

Klasifikasi

Data Mining

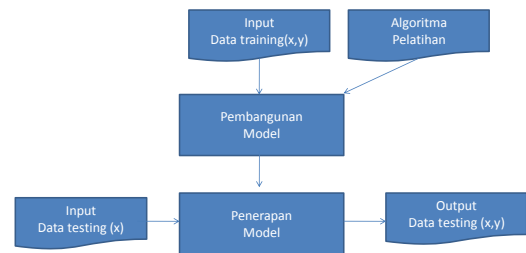
Konsep Klasifikasi

- Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia (Eko Prasetyo,2012).

Pekerjaan Klasifikasi

1. Pembangunan model sebagai prototipe untuk disimpan sebagai memory.
2. Penggunaan model untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut berada berdasarkan model yang ada.

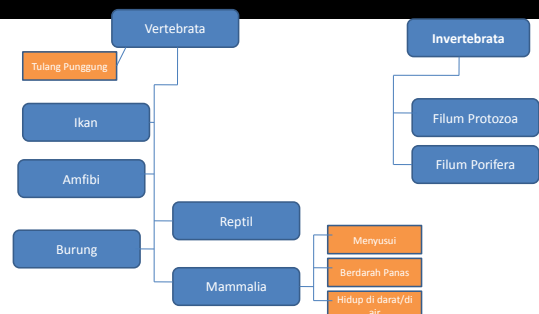
Pekerjaan Klasifikasi



Contoh

- Pengklasifikasian jenis hewan yang mempunyai sejumlah atribut. Dengan atribut tsb jika ada hewan baru kelas dari hewan tsb bisa langsung diketahui.

Klasifikasi



Model Klasifikasi

- Pemodelan **Deskriptif**
Model Klasifikasi yang dpt berfungsi sbg alat **penjelasan** untuk membedakan **obyek-obyek** dalam **kelas-kelas** yang berbeda.
- Pemodelan **Prediktif**
Model klasifikasi yang dapat digunakan untuk **memprediksi label kelas** yang tidak diketahui pada suatu **object/record**.

Algoritma

- Pembangunan model selama proses pelatihan memerlukan algoritma pelatihan
- Berdasarkan cara pelatihan:
 - Eager learner
Melakukan pembacaan pd data training untuk menghasilkan model. Proses prediksi dilakukan dengan model tsb. (contoh ANN,SVM,Decision Tree,Regresi Linear)
 - Lazy learner
Sedikit melakukan pelatihan(tdk sama sekali) menggunakan sebagian/seluruh data training dalam proses prediksi (contoh K-Nearest Neighbour,Fuzzy K-NN,Bayesian)

Nearest Neighbor (K-NN)

- Metode k-NN pertama kali dikenalkan pada awal tahun 1950an. Metode ini bekerja lambat bila diberi pelatihan(data training) besar. Tahun 1960an ketika daya komputasi meningkat mulai banyak digunakan di bidang pengenalan pola.

KNN

- Algoritma K-NN merupakan algoritma yang melakukan klasifikasi berdasarkan kedekatan jarak suatu data dengan data lainnya.
- Kedekatan jarak dihitung dengan jarak euclidean,manhattan

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2}$$

KNN

- Algorithm ini berusaha mengklasifikasikan data baru yang belum diketahui class-nya dengan memilih data sejumlah k yang letaknya terdekat dari data baru tersebut.
- Class terbanyak dari data terdekat sejumlah k tersebut dipilih sebagai class yang diprediksikan untuk data yang baru.
- k umumnya ditentukan dalam jumlah ganjil untuk menghindari munculnya jumlah jarak yang sama dalam proses pengklasifikasian.

Algoritma K-NN

1. Cari sejumlah K-tetangga terdekat pada data training.
2. Ambil class dengan jumlah data terbesar untuk k-tetangga

Contoh Klasifikasi dg K-NN

Data Training				Data testing			
No	Atribut 1	Atribut 2	Class	No	Atribut 1	Atribut 2	Class
1	0,8	6,3	-	21	16,1	6,8	+
2	1,4	8,1	-	22	17,0	4,9	-
3	2,1	7,4	-	23	18,9	6,1	-
4	2,6	14,3	+	24	19,0	5,4	-
5	6,8	12,6	-	25	17,6	11,1	+
6	8,8	9,8	+				
7	9,2	11,6	-				
8	10,8	9,5	-				
9	11,4	9,8	+				
10	12,4	6,5	+				
11	12,8	1,1	-				
12	15	10,9	-				
13	14,2	18,5	-				
14	15,6	17,4	-				
15	16,8	19,2	-				
16	16,6	6,7	+				
17	17,4	4,5	+				
18	18,2	6,9	+				
19	19	3,4	-				
20	19,6	11,1	+				

K-NN

- Cari sejumlah K-tetangga terdekat pada data training, dengan cara:
 - Hitung jarak setiap data testing dengan setiap data training
 - Tentukan K (jml tetangga terdekat), misl: K=10

K-NN

Data Training				Jarak untuk data ke-21 (data testing) dengan masing-masing data di data training				
No	Atribut 1	Atribut 2	Class	No	Atribut 1	Atribut 2	Class	d(21)
1	0,8	6,3	-	21	16,1	6,8	+	13,71
2	1,4	8,1	-	22	17,0	4,9	-	14,76
3	2,1	7,4	-	23	18,9	6,1	-	14,01
4	2,6	14,3	+	24	19,0	5,4	-	15,44
5	6,8	12,6	-	25	17,6	11,1	+	10,96
6	8,8	9,8	+					7,89
7	9,2	11,6	-					8,41
8	10,8	9,5	-					5,99
9	11,4	9,8	+					5,20
10	12,4	6,5	+					3,71
11	12,8	1,1	-					6,59
12	15	10,9	-					13,27
13	14,2	18,5	-					11,89
14	15,6	17,4	-					10,61
15	16,8	19,2	-					5,41
16	16,6	6,7	+					0,51
17	17,4	4,5	+					2,64
18	18,2	6,9	+					2,10
19	19	3,4	-					4,47
20	19,6	11,1	+					5,54

K-NN

- Ambil class dengan jumlah data terbesar, dengan cara:
 - Hitung jumlah masing-masing class pada ke-10 data tsb
 - Pilih nilai class dengan jumlah nilai terbesar

K-NN

Dipilih 10 tetangga terdekat				Hitung jumlah masing-masing class			
No	Atribut 1	Atribut 2	Class	Class	Jumlah		
16	16,6	6,7	+	+	5		
18	18,2	6,9	+	+	3		
17	17,4	4,5	+	+	2		
19	19,0	3,4	-	-	1		
20	19,6	11,1	+	+	1		
10	12,4	6,5	+	+	1		
9	11,4	9,8	+	+	1		
15	16,8	19,2	-	-	1		
25	17,6	11,1	+	+	1		
8	10,8	9,5	-	-	1		
11	12,8	1,1	-	-	1		
6	8,8	9,8	+	+	1		
7	9,2	11,6	-	-	1		
14	15,6	17,4	-	-	1		
5	6,8	12,6	-	-	1		
13	14,2	18,5	-	-	1		
12	15	10,9	-	-	1		
3	2,1	7,4	-	-	1		
2	1,4	8,1	-	-	1		
1	0,8	6,3	-	-	1		
4	2,6	14,3	+	+	1		

Latihan

- Prediksikan class (y) dengan algoritma K-NN dengan K=7
- untuk data sbb:

x1	x2	y
8	8	
12	9	

Obyek	Promosi (x ₁)	Harga (x ₂)	Keputusan Konsumen (y)
1	10	7	A
2	2	3	B
3	4	2	B
4	6	4	B
5	8	6	A
6	7	5	A
7	4	3	B
8	6	3	B
9	7	4	A
10	6	3	A

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 typedef float Matrix[10][10];
6 typedef float Arr[3];
7
8 int main()
9 {
10 //data seeding
11 Matrix A={{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0},{1,1,0,1,1,1,0,1,1,0}};
12 //data seeding
13 Matrix B;
14 Arr C[3]={1,4,0};
15 Arr X;
16 float hasil;
17 int j, k, m, n;
18 //data seeding
19 printMatrix(A, 10, 10);
20 hitungJarak(A, B, C, X, 10);
21 printf("%d\n", hasil);
22 printMatrix(B, 10, 10);
23 printMatrix(C, 10, 10);
24 urut(B, 10);
25 printMatrix(B, 10, 10);
26 j=0; k=0;
27 k=0;
28 }

```

```

31 for (j=0; j<n; j++)
32 if (B[j][k]==1)
33 j++;
34 if (B[j][k]==0)
35 j--;
36 }
37 if (j>0)
38 printf("hasil:");
39 else
40 printf("hasil:0");
41 return 0;
42 }
43
44 void printArr(Arr X, int m) {
45 int i, j;
46 for (i=0; i<m; i++)
47 printf("%d.2f\t", X[i]);
48 printf("\n");
49 }
50
51 void urutMatrix(A, int m) {
52 int i, j;
53 for (i=0; i<m-1; i++)
54 for (j=i+1; j<m; j++)
55 if (A[i][0]>A[j][0])
56 tukar(A[i][0], A[j][0]);
57 tukar(A[i][1], A[j][1]);
58 }
59 }
60
61
62 void urut2(Matrix A, int m) {
63 int i, j;
64 for (i=0; i<m-1; i++)
65 for (j=i+1; j<m; j++)
66 if (A[i][1]>A[j][1])
67 tukar(A[i][1], A[j][1]);
68 }
69
70
71
72
73
74
75 void hitungJarak(Matrix A, Matrix B, Arr X, int m, int n) {
76 int i, j;
77 float sum;
78 for (i=0; i<m; i++)
79 sum+=
80 for (j=0; j<n; j++)
81 sum+=pow(abs(X[j]-A[i][j]), 2);
82
83 B[i][0]=sqrt(sum);
84 B[i][1]=A[i][1];
85 }
86
87 void printMatrix(Matrix A, int m, int n) {
88 int i, j;
89 for (i=0; i<m; i++)
90 for (j=0; j<n; j++)

```

```

87 void printMatrix(Matrix A, int m, int n) {
88 int i, j;
89 for (i=0; i<m; i++)
90 printf("%d.2f\t", A[i][j]);
91 }
92 for (j=0; j<n; j++)
93 printf("%d.2f\t", A[i][j]);
94 }
95 printf("\n");
96 }
97
98 void tukar(float *a, float *b) {
99 float temp;
100 temp=*a;
101 *a=*b;
102 *b=temp;
103 }

```

D:\perguruan\DM\Apit\test\test\Debug\Mat

0	1.0000	1.0000	0.0000
1	1.0000	1.0000	0.0000
2	1.0000	1.0000	0.0000
3	1.0000	1.0000	0.0000
4	1.0000	1.0000	0.0000
5	1.0000	1.0000	0.0000
6	1.0000	1.0000	0.0000
7	1.0000	1.0000	0.0000
8	1.0000	1.0000	0.0000
9	1.0000	1.0000	0.0000
10	1.0000	1.0000	0.0000
11	1.0000	1.0000	0.0000
12	1.0000	1.0000	0.0000
13	1.0000	1.0000	0.0000
14	1.0000	1.0000	0.0000
15	1.0000	1.0000	0.0000
16	1.0000	1.0000	0.0000
17	1.0000	1.0000	0.0000
18	1.0000	1.0000	0.0000
19	1.0000	1.0000	0.0000
20	1.0000	1.0000	0.0000
21	1.0000	1.0000	0.0000
22	1.0000	1.0000	0.0000
23	1.0000	1.0000	0.0000
24	1.0000	1.0000	0.0000
25	1.0000	1.0000	0.0000
26	1.0000	1.0000	0.0000
27	1.0000	1.0000	0.0000
28	1.0000	1.0000	0.0000
29	1.0000	1.0000	0.0000
30	1.0000	1.0000	0.0000