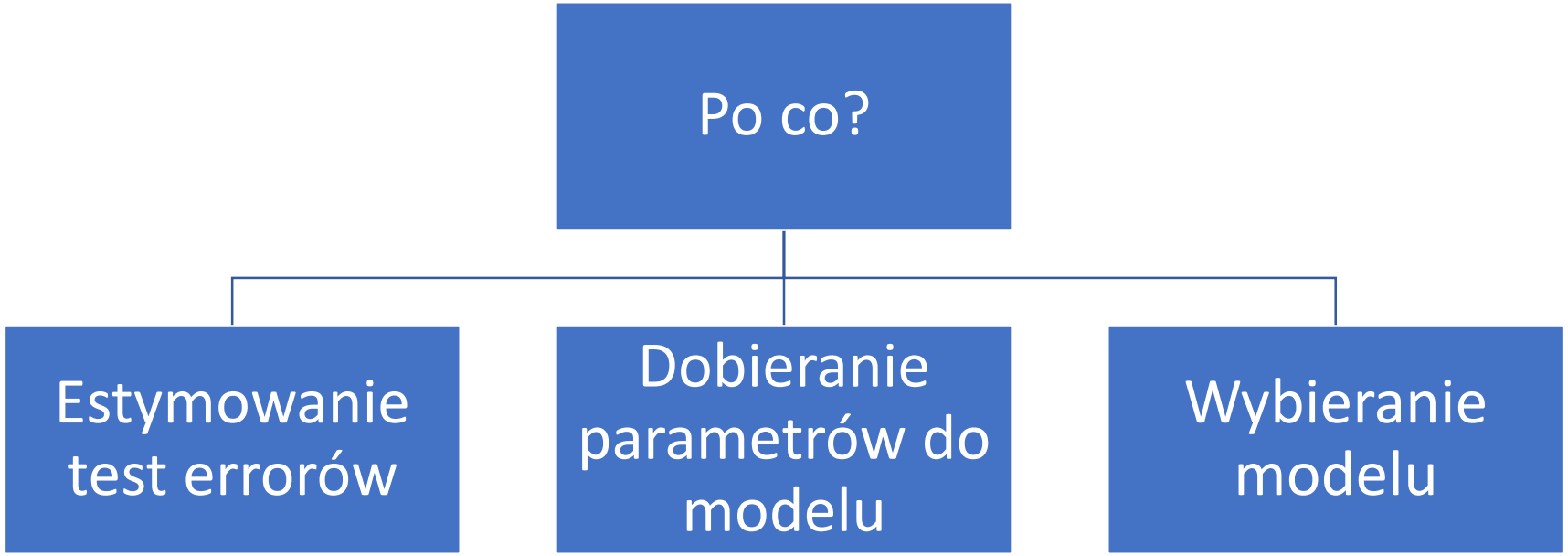
Resampling methods
Cross-Validation

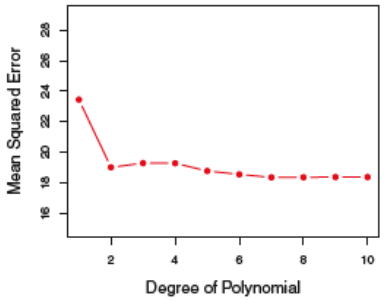
Igor Adamiec

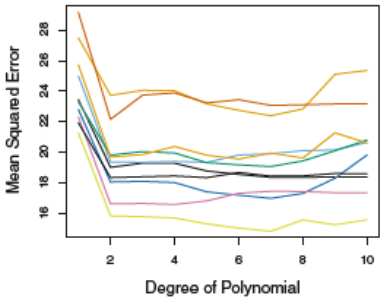
Metody ponownego próbkowania



Wady: wymaga dużej mocy obliczeniowej

Cross-validation



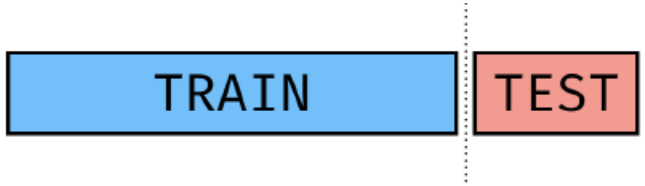


Dzielimy zbiór danych na zbiór treningowy (training set), na którym budujemy model, oraz na zbiór testowy (test set, a w książce validation set), na którym model testujemy.

Robimy to, ponieważ gdybyśmy testowali model na tych samych danych, na których go budowaliśmy, to byłoby to bezsensowne.

Testując zbiór na zbiorze testowym (czyli na danych, których nigdy nie widział) badamy jak model zachowa się przy nowych danych. To właśnie ta zdolność predykcyjna jest dla nas najważniejsza

Podział na zbiory treningowy i testowy powinien być losowy.

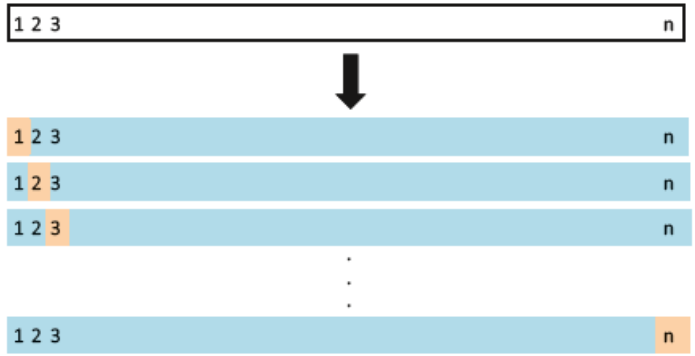


Leave-One-Out CV

LOOCV wymaga dużo mocy obliczeniowej, ponieważ zakłada stworzenie n modeli (tyle modeli ile obserwacji w zbiorze).

Każdy model trenowany jest na n-1 obserwacji, a na tej wyrzuconej jest trenowany.
Oznacza to, że pierwszy model trenujemy go na obserwacjach 2-n, a testujemy na 1. Drugi model trenujemy na obserwacjach 1 i 3-n, a testujemy na 2.
Ostatni model trenujemy na obserwacjach 1- n-1, a testujemy na n.

Wybieramy dowolną statystykę (np. MSE) i na koniec wyliczamy średnią ze wszystkich modeli



$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i.$$

Dla regresji liniowej najmniejszych kwadratów i regresji wielomianowej nie trzeba obliczać n modeli. Istnieje magiczny wzór:

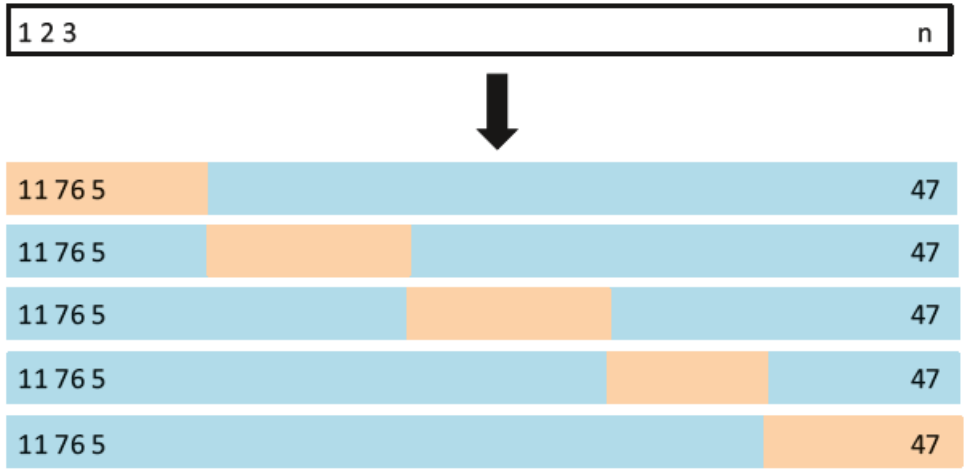
$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2,$$

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2}.$$

k-fold CV

Dzielimy zbiór na k równych części i tworzymy k modeli. Za każdym razem inny podzbiór pełni rolę zbioru testowego, a model buduje się na pozostałych, k-1, podzbiorach.

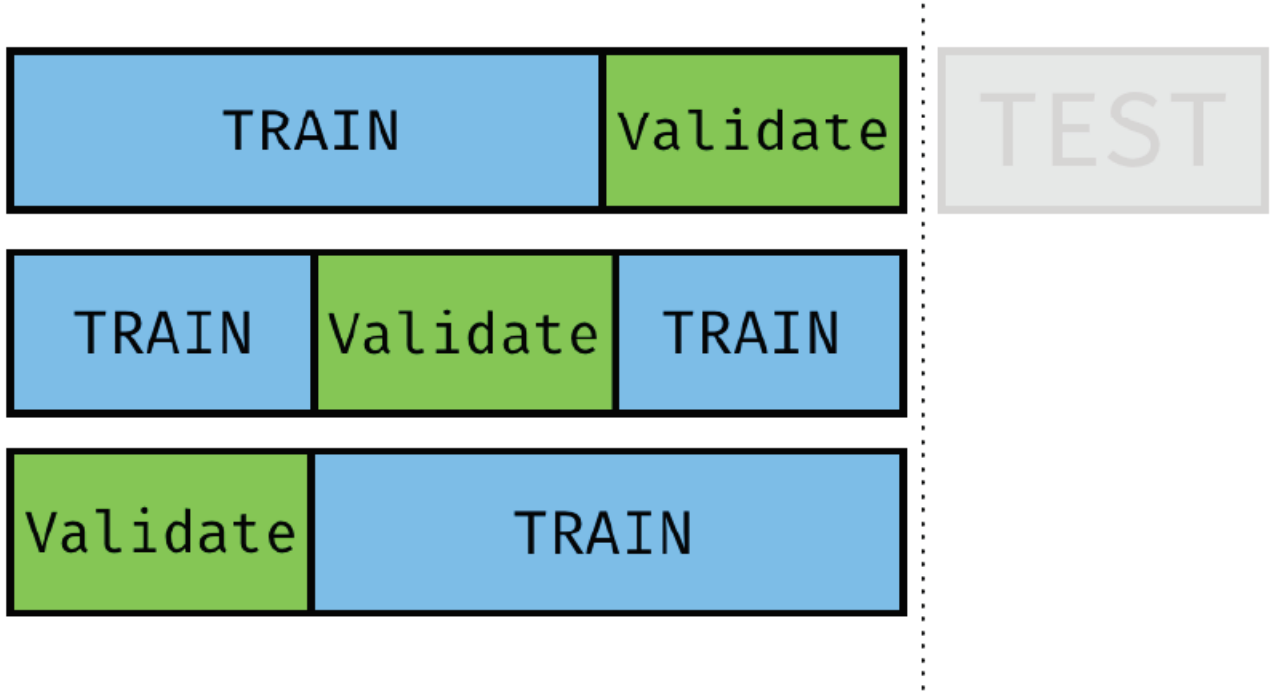
Model ocenia się tak samo – wylicza się średnią z dowolnej statystyki testowej



$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i.$$

Rzeczywiste podejście

Zazwyczaj zawsze dzieli się zbiór na treningowy i testowy i to na zbiorze treningowym możemy wykonać jakąkolwiek CV. To nam pozwala ocenić jaki model z jakimi parametrami będzie najlepszy do naszych danych.



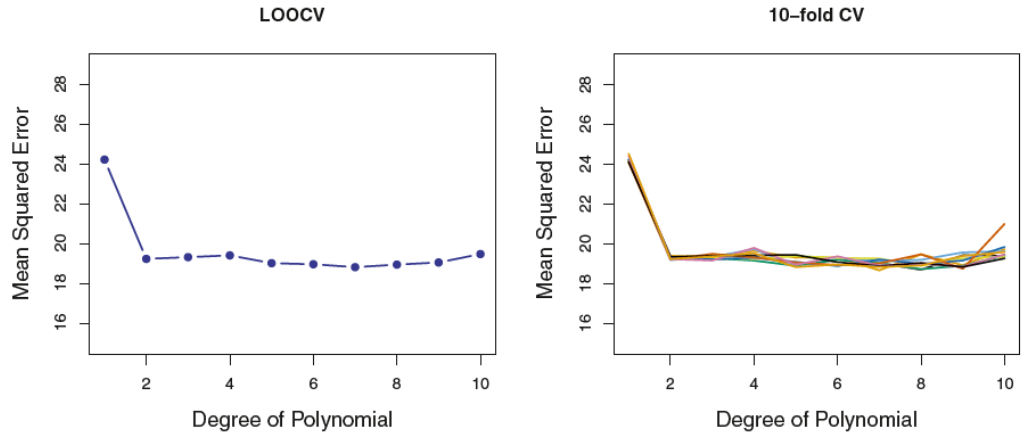


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10*-fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.*

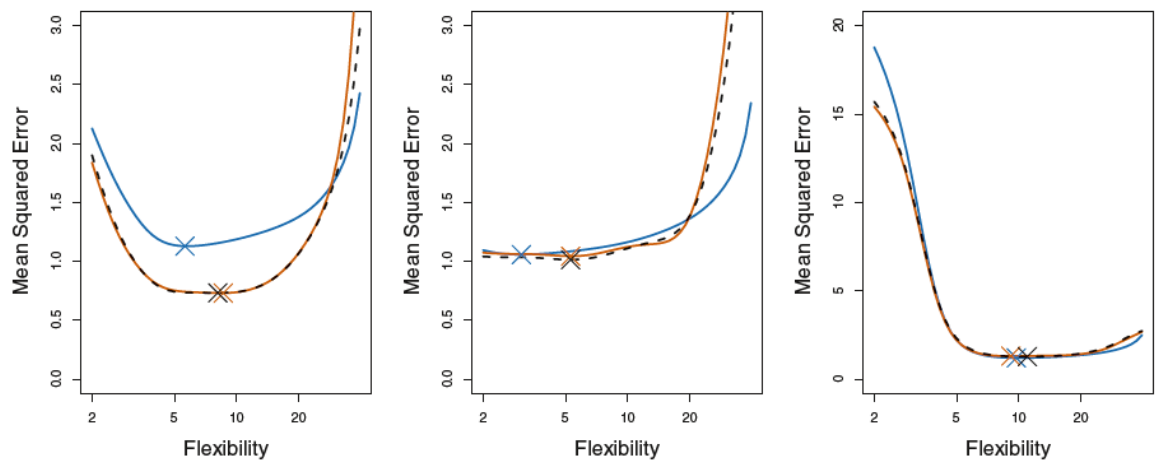


FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9 (*left*), 2.10 (*center*), and 2.11 (*right*). The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the* 10*-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.*

Bias – Variance trade off

Przy LOOCV bias jest nieduży, ponieważ każdy model tworzony jest prawie na wszystkich dostępnych danych. Natomiast wariancja jest duża, bo każdy model jest tworzony praktycznie na tych samych danych, co oznacza, że są one ze sobą skorelowane.

K-fold CV rozbija zbiór danych na kilka różnych podzbiorów, które na siebie nie nachodzą, więc nie są aż tak skorelowane.

W rzeczywistości używa się K-fold CV z k = 5 lub k = 10.

CV dla klasyfikacji

CV dla klasyfikacji wylgąda tak samo jak dla regresji. Wystarczy wybrać dowolną statystykę – może to być accuracy albo error rate (odwrotność accuracy).

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{Err}_i,$$

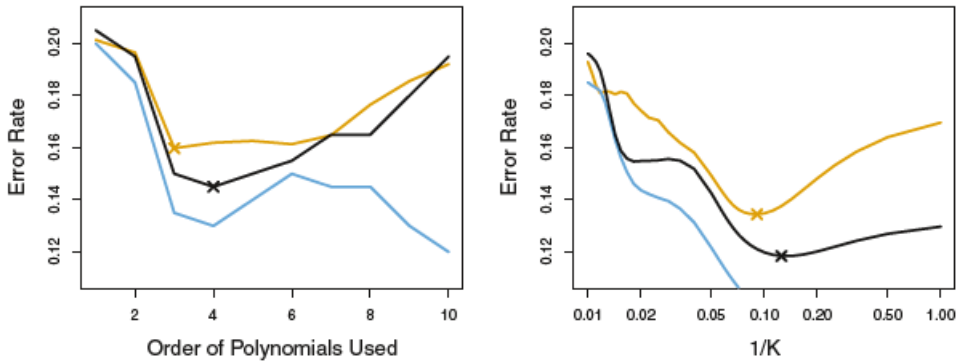


FIGURE 5.8. *Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.*

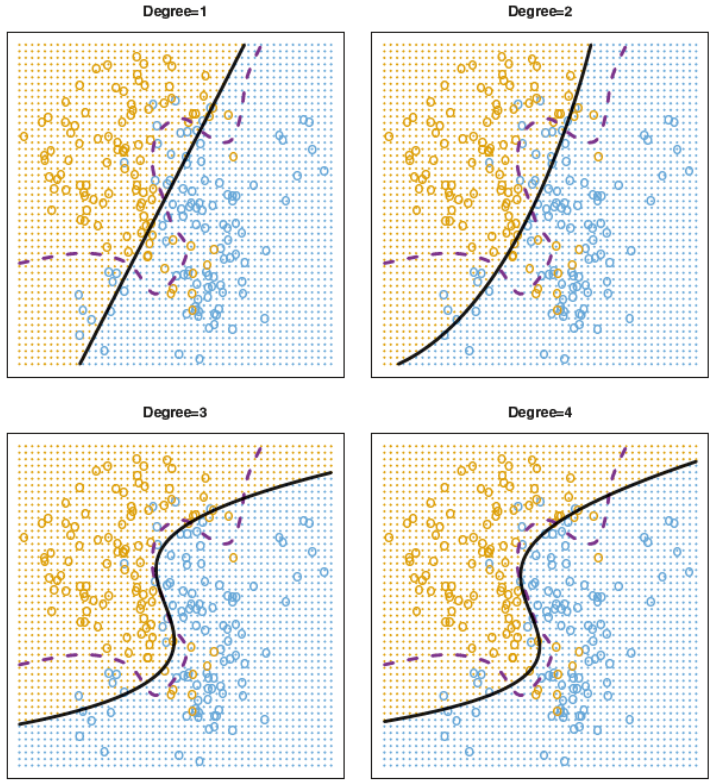


FIGURE 5.7. *Logistic regression fits on the two-dimensional classification data displayed in Figure 2.13. The Bayes decision boundary is represented using a purple dashed line. Estimated decision boundaries from linear, quadratic, cubic and quartic (degrees 1–4) logistic regressions are displayed in black. The test error rates for the four logistic regression fits are respectively* 0.201, 0.197, 0.160, *and* 0.162, *while the Bayes error rate is* 0.133.