K-Nearest Neighbor

Pendahuluan

- K-Nearest Neighbour atau KNN adalah salah dari algoritma instance based learning atau case-based reasoning.
- Definisi case based reasoning:

What is case-based reasoning?

- Case-based reasoning is another methodology for, among other things, identifying clusters of *similar* events in large databases

- KNN digunakan dalam banyak aplikasi data mining, statistical pattern recognition, image processing, dll.
- Beberapa aplikasinya meliputi: pengenalan tulisan tangan, satellite image dan ECG pattern. **ECG** produces a **pattern** reflecting the electrical activity of the heart.

Apa itu is K-Nearest Neighbor (KNN) Algorithm?

- K-nearest neighbor adalah algoritma supervised learning dimana hasil dari instance yang baru diklasifikasikan berdasarkan mayoritas dari kategori K-tetangga terdekat.
- Tujuan dari algoritma ini adalah untuk mengklasifikasikan obyek baru berdasarkan atribut dan sampel2 dari data training.
- Algoritma K Nearest neighbor menggunakan neighborhood classification sebagai nilai prediksi dari nilai instance yang baru.

Contoh kasus

- Data didapatkan dari kuesioner dengan obyek pengujian berupa dua atribut (daya tahan keasaman dan kekuatan) untuk mengklasifikasikan apakah sebuah kertas tissue tergolong bagus atau jelek. Berikut ini contoh datanya:

X1 = Daya tahan keasaman (detik)	X2 = Kekuatan (kg/meter persegi)	Klasifikasi
7	7	Jelek
7	4	Jelek
3	4	Bagus
1	4	Bagus

Contoh kasus

- Sebuah pabrik memproduksi kertas tissue baru yang memiliki X1 = 3 dan X2 = 7.
- Dapatkah kita melakukan prediksi termasuk klasifikasi apa (bagus atau jelek) kertas tissue yang baru ini?
- Algoritma K nearest neighbor (KNN) digunakan untuk ini.

Bagaimana cara kerja Algoritma K-Nearest Neighbor (KNN)?

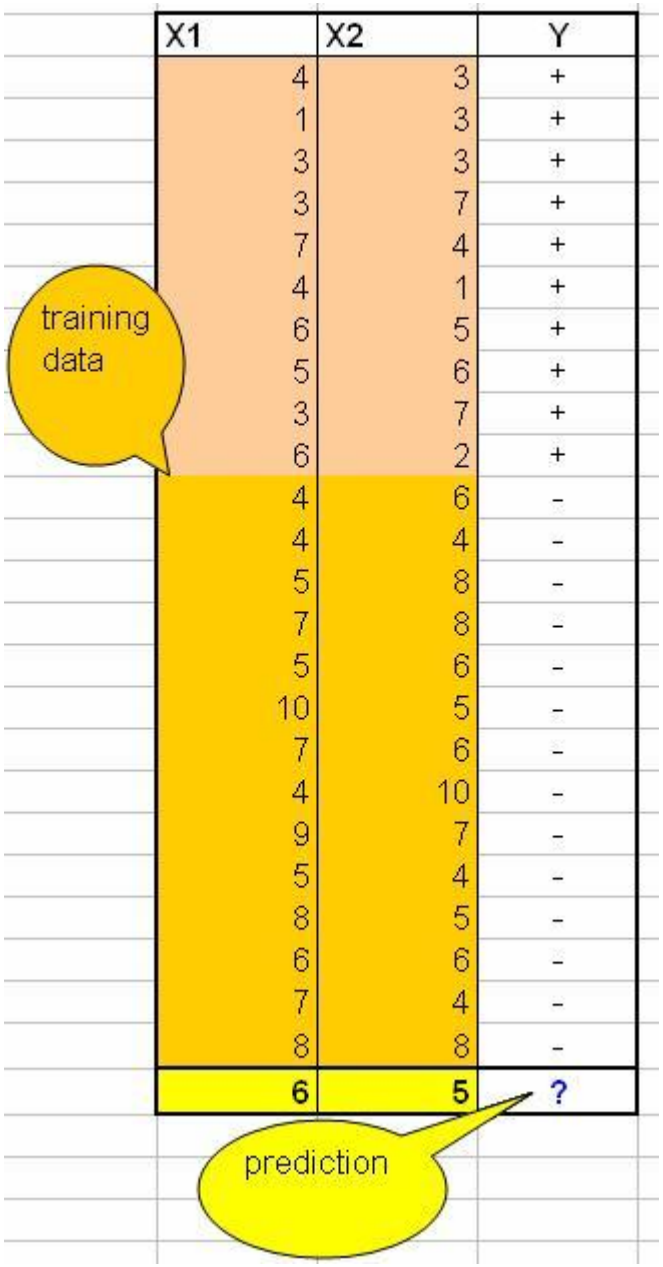
- K nearest neighbor bekerja berdasarkan jarak minimum dari data baru ke data training samples untuk menentukan K tetangga terdekat.
- Setelah itu, kita dapatkan nilai mayoritas sebagai hasil prediksi dari data yang baru tersebut.

Bagaimana cara kerja Algoritma K-Nearest Neighbor (KNN)?

- Data berisi banyak atribut (X1, X2, …, Xn) digunakan untuk mengklasifikasikan atribut target Y.

Bagaimana cara kerja Algoritma K-Nearest Neighbor (KNN)?

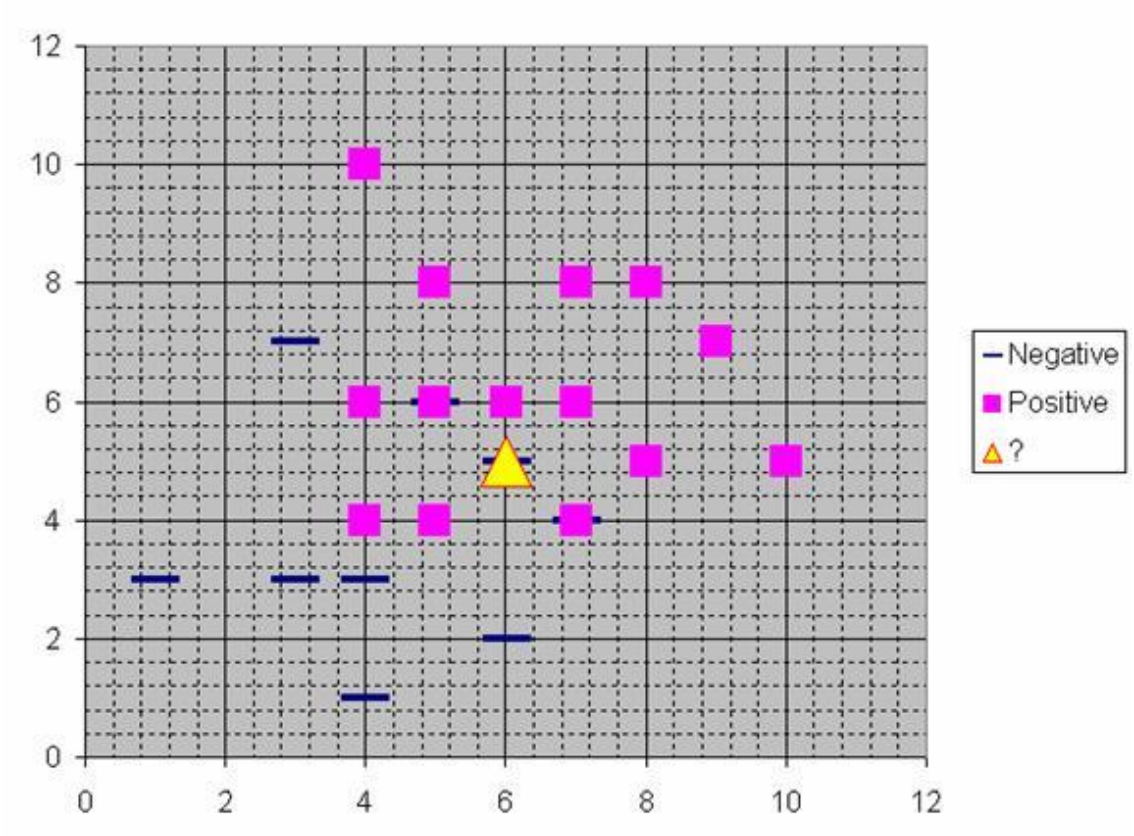
- Misal kita punya data training seperti ini:
- Data pada baris terakhir adalah data baru yang akan diprediksi nilai dari atribut Y.



X1	X2	Y
4	3	+
1	3	+
3	3	+
3	7	+
7	4	+
4	1	+
6	5	+
5	6	+
3	7	+
6	2	+
4	6	-
4	4	-
5	8	-
7	8	-
5	6	-
10	5	-
7	6	-
4	10	-
9	7	-
5	4	-
8	5	-
6	6	-
7	4	-
8	8	-
6	5	?

Bagaimana cara menyelesaikan masalah ?

- Graph dari persoalan tersebut ditunjukkan oleh gambar berikut



Contoh perhitungan numerik (perhitungan dengan cara manual)

- Berikut ini langkah-langkah dari algoritma K-nearest neighbors (KNN):
  1. Tentukan parameter K = jumlah banyaknya tetangga terdekat
  2. Hitung jarak antara data baru dan semua data yang ada di data training.
  3. Urutkan jarak tersebut dan tentukan tetangga mana yang terdekat berdasarkan jarak minimum ke-K.
  4. Tentukan kategori dari tetangga terdekat.
  5. Gunakan kategori mayoritas yang sederhana dari tetangga yang terdekat tersebut sebagai nilai prediksi dari data yang baru.

Contoh perhitungan numerik (perhitungan dengan cara manual)

- **Contoh**
- Data didapatkan dari kuesioner dengan obyek pengujian berupa dua atribut (daya tahan keasaman dan kekuatan) untuk mengklasifikasikan apakah sebuah kertas tissue tergolong bagus atau jelek. Berikut ini contoh datanya:
-

X1 = Daya tahan keasaman (detik)	X2 = Kekuatan (kg/meter persegi)	Klasifikasi
7	7	Jelek
7	4	Jelek
3	4	Bagus
1	4	Bagus

- Sebuah pabrik memproduksi kertas tissue baru yang memiliki X1 = 3 dan X2 = 7. Kita gunakan algoritma KNN untuk melakukan prediksi termasuk klasifikasi apa (bagus atau jelek) kertas tissue yang baru ini.

Contoh perhitungan numerik (perhitungan dengan cara manual)

1. Tentukan parameter K = jumlah banyaknya tetangga terdekat. Misal K=3
2. Hitung jarak antara data baru dan semua data yang ada di data training. Misal digunakan square distance dari jarak antara data baru dengan semua data yang ada di data training

X1 = Daya tahan keasaman (detik)	X2 = Kekuatan (kg/meter persegi)	**Square Distance ke data baru (3, 7)**
7	7	$(7-3)^2+(7-7)^2=16$
7	4	$(7-3)^2+(4-7)^2=25$
3	4	$(3-3)^2+(4-7)^2=9$
1	4	$(1-3)^2+(4-7)^2=13$

Contoh perhitungan numerik (perhitungan dengan cara manual)

3. Urutkan jarak tersebut dan tentukan tetangga mana yang terdekat berdasarkan jarak minimum ke-K.

X1 = Daya tahan keasaman (detik)	X2 = Kekuatan (kg/meter persegi)	Square Distance to query instance (3, 7)	**Urutan (ranking) jarak**	**Apakah termasuk 3-NN?**
7	7	$(7-3)^2+(7-7)^2=16$	**3**	**YA**
7	4	$(7-3)^2+(4-7)^2=25$	**4**	**TIDAK**
3	4	$(3-3)^2+(4-7)^2=9$	**1**	**YA**
1	4	$(1-3)^2+(4-7)^2=13$	**2**	**YA**

Contoh perhitungan numerik (perhitungan dengan cara manual)

4. Tentukan kategori dari tetangga terdekat. Perhatikan pada baris kedua pada kolom terakhir: katagori dari tetangga terdekat (Y) tidak termasuk karena ranking dari data ini lebih dari 3 (=K).

X1 = Daya tahan keasaman (detik)	X2 = Kekuatan (kg/meter persegi)	Square Distance to query instance (3, 7)	**Urutan (ranking) jarak**	**Apakah termasuk 3-NN?**	**Y = Category of nearest Neighbor**
7	7	$(7-3)^2+(7-7)^2=16$	3	**YA**	**Jelek**
7	4	$(7-3)^2+(4-7)^2=25$	4	**TIDAK**	**-**
3	4	$(3-3)^2+(4-7)^2=9$	1	**YA**	**Bagus**
1	4	$(1-3)^2+(4-7)^2=13$	2	**YA**	**Bagus**

Contoh perhitungan numerik (perhitungan dengan cara manual)

5. Gunakan kategori mayoritas yang sederhana dari tetangga yang terdekat tersebut sebagai nilai prediksi dari data yang baru.

   – Kita punya 2 kategori Bagus dan 1 kategori Jelek, karena 2>1 maka kita simpulkan bahwa kertas tissue baru tadi yang memiliki X1 = 3 dan X2 = 7 termasuk dalam kategori **Bagus.**

Kelebihan dan kelemahan dari Algoritma K-Nearest Neighbour

- Kelebihan dari Algoritma K-Nearest Neighbor :
  - Robust terhadap data yang noisy
  - Efektif jika training data berjumlah banyak
- Kekurangan dari Algoritma K-Nearest Neighbor :
  - Perlu menunjukkan parameter K (jumlah tetangga terdekat)
  - Berdasarkan perhitungan nilai jarak (Distance based learning), tidak jelas perhitungan jarak mana yang sebaiknya digunakan dan atribut mana yang memberikan hasil yang baik.
  - Nilai komputasinya tinggi karena kita perlu menghitung jarak dari nilai baru ke semua data yang ada di data training. Beberapa cara pengindexan (K-D tree) dapat digunakan untuk mereduksi biaya komputasi.

Latihan soal (1)

1. Misal kita punya data training seperti pada gambar berikut ini, data pada baris terakhir adalah data baru yang akan diprediksi nilai dari atribut Y.
   a. Gunakan Algoritma K-Nearest Neighbour dengan K=10 (10-Nearest Neighbour) untuk mencari prediksi dari nilai Y pada data baru (baris terakhir).

Latihan soal (2)

b. Jika digunakan K=1 (1-Nearest Neighbour) berapa nilai prediksi Y pada data baru (baris terakhir)?