

TUGAS M-5

KNOWLEDGE DISCOVERY

CLUSTERING & CLUSTER ANALYSIS

Nama : Ni Putu Devira Ayu Martini
Kelas : S2 Elektro 2020
NRP : 1120800012
Pengampu : P. Ali Ridho Barakbah

Tugas

1. Convert that data into the numerical values
2. Impute the missing data with the mean values of same attribute in the same class
3. Hide the class label of the supervised data
4. Cluster the data using K-Means or Hierarchical Clustering into 2 groups
5. Do cluster analysis
6. What is your analysis from the experiment?
7. Make the report and upload to the website (M5)

DATASET 'Hepatitis'

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
#	Age	Sex	Steroid	Antivirals	Fatigue	Malaise	A0rexia	1r Big	1r Firm	Spleen Palpable	Speiders	Ascites	Varices	Bilirubin	Alk Phosphate	SGOT	Albumin	Protime	Histology	CLASS
1	30	1	0	1	1	1	1	0	1	1	1	1	1	1	85	18	4	0	0	1
2	50	0	0	1	0	1	1	0	1	1	1	1	1	0.9	135	42	3.5	0	0	1
3	78	0	1	1	0	1	1	1	1	1	1	1	1	0.7	96	32	4	0	0	1
4	31	0	0	0	1	1	1	1	1	1	1	1	1	0.7	46	52	4	80	0	1
5	34	0	1	1	1	1	1	1	1	1	1	1	1	1	0	200	4	0	0	1
6	34	0	1	1	1	1	1	1	1	1	1	1	1	0.9	95	28	4	75	0	1
7	51	0	0	1	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0
8	23	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
9	39	0	1	1	0	1	1	1	0	1	1	1	1	0.7	0	48	4.4	0	0	1
10	30	0	1	1	1	1	1	1	1	1	1	1	1	1	0	120	3.9	0	0	1
11	39	0	0	0	1	1	1	0	0	1	1	1	1	1.3	78	30	4.4	85	0	1
12	32	0	1	0	0	1	1	1	0	1	0	1	1	1	59	249	3.7	54	0	1
13	41	0	1	0	0	1	1	1	0	1	1	1	1	0.9	81	60	3.9	52	0	1
14	30	0	1	1	0	1	1	1	0	1	1	1	1	2.2	57	144	4.9	78	0	1
15	47	0	0	0	1	1	1	1	1	1	1	1	1	0	0	60	0	0	0	1
16	38	0	0	1	0	0	0	1	1	1	1	0	1	2	72	89	2.9	46	0	1
17	66	0	1	1	0	1	1	1	1	1	1	1	1	1.2	102	53	4.3	0	0	1
18	40	0	0	1	0	1	1	1	0	1	1	1	1	0.6	62	166	4	63	0	1
19	38	0	1	1	1	1	1	1	1	1	1	1	1	0.7	53	42	4.1	85	1	1

- 19 Attribut
- 155 Data
- 2 Kelas (0 dan 1)

PERALATAN

1. Laptop/PC
2. Software Bahasa C
3. Data 'Hepatitis'

PROSES

1. Mendeklarasikan data input 'Hepatitis'

```
freopen("task1.txt", "r", stdin); //if You want to notepad , with name input.txt
for( int i = 1 ; i <=155; i++ )
{
    for(j=1; j<=19; j++)
    {
        scanf("%f", &I[i][j]);
    }
    scanf("%f",&kelas[i]);
}
```

2. Menormalisasi data input 'Hepatitis'

```
//Data Baru yang sudah diNormalisasi Min-Max(0-1)
//Data Learning
newmax=1;
newmin=0;|
for(i=1; i<=155; i++)
{
    I[i][1]=((I[i][1]-7)*(newmax-newmin))/((78-7)+newmin);
    I[i][14]=((I[i][14]-1)*(newmax-newmin))/((8-1)+newmin);
    I[i][15]=((I[i][15]-26)*(newmax-newmin))/((295-26)+newmin);
    I[i][16]=((I[i][16]-14)*(newmax-newmin))/((648-14)+newmin);
    I[i][17]=((I[i][17]-3)*(newmax-newmin))/((5-3)+newmin);
    I[i][18]=((I[i][18]-0)*(newmax-newmin))/((100-0)+newmin);
    //printf("%d = %f %f %f %f %f %f\n",i, Age[i], Bilir[i], Alk[i], Sgot[i], Albumin[i], Protine[i]);
}
```

3. Mendeklarasikan nilai Cluster yaitu dengan cara di Random antara nilai terkecil dan terbesar dari data 'Hepatitis'

```
for(i=1; i<=2; i++)
{
    for(j=1; j<=19; j++)
    {
        c[i][j]=rand()%100/99.1938129;
        //printf("c[%d][%d]= %f\n",i,j,c[i][j]);
    }
}
```

4. Membuat nilai looping

```
loop=10;
for(iterasi=1; iterasi<=loop; iterasi++)
{
```

5. Menghitung jarak setiap data pada atribut dalam setiap kelas dengan nilai setiap cluster

```
for(i=1; i<=155; i++)
{
    for(j=1; j<=2; j++) //Cluster
    {
        jarak[j]=0;
        for(k=1; k<=19; k++) //Penjumlahan jarak
        {
            jarak[j]=jarak[j]+pow(I[i][k]-c[j][k],2);
        }
        jarak[j]=sqrt(jarak[j]);
        //printf("jarak[%d][%d]=%f\n",i,j,jarak[j]);
    }
}
```

6. Menentukan ranking cluster pada setiap data 'Hepatitis'

```
for(j=1; j<=2; j++)
{
    rank[j]=1;
    for(b=1; b<=2; b++)
    {
        if(jarak[j]>jarak[b])
        {
            rank[j]++;
        }
    }
}
//printf("Nilai Ujian\t Ranking\n");

for(j=1; j<=2; j++)
//printf("%f\t\t %f\n",jarak[j], rank[j]);
```

7. Menghitung jumlah data yang termasuk Cluster 1 atau 2

```
for(j=1; j<=2; j++)
{
    if(rank[j]==1 && j==1)
    {
        jumlah[j]=jumlah[j]+1;
        ranking[i][j]=j;
    }
    else if(rank[j]==1 && j==2)
    {
        jumlah[j]=jumlah[j]+1;
        ranking[i][j]=j;
    }
    else
    {
        ranking[i][j]=0;
    }
}
```

8. Menghitung total jumlah data yang termasuk Cluster 1 atau 2

```
//printf("\n\n");
for(i=1; i<=2; i++)
{
    //printf("jumlah[%d] = %f\n",i,jumlah[i]);
}
```


9. Menghitung penjumlahan data yang termasuk kedalam cluster 1 atau 2 untuk keperluan perhitungan Mean

```
for(i=1; i<=155; i++)
{
    printf("i= %d    ",i);
    for(j=1; j<=2; j++)
    {
        if(ranking[i][j]==1)
        {
            kelastesting[i]=ranking[i][j];
            printf("class[%d] = %f\n",i,kelastesting[i]);
            mean[ranking[i][j]][1]=mean[ranking[i][j]][1]+I[i][1];
            mean[ranking[i][j]][2]=mean[ranking[i][j]][2]+I[i][2];
            mean[ranking[i][j]][3]=mean[ranking[i][j]][3]+I[i][3];
            mean[ranking[i][j]][4]=mean[ranking[i][j]][3]+I[i][4];
            mean[ranking[i][j]][5]=mean[ranking[i][j]][3]+I[i][5];
            mean[ranking[i][j]][6]=mean[ranking[i][j]][3]+I[i][6];
            mean[ranking[i][j]][7]=mean[ranking[i][j]][3]+I[i][7];
            mean[ranking[i][j]][8]=mean[ranking[i][j]][3]+I[i][8];
            mean[ranking[i][j]][9]=mean[ranking[i][j]][3]+I[i][9];
            mean[ranking[i][j]][10]=mean[ranking[i][j]][3]+I[i][10];
            mean[ranking[i][j]][11]=mean[ranking[i][j]][3]+I[i][11];
            mean[ranking[i][j]][12]=mean[ranking[i][j]][3]+I[i][12];
            mean[ranking[i][j]][13]=mean[ranking[i][j]][3]+I[i][13];
            mean[ranking[i][j]][14]=mean[ranking[i][j]][3]+I[i][14];
            mean[ranking[i][j]][15]=mean[ranking[i][j]][3]+I[i][15];
            mean[ranking[i][j]][16]=mean[ranking[i][j]][3]+I[i][16];
            mean[ranking[i][j]][17]=mean[ranking[i][j]][3]+I[i][17];
            mean[ranking[i][j]][18]=mean[ranking[i][j]][3]+I[i][18];
            mean[ranking[i][j]][19]=mean[ranking[i][j]][3]+I[i][19];
        }
    }
}
```

10. Menghitung total penjumlahan data yang termasuk kedalam cluster 1 atau 2 untuk keperluan perhitungan Mean

```
for(i=1; i<=2; i++)
{
    for(j=1; j<=19; j++)
    {
        //printf("mean[%d][%d] = %f\n",i,j,mean[i][j]);
    }
}
```

11. Menghitung Mean

```
//printf("\n\n");
for(i=1; i<=2; i++)
{
    if(i==1)
    {
        for(j=1; j<=19; j++)
        {
            c[i][j]=mean[i][j]/jumlah[i];
            printf("c[%d][%d] = %f\n",i,j,c[i][j]);
        }
    }
    else if(i==2)
    {
        for(j=1; j<=19; j++)
        {
            c[i][j]=mean[i][j]/jumlah[i];
            printf("c[%d][%d] = %f\n",i,j,c[i][j]);
        }
    }
}
```

12. Menghitung Error menggunakan perhitungan 'Error Ratio'

```
int error;
error=0;
for(i=1; i<=155; i++)
{
    if(kelas[i]==kelastesting[i])
    {
        error=error+0;
    }
    else if(kelas[i]!=kelastesting[i])
    {
        error=error+1;
    }
}

printf("error = %d\n",error);
float totalerror = error*100/155;
printf("totalerror = %f\n",totalerror);
```

ANALISA

- Praktikum kali ini dilakukan menggunakan Algoritma Clustering K-Means untuk melakukan pengelompokan data dengan cara membagi data ke dalam jumlah cluster -2, dan memanfaatkan perhitungan jarak untuk mengukur kemiripan antar data.
- Untuk Analisa Cluster menggunakan perhitungan Error Ratio. Dimana perhitungan tersebut membutuhkan data supervised (data yang sudah ada kelas labelnya). Perhitungan Error Ratio digunakan untuk melihat presisi kinerja dari Algoritma Clustering K-Means.
- Error yang didapat dengan menggunakan data 'Hepatitis' dengan 2 Cluster dalam praktikum ini adalah 25%.