An introduction to Statistical Learning - chapter 2

Lila Gmerek / Week 2 / 14.05.2019

Links

- Book http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf
- Github: https://github.com/erg0-0/ML-study-group-pl-fc
- Link do pliku z podzialem rozdzialow: https://docs.google.com/document/d/10DWwfxazT9RiKX0ZV7xAcYVtZspaalqn8S6lCgzmMdE/edit

the Trade-Off Between Prediction Accuracy and Model Interpretability (Lila)

The Trade-Off Between Prediction Accuracy and Model Interpretability

kompromis pomiędzy dokładnością predykcji a interpretowalnością modelu

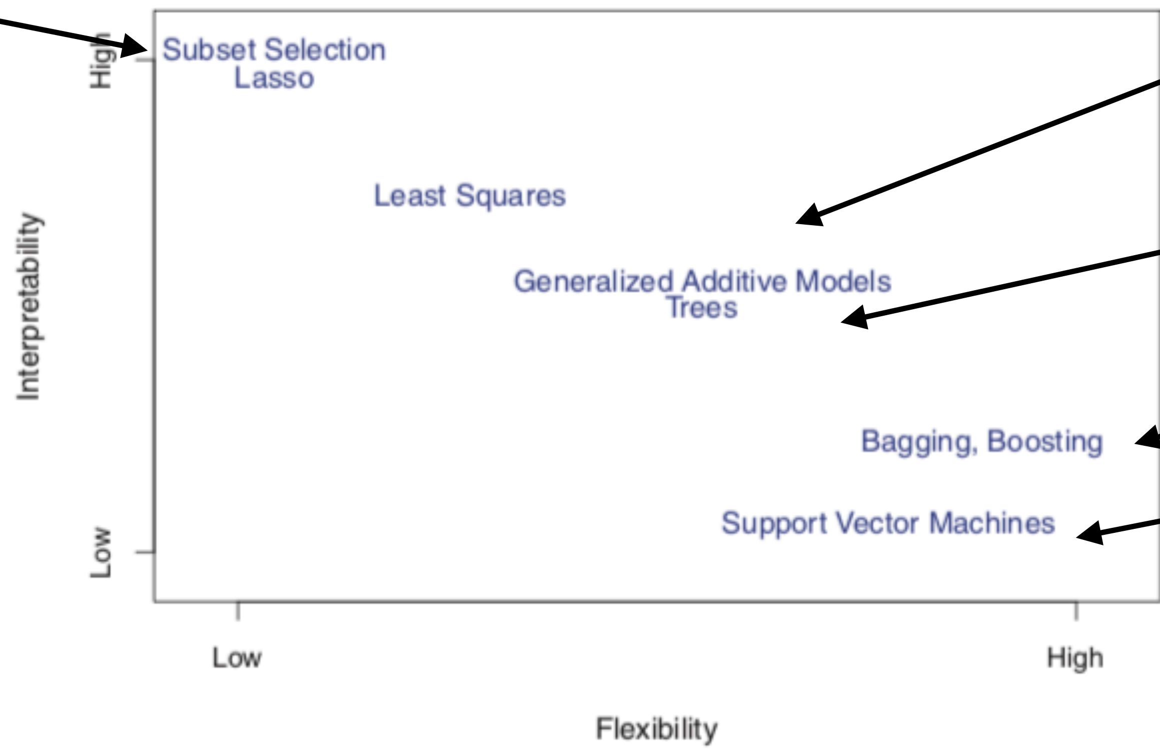
Nie można mieć wszystkiego.
Albo model jest łatwy w interpretacji, ale bardzo restrykcyjny;
albo jest trudny w interpretacji, ale dużo bardziej elastyczny.

Generalized Additive models (GAMs) (Chapter 7) extend the linear model to allow for certain non-linear relationships. Consequently, GAMs are more flexible than linear regression. They are also somewhat less interpretable than linear regression, because the relationship between each predictor and the response is now modeled using a curve.

2.1 What Is Statistical Learning? 25

Regresje

The lasso (Chapter 6) relies upon the linear model, but uses an alternative fitting procedure for estimating the coefficients $\beta_0, \beta_1, \ldots, \beta_p$. The new procedure is more restrictive in estimating the coefficients, and sets a number of them to exactly zero. Hence in this sense the lasso is a less flexible approach than linear regression. It is also more interpretable than linear regression, because in the final model the response variable will only be related to a small subset of the predictors—namely, those with nonzero coefficient estimates.



Decision Trees
Random Forest

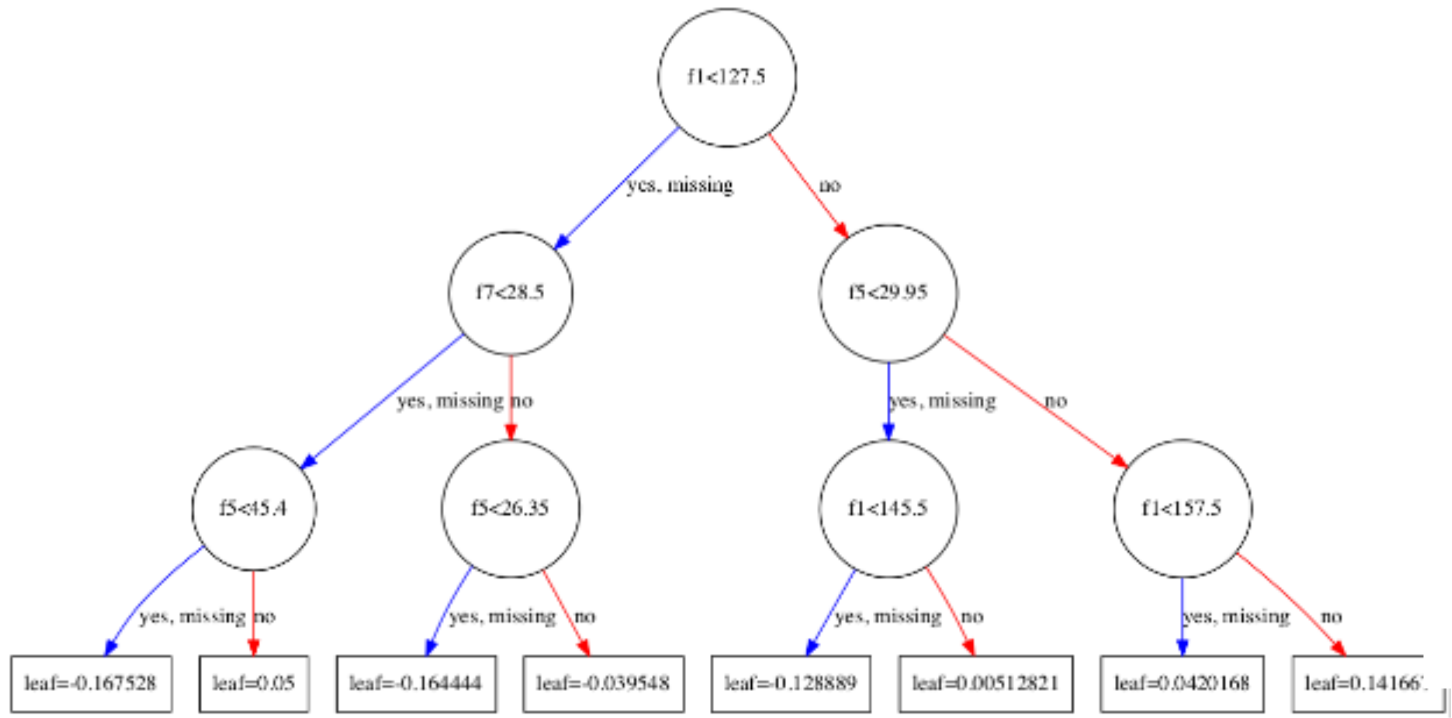
XGboost

tzw. SVM

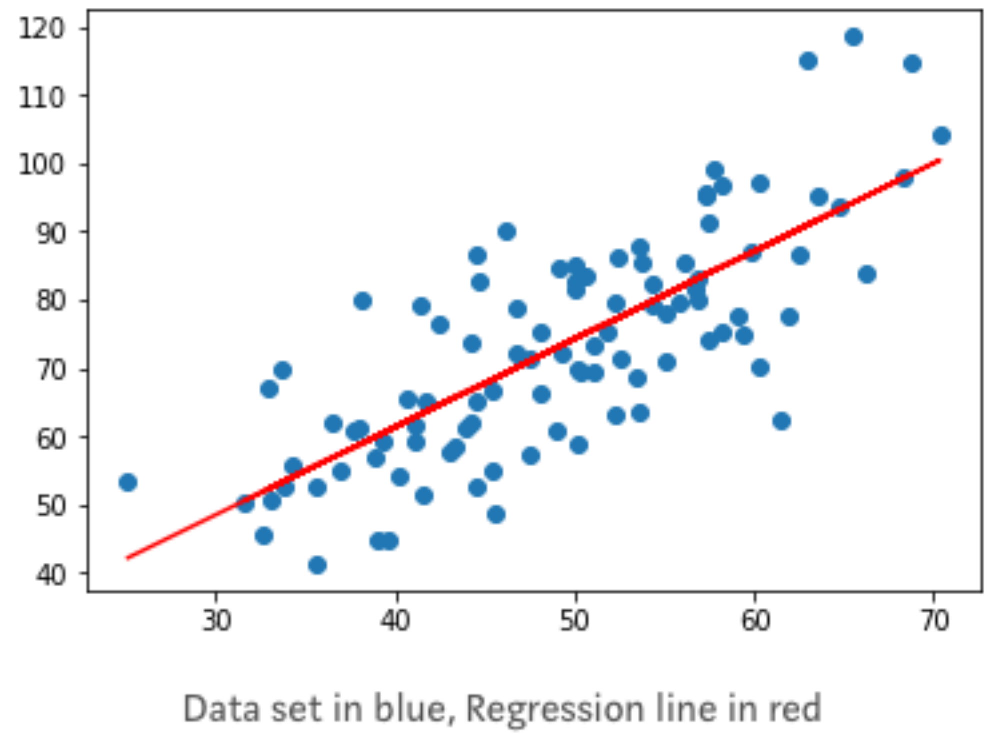
FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

XGboost (Bagging, Boosting), SVM - kernels are non-linear therefore models are highly flexible but very difficult to interpret.

Przyklady



Wizualizacha modelu XGboost (Extrem Gradient Boosting, pojedyncze drzewo)
https://machinelearningmastery.com/visualize-gradient-boosting-decision-trees-xgboost-python/



Wizualizacja
Regresja Liniowa
https://towardsdatascience.com/linear-regression-in-6-lines-of-python-5e1d0cd05b8d

https://playground.tensorflow.org/

The Trade-Off Between Prediction Accuracy and Model Interpretability

Po co mi w ogóle model nieelastyczny (restryktywny)?
Dobrze sprawdzają się w przypadku problemów opartych na wnioskowaniu lub badaniu zależności różnych zmiennych X na Y. Łatwa w interpretacji regresja pomaga szybko to ocenić. Po co zatem komplikować sobie życie i tracić czas na skomplikowane modele, zwłaszcza, że ich rezultaty mogą być dużo bardziej niejednoznaczne.

Przy okazji, łatwiej wytłumaczyć taki łatwiejszy model szefowi lub klientowi, którzy za te nasze statystyki płacą - co nierzadko przemawia za ich zastosowaniem.

Czy zawsze bardziej elastyczne modele wpływają na lepszą predykcję?

Istnieje pewien paradoks - czasami lepiej zastosować bardziej restrykcyjny model aby uzyskać lepszą predykcję. Okazuje się, że bardziej skomplikowane modele (boosting czy nawet te popularne sieci neuronowe i cały deep learning) dużo częściej cierpią na problem z tzw. overfittingiem (czyli przetrenowaniem modelu). W rezultacie, fantastycznie radzą sobie na danych treningowych, ale dużo gorzej na danych na których są testowane.

For instance, when inference is the goal, the linear model may be a good choice since it will be quite easy to understand the relationship between Y and $X_1, X_2, \ldots, X_p$. In contrast, very flexible approaches, such as the splines discussed in Chapter 7 and displayed in Figures 2.5 and 2.6, and the boosting methods discussed in Chapter 8, can lead to such complicated estimates of f that it is difficult to understand how any individual predictor is associated with the response.

Assessing model accuracy - measuring quality fit

Why is it necessary to introduce so many different statistical learning approaches, rather than just a single best method?

There is no free lunch in statistics: no one method dominates all others over all possible data sets.

Hence it is an important task to decide for any given set of data which method produces the best results. Selecting the best approach can be one of the most challenging parts of performing statistical learning in practice.

Ockham chooses a razor

Measuring quality of fit

$$\begin{aligned} E(Y-\hat{Y})^2 &= E[f(X)+\epsilon-\hat{f}(X)]^2 \\ &= \underbrace{[f(X)-\hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible}}, \end{aligned}$$

Jak ocenić czy model radzi sobie dobrze czy źle?

$$\hat{Y} = \hat{f}(X),$$

Należy sprawdzić jak wartość predykowana/modelowana(y z daszkiem) różni się od tej prawdziwej (y bez daszka).

Przyklad różnych metryk używanych w regresji [Python]:

y_test (y ze zbioru testowego - wartość modelowana z datasetu testowego/model nie byl na nim trenowany)

predictions_dtr (=y predykcja wartosc przewidziana przez model dla testowanego datasetu)

```
dtr_mse=round(mean_squared_error(y_test, predictions_dtr))
dtr_medae=round(median_absolute_error(y_test, predictions_dtr))
dtr_mae = round(mean_absolute_error(y_test,predictions_dtr ))
dtr_rmse= round(sqrt(mean_squared_error(y_test,predictions_dtr)))

print('Mean Squared Error: {}'.format(dtr_mse))
print("Median_absolute_error: {}.".format(dtr_medae))
print("Mean Absolute Error:  {}".format(dtr_mae))
print("Root Mean Squared Error: {}". format(dtr_rmse))
```

```
Mean Squared Error: 3280320019.0
Median_absolute_error: 39590.0.
Mean Absolute Error:  47583.0
Root Mean Squared Error: 57274
```

Measuring quality of fit

$$\hat{Y} = \hat{f}(X),$$

feature x1	feature x2	feature x3	cos co chcemy przewidywac (y), (y^)
X_train	X_train	X_train	y_train
X_train	X_train	X_train	y_train
X_test	**X_test**	**X_test**	**y_PREDYKCJA**

n

y_test

Mean Squared Error
błąd średniokwadratowy

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

Jedną z najpopularniejszych metryk oceniających model jest:

MSE (Mean Squared Error) / błąd średniokwadratowy / średni błąd kwadratowy

błąd estymatora f^(x i) nieobserwowanego parametru y.

MSE jest wartością oczekiwaną kwadratu „błędu", czyli różnicy pomiędzy estymatorem oraz wartością estymowaną.

funkcja f^ x i - to predykcja jaką ta funkcja wyznaczyła dla *i obserwacji.*

Obciążenie estymatora jest różnicą między wartością oczekiwaną estymatora a wartością szacowanego parametru.
Błąd będzie mały jeśli - przewidywana wartość będzie blisko prawdziwej wartości.
Błąd będzie wysoki jeśli dla niektórych obserwacji, przewidywana wartość oraz prawdziwa wartość będą od siebie odbiegać.

MSE jest wyliczany w oparciu o dane treningowe, dlatego powinno się go raczej określać jako MSE treningowe. Zasadniczo jednak nie interesuje nas jak dobrze ten błąd wypada na danych treningowych. Interesuje nas **jak wysoki ten błąd będzie dla danych, których model jeszcze nie widział.**

[Przyklad - ksiazka]. Projektujesz algorytm do przewidywania cen akcji na gieldzie, bazujac na danych historycznych. Trenujesz model na danych za ostatnie 6 miesiace. Nie interesuje Cie jak dobrze model przewidzial zmiany na gieldzie w zeszlym tygodniu, ale jak dobrze bedzie przewidywac je w przyszlosci.

[przyklad ksiazka] masz do dyspozycji badania krwi pacjentow chorych na cukrzyce. Wykorzystujesz ich dane do wytrenowania modelu. Zasadniczo jednak interesuje Cie przewidywanie cukrzycy u przyszlych pacjentow, a nie u tych o ktorych juz wiesz ze maja cukrzyce.

Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

Matematycznie:

wyobrazmy sobie ze dostosowujemy model uzywajac do tego obserwacji treningowych $\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$,

dzięki temu uzyskujemy funkcję f (estymator).

W ten sposob mozna wyliczyc $f(x_1),f(x_2),...,f(x_n)$.

Jezeli sa one mniej więcej równe y1, y2 … yn, wówczas MSE treningowe będzie niewielkie.

Jednak nie interesuje nas czy $f(x_i) \approx y_i$; Zamiast tego chcemy wiedziec czy $f(x_0)$ jest wzglednie rowna y_0,

przy zalozeniu ze **obserwacje(x_0, y_0) naleza do nie widzianego wczesniejej testowego zbioru obserwacji i** nie zostaly uzyte do trenowania modelu statystycznego.

Chcemy wybrac taki model ktory daje w rezultacie najnizsze MSE, w opozycji do najniższego treningowego MSE. Innymi slowy - jesli mamy duza liczbe obserwacji testowych, mozemy obliczyc:

$$\text{Ave}(y_0 - \hat{f}(x_0))^2,$$

the average squared prediction error for these test observations (x_0,y_0).

We’d like to select the model for which the average of this quantity—the test MSE—is as small as possible.

Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2, \qquad \mathrm{Ave}(y_0 - \hat{f}(x_0))^2,$$

W jaki sposob mozemy wybrac model ktorego MSE jest najnizszy?

Najlepiej w tym celu wykorzystac zestaw danych testowych, ktorych model nie widzial.

Jeśli nie ma takiego zestawu danych, nalezy go sztucznie wydzielic z istniejacego data setu. Jesli tego nie zrobisz, model bedzie przeuczony (tzw. overfitting). Innymi slowy, bedzie doskonale przewidywac wartosci wytrenowane (niski MSE), ale poradzi sobie duzo gorzej na niewidzianych wczesniej danych.

niebieski/zielony smoothing splines (rozdzial 7)

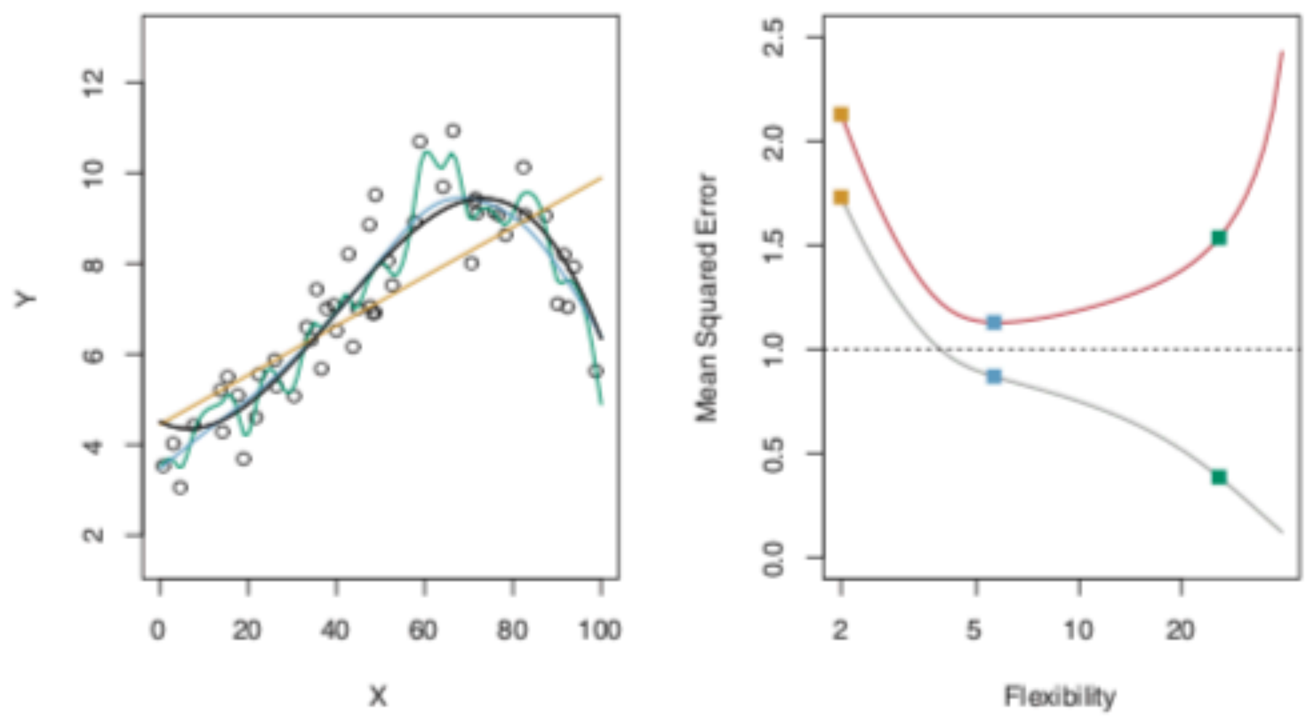


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

szara linia = sredni blad kwadratowy (training MSE) jako funkcja elastycznosci, funkcja liczby stopni swobody (degrees of freedom). Stopnie swobody to liczba ktora okresla elastycznosc krzywej (wiecej w rozdziale 7)

Zolte/niebieskie/zielone kwadraty = treningowe oraz testowe MSE funkcji z wykresu po lewej stronie w odpowiadajacym kolorze

Krzywa treningowego MSE opada monotonicznie w miare wzrostu elastycznosci modelu (zolta regresja liniowa bardzo restrykcyjna a przez to o bardziej gladkiej linii, model z zielonej lini bardzo pokrzywionej linii o duzej wariancji —> bardzo elastyczny model).

W tym przykladzie prawdziwa funkcja f nie jest linearna, dlatego zolta krzywa regresji liniowej jest zbyt restrykcyjna i zle dopasowana do danych(lewy wykres). Zielona funkcja jest najbardziej dopasowana do danych, ale wypada najgorzej w testowym MSE/zielony kwadrat na czerwonej linii (jest przeuczona / overfitting)

Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

$$\text{Ave}(y_0 - \hat{f}(x_0))^2,$$

[wikipedia]
Liczba stopni swobody, **df** (ang. *degrees of freedom*) – liczba niezależnych wyników obserwacji pomniejszona o liczbę związków, które łączą te wyniki ze sobą.

Liczbę stopni swobody można utożsamiać z liczbą niezależnych zmiennych losowych, które wpływają na wynik.
Inną interpretacją liczby stopni swobody może być: liczba obserwacji minus liczba parametrów estymowanych przy pomocy tych obserwacji.
LINK https://pl.wikipedia.org/wiki/Liczba_stopni_swobody_(statystyka)

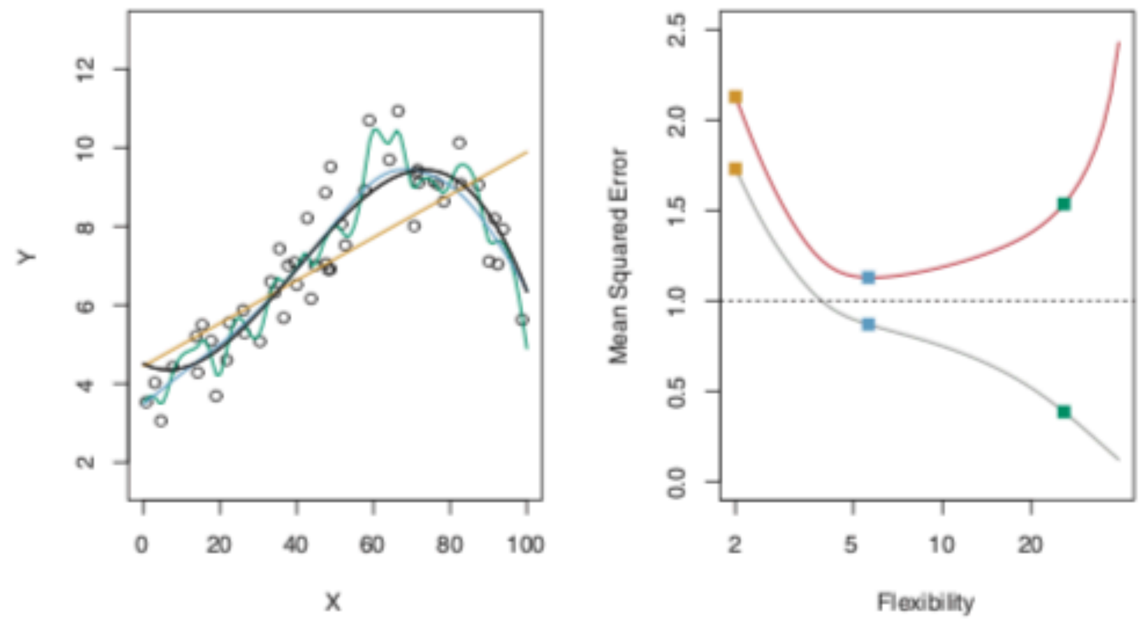


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

(lewy wykres): funkcja zielona oraz zolta(regresja l.) maja najwyzsze MSE testowe (czerwona linia, kwadraty zolty+zielony)
FUnkcja granatowa ma najnizsze MSE testowe, minimalizuje testowy MSE ==> nie powinno to zaskakiwac, na lewym wykresie widac, ze najlepiej dopasowuje sie do danych.

prawy wykres - horyzontalna linia przerywana wskazuje na nieredukowalny blad Var(ε)
oraz koresponuje z najnizszym osiagnietym testowym MSE sposrod wszystkich metod

Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

$$\text{Ave}(y_0 - \hat{f}(x_0))^2,$$

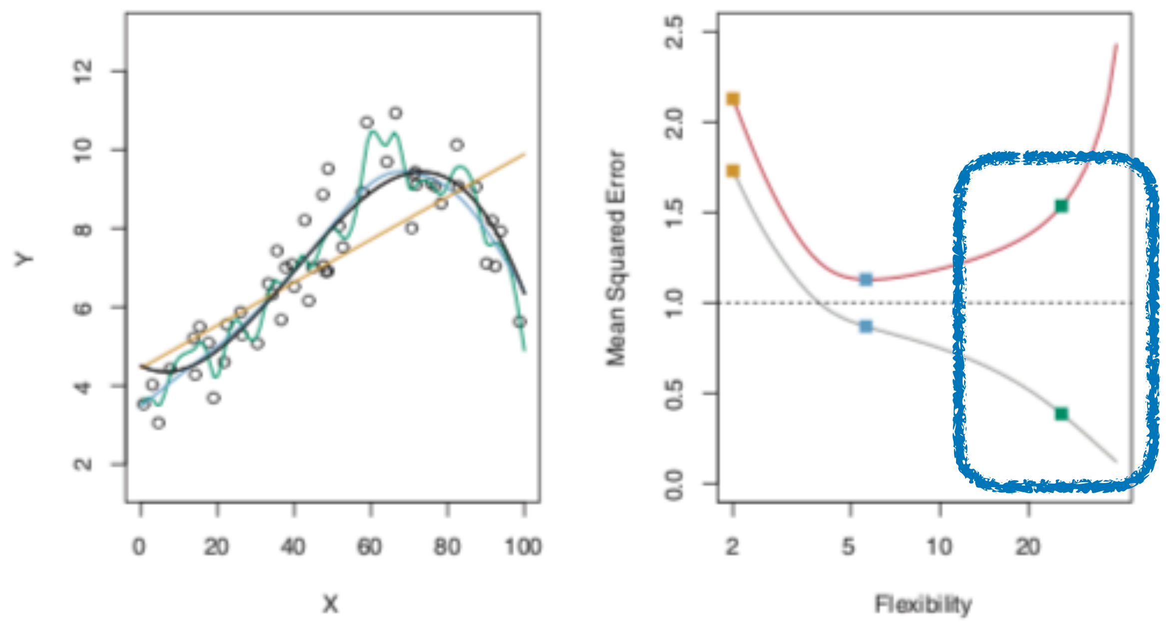


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

Overfitting
Przeuczony model

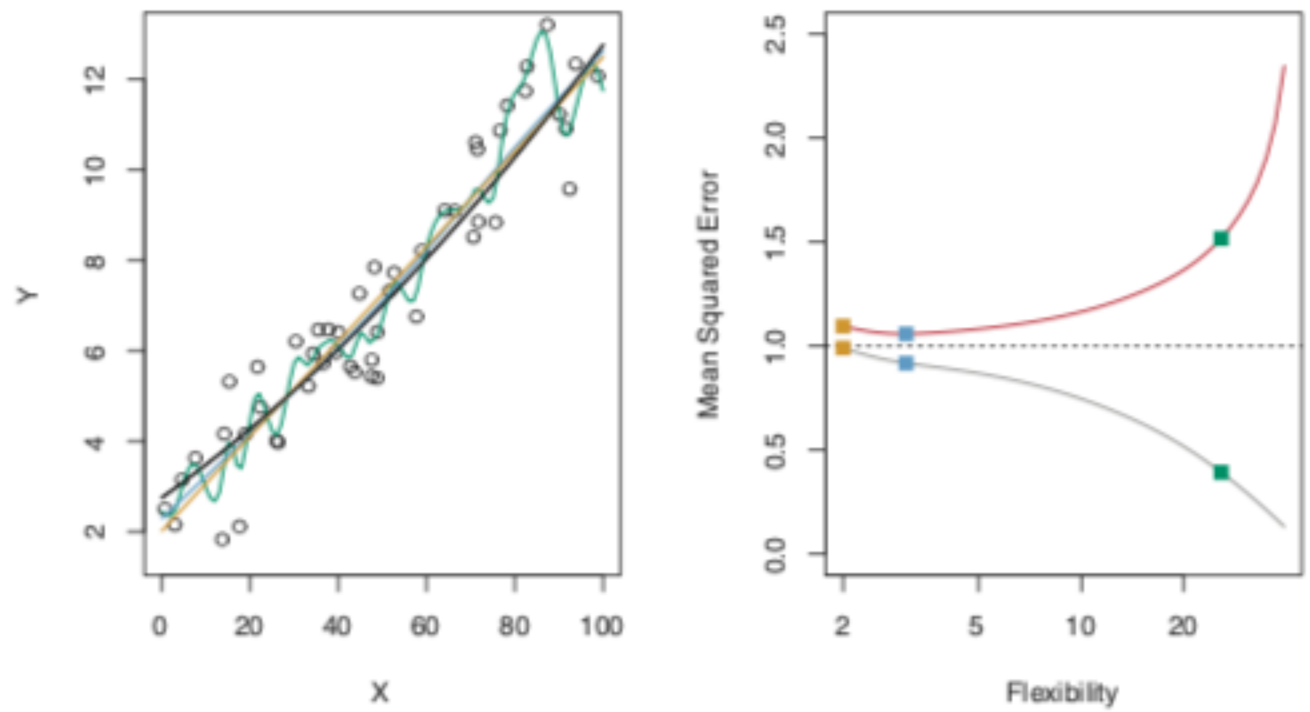
A monotone decrease in the training MSE and a U-shape in the test MSE. This is a fundamental property of statistical learning that holds regardless of the particular data set at hand and regardless of the statistical method being used. As model flexibility increases, training MSE will decrease, but the test MSE may not.

This happens because our statistical learning procedure is working too hard to find patterns in the training data, and may be picking up some patterns that are just caused by random chance rather than by true properties of the unknown function f.

Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

$$\mathrm{Ave}(y_0 - \hat{f}(x_0))^2,$$



Inny dataset

Na tym przykladzie widac, ze regresja liniowa (zolta linia) lepiej o dpowiada danym, a kwadrat roznic mse treningowego i testowego jest mniejszy (2 zolte kwadraty na prawym wykresie)

FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

Reference: cross-validation chapter 5

The Bias-Variance Trade-Off - Mateusz Podlasin