

# Evaluasi Model

Ali Akbar Septiandri

Universitas Al-Azhar Indonesia

*aliakbars@live.com*

October 22, 2017

# Selayang Pandang

- 1 Generalisasi
- 2 Optimasi Model dari Dataset
- 3 Metrik Evaluasi

## Bahan Bacaan

- 1 VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media. <https://jakevdp.github.io/PythonDataScienceHandbook/05.03-hyperparameters-and-model-validation.html>
- 2 Tan, P. N. (2006). Introduction to data mining. Pearson Education India. (Chapter 4. Classification)

# Generalisasi

# Generalisasi Error

- Tujuan kita adalah menghasilkan model yang dapat bekerja baik pada **semua data**
- **Tidak mungkin** mendapatkan semua data
- Solusi: Gunakan **data latih** dan **data uji**

# Generalisasi Error

- *Training data*:  $\{x_i, y_i\}$
- *Future data*:  $\{x_i, ?\}$
- Target: Model bekerja baik pada **future data**

Mengapa?

# Overfitting

- Model terlalu kompleks, **terlalu fleksibel**
- Mengenali dan memasukkan *noise* dari dalam data latih ke dalam model
- Mengenali pola yang *tidak akan muncul lagi*



# Overfitting: Definisi

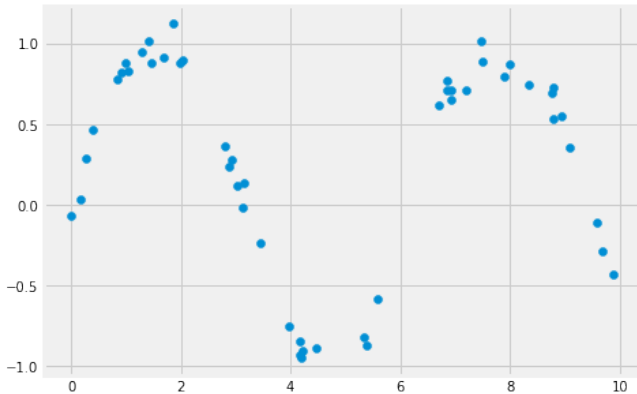
Model  $F$  dikatakan *overfitting* jika:

- 1 kita dapat menemukan model lain  $F'$
- 2 dengan error lebih besar pada data latih:  $E_{train}(F') > E_{train}(F)$
- 3 tetapi error lebih kecil pada data uji:  $E_{gen}(F') < E_{gen}(F)$

# Underfitting

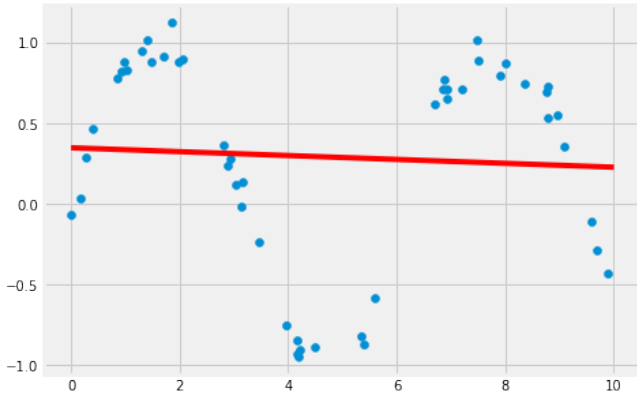
- Model terlalu kaku, **terlalu simpel**
- Tidak berhasil menemukan pola yang penting
- Masih ada model yang bisa menghasilkan  $E_{train}$  dan  $E_{gen}$  lebih rendah

## Contoh pada Regresi



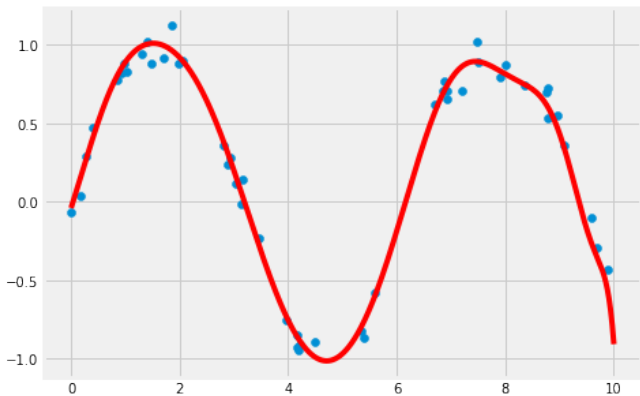
**Gambar:** Bagaimana kira-kira hasil regresi pada data seperti ini?

## Contoh pada Regresi



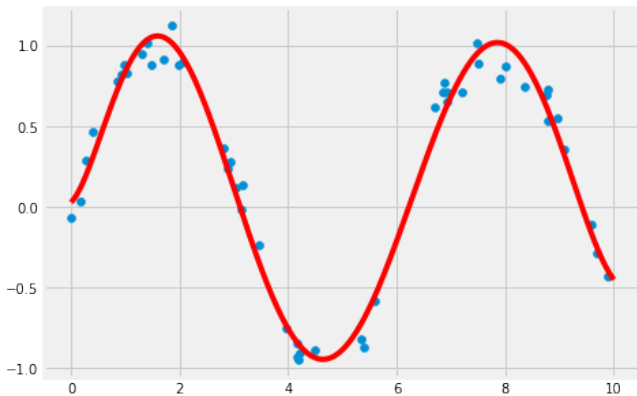
Gambar: Regresi polinomial dengan  $p = 1$  (linear)

## Contoh pada Regresi



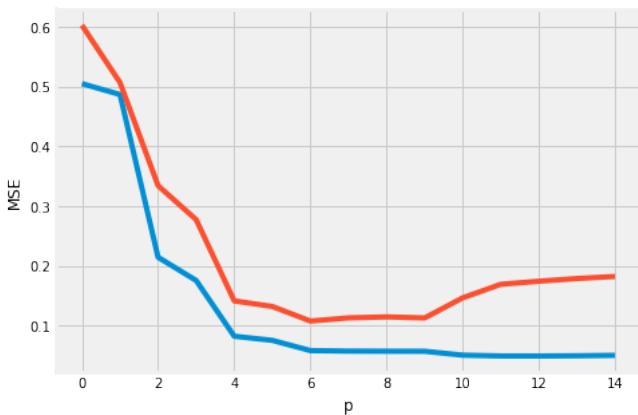
Gambar: Regresi polinomial dengan  $p = 15$

## Contoh pada Regresi



Gambar: Regresi polinomial dengan  $p = 7$

## Contoh pada Regresi



**Gambar:** Kapan *regressor* yang kita latih dikatakan mengalami *overfitting* atau *underfitting*?

# Fleksibilitas Prediktor

- Setiap dataset perlu prediktor dengan **fleksibilitas yang berbeda**, tergantung kesulitannya dan data yang tersedia



# Fleksibilitas Prediktor

- Setiap dataset perlu prediktor dengan **fleksibilitas yang berbeda**, tergantung kesulitannya dan data yang tersedia
- Diperlukan **kenop** untuk mengubah fleksibilitasnya, e.g.
  - regresi: orde polinomial
  - NB: jumlah atribut,  $\epsilon$
  - decision tree: jumlah simpul dalam pohon

# Fleksibilitas Prediktor

- Setiap dataset perlu prediktor dengan **fleksibilitas yang berbeda**, tergantung kesulitannya dan data yang tersedia
- Diperlukan **kenop** untuk mengubah fleksibilitasnya, e.g.
  - regresi: orde polinomial
  - NB: jumlah atribut,  $\epsilon$
  - decision tree: jumlah simpul dalam pohon
- Idenya, memutar kenop tersebut untuk **menghasilkan error yang rendah secara umum**

# Error Latihan vs General

- Error latihan:

$$E_{train} = \frac{1}{n} \sum_{i=1}^n error(f_D(\mathbf{x}_i), y_i)$$

- Error general:

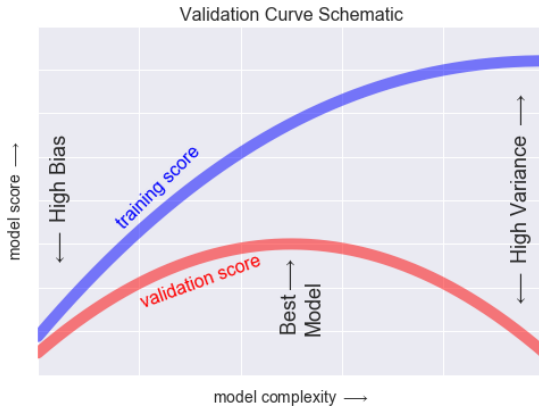
$$E_{gen} = \int error(f_D(\mathbf{x}), y) p(y, \mathbf{x}) d\mathbf{x}$$

- Kita hanya tahu **jangkauan** dari  $\{x, y\}$

Estimasi nilainya dengan

$$E_{test} = \frac{1}{n} \sum_{i=1}^n error(f_D(\mathbf{x}_i), y_i)$$

# Validasi Model



Gambar: Perubahan nilai *metric* sesuai dengan kompleksitas model

# Optimasi Model dari Dataset

# Training, Validation, Testing sets

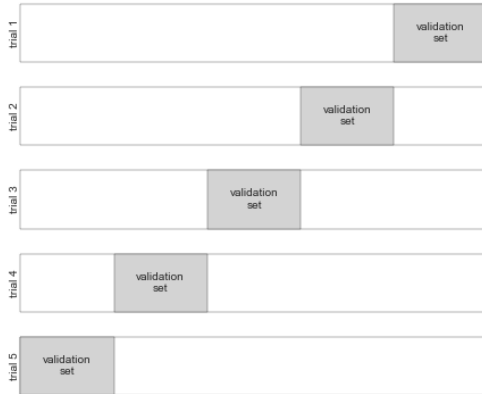
- **Data latih:** konstruksi *classifier*
- **Data validasi:** memilih algoritma dan *parameter tuning*
- **Data uji:** mengestimasi *error rate* secara umum
- Catatan: Bagi datanya secara **acak!**

# Cross-validation

- Datanya kadang tidak cukup banyak untuk dibagi!
- Ide: latih dan uji secara bergantian
- Umumnya: 10-fold cross-validation



# Cross-validation



Gambar: 5-fold cross-validation

# Leave-one-out

*n-fold cross-validation*

## Pros

Menghasilkan *classifier* terbaik

## Cons

- Ongkos komputasi tinggi
- Kelas tidak seimbang → *stratification*

# Metrik Evaluasi

# Unbalanced Dataset

- e.g. Prediksi apakah akan terjadi gempa atau tidak!

# Unbalanced Dataset

- e.g. Prediksi apakah akan terjadi gempa atau tidak!
- Jika selalu diklasifikan sebagai “tidak”, akurasi akan maksimal, error akan minimal.

# Unbalanced Dataset

- e.g. Prediksi apakah akan terjadi gempa atau tidak!
- Jika selalu diklasifikasikan sebagai “tidak”, akurasi akan maksimal, error akan minimal.
- Solusi: Gunakan metrik lain

## Misses & False Alarms

- False Alarm rate = False Positive rate =  $FP/(FP + TN)$
- Miss rate = False Negative rate =  $FN/(TP + FN)$
- Recall = True Positive rate = Sensitivity =  $TP/(TP + FN)$
- Precision =  $TP/(TP + FP)$
- Specificity =  $1 - FPR = TN/(TN + FP)$
- Harus dilaporkan berpasangan!

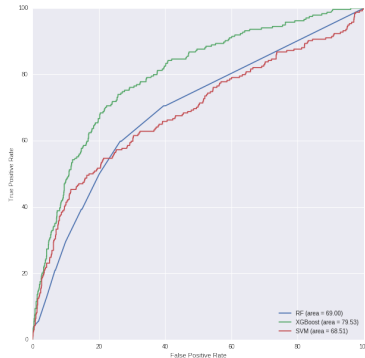
## Utility & Cost

- Terkadang perlu satu angka untuk pembandingan antarmodel
- **Detection cost:**  $cost = c_{FP} \times FP + c_{FN} \times FN$
- **F-measure:**  $F_1 = 2 \times \frac{precision \times recall}{precision + recall}$



# ROC Curves

Receiver Operating Characteristic: TPR vs FPR dengan perubahan *threshold*



Menghitung Area Under the Curve (AUC) sebagai pengganti akurasi

Terima kasih