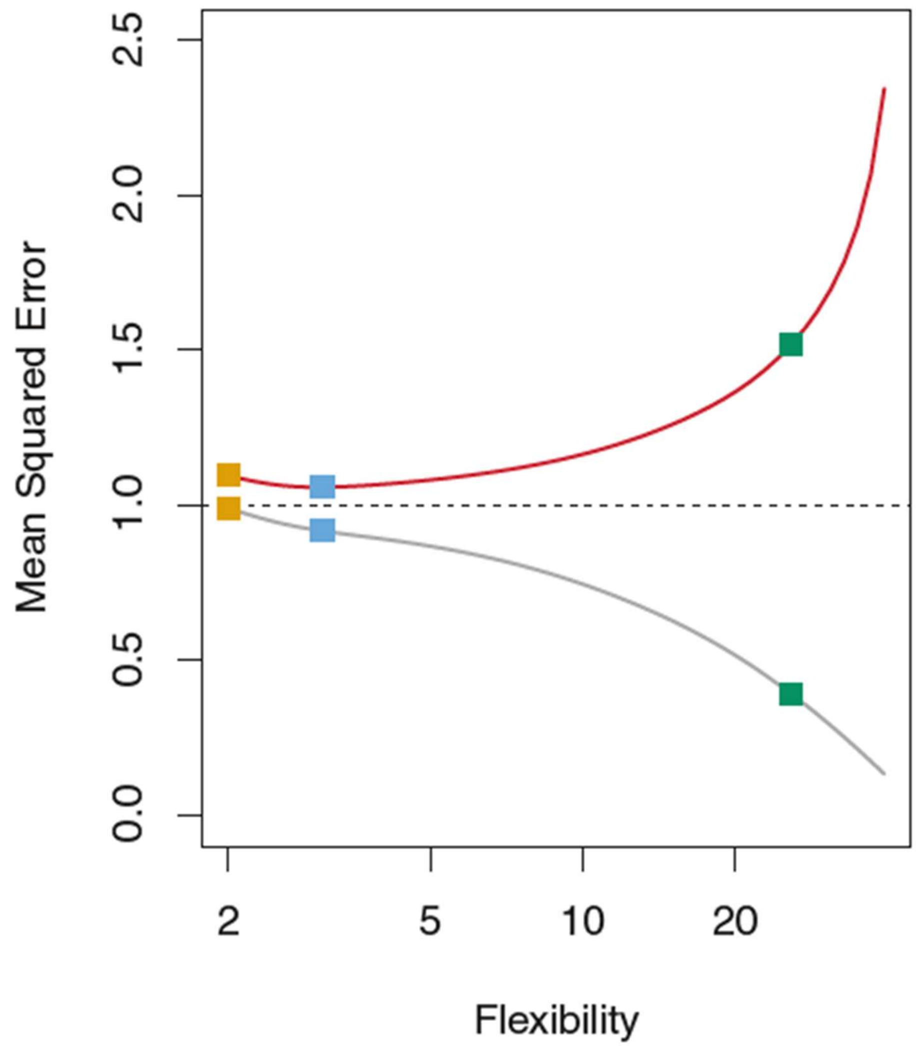
- **As the flexibility increases, the training MSE will decrease, but the test MSE may not.**

- **When we have a small training MSE and a large test MSE -> overfitting!**

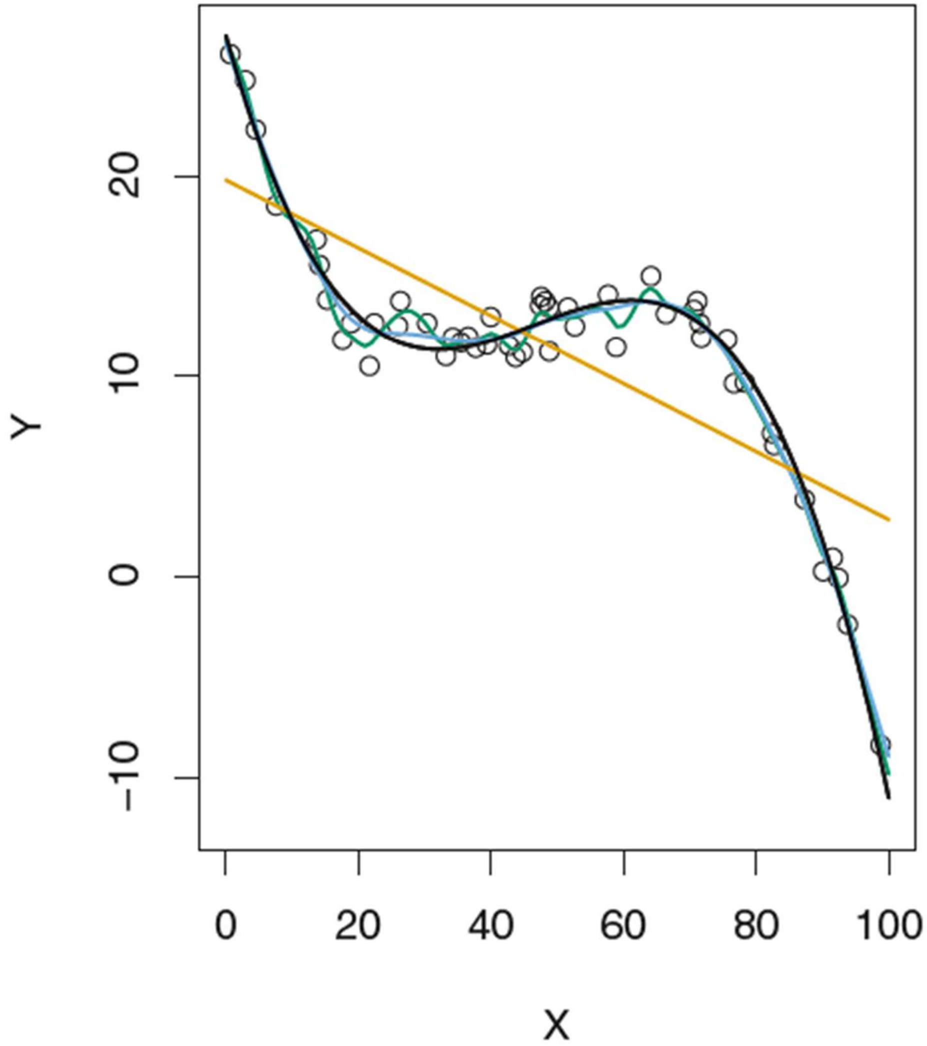
  - The method **may be picking patterns from the training data** that are caused **by random chance** and **those patterns don't actually exist in the test data** -> increased test MSE.

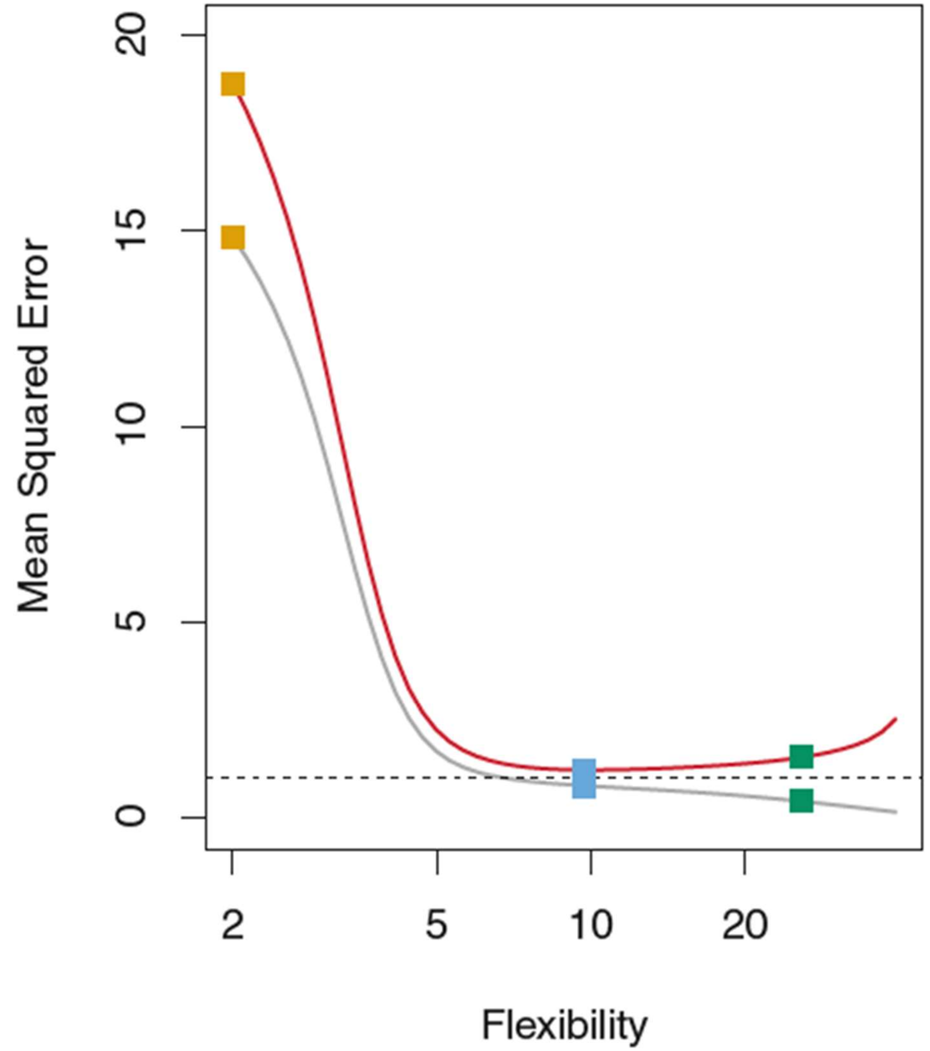
Other examples





Other examples





The bias-variance trade-off

- Two **competing properties** of statistical learning approaches

- The test MSE for a given value x_0 can be decomposed into the sum of three fundamental measures:

$$\underbrace{E\left(y_0 - \hat{f}(x_0)\right)^2}_{\text{expected test } MSE} = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$

The bias-variance trade-off

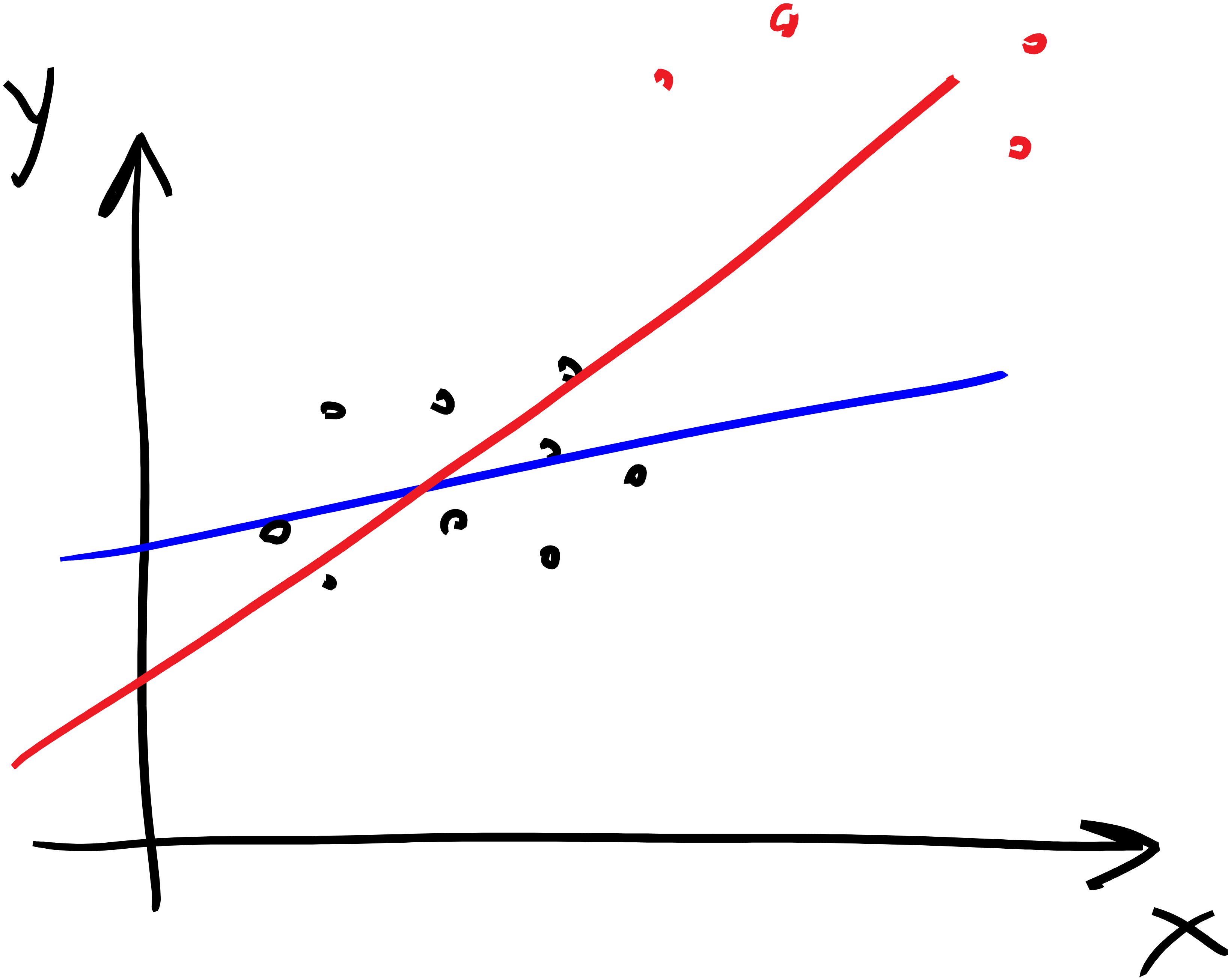
$$\underbrace{E\left(y_0 - \hat{f}(x_0)\right)^2}_{\text{expected test } MSE} = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$

- The overall expected test $\boldsymbol{MSE}$ can be computed by averaging this measure for all possible values of x_0 in the test data.
- **What does this equation infer?**
  - In order to **minimize the expected test error**, we need to choose a method that would satisfy **low variance and low bias**.
  - Note that the **expected test $\boldsymbol{MSE}$ cannot be less than the irreducible error** given by $Var(\in)$.

The bias-variance trade-off

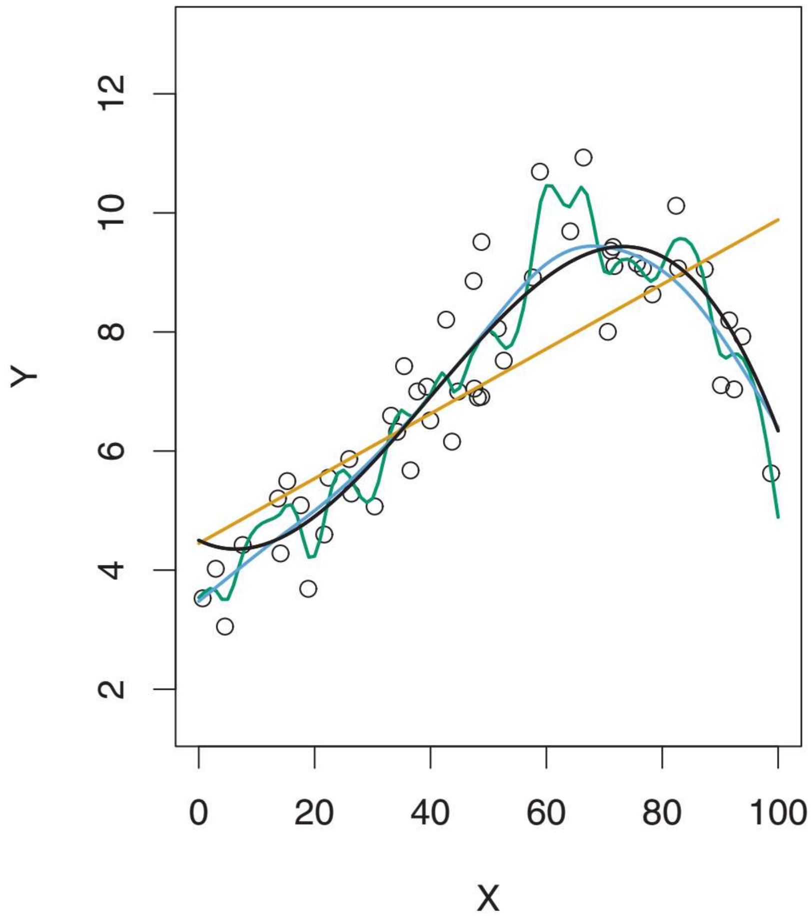
$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \underbrace{\mathrm{Var}(\hat{f}(x_0))}_{\substack{\text{amount of variation} \\ \text{of } \hat{f} \text{ using different} \\ \text{training datasets}}} + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$

- Ideally the estimate of f **should not vary much** between different training datasets.
- A **high variance** means that **small data changes -> large changes in $\hat{f}$**.
- Generally, **more flexible** methods -> **higher variance**. **Why?**


y
x

The bias-variance trade-off

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \underbrace{\mathrm{Var}(\hat{f}(x_0))}_{\text{amount of variation of } \hat{f} \text{ using different training datasets}} + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$



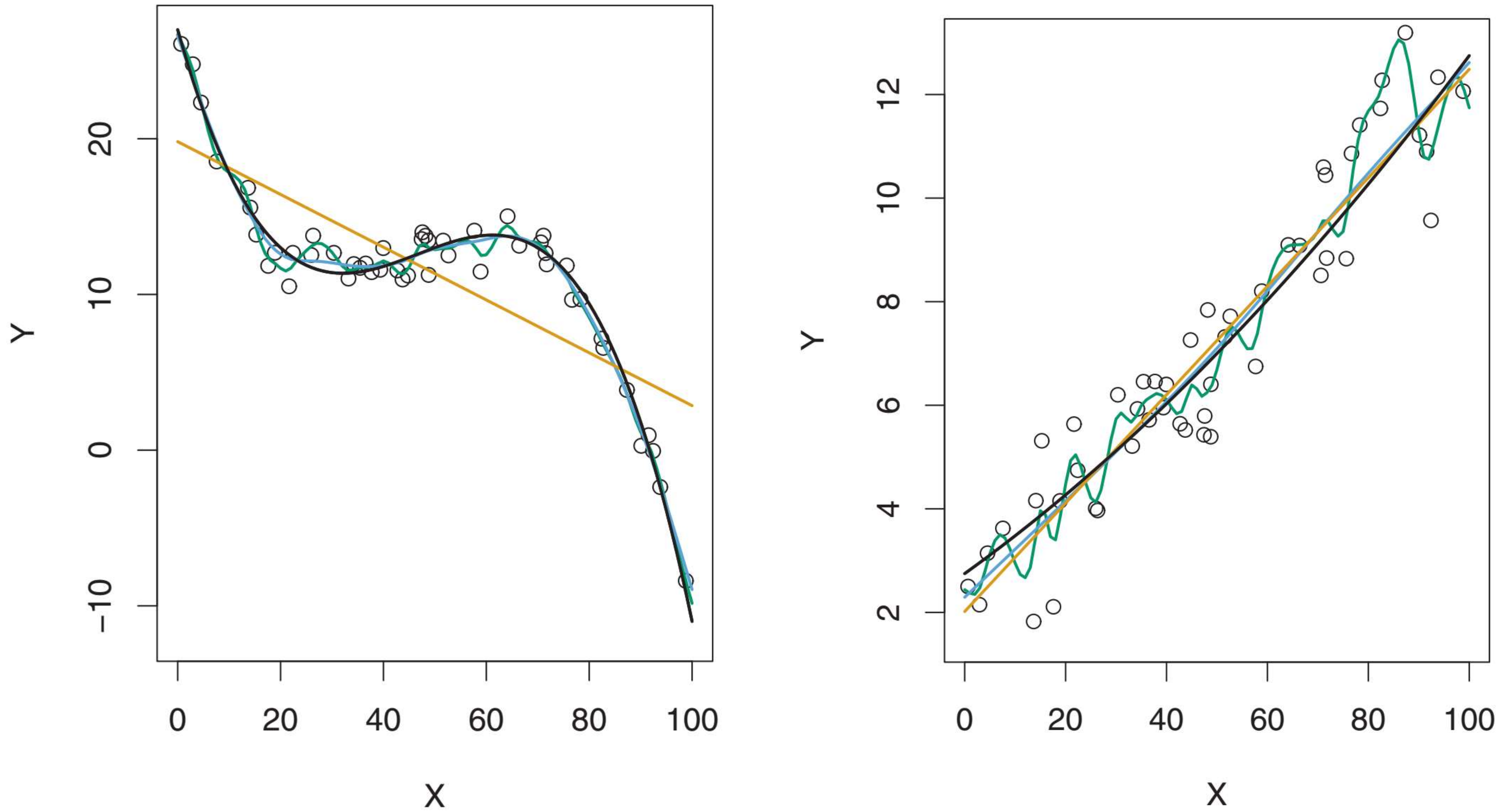
- The **green** curve has a very close trend as the data observations – it is very flexible.
  - Changing data -> higher variance.
- The **orange** curve is less flexible,
  - Moving a single observation would poorly affect the position of the line.

The bias-variance trade-off

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + \underbrace{[\mathrm{Bias}(\hat{f}(x_0))]^2} + \mathrm{Var}(\epsilon)$$

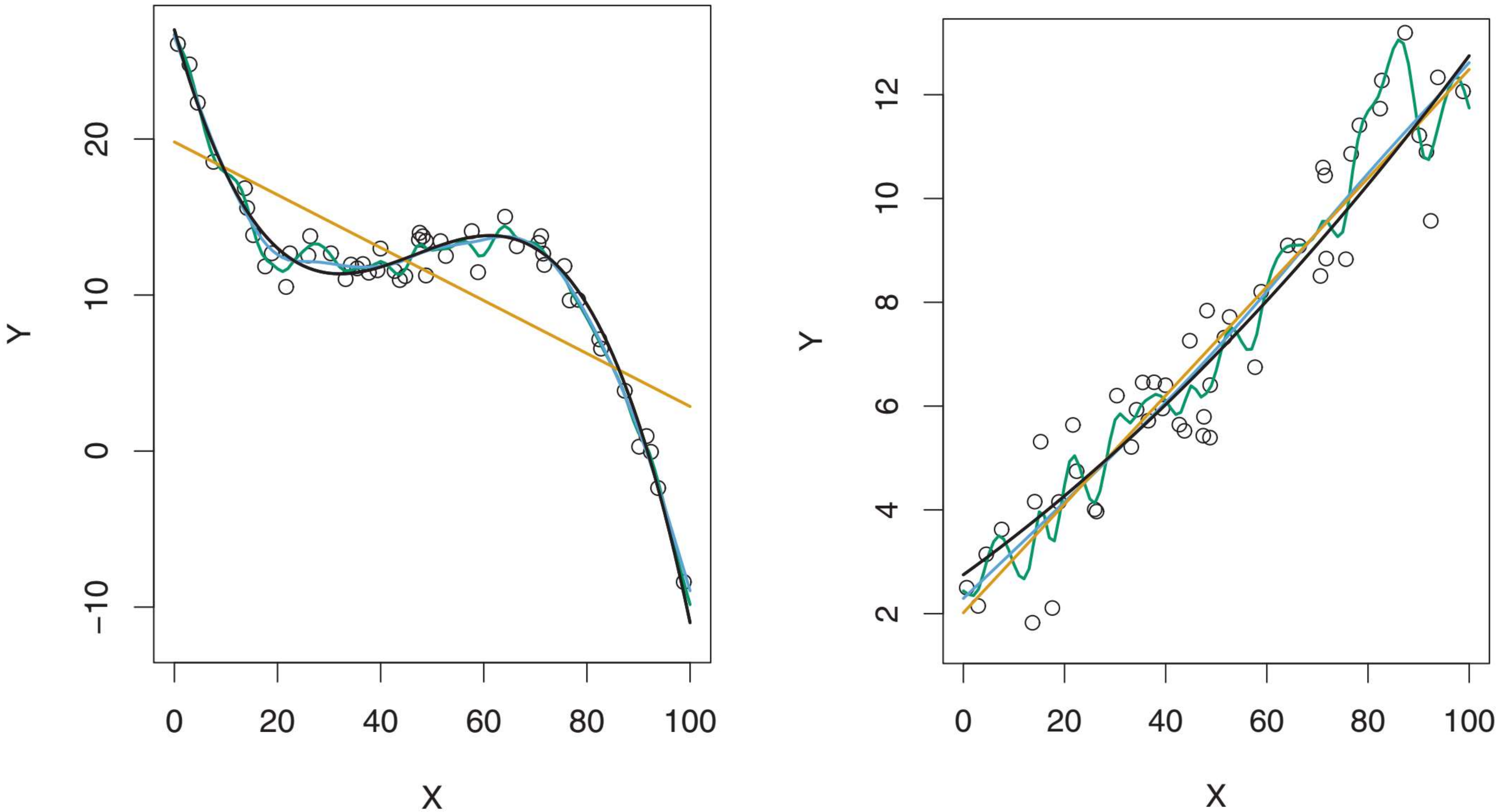
error introduced by approximating a very complicated problem by a much simpler problem

The bias-variance trade-off



- **More flexible methods -> __________ (less or more) bias?**

The bias-variance trade-off



- For instance, it is less likely that any real-life problem has a truly simple linear relationship between Y and $X_1, X_2, \ldots, X_p$.
- Performing linear regression on such problems would certainly introduce some bias in $\hat{f}$.
- In general, for **more flexible methods -> less bias**.

The bias-variance trade-off – summary

- **As a general rule, as flexibility increases, variance increases and bias decreases.**
- The corresponding change of rates defines how MSE changes.
- The **bias usually decreases faster** than the variance increase.
  - Expected test MSE decreases.
  - But at some point, the flexibility does not affect the bias anymore, but it starts to significantly increase the variance.

Classification

- In this setting, $\boldsymbol{y_i}$ **is not numerical**.
- We want to estimate f based on training observations

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}$$

where $y_1, \ldots, y_n$ are **qualitative**.

Classification

- In this setting, $\boldsymbol{y_i}$ **is not numerical**.
- We want to estimate f based on training observations

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}$$

where $y_1, \ldots, y_n$ are **qualitative**.

- The most common way to measure the **accuracy of $\hat{\boldsymbol{f}}$** is the **error rate**.

  - For training data, it is the **proportion of mistakes that are made** if we apply $\hat{f}$ to training data.

training error rate: $$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

equals 1 if $y_i = \hat{y}_i$, and 0 otherwise.

the predicted class label of the i^{th} observation using $\hat{f}$

Classification

training error rate: $$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

equals 1 if $y_i = \hat{y}_i$, and 0 otherwise.

the predicted class label of the i^{th} observation using $\hat{f}$

- We are interested in the test error rather than the training error.

testing error rate: $$\mathrm{Ave}\left(I(y_0 \neq \hat{y}_0)\right)$$

corresponding to the average error rate on unseen observations (x_0, y_0)

Bayes Theorem

$Pr(Y = \text{Default} \mid x_0) = 0.15$

$Pr(Y = \text{Not Def.} \mid x_0) = 0.85$

$$\underbrace{Pr\left(Y = c_{i} \mid X = x_0\right)}_{\text{Posterior prob.}} = \frac{Pr(x_0 \mid c_i)\, Pr(c_i)}{Pr(x_0)}$$

$c_1 \equiv \text{Default}$

$c_2 \equiv \text{Not Default}$

$x_0 = (x_1, x_2)$

$= (32, 10)$

age

years of education

Bayes classifier

- **A very simple classifier minimizes this test error on average.**

- **That is achieved by assigning each observation to the most likely class, given its predictor values.**

- A test observation with predictor vector x_0 is assigned to class j for which the following probability is the largest:

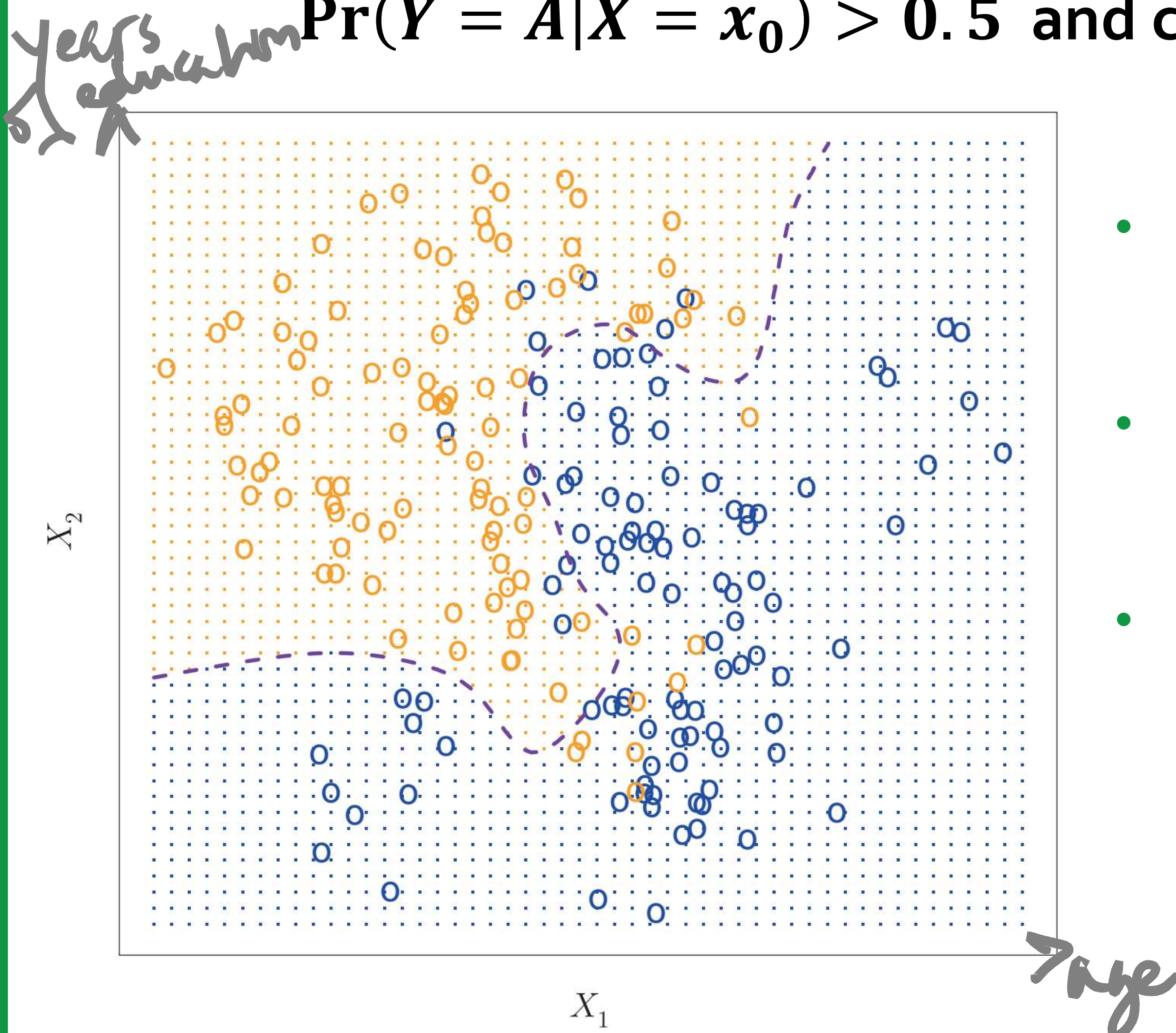
$$\Pr(Y = j | X = x_0)$$

conditional probability that $\mathbf{Y} = j$, given x_0

Bayes classifier – two classes

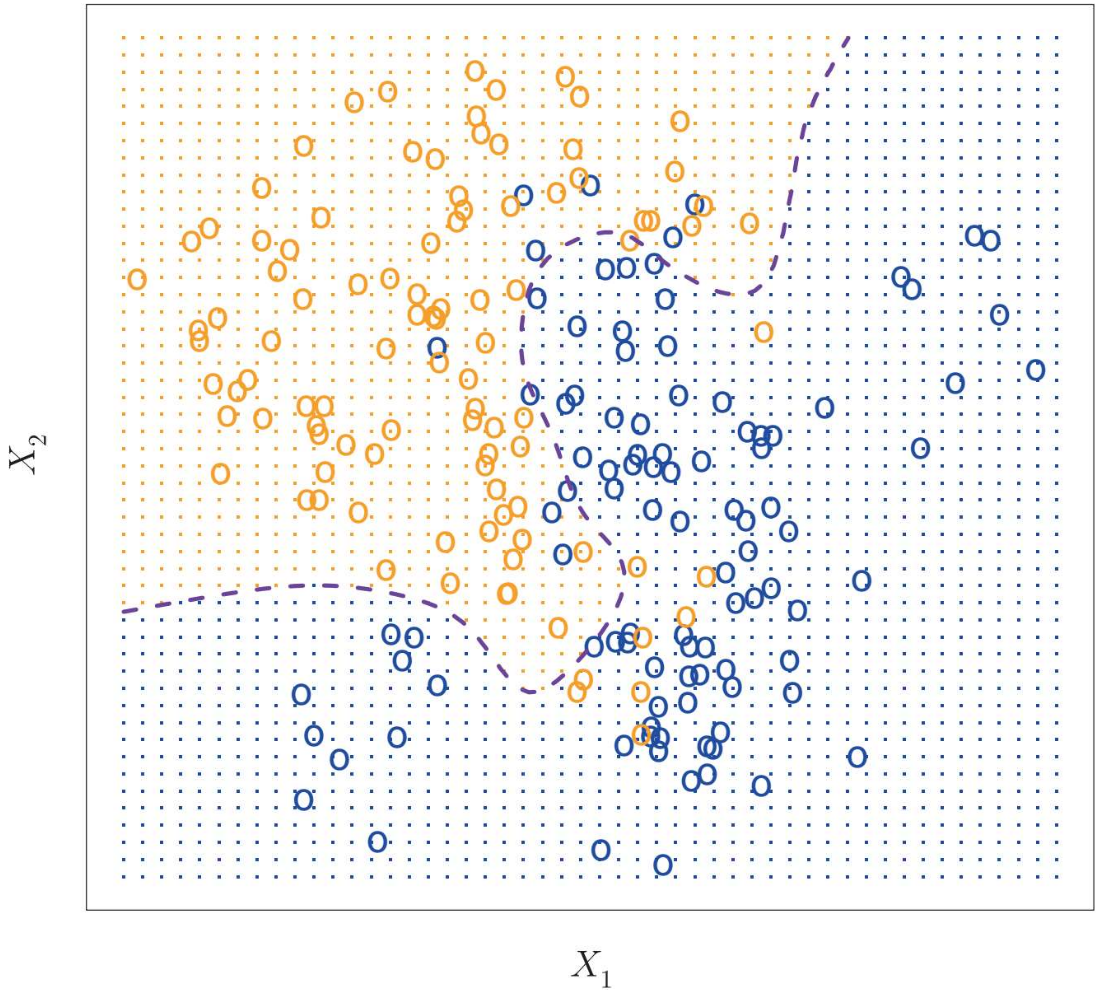
- Suppose we have a two-class (A or B) problem, the Bayes classifier **assigns x_0 to A if:**

$$\mathbf{Pr}(\boldsymbol{Y} = \boldsymbol{A} | \boldsymbol{X} = \boldsymbol{x_0}) > \mathbf{0.5} \text{ and class } \boldsymbol{B} \text{ otherwise}$$



- Simulated dataset in $2D$ space of predictors X_1 and X_2.
- Two classes represented by orange and blue circles.
- We can see that for each value of X_1 and X_2, there is a different probability of the response (orange or blue).

Bayes classifier – two classes



- The **orange region** reflects the set of point for which $\Pr(Y = orange \mid X) > 0.5$.
- The **blue region** indicates that this probability is less than 0.5.
- The dashed line separating both regions is called the **Bayes decision boundary** and it is where this probability is **exactly $\mathbf{0.5}$**.

Bayes classifier – two classes

- The Bayes classifier produces the **lowest possible test error** (Bayes error rate).
  - always chooses class for which the probability is largest.

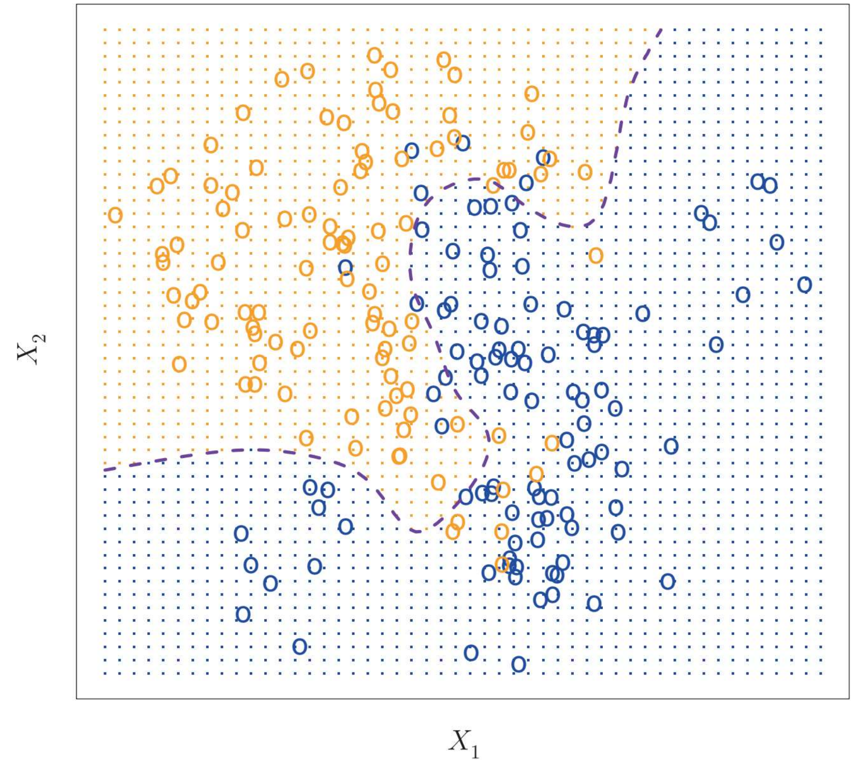
- The error rate at $X = x_0$ is:

$$1 - max_j \Pr(Y = j | X = x_0)$$

- The overall Bayes error rate is:

$$1 - E(max_j \Pr(Y = j | X))$$

Equivalent to the irreducible error.
It is the lowest possible classification error that can be reached.



K-nearest neighbors classifier

- **We do not always know the conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{X}$** -> using Bayes classifier is not possible.

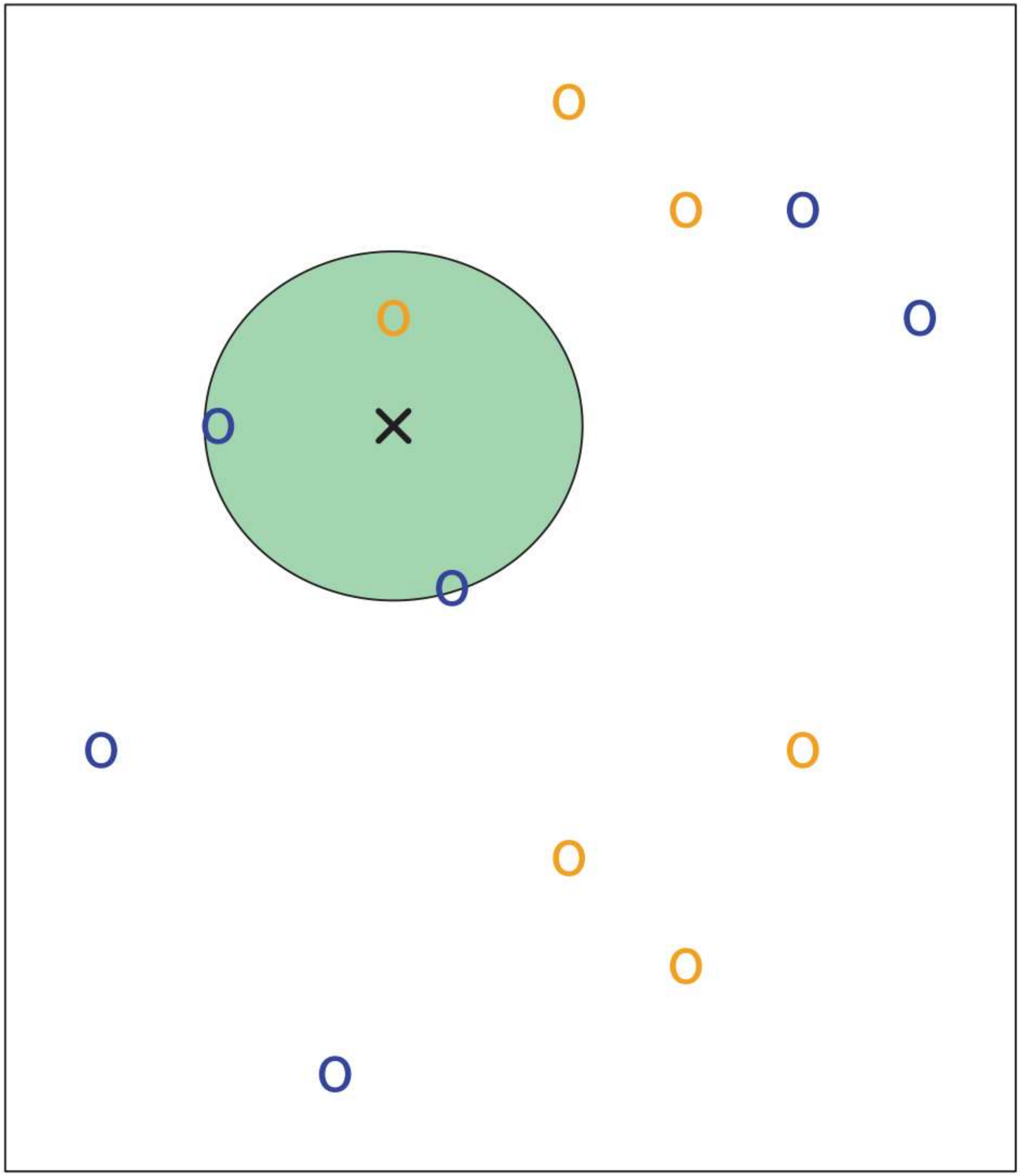
- In a way, the **Bayes classifier serves as an unattainable gold standard** to which we can compare the performance of other approaches.

- The K-nearest neighbors (**KNN**) classifier thus seeks:

  - **to estimate the conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{X}$**

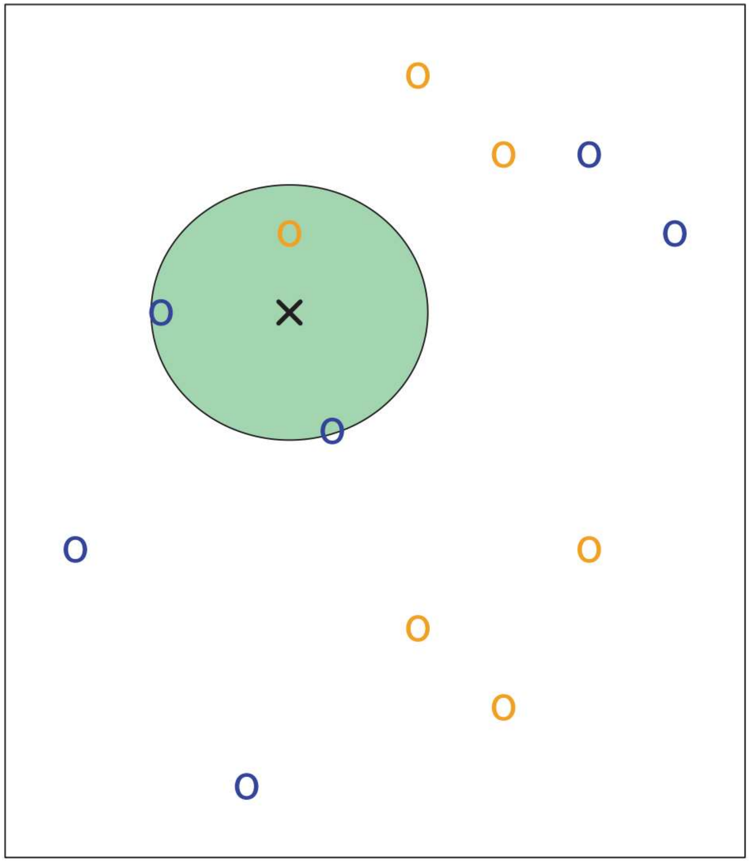
  - **then to classify an observation to the class with the highest estimated probability.**

3-nearest neighbors classifier



K-nearest neighbors classifier

- Small data consisting of **6 blue and 6 orange observations**.
- Goal: make a **prediction for** the point represented by a **black x**.
- Suppose that $\boldsymbol{K = 3}$,
  1. the classifier identifies the **three observations that are the closest** to x.
  2. the nearest neighbors result in **estimated probabilities of $2/3$ for the blue class** and $1/3$ orange class.
  3. KNN will predict that x belongs to the **blue class**.



K-nearest neighbors classifier

Given a positive integer K and a test observation x_0:

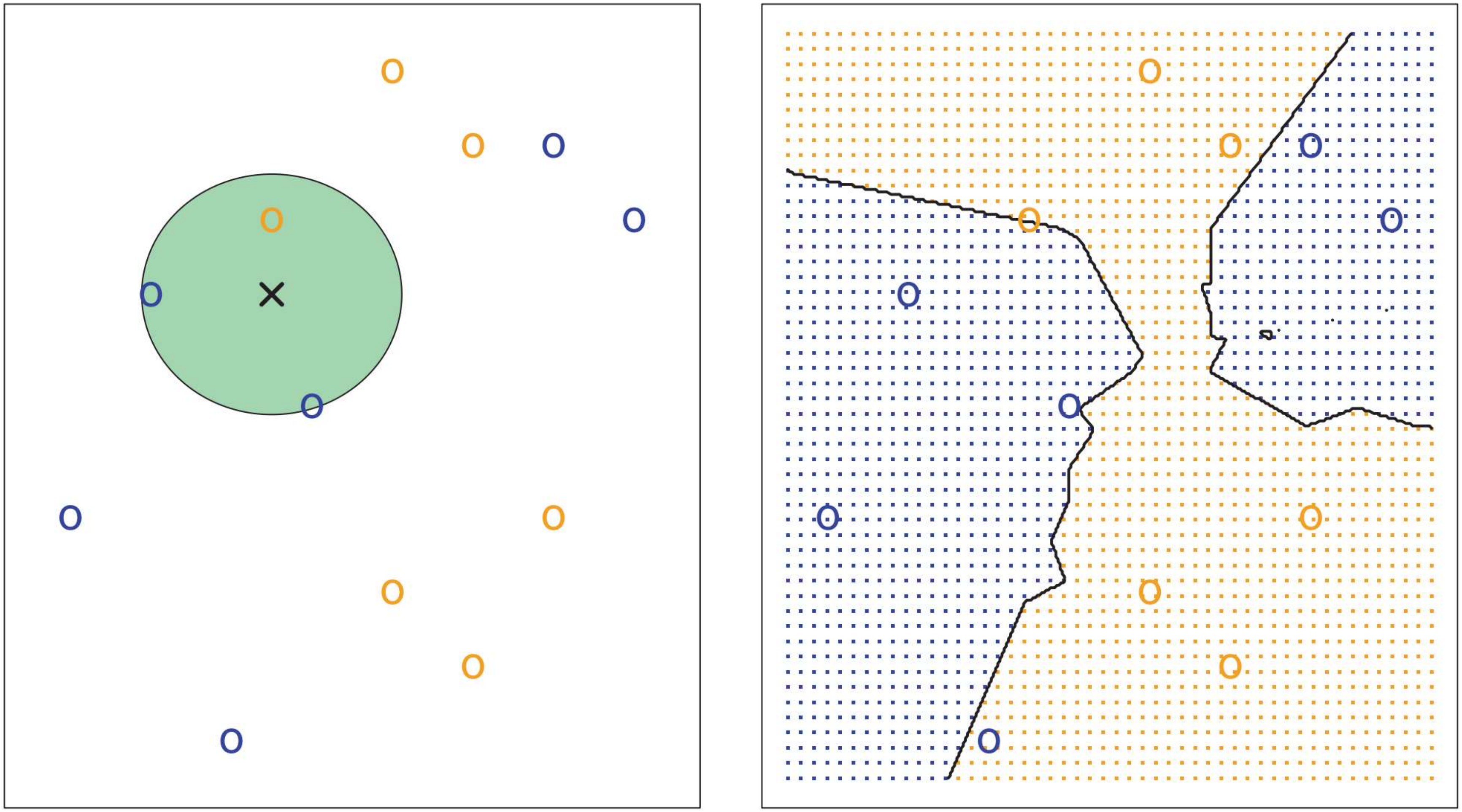
1. The classifier **identifies $\boldsymbol{\mathcal{N}_0}$** consisting of **$\boldsymbol{K}$ points in the training data that are the closest to $\boldsymbol{x_0}$**.

2. Then, it **estimates the conditional probability for class $\boldsymbol{j}$** as the fraction of points in $\mathcal{N}_0$ with response values equal to j:

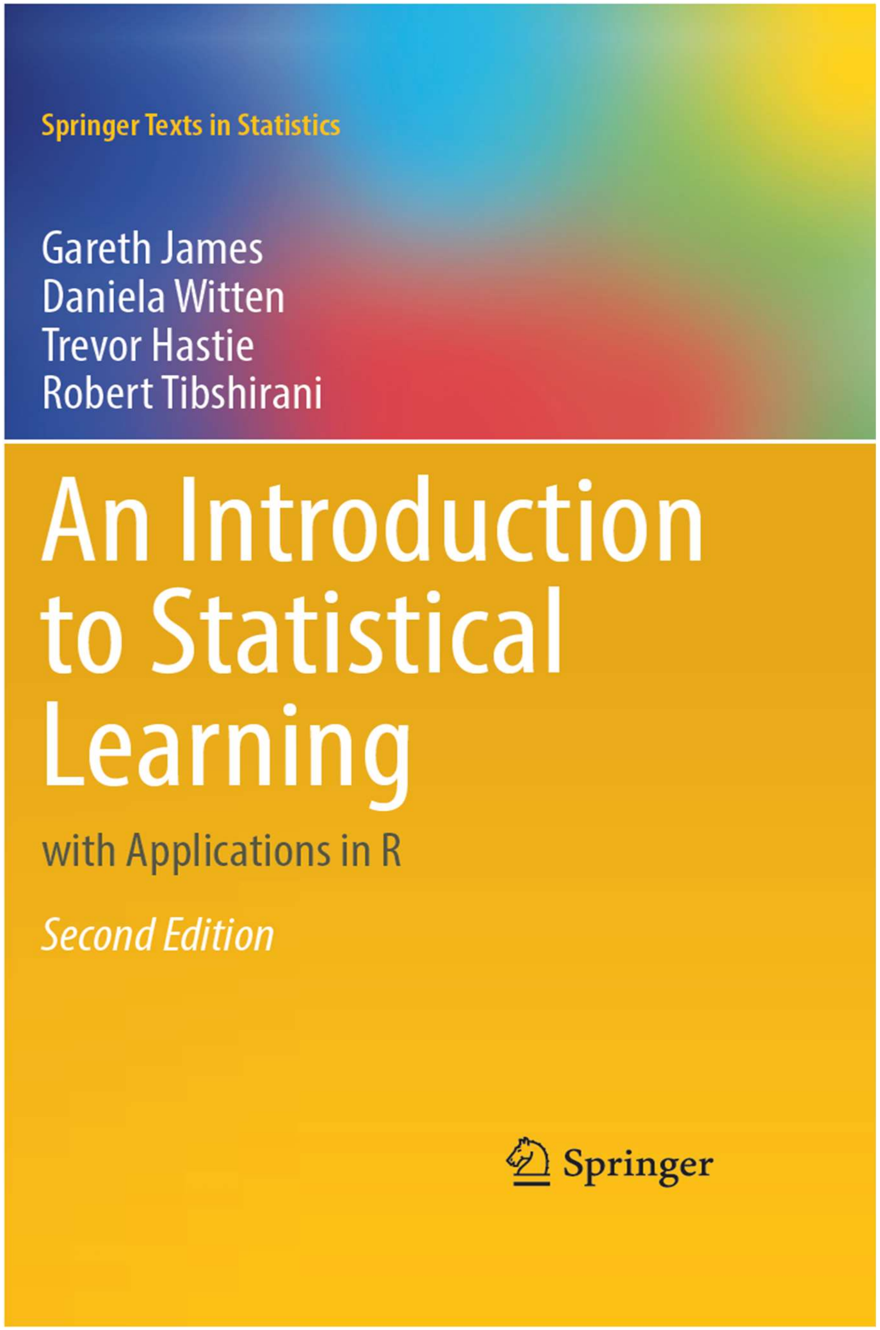
$$\Pr(Y = j | X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

3. It applies **Bayes rule** and classifies x_0 to the class with the **largest probability**.

K-nearest neighbors classifier



Reference


Springer Texts in Statistics
Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani
An Introduction to Statistical Learning
with Applications in R
Second Edition
Springer