A Not So Short Introduction To Statistical Machine Learning

SOSC 2022

Prof. Luca Scrucca
Università degli Studi di Perugia

✉ luca.scrucca@unipg.it
http://www.stat.unipg.it/luca
luca-scr
luca_scr

Perugia, 30 November 2022

All slides © Luca Scrucca
All slides can be used for educational purposes if this copyright notice is included
Permissions must be obtained from the copyright holder(s) for any other use

Data Science, Statistics, Machine Learning, AI, ...

- **Statistics** is the discipline that studies the collection, analysis, interpretation, presentation, and organization of data.

  Classical statistical methods:
  - Point estimation
  - Confidence intervals
  - Hypothesis testing
  - Statistical models for regression and classification
  - Likelihood-based inference
  - Bayesian inference
  - ...

- Leo Breiman ("Statistical Modeling: The Two Cultures", Statistical Science, 2001) described "two cultures":

  1. "generative" modeling culture which seeks to develop stochastic models that fit the data, and then make inferences about the data-generating mechanism based on the structure of those models. Implicit is the notion that there is a true model generating the data, and often a "best" way to analyze the data.

  2. "predictive" modeling culture which focuses on predictions, ignoring the underlying data generating mechanism, and discuss only accuracy of predictions made by different algorithms.

- According to Breiman "Statistics starts with data. Think of the data as being generated by a black box [...]"

  Two main goals can be pursued when analyzing data:

  - Prediction, i.e to be able to predict what the responses are going to be to future input variables;
  - Inference, i.e to infer how nature is associating the response variables to the input variables.

- **Machine learning** is the study of computer algorithms that improve automatically through experience and by the use of data.

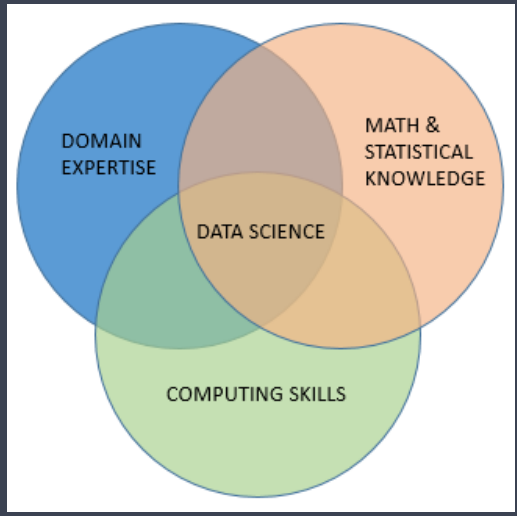
- In the context of predictive modelling, the difference between machine learning and statistical learning is blurred.
  - Machine learning tends to be focused more on developing efficient algorithms that scale to large data in order to optimize a predictive model.
  - Statistical learning generally pays more attention to the probabilistic structure of the model in order to provide an assessment of the uncertainty.

References:

- Murphy K. P. (2022) Probabilistic Machine Learning: An Introduction, MIT Press https://probml.github.io/pml-book/book1.html
- Murphy K. P. (2023) Probabilistic Machine Learning: Advanced Topics, MIT Press https://probml.github.io/pml-book/book2.html

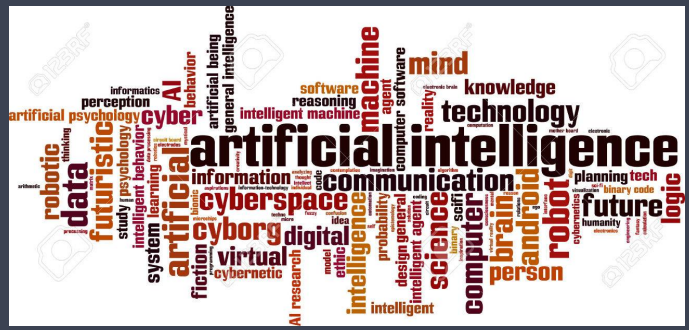
- **Data Science** is a vaguely defined, constantly changing, cross-disciplinary field.
- From a statistician point of view, data science can be seen as a broader view of statistics.

  When physicists do mathematics, they don't say they're doing "number science". They're doing math. If you're analyzing data, you're doing statistics. You can call it data science or informatics or analytics or whatever, but it's still statistics.
  — Karl Broman (U of Wisconsin)



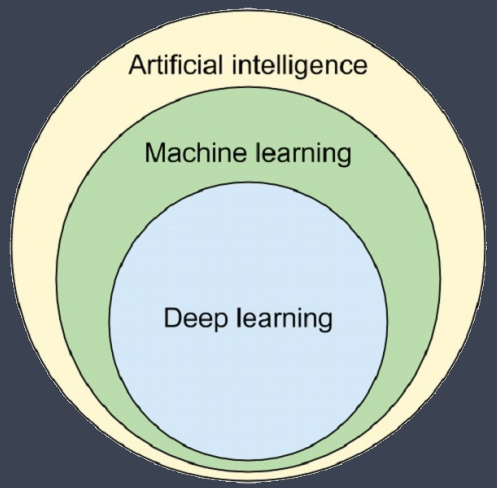
- **Big Data** refers to data sets that are too large or complex to be dealt with by traditional data analysis software.
  Big data are usually described in terms of three key concepts: volume, variety, and velocity.

- **Artificial Intelligence** (AI) refers to the ability of digital machines to perform tasks that typically require human intelligence.



Can machines think?
– Alan Turing ("Computing Machinery and Intelligence", 1950)

- AI can be seen as a branch of computer science, and in the years since its introduction in 1950s has experienced several waves of optimism followed by disappointment and the loss of funding (aka "AI winter"), followed by new approaches, success and renewed funding.
- There is a large debate and no single definition of the AI field that is universally accepted.
- In a broad sense, AI is an interdisciplinary science with multiple approaches, where advancements in machine learning and deep learning play a central role.



Supervised Learning

- Suppose we collected data for a sample of n observations. The **training set** is made of pairs of input and output variables:

$$\mathcal{D}_{\text{train}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$$

- Assume there exists a dependency between them, so the output y_i can be expressed as a function of the input variables $\boldsymbol{x}_i$ and some other unobservable (latent) variables $\boldsymbol{z}_i$:

$$y_i = f(\boldsymbol{x}_i, \boldsymbol{z}_i)$$

- The aim of **supervised learning** is to fit a model to learn the mapping from the observable input to the output

$$\widehat{y}_i = g(\boldsymbol{x}_i \mid \boldsymbol{\theta})$$

where $g(.)$ is a statistical model and $\boldsymbol{\theta}$ the unknown parameters.

- The learning task corresponds to finding the parameters that minimize a loss function measuring the deviation of our prediction $\widehat{y}_i$ from the observed output y_i:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} L(y_i, \widehat{y}_i) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} L(y_i, g(\boldsymbol{x}_i \mid \boldsymbol{\theta}))$$

- Different supervised learning algorithms differ in the models or the loss functions they assume, or the procedures they use in optimization.
- In **regression** problems
  - the output y_i is a numerical value (quantitative response)
  - $g(.)$ is a regressor function
  - loss is often the squared error, so the aim is to find the best $\boldsymbol{\theta}$ that minimize the fitting error.
- In **classification** problems
  - the output y_i is a discrete label (qualitative response)
  - $g(.)$ is a discriminant/classification function
  - loss is usually a 0/1 function, so the aim is to minimize the total number of misclassifications.

- Popular supervised learning models are:
  - Linear regression
  - Logistic regression
  - Generalized Linear Models (GLM)
  - Generalized Additive Models (GAM)
  - Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA)
  - Naive Bayes methods
  - Mixture models (e.g. Gaussian mixtures)
  - Decision Trees (Regression and Classification Trees)
  - Ensemble methos (Bagging, Random Forests, Boosting)
  - Support Vector Machines (SVM)
  - Neural Networks (NN) and Deep Learning (DL)

Unsupervised Learning

- Suppose we collected a dataset $\mathcal{D}_{\text{train}} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ composed of only a set of variables drawn from some unknown probability/density function

$$\boldsymbol{x}_i \sim p(\boldsymbol{x})$$

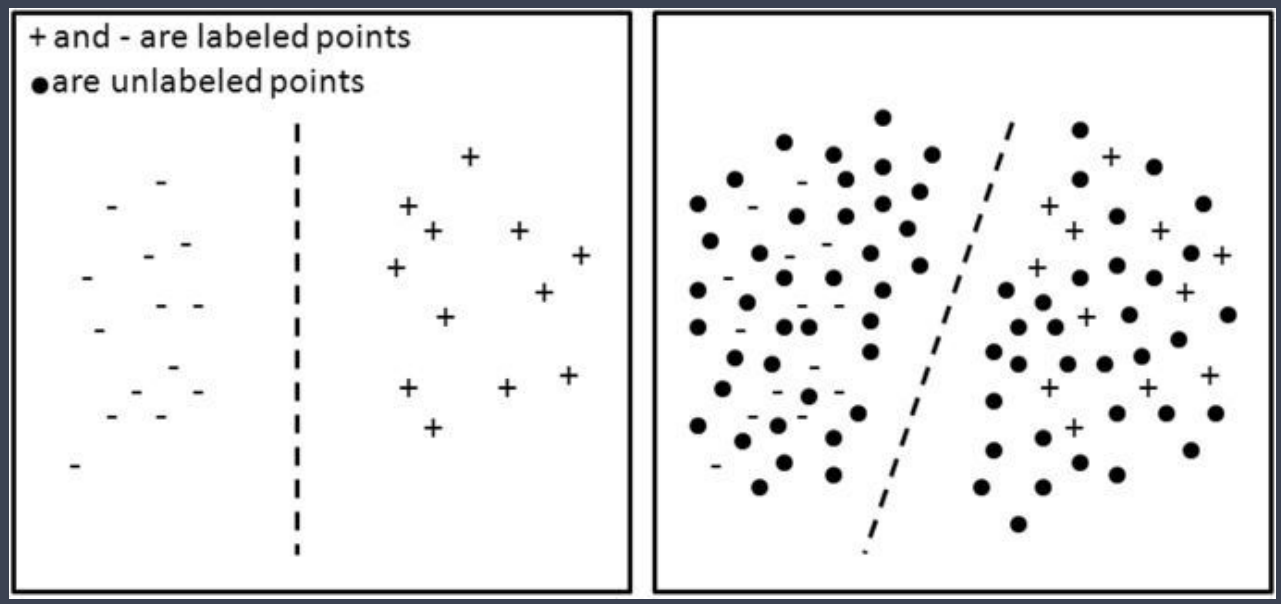
- In **unsupervised learning** for each case only the predictors vector $\boldsymbol{x}_i$ is observed, but there is no response y_i ($i = 1, \ldots, n$). Thus, we lack a response variable that can supervise our analysis.
- The aim is to estimate a model with parameters $\boldsymbol{\theta}$

$$\boldsymbol{x}_i \approx q(\boldsymbol{x} \mid \boldsymbol{\theta})$$

  where $q(.)$ is some working distribution depending on parameters $\boldsymbol{\theta}$.
- The learning task corresponds to finding the parameters that makes $q(.)$ as close as possible to the unknown $p(.)$ and from that understand the relationships between the variables or between the observations.
- Cluster analysis is a typical unsupervised learning task: look for the presence of one or more distinct groups of observations with no explicit assessment criterion because truth is not known (e.g. market segmentation to detect groups of customers).

Semi-supervised Learning

- Many problems fall naturally into the supervised or unsupervised learning paradigms.
  However… sometimes the question is less clear-cut.
- There can be situations where for a subset of $m < n$ observations we have information on both the predictors and the response variable, and for the remaining $n - m$ observations we have only predictor measurements but no response measurement.



- Such a scenario is referred to as **semi-supervised learning**.

- In general, suppose that we have observed
  - a quantitative response (dependent variable, output, ...) Y, and
  - a set of p different predictors (independent variables, covariates, features, ...) collected in the input vector $X = (X_1, X_2, \ldots, X_p)^\top$.
- Further assume that there is some relationship between Y and X, i.e.

$$Y = f(X) + \epsilon$$

  where
  - $f()$ represents the systematic information that X provides about Y;
  - ϵ is a random error term, which captures measurement errors and other discrepancies, independent of X and with zero mean.
- There are two main reasons to estimate $f()$:
  1. inference
  2. prediction

Inference

- In descriptive or explanatory modelling we want to understand how Y changes as a function of $(X_1, \ldots, X_p)$.
- Interesting questions:
  - Which predictors are associated with the response?
  - What is the relationship between the response and each predictor?
  - What is the functional form of the relationship between Y and each predictor?

Prediction

- In predictive modelling the goal is to predict the response variable based on the observed values of the predictors:

$$\widehat{Y} = \widehat{f}(X)$$

- $\widehat{f}()$ is often treated as a **black box**: we are not interested in knowing the exact form of $f()$, provided that it yields accurate predictions for Y.
- The prediction error of estimating Y using $\widehat{Y}$ can be decomposed as

$$Y - \widehat{Y} = f(X) + \epsilon - \widehat{f}(X) = (f(X) - \widehat{f}(X)) + (Y - f(X))$$

- Suppose that both $f()$ and X are fixed, then recalling that $\mathrm{E}[\epsilon] = 0$, the expected prediction error (under squared error loss) is given by

$$\begin{aligned}
\mathrm{E}[(Y - \widehat{Y})^2] &= \mathrm{E}[(f(X) + \epsilon - \widehat{f}(X))^2] \\
&= \mathrm{E}[(f(X) - \widehat{f}(X))^2] + \mathrm{E}[\epsilon^2] + 2\,\mathrm{E}[\epsilon(f(X) - \widehat{f}(X))] \\
&= (f(X) - \widehat{f}(X))^2 + \mathrm{V}[\epsilon] = \text{reducible error} + \text{irreducible error}
\end{aligned}$$

- Main goal: estimate $f()$ with the aim of minimizing the reducible error.
- The irreducible error provides a lower bound on the accuracy of our prediction for Y, and it is almost always unknown in practice.

Estimating $f()$

- Estimation (or learning in ML) is the process of applying a statistical/machine learning method to the training data to estimate the unknown function $f()$.
- Several approaches are available, both parametric and non-parametric.

Parametric methods

A two-step model-based approach:

1. Select the functional form, or shape, of $f()$.
   For example, the **linear model** assumes that $f()$ is linear in X:
   $$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$
2. Select a procedure that uses the training data to fit or train the model.
   For example, in the linear model case we only need to **estimate the parameters** $(\beta_0, \beta_1, \beta_2, \ldots, \beta_p)$. A popular approach is (ordinary) least squares (OLS), but many other exists (maximum likelihood, regularized ML, Bayesian estimation, ...).

- This model-based approach is called parametric because it reduces the problem of estimating $f()$ down to one of estimating a set of parameters (the coefficients of the model).

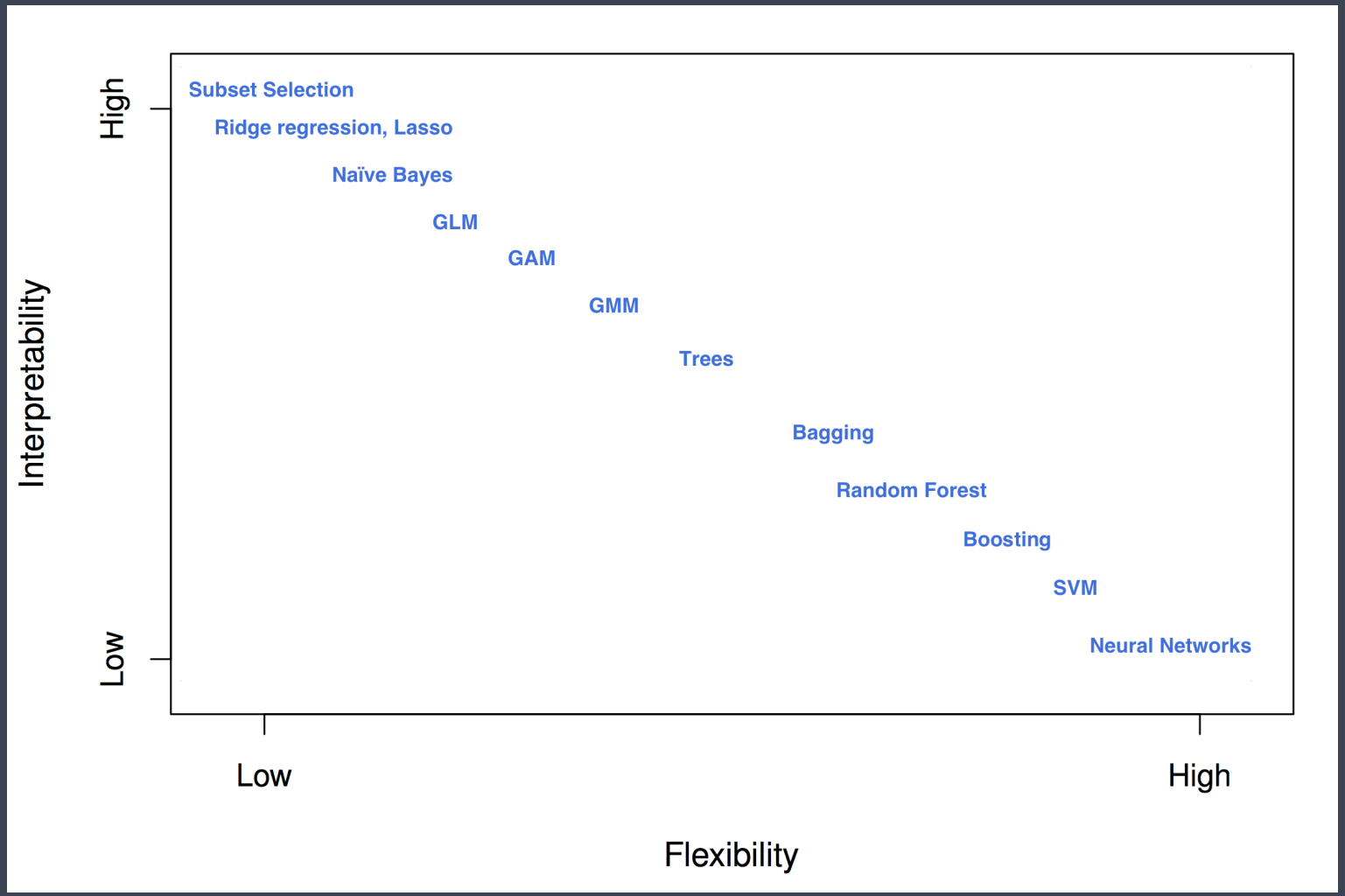
- Pros: Generally is much easier to estimate a set of parameters than it is to fit an entirely arbitrary function $f()$.
- Cons: the selected model can be a poor approximation of true unknown form of $f()$.

Non-parametric methods

- Non-parametric methods do not make explicit assumptions about the functional form of $f()$.
- They try to estimate $f()$ getting as close to the data points as possible without being too rough or wiggly.
- Pros: avoid the assumption of a particular functional form for $f()$, so they have the potential to accurately fit a wider range of possible shapes for $f()$.
- Cons: a large number of observations is required to accurately estimate $f()$.

Trade-off between model interpretability and flexibility

- If we are mainly interested in explanatory inference, then simple models (e.g. Linear Models, Logistic Regression) are much more interpretable than black-box models (e.g. Random Forest, SVM, Neural Networks).
- Flexible models allow to fit many different possible functional forms for $f()$, but usually require estimating a larger number of parameters.
- In general, as the flexibility of a model/algorithm increases, its interpretability decreases.
- **Overfitting** is the main risk, i.e. to follow the observed data (including the error/noise component) too closely.
- If we are only interested in prediction, then the interpretability of the predictive model may be simply not of interest.
- Flexible models may provide good fit but there is the risk of overfitting.
- Models involving fewer variables are often preferred over more complicated models involving several variables or features.



Interpretability vs flexibility using different statistical/machine learning methods

Assessing model accuracy

- Suppose we fit a model $f(x)$ to some training data $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, and we wish to assess its performance.
- Compute the **average squared prediction error** over $\mathcal{D}_{\text{train}}$:

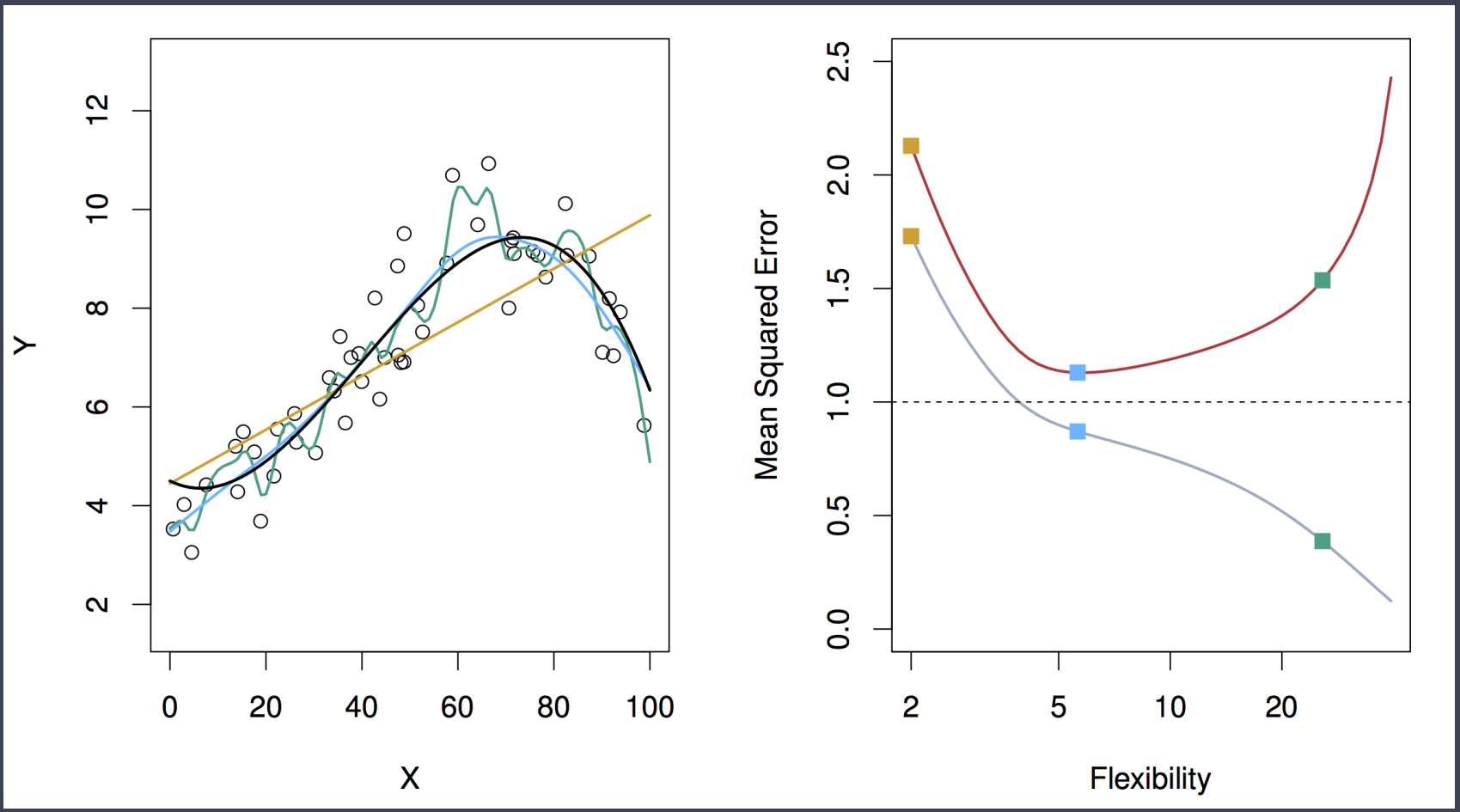
$$\text{MSE}_{\text{train}} = \frac{1}{n} \sum_{i \in \mathcal{D}_{\text{train}}} [y_i - \widehat{f}(\boldsymbol{x}_i)]^2$$

  - Since the same data is used both for "learning" and for "evaluating" the fit of a model, this gives an optimistic evaluation of model accuracy.
  - If used for selecting the complexity of a statistical model, it is biased toward overfitting models.

- Compute the MSE on a **test set** $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, i.e. a fresh dataset not used for parameters estimation:

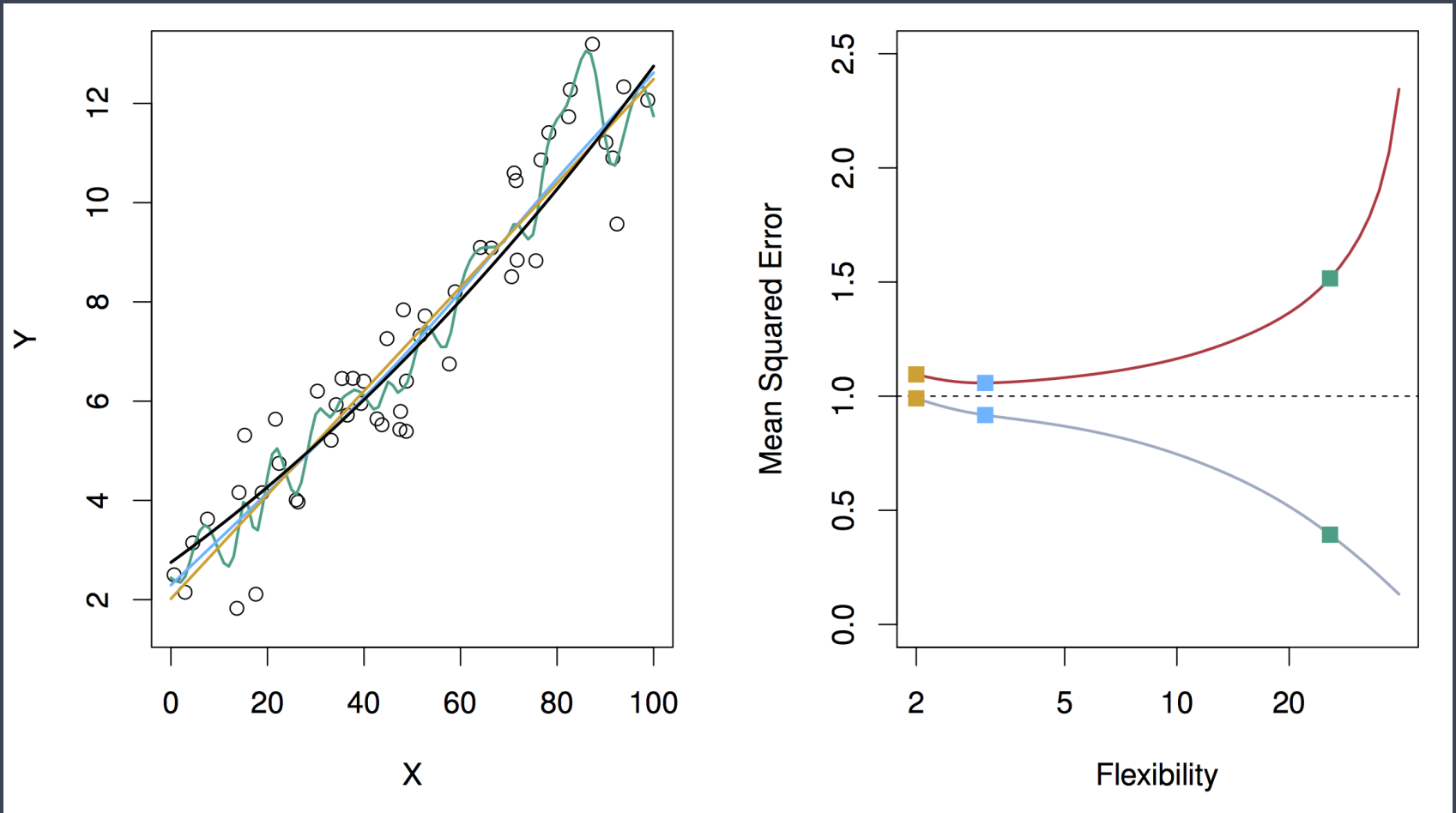
$$\text{MSE}_{\text{test}} = \frac{1}{m} \sum_{i \in \mathcal{D}_{\text{test}}} [y_i - \widehat{f}(\boldsymbol{x}_i)]^2$$

  - This is a more realistic measure of how accurately an algorithm is able to predict outcome values for previously unseen data.



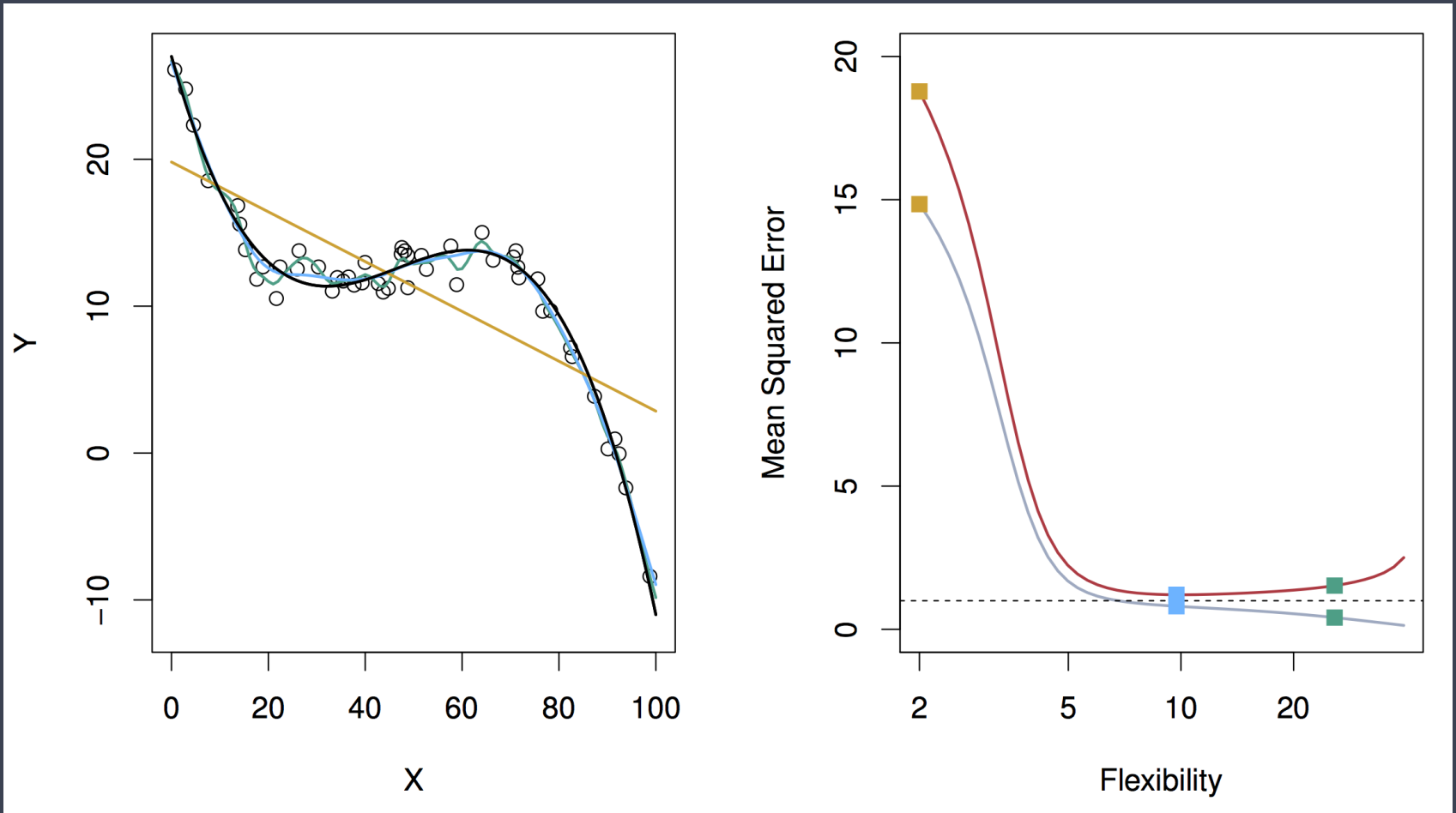
Models: 1) true model —— 2) linear regression fit —■— 3) smoothing spline fit with medium flexibility —■— 4) smoothing spline fit with high flexibility —■—

Accuracy measure: (i) training MSE —— (ii) test MSE —— (iii) irreducible MSE – – –



Models: 1) true model —— 2) linear regression fit —■— 3) smoothing spline fit with medium flexibility —■— 4) smoothing spline fit with high flexibility —■—

Accuracy measure: (i) training MSE —— (ii) test MSE —— (iii) irreducible MSE – – –



Models: 1) true model —— 2) linear regression fit —■— 3) smoothing spline fit with medium flexibility —■— 4) smoothing spline fit with high flexibility —■—

Accuracy measure: (i) training MSE —— (ii) test MSE —— (iii) irreducible MSE – – –

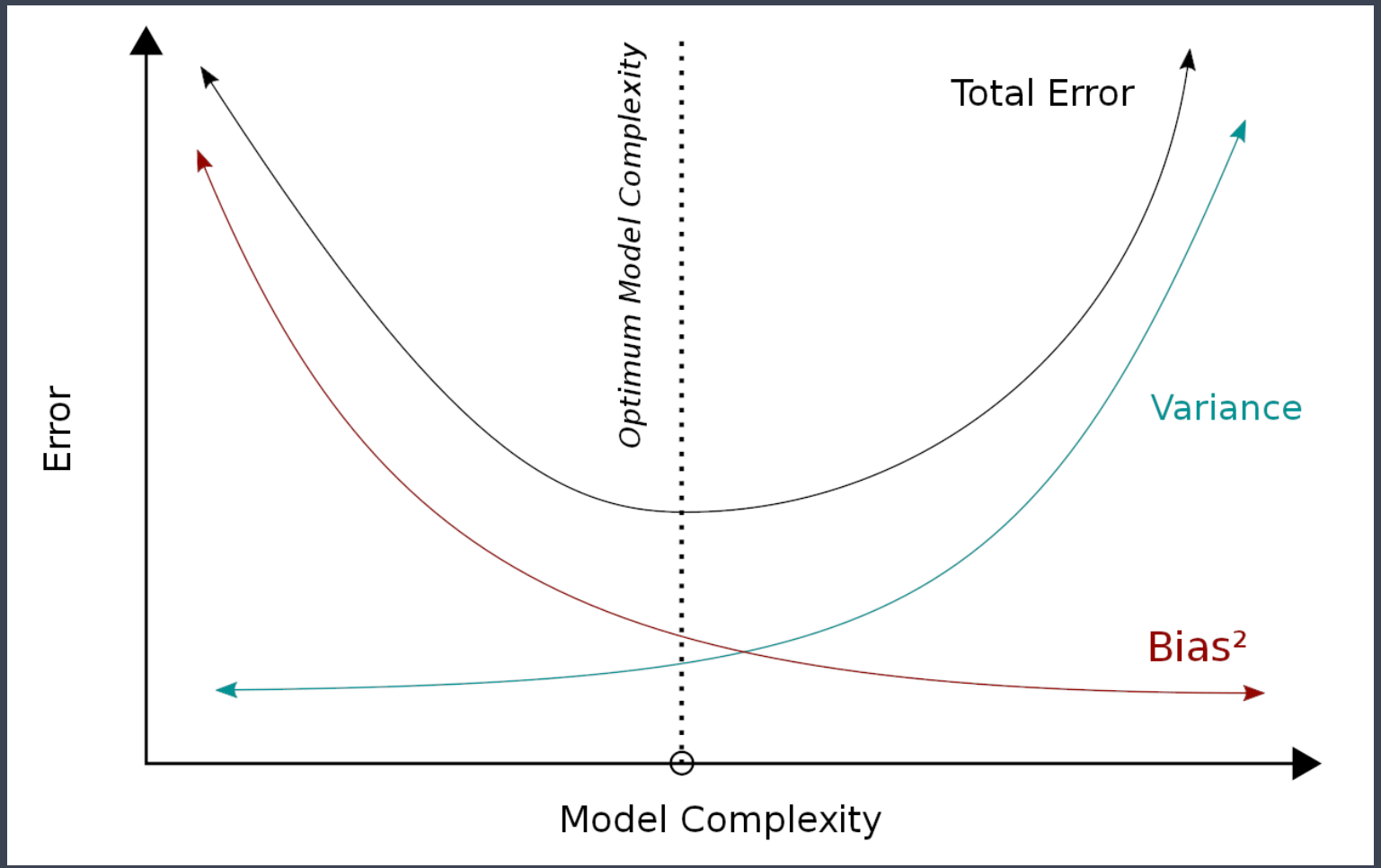
Bias-Variance trade-off

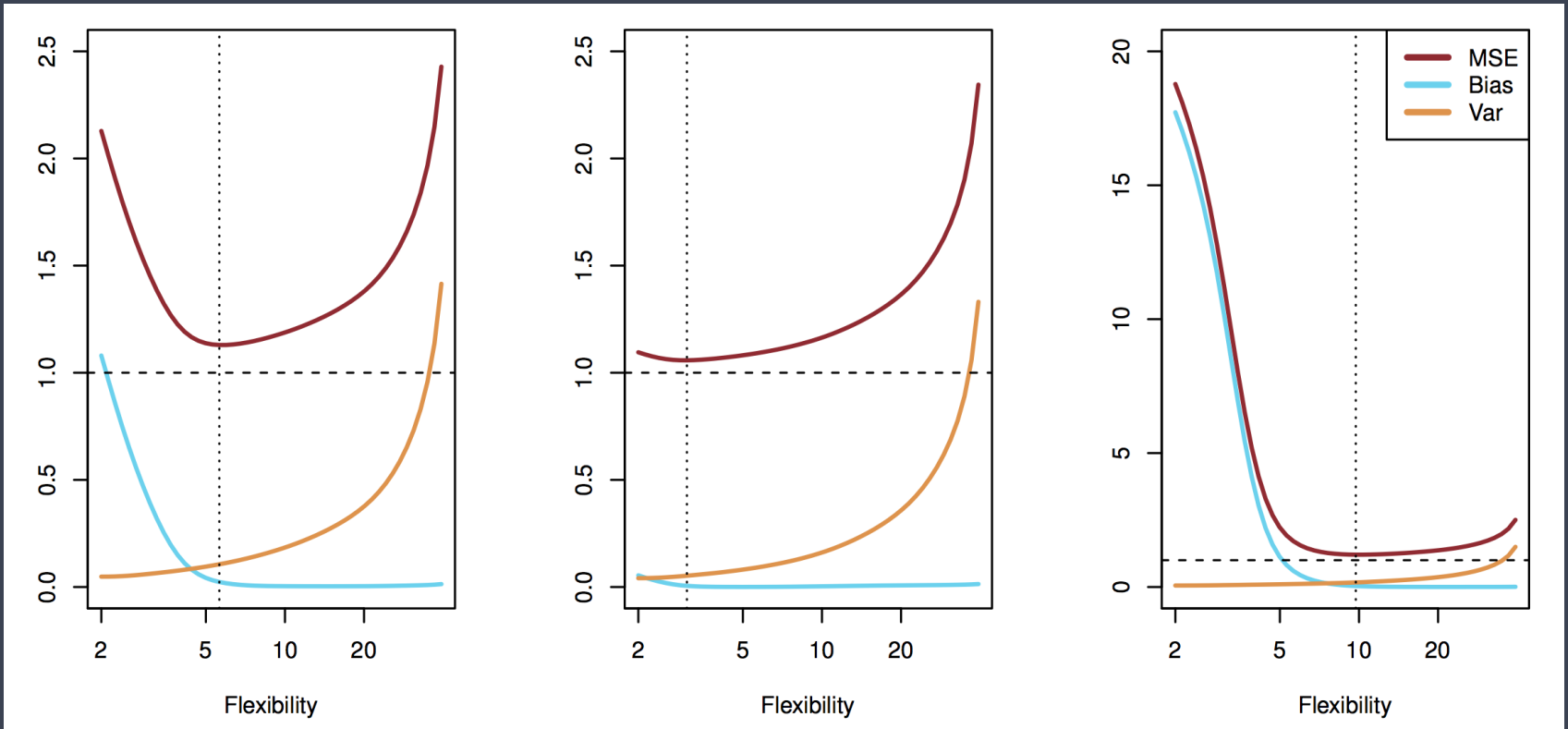
- The expected test error for a new observation value $\boldsymbol{x}_0$ can always be decomposed as

$$\mathrm{E}[y_0 - \widehat{f}(\boldsymbol{x}_0)]^2 = \mathrm{V}[\widehat{f}(\boldsymbol{x}_0)] + \mathrm{B}[\widehat{f}(\boldsymbol{x}_0)]^2 + \mathrm{V}[\epsilon]$$

  where
  - $\mathrm{V}[\widehat{f}(\boldsymbol{x}_0)]$ is the variance expressing the amount by which $\widehat{f}()$ would change if we estimated it using a different training dataset;
  - $\mathrm{B}[\widehat{f}(\boldsymbol{x}_0)]$ is the bias expressing the error that is introduced by approximating the data distribution by a statistical model;
  - $\mathrm{V}[\epsilon]$ is the irreducible error.

- The expected test error can never be smaller than the irreducible error.
- In general, more flexible statistical methods have higher variance and smaller bias. On the contrary, simpler models have smaller variance but higher bias.
- To minimize the expected test error, we need to select a statistical learning method that simultaneously achieves low variance and low bias.


Error
Optimum Model Complexity
Total Error
Variance
Bias²
Model Complexity



MSE and the bias-variance trade-off

- Suppose that we seek to estimate $f()$ on the basis of the training observations $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, where in this case $y_i \in \{C_1, C_2, \ldots, C_K\}$ is the class or label associated with the ith observation.

- **Training classification error rate** is the proportion of misclassified observations, i.e.

$$\text{CE}_{\text{train}} = \frac{1}{n} \sum_{i \in \mathcal{D}_{\text{train}}} \mathbb{1}(y_i \neq \widehat{y}_i)$$

  where

  - $\widehat{y}_i$ is the predicted class label for the ith observation using $\widehat{f}()$;
  - $\mathbb{1}(y_i \neq \widehat{y}_i)$ is the indicator function that returns 1 if $y_i \neq \widehat{y}_i$ and 0 otherwise.

- The **test classification error rate** associated with a set of test observations $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$ is given by

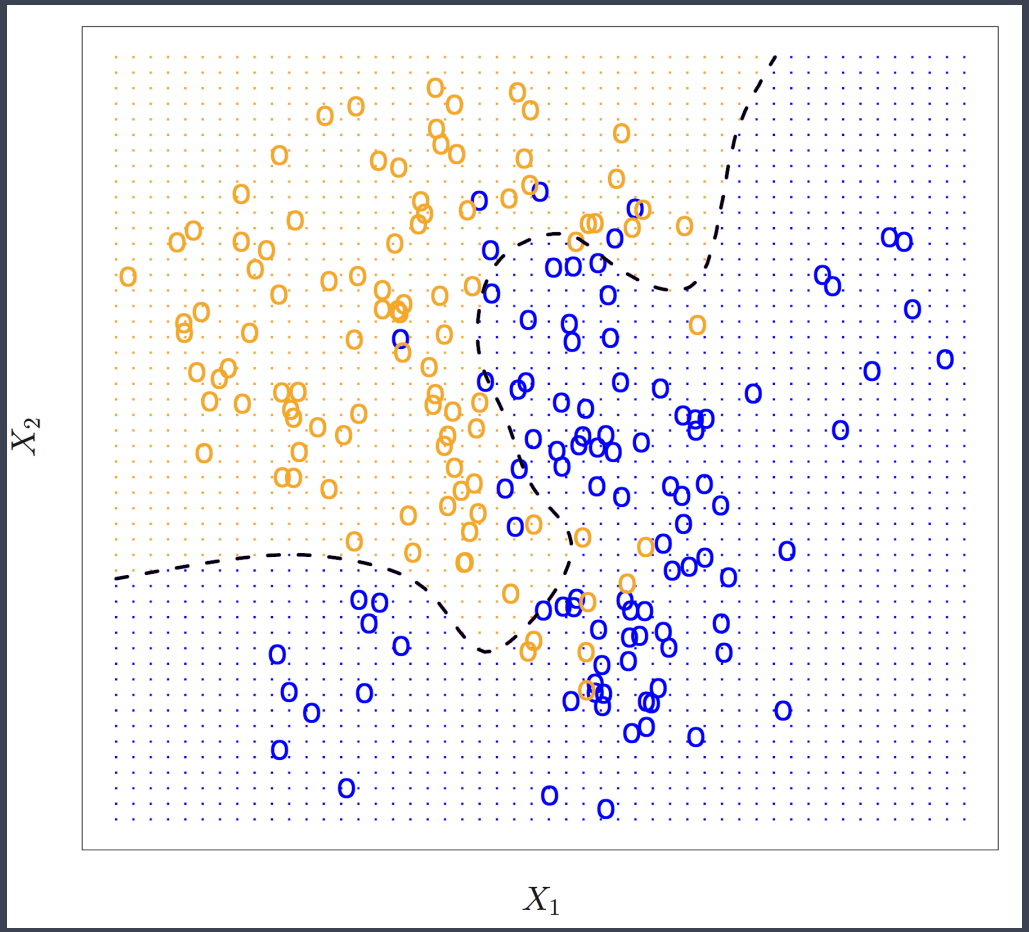
$$\text{CE}_{\text{test}} = \frac{1}{m} \sum_{i \in \mathcal{D}_{\text{test}}} \mathbb{1}(y_i \neq \widehat{y}_i)$$

The Bayes Classifier

- The test error rate CE_{test} is minimized, on average, by a very simple classifier that assigns each observation to the most likely class, given its predictor values.
- According to the **Bayes classifier**, a test observation with predictor vector $\boldsymbol{x}_0$ should be assigned to the class C_k (with $k = 1, \ldots, K$) for which

$$\Pr[Y = C_k \mid \boldsymbol{x}_0] \qquad \text{is maximum}$$

- The Bayes error rate is the lowest possible test error rate produced by the Bayes classifier:
  - The error rate at $\boldsymbol{x}_0$ is $1 - \max_k \Pr[Y = C_k \mid \boldsymbol{x}_0]$.
  - The overall Bayes error rate is $1 - \text{E}\left[\max_k \Pr[Y = C_k \mid X]\right]$
  - The Bayes error rate is analogous to the irreducible error for classification tasks.
- The Bayes decision boundary defines the regions in which a test observation will be assigned to one of the K classes.



Two-class simulated dataset. The dashed line represents the Bayes decision boundary with Bayes error rate $\approx$ 13%

- For real data, we do not know the conditional distribution of Y given X, and so computing the Bayes classifier is impossible.

Model validation

- Often models require the tuning of hyper-parameters (k in KNN, number of components in GMM, smoothing parameter, lasso and/or ridge parameters, number of hidden layers and number of nodes in NNET, etc.).
- Sometimes we have no test data available for estimating MSE, classification error, etc.
- In all these cases, a separated validation dataset $\mathcal{D}_{\text{val}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{V}$ should be used.
- However, instead of setting aside a validation set, it is preferable to use resampling methods, such as the cross-validation.
- No free lunch theorem in statistics: *no method/algorithm/model dominates all others over all possible datasets*.
- Realistically, we should decide for any given set of data which method produces the best results.
- This is the most challenging part of statistical learning in practice.

Resampling

- Resampling methods are a fundamental tool in modern statistics.
- They involve repeatedly drawing samples from a training set and refitting a model of interest on each sample to obtain additional information about the fitted model.
- They can be computationally expensive, because the same statistical model must be fitted multiple times using different subsets of the training data.
- Goals
  - Model assessment (evaluating model's performance)
  - Model selection (selecting the level of flexibility of a model, i.e. hyperparameters tuning)
  - Model inference (provide a measure of accuracy of a parameter estimate or of a given statistical learning method)

- Several possible **performance metrics** can be adopted.
- For regression problems, the error is usually measured by the **root mean square error**:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_i [y_i - \widehat{f}(x_i)]^2}$$

  or directly using the MSE.
- For classification problems, the error can be measured by the **classification error**:

$$\text{CE} = \frac{1}{n}\sum_i \mathbb{1}(y_i \neq \widehat{y}_i)$$

  Many other measures are available: sensitivity/specificity, ROC-AUC, precision/recal, F-score, log-loss or cross-entropy, Brier score, etc.
- If a validation set is not available, an estimate of the true error must be obtained by resampling methods.

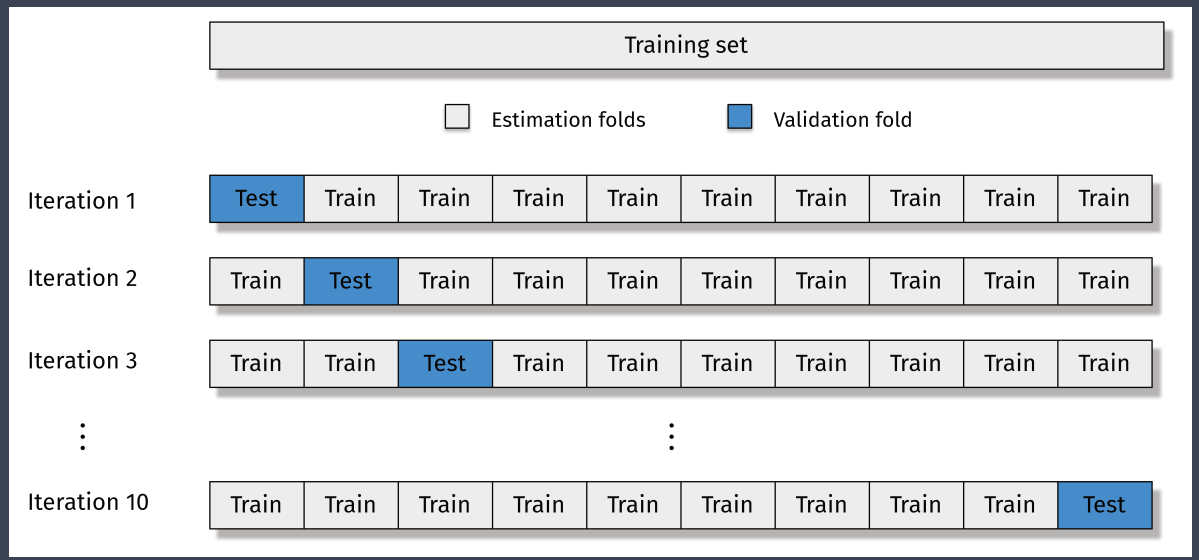
Cross-validation

- Cross-validation is a widely used resampling approach for estimating the performance of a statistical/machine learning model/algorithm.

V-fold cross-validation

The set of training observations is randomly splitted into V parts or folds. The model is trained using all but the vth fold, then the remaining vth fold is used as validation set. This is done in turn for each fold $v = 1, \ldots, V$, and then the results are combined.

10-fold cross-validation scheme



- When $V = n$, the procedure is called **leave-one-out cross-validation** (LOOCV), because we leave out one data point at a time.

V-fold cross-validation algorithm for a classification task

1. Divide the training set $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ into V folds of roughly the equal size (and balanced): $F_1, \ldots, F_V$, where fold F_v has size n_v ($n = n_1 + n_2 + \ldots + n_V$).
2. For $v = 1, \ldots, V$ do
   - define the training set as $\{(\boldsymbol{x}_i, y_i); i \notin F_v\}$
     and the validation set as $\{(\boldsymbol{x}_i, y_i); i \in F_v\}$
   - train the classifier using all but fold F_v
   - validate the classifier by computing the classification error on fold F_v

$$\text{CE}_v = \frac{1}{n_v} \sum_{i \in F_v} \mathbb{1}(y_i \neq \widehat{y}_i)$$

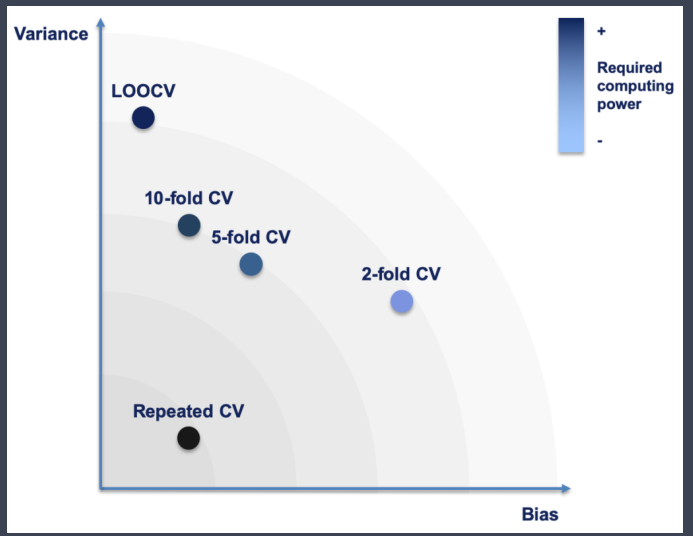
3. Compute the (weighted) average error over all folds:

$$\text{CE}_{\text{CV}} = \sum_{v=1}^{V} \frac{n_v}{n} \text{CE}_v = \frac{1}{n} \sum_{v=1}^{V} \sum_{i \in F_v} \mathbb{1}(y_i \neq \widehat{y}_i)$$

- Leave-one-out CV (**LOOCV**) follows the same algorithm outlined above by removing one observation at time (i.e. set $V = n$).
  LOOCV has the potential to be computationally expensive because the model has to be fit n times.
- **Repeated V-fold CV** creates multiple versions of the folds and aggregates the results. Research indicates that this procedure can be used to effectively increase the precision of the estimates while still maintaining a small bias.
- Leave-group-out or Monte-Carlo CV (**LGOCV**) repeatedly splits the data into training and validation sets. Usually, 70–80% of the sample is used for training.

Bias-variance trade-off for V-fold CV

- LOOCV has less bias than V-fold CV, but much higher variance.
- LGOCV has less variability than V-fold CV, but larger bias.
- V-fold CV often gives more accurate estimates of the test error rate than LOOCV.



- Empirically it has been shown that V-fold CV with $V = 5$ or $V = 10$ yields test error rate estimates that have neither excessively high bias nor high variance.

Bootstrap

- The **bootstrap** is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
  For example, it can provide an estimate of the standard error of a coefficient of a regression model, or a confidence interval for that coefficient.
- Bootstrap takes random samples with replacement of the same size as the original data set.
- Since sampling is made with replacement, some observations may be selected more than once and each observation has a 63.2% chance of showing up at least once.

  The probability for an observation of not being selected in any of n draws from n samples with replacement is $(1 - 1/n)^n$.
  Then $\lim_{n\to\infty}(1-1/n)^n = e^{-1} \approx 0.368$, *and the probability of being selected at least once is* $1 - e^{-1} \approx 0.632$.
- The observations not selected (approximately 1/3 of the sample) are usually referred to as the out-of-bag observations.

Bootstrap algorithm for a classification task

1. Create a bootstrap sample by random sampling with replacement;
2. Fit a classifier using the bootstrap sample as training set;
3. Predict out-of-bag observations to get bootstrap classification error;
4. Repeated steps 1-3 multiple times (usually $30 - 100$) and then combine the results.

Bias-variance trade-off for bootstrap

- The bootstrap estimates of error rate have less variability than V-fold CV, but larger bias (similar to 2-fold CV).
  If the training set size is small, this bias may be problematic, but will decrease as the training set sample size becomes larger.
- The "632" bootstrap method tries to reduce the bias by creating a performance estimate that is a combination of the simple bootstrap estimate and the estimate from predicting the training set:

$$(0.632 \times \text{bootstrap error rate}) + (0.368 \times \text{training error rate})$$

One standard error rule

- Instead of selecting the model with the “best” tuning parameter value, other schemes for selecting a single model can be used.
- A popular choice is the so-called “one standard error rule”:

  “all else equal (up to one standard error), go for the simpler (more regularized/parsimonious) model”

- In practice:
  - the model with the best performance value is identified;
  - an estimate of the standard error of performance is computed by a resampling method;
  - the final model is the simplest model whose estimated performance is within one standard error from the best model performance.

Steps of Statistical/Machine Learning process

Tidy datasets are all alike, but every messy dataset is messy in its own way.
— Hadley Wickham (RStudio)

1. **Data collection**
   - The data collection step involves gathering the data that will be used to generate actionable knowledge.
   - The quality of any SL/ML project is based largely on the quality of its input data ("Garbage in–Garbage out").
2. **Data preparation and exploration**
   - Data preparation involves fixing or cleaning so-called "messy" data, eliminating unnecessary data, and recoding the data to conform to the learner's expected inputs. Often this may take a lot more time than modelling.
   - Data exploration is the process of learning more about the data and the problem(s) to be solved using data.
   - Selection or relevant variables/features, transformation, normalization (feature engineering) is often one of the most important part of any statistical/data science project.
   - It is often said that 80% of data analysis is spent on the cleaning and preparing data.

3. **Model training**
    - By the time the data has been prepared for analysis, the specific statistical/machine learning task chosen will inform the selection of an appropriate algorithm, and the algorithm will represent the data in the form of a model.
4. **Model evaluation**
    - Because each statistical/machine learning algorithm results in a biased solution to the learning problem, it is important to evaluate how well the algorithm learns from its experience.
    - Depending on the type of model used, you might be able to evaluate the accuracy of the model using a test dataset or you may need to develop measures of performance specific to the intended application.
5. **Model improvement**
    - If better performance is needed, it becomes necessary to utilize more advanced strategies to augment the performance of the model.
    - Sometimes, it may be necessary to switch to a different type of model altogether.
    - You may need to supplement your data with additional data or perform additional preparatory work as in step two of this process.

Statistical/Machine Learning Pipeline