

k-Nearest Neighbours

Ali Akbar Septiandri

October 17, 2017

Universitas Al Azhar Indonesia

1. k-Nearest Neighbours
2. Evaluasi dan Generalisasi

k-Nearest Neighbours

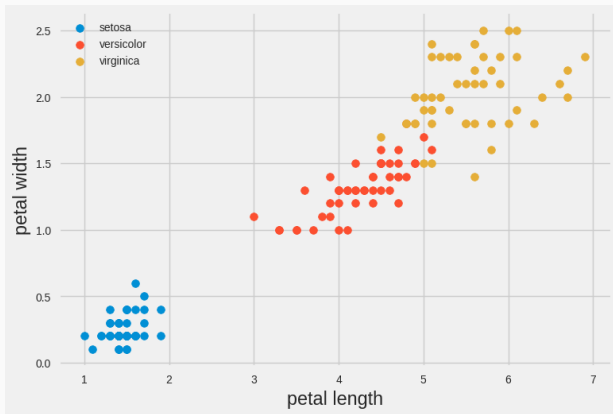
Deskripsi Dataset

- Iris dataset
- Pembuat: R.A. Fisher (1936)
- <http://archive.ics.uci.edu/ml/>
- 4 atribut: sepal length, sepal width, petal length, petal width
- 3 label: Iris Setosa, Iris Versicolour, Iris Virginica

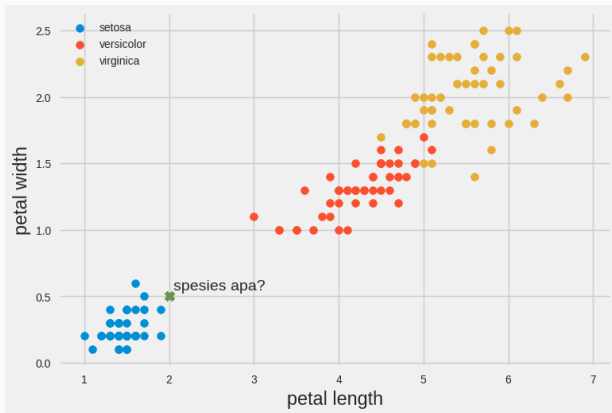


Figure 1: Tanaman Iris

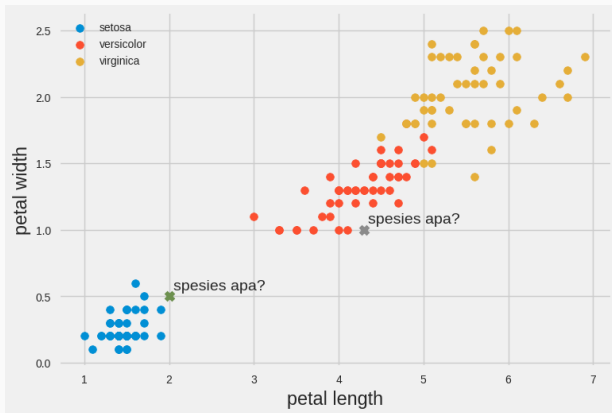
Iris Dataset



Data Baru



Data Baru



Nearest Neighbour

- Mencari referensi dari tetangga terdekat

Nearest Neighbour

- Mencari referensi dari tetangga terdekat
- Apa definisi “terdekat”?

Nearest Neighbour

- Mencari referensi dari tetangga terdekat
- Apa definisi “terdekat”?
- Metode umum: **Euclidean distance**

Euclidean Distance

$$d([x_1, x_2, \dots, x_d], [y_1, y_2, \dots, y_d]) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

Masalah

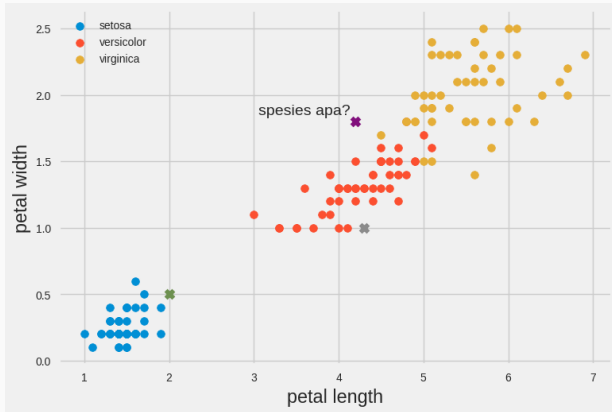


Figure 2: Seberapa yakin kita dengan referensi terdekat?

k-Nearest Neighbours

- Mencari referensi dari beberapa (k) tetangga terdekat

k-Nearest Neighbours

- Mencari referensi dari beberapa (k) tetangga terdekat
- Melihat label mayoritas dari tetangga terdekat

k-Nearest Neighbours

- Mencari referensi dari beberapa (k) tetangga terdekat
- Melihat label mayoritas dari tetangga terdekat
- Perhatikan bahwa harus dihitung jaraknya dengan semua data yang ada

k-Nearest Neighbours

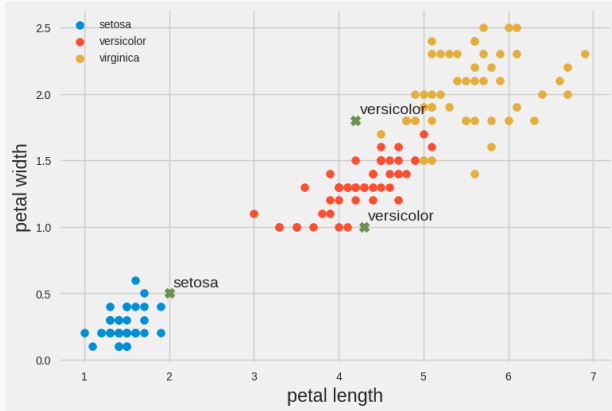
- Mencari referensi dari beberapa (k) tetangga terdekat
- Melihat label mayoritas dari tetangga terdekat
- Perhatikan bahwa harus dihitung jaraknya dengan semua data yang ada
- Kompleksitas: $O(nd)$

k-Nearest Neighbours

- Mencari referensi dari beberapa (k) tetangga terdekat
- Melihat label mayoritas dari tetangga terdekat
- Perhatikan bahwa harus dihitung jaraknya dengan semua data yang ada
- Kompleksitas: $O(nd)$
- *Tidak mungkin kita hitung sendiri!*

- Diketahui
 - data latih $\{x_i, y_i\}$
 - x_i : nilai atribut
 - y_i : label kelas
 - *instance* uji x
- Algoritma:
 1. Hitung jarak $D(x, x_i)$ untuk semua x_i
 2. Pilih k tetangga terdekat dengan labelnya
 3. \hat{y} = mayoritas dari label tetangga terdekat

Prediksi



Klasifikasi k-NN

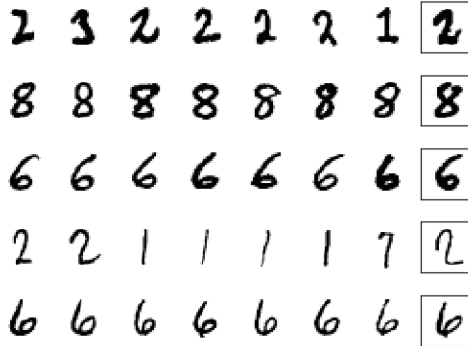


Figure 3: 7-NN pada data MNIST dengan data uji di paling kanan

- Diketahui
 - data latih $\{x_i, y_i\}$
 - x_i : nilai atribut
 - y_i : nilai numerik sebenarnya
 - *instance* uji x
- Algoritma:
 1. Hitung jarak $D(x, x_i)$ untuk semua x_i
 2. Pilih k tetangga terdekat dengan labelnya
 3. $\hat{y} = f(x) = \frac{1}{k} \sum_{j=1}^k y_{ij}$ (nilai rata-rata)

Regresi k-NN

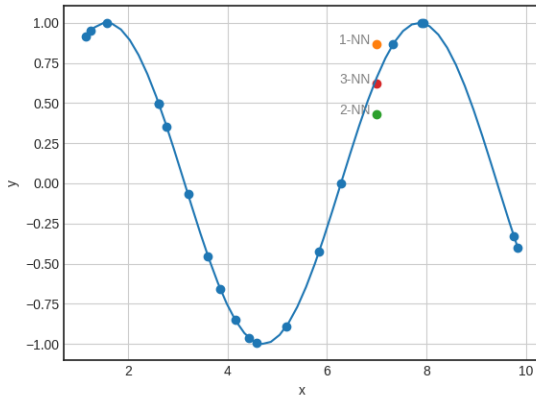


Figure 4: Interpolasi dengan $\{1,2,3\}$ -NN

Regresi k-NN

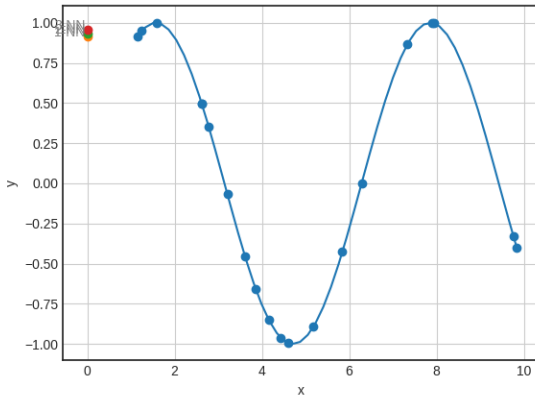


Figure 5: Ekstrapolasi dengan $\{1,2,3\}$ -NN

Bagaimana cara memilih nilai k ?

- Nilai yang besar $\rightarrow P(y)$ atau \bar{y}
- Nilai yang kecil \rightarrow terlalu variatif, batas keputusan yang tidak stabil

- Nilai yang besar $\rightarrow P(y)$ atau \bar{y}
- Nilai yang kecil \rightarrow terlalu variatif, batas keputusan yang tidak stabil
- **Solusi:** Gunakan data validasi!

Batas Keputusan

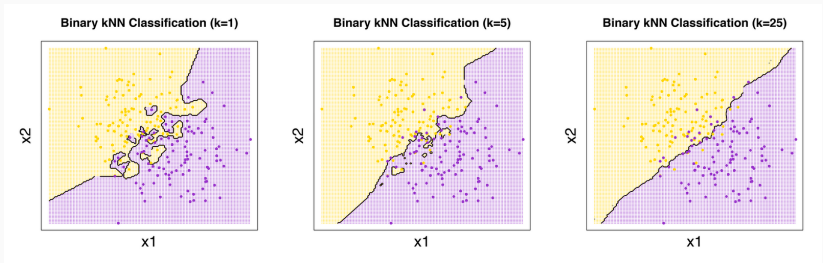


Figure 6: Pengaruh nilai k pada batas keputusan [DeWilde, 2012]

- Hasil seri:
 1. Gunakan jumlah k ganjil
 2. Acak, lemparan koin
 3. *Prior probability*
 4. 1-NN
- *Missing values*: harus diganti (*impute*)
- Rentan terhadap perbedaan rentang variabel

Perbedaan Rentang

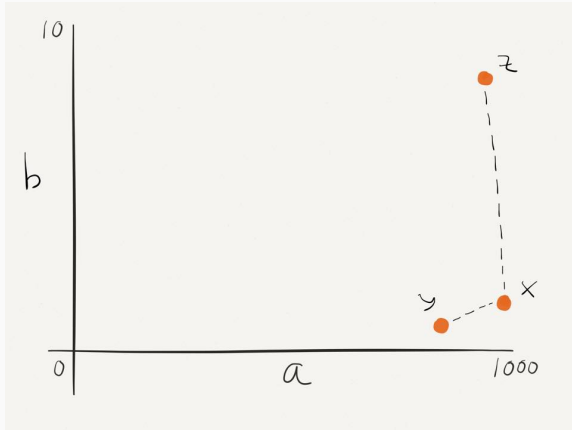


Figure 7: Perbedaan rentang variabel bisa mengacaukan klasifikasi k-NN
[Wibisono, 2015]

- Pros:
 - Tidak ada asumsi terhadap data, non-parametrik
 - *Asymptotically correct*
- Cons:
 - Harus mengganti nilai yang hilang
 - Sensitif terhadap kelas pencilan (data latih yang salah dilabeli)
 - Sensitif terhadap atribut yang irelevan
 - **Mahal secara komputasi** $O(nd)$

Evaluasi dan Generalisasi

- 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

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!**

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

- k-Nearest Neighbours merupakan algoritma yang dapat dipakai untuk **klasifikasi dan regresi**
- Nilai **k** dalam algoritma k-NN **perlu divalidasi**
- k-NN bersifat **non-parametrik**
- Kita dapat menghindari **overfitting** atau **underfitting** dengan menggunakan data latih, validasi, dan uji secara tepat
- Saat jumlah data tidak cukup banyak, **cross-validation** dapat menjadi alternatif

Pertemuan berikutnya

- Unsupervised learning
- k-Means clustering



Burton DeWilde (26 Oktober 2012)

Classification of Hand-written Digits (3)

<http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/>



Okiriza Wibisono (16 September 2015)

kNN: Perhitungan Jarak, serta Batasan dan Keunggulan

<https://tentangdata.wordpress.com/2015/09/16/knn-perhitungan-jarak-serta-keunggulan-dan-batasan/>

Terima kasih