ITERA

Modul 5 Praktikum
Statistika Sains Data

k-nearest neighbor (KNN)

Program Studi Sains Data
Fakultas Sains
Institut Teknologi Sumatera

2024

A. Tujuan Praktikum

1. Mahasiswa mampu mengklasifikasikan objek baru berdasarkan atribut dan training samples menggunakan algoritma KNN.
2. Mahasiswa mampu mengimplementasi metode K-nearest neighbor (KNN).

B. Teori Dasar

Algoritma k-nearest neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN termasuk algoritma supervised learning dimana hasil dari query instance yang baru diklasifikan berdasarkan mayoritas dari kategori pada KNN. Nanti kelas yang paling banyak muncullah yang akan menjadi kelas hasil klasifikasi.

Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru bedasarkan atribut dan training sample. Classifier tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik query, akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik query. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek. Algoritma k-nearest neighbor (KNN) menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari query instance yang baru. Kita akan menerapkan KNN menggunakan data "Growth of Orange Trees". Growth of Orange Trees adalah dataset yang menjelaskan tentang pertumbuhan pohon jeruk yang dilihat dari usia pohon serta lingkar batang pada pohon. Kerangka data Orange memiliki 35 baris dan 3 kolom catatan tentang pertumbuhan pohon jeruk.

```
library(class)
Orange
summary(Orange)
```

Pertama kita akan mengaktifkan *package library(class)*. Kemudian memanggil data *orange* yang sudah dijelaskan sebelumnya dan mencari informasi mengenai data tersebut menggunakan sintaks *summary(Orange)*. Berikut adalah *output*-nya:

```
> summary(Orange)
 Tree       age         circumference
 3:7   Min.   : 118.0   Min.   : 30.0
 1:7   1st Qu.: 484.0   1st Qu.: 65.5
 5:7   Median :1004.0   Median :115.0
 2:7   Mean   : 922.1   Mean   :115.9
 4:7   3rd Qu.:1372.0   3rd Qu.:161.5
       Max.   :1582.0   Max.   :214.0
```

Tree merupakan sebuah vaktor dengan level 1 sampai 5 yang menunjukan eksperimen yang bisa diterima oleh pohon jeruk dan terdapat 7 pohon jeruk pada masing-masing eksperimen. *Age* merupakan umur pohon jeruk dengan umur terendah yaitu 118 hari dan umur terpanjang yaitu 1582 hari. Sedangkan *Circumference* merupakan lingkar pada masing-masing batang pohon jeruk, dan lingkar pada pohon jeruk terkecil yaitu 30cm dengan lingkar batang pohon jeruk terbesar yaitu 214cm.

Selanjutnya saya akan membuat kelas data menggunakan sintaks berikut:

```
Orange.kelas<-c(rep("1",7),rep("2",7),rep("3",7),rep("4",7),rep("5",7))Orange.data<-
data.frame(Orange[,2:3],tree=Orange.kelas)Orange.data
```

```
> Orange.data
    age circumference tree
1   118            30    1
2   484            58    1
3   664            87    1
4  1004           115    1
5  1231           120    1
6  1372           142    1
7  1582           145    1
8   118            33    2
9   484            69    2
10  664           111    2
11 1004           156    2
12 1231           172    2
13 1372           203    2
14 1582           203    2
15  118            30    3
16  484            51    3
17  664            75    3
18 1004           108    3
19 1231           115    3
20 1372           139    3
21 1582           140    3
22  118            32    4
23  484            62    4
24  664           112    4
25 1004           167    4
26 1231           179    4
27 1372           209    4
28 1582           214    4
29  118            30    5
30  484            49    5
31  664            81    5
32 1004           125    5
33 1231           142    5
34 1372           174    5
35 1582           177    5
```

Gambar diatas merupakan kelas data pada *dataset "Growth of Orange Trees"* dengan menggunakan 2 variabel yaitu variabel "*Age*" dan variabel "*circumference*".

```
Orange.knn<-knn(Orange.latihan[,-3],Orange.uji[,-3],Orange.latihan[,3],k=3)
(table(Orange.knn,Orange.uji[,3]))
```

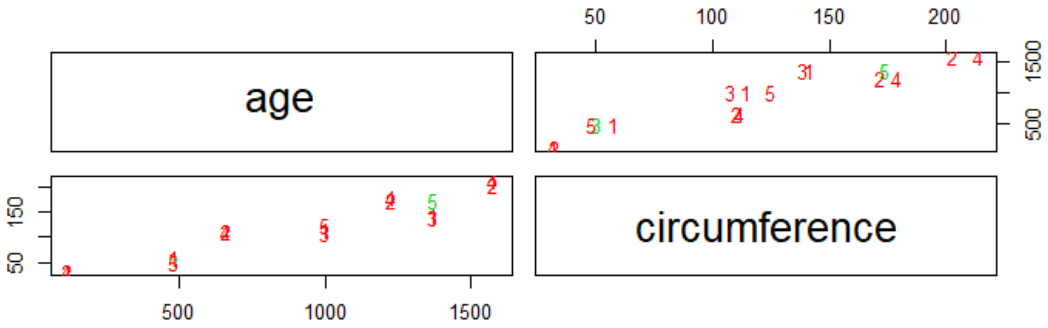
Sintaks diatas digunakan untuk menampilkan klasifikasi.

```
Orange.knn 1 2 3 4 5
         1 0 0 0 1 1
         2 1 0 2 0 0
         3 0 1 1 2 0
         4 2 0 0 0 1
         5 0 3 0 1 1
```

Dari *output* diatas dapat dilihat bahwa pohon pertama benar sebanyak 0 dengan salah sebanyak 1, pohon kedua benar sebanyak 0 dan salah sebanyak 3, pohon ketiga benar sebanyak 1 dan salah sebanyak 5, pohon 4 benar sebanyak 0 dan salah sebanyak 3, serta pohon lima benar sebanyak 1 dan salah sebanyak 3.

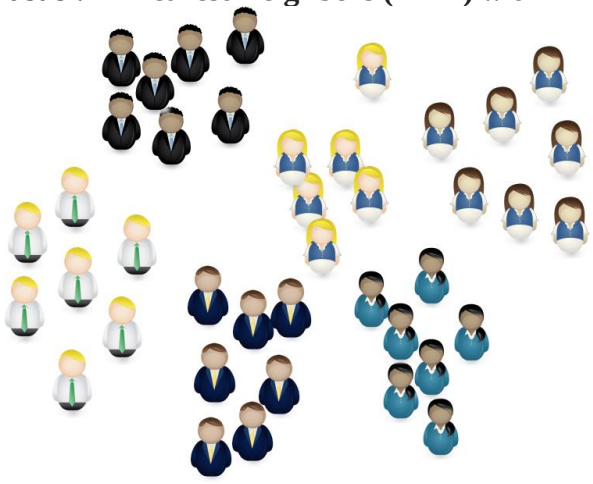
Kemudian kita akan *plot cluster*-nya.

```
pairs(Orange.uji[,1:2],pch=as.character(Orange.uji[,3]),col=c(3,2)
      [(Orange.uji$tree!=Orange.knn)+1])
```



Dari *plot* diatas terlihat bahwa terdapat warna merah dari variabel "*Age*" serta "*circumference*" yang berwarna hijau hanya terdapat pada pohon 3 dan 5 dari setiap variabel "*Age*" maupun "*Circumference*".

Contoh Kasus : K-Nearest Neighbors (KNN) with R



K-Nearest Neighbors atau KNN adalah algoritma yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (*train data sets*), yang diambil dari **k** tetangga terdekatnya (*nearest neighbors*). Dengan **k** merupakan banyaknya tetangga terdekat.

Weight versus age of chicks on different diets merupakan merupakan salah satu dataset yang terdapat dalam R. *Weight versus age of chicks on different diets* adalah kerangka data *ChickWeight* memiliki 578 baris dan 4 kolom dari percobaan tentang efek diet pada pertumbuhan awal anak ayam.

Variabel	Keterangan
Weight (gram)	Vektor numerik yang memberikan bobot tubuh anak ayam.
Time	Vektor numerik yang memberikan jumlah hari sejak lahir saat pengukuran dilakukan
Chick	faktor terurut dengan level 18 < x < 48 memberikan pengidentifikasi unik untuk anak ayam. Pengurutan kelompok level dilakukan bersama-sama pada diet yang sama dan memesannya sesuai dengan berat akhir mereka (paling ringan hingga terberat) dalam diet.
Diet	sebuah faktor dengan level 1, 2, 3, 4 yang menunjukkan diet eksperimental mana yang diterima anak ayam.

Dataset Weight versus age of chicks on different diets terdiri dari 578 kolom dan 4 baris. Untuk melakukan pengecekan jumlah baris dan kolom dapat dilakukan dengan perintah "dim(ChickWeight)".

```
library(class)
ChickWeight <-ChickWeight
ChickWeight
str(ChickWeight)
dim(ChickWeight)
```

Untuk mengetahui deskriptif dari dataset *Weight versus age of chicks on different diets* dapat dilakukan dengan menggunakan perintah "summary(ChickWeight)".

```
summary(ChickWeight)
```

```
> summary(ChickWeight)
     weight           Time           Chick     Diet
 Min.   : 35.0   Min.   : 0.00   13     : 12   1:220
 1st Qu.: 63.0   1st Qu.: 4.00   9      : 12   2:120
 Median :103.0   Median :10.00   20     : 12   3:120
 Mean   :121.8   Mean   :10.72   10     : 12   4:118
 3rd Qu.:163.8   3rd Qu.:16.00   17     : 12
 Max.   :373.0   Max.   :21.00   19     : 12
                                 (Other):506
```

Weight yang merupakan vector numerik yang memberikan bobot pada tubuh anak ayam dengan satuan gram memiliki bobot tubuh ayam terberat sebesar 373 grm, bobot tubuh anak ayam terkecil sebesar 35 grm, dan rata-rata bobot anak ayam sebesar 121.8 gram.

Time yang merupakan vaktor numerik dari jumlah hari saat ayam menetas saat dilakukan pengukuran memiliki jumlah hari telama sebesar 21 hari dengan rata-rata hari menetas sebesar 10.72.

Diet yang merupakan sebuah faktor dengan level 1 sampai 4 yang menunjukkan diet eksperimen yang bisa di terima oleh anak ayam. Pada level pertama terdapat 220 anak ayam yang menerima, level kedua dan ketiga terdapat 120 anak ayam yang menerima, dan level keempat terdapat 118 anak ayam yang menerima.

Dalam pembuatan *K-Nerest Neaighbors* diperlukan pembagian data yaitu data train dan data test atau yang biasa dikenal dengan metode *cross validation*. Pada umumnya data train sebesar 80% dan data test sebesar 20%. Untuk membagi data dapat dilakukan dengan menggunakan perintah :

```
indexes = sample(1:nrow(ChickWeight), size = 0.2*nrow(ChickWeight))
#Test
test = ChickWeight[indexes,]
head(test)
#Dimensi Test
dim(test)
#Train
train = ChickWeight[-indexes,]
head(train)
```

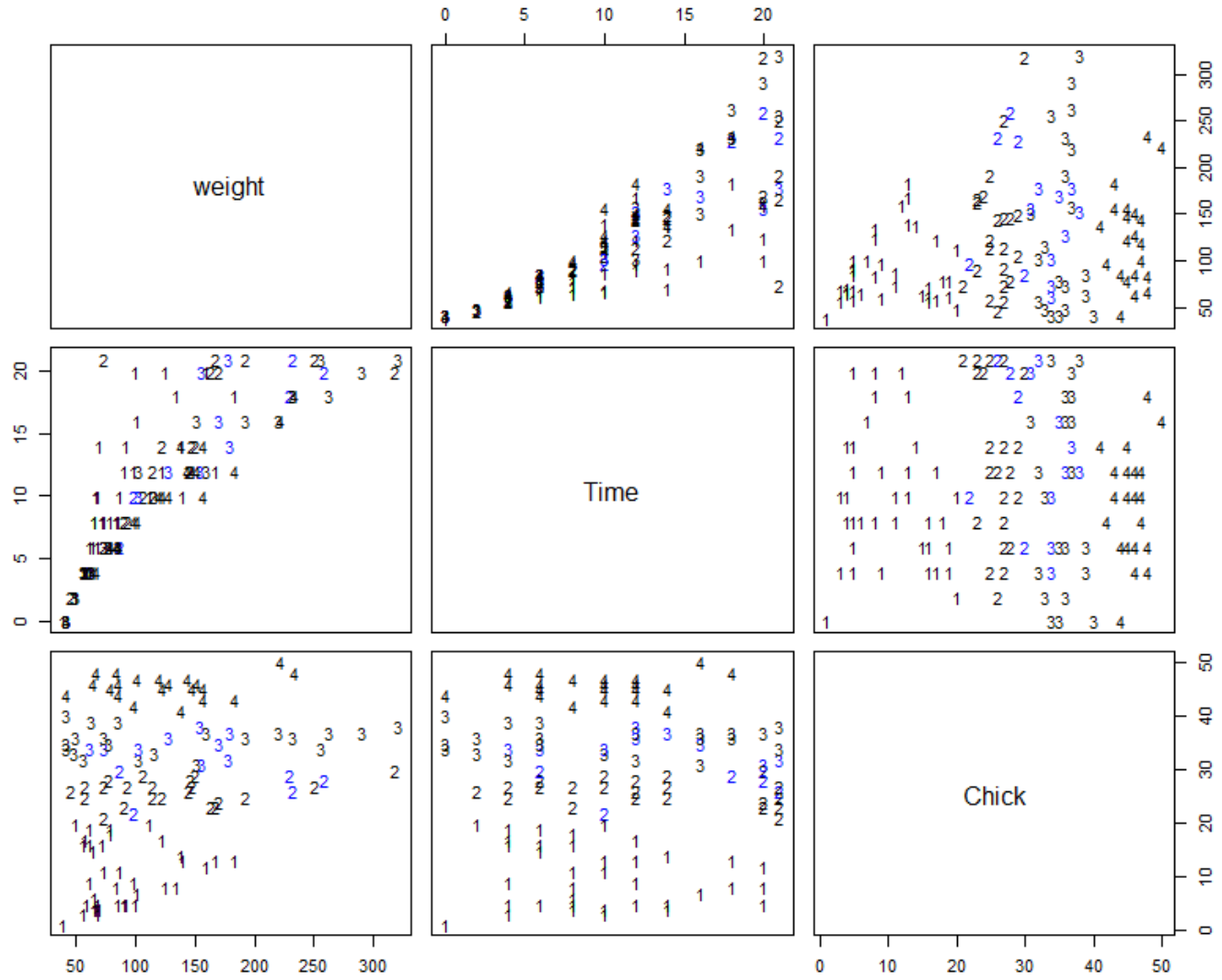
Selanjutnya, setelah data *test* dilakukan proses silang dimana data pengujian lantas dijadikan data *train* ataupun sebaliknya, data *train* sebelumnya dijadikan kini menjadi data *test*. Berikut sintaks yang digunakan:

```
#Dimensi Train
dim(train)
ChickWeight.KNN<-knn(train[,-4],test[,-4],train[,4],k=4)
ChickWeight.KNN
(table(ChickWeight.KNN,test[,4]))
```

```
> (table(ChickWeight.KNN,test[,4]))

ChickWeight.KNN  1  2  3  4
              1 37  2  0  0
              2  0 22  5  0
              3  0  3 21  0
              4  0  0  4 21
```

Berdasarkan *crosstab validation* diatas dapat dilihat bahwa hasil diet level 1 benar sebesar 37 dan mengalami kesalahan sebesar 2, diet level 2 benar sebesar 22 dan mengalami kesalahan sebesar 5, diet level 3 benar

sebesar 21 dan mengalami kesalahan sebesar 3, dan diet level 4 benar sebesar 21 dan mengalami kesalahan sebesar 4.



Untuk mendapatkan hasil visualisasi KNN seperti pada gambar 1.3 maka dapat menggunakan perintah :

```
pairs(test[,1:3],pch=as.character(test[,4]),col=c(1,4)[(test$Diet!=ChickWeight.KNN)+1])
```

Visualisasi seperti gambar diatas merupakan visualisasi *pairs plot* atau yang biasa dikenal dengan matrik *scatterplot*. Matriks *scatterplot* adalah cara yang bagus untuk menentukan secara kasar apakah terdapat korelasi linier antara beberapa variabel. Ini sangat membantu dalam menentukan dengan tepat variabel-variabel tertentu yang mungkin memiliki korelasi serupa dengan data *genom* atau *proteomic*.

Berdasarkan gambar 1.3 dapat dilihat bahwa terdapat cerimanan *scatterplot* pada garis diagonal. Jika dilihat bahwa terdapat korelasi linier pada variabel *weight* dan *time* karena dapat dilihat bahwa plot nya membentuk seperti garis. Sedangkan pada variabel *weight* dengan *chick,* dan *time* dengan *chinck* tidak terdapat korelasi linier karena terlihat penyebaran data (level diet).

C. Latihan Praktikum

Data yang digunakan dalam Praktikum kali ini adalah **Pima Indians Diabetes**
Database Sumber: https://www.jair.org/index.php/jair/article/view/10129

Langkah 1: Memanggil Package yang digunakan

Pada praktikum ini akan digunakan `ggplot2`, `caret`, `class`, `mvtnorm`, `MASS`, dan `gridExtra`.

```
library(caret)
library(class)
library(ggplot2)
```

Langkah 2: Import & Explor Data

```
# alamat <- "ganti alamat dengan lokasi tempat data disimpan"
alamat <- 'D:/pima-indians-diabetes.csv'
diabetes <- read.csv(alamat, stringsAsFactors = TRUE)
str(diabetes)
## 'data.frame':    768 obs. of  9 variables:
##  $ preg : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ plas : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ pres : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ skin : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ insu : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ mass : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ pedi : num  0.627 0.351 0.672 0.167 2.288 ...
##  $ age  : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ class: Factor w/ 2 levels "tested_negative",..: 2 1 2 1 2 1 2 1 2 2 ...
diabetes$class <- as.factor(diabetes$class)
```

```
table(diabetes$class)
```

```
##
## tested_negative tested_positive
##             500             268
```

```
mean(diabetes$age)
```

```
## 33.24089
```

```
summary(diabetes$age)
```

```
     preg             plas             pres
Min.   : 0.000   Min.   :  0.0   Min.   :  0.00
1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00
Median : 3.000   Median :117.0   Median : 72.00
Mean   : 3.845   Mean   :120.9   Mean   : 69.11
3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00
Max.   :17.000   Max.   :199.0   Max.   :122.00
     skin            test            mass
Min.   : 0.00   Min.   :  0.0   Min.   : 0.00
1st Qu.: 0.00   1st Qu.:  0.0   1st Qu.:27.30
Median :23.00   Median : 30.5   Median :32.00
Mean   :20.54   Mean   : 79.8   Mean   :31.99
3rd Qu.:32.00   3rd Qu.:127.2   3rd Qu.:36.60
Max.   :99.00   Max.   :846.0   Max.   :67.10
     pedi             age        class
Min.   :0.0780   Min.   :21.00   0:500
1st Qu.:0.2437   1st Qu.:24.00   1:268
Median :0.3725   Median :29.00
Mean   :0.4719   Mean   :33.24
3rd Qu.:0.6262   3rd Qu.:41.00
Max.   :2.4200   Max.   :81.00
```

Langkah 3: k-Nearest Neighbor (k-NN)

k-NN melakukan klasifikasi berdasarkan k tetangga terdekat, sehingga sangat tergantung pada jarak. Sehingga, jika skala dan rentang peubah yang digunakan berbeda-beda, perlu melakukan standarisasi terhadap peubah tersebut. Umumnya Standarisasi yang paling sering digunakan adalah Standarisasi [0,1].

```
train=diabetes[1:500,]

test=diabetes[501:768,]

pred_test=knn(train[,-9],test[,-9],train$class,k=2)

pred_test
```

```
  [1] 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0
 [31] 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 0
 [61] 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 0 1 1 0 0 0 1 0
 [91] 1 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 1 1 0 1 0 0 0 1 0
[121] 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0
[151] 0 0 0 1 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1
[181] 0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 1 1 1 0 1 1 0 1 1 0 0 0 0 1 0
[211] 1 1 1 0 1 1 1 0 1 1 0 1 0 0 0 1 0 1 1 0 0 0 1 0 1 0 0 0 0 1
[241] 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0 0 0
Levels: 0 1
```

confusion=table(pred_test,test$class)

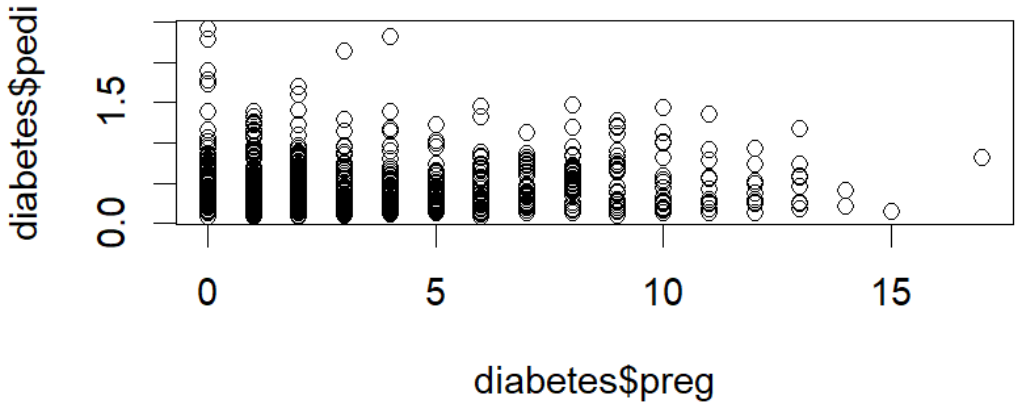
sum(diag(confusion))/nrow(test)

```
[1] 0.6716418
```

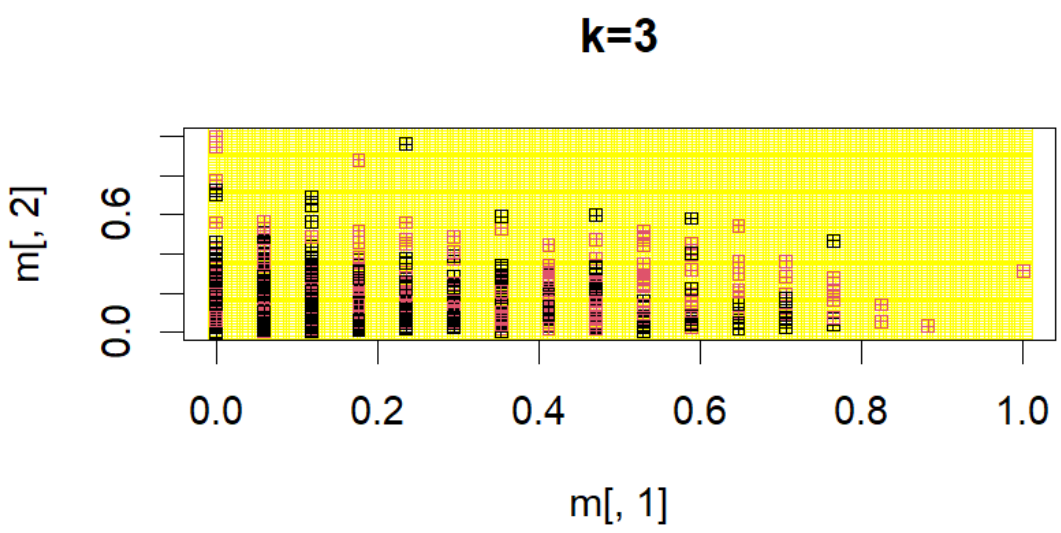
confusionMatrix(pred_test,test$class)

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 129  35
         1  53  51

               Accuracy : 0.6716
                 95% CI : (0.6119, 0.7276)
    No Information Rate : 0.6791
    P-Value [Acc > NIR] : 0.63082

                  Kappa : 0.286

 Mcnemar's Test P-Value : 0.06995

            Sensitivity : 0.7088
            Specificity : 0.5930
         Pos Pred Value : 0.7866
         Neg Pred Value : 0.4904
             Prevalence : 0.6791
         Detection Rate : 0.4813
   Detection Prevalence : 0.6119
      Balanced Accuracy : 0.6509

       'Positive' Class : 0
```

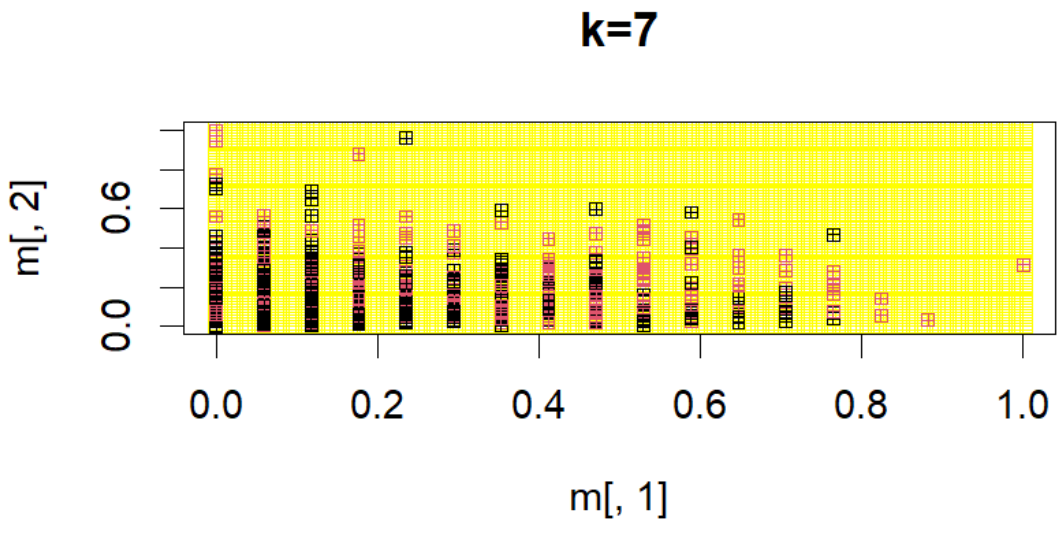
plot(diabetes$preg,diabetes$pedi)



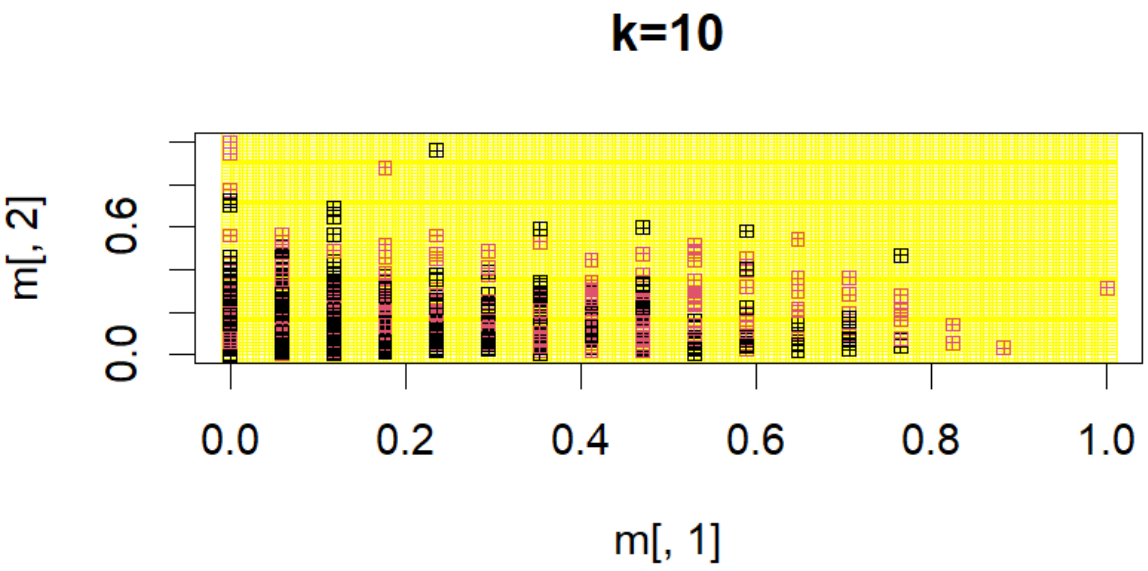

```
#standardize
stdmaxmin <- function(X) (X-min(X))/(max(X)-min(X))
preg1 <- stdmaxmin(diabetes$preg)
pedi1 <- stdmaxmin(diabetes$pedi)

m <-NULL; a <-b <-seq(0, 1, length.out = 70)
for (i in a) for (j in b) m <-rbind(m, c(i, j))
#k=3
prediksi<-knn(cbind(preg1,pedi1), m, diabetes$class, k = 3)
plot(m[,1], m[,2], col=ifelse(prediksi=="tested_positive", "cyan","yellow"),
    pch=ifelse(prediksi=="tested_positive",17,12), main="k=3")
points(preg1, pedi1, col=diabetes$class,
      pch=ifelse(diabetes$class=="tested_positive",17,12), cex=.7)
```




```
#k=7
prediksi<-knn(cbind(preg1,pedi1), m, diabetes$class, k = 7)
plot(m[,1], m[,2], col=ifelse(prediksi=="tested_positive", "cyan","yellow"),
     pch=ifelse(prediksi=="tested_positive",17,12), main="k=7")
points(preg1, pedi1, col=diabetes$class,
       pch=ifelse(diabetes$class=="tested_positive",17,12), cex=.7)
```



```
#k=10
prediksi<-knn(cbind(preg1,pedi1), m, diabetes$class, k = 10)
plot(m[,1], m[,2], col=ifelse(prediksi=="tested_positive", "cyan","yellow"),
     pch=ifelse(prediksi=="tested_positive",17,12), main="k=10")
points(preg1, pedi1, col=diabetes$class,
       pch=ifelse(diabetes$class=="tested_positive",17,12), cex=.7)
```



Dari hasil visualisasi dengan k yang berbeda-beda, terlihat hasilnya juga berbeda-beda. Semakin besar k, maka luasan wilayah akan semakin besar dan jarak batas umumnya lebih jauh dari data. Namun jika dicermati lebih jauh, semakin banyak juga data yang diklasifikasikan berbeda.