Departemen Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Pertanian Bogor

PEMODELAN KLASIFIKASI

PERTEMUAN #7

DISKRETISASI

Bagus Sartono
bagusco@ipb.ac.id bagusco@gmail.com

Outline

Pendahuluan dan Motivasi

Pengelompokan Metode Diskretisasi

Unsupervised Discretization

Supervised Discretization

Pendahuluan

Data terdiri atas banyak variabel dengan berbagai format/tipe:

- Numerik diskret
- Numerik kontinu
- Kategorik ordinal
- Kategorik nominal

Variabel yang bertipe numerik dapat diubah menjadi kategorik (ordinal) → prosesnya dikenal sebagai diskretisasi, ada juga yang menyebut sebagai binning

Diskretisasi ini sering membantu dalam pemodelan prediktif

Diskretisasi

Andaikan dataset berisi N observasi, proses diskretiasi terhadap variabel numerik A adalah mengubah nilai variabel tersebut menjadi m interval $D = \{[d_0, d_1], (d_1, d_2], \ldots, (d_{m-1}, d_m]\}$, dengan d_0 adalah nilai terkecil, d_m adalah nilai terbesar, dan $d_i < d_{i+i}$, untuk $i = 0, 1, \ldots, m-1$.

Diskretisasi

6.58
15.35
14.24
6.22
1.82
2.11
13.77
5.65
15.58
12.46
13.05
11.64
10.91
14.31
7.42

$X \leq 5$

$5 < X \leq 10$

$10 < X \leq 15$

$X > 15$

Motivasi

- Without data discretization some rules would be difficult to establish.
- Several existing data mining systems cannot handle continuous variables without discretization.
- Data discretization significantly improves the quality of the discovered knowledge.
- New methods of discretization needed for tables with rare events.
- Data discretization significantly improves the performance of data mining algorithms. Some studies reported ten fold increase in performance. However:

- *Any discretization process generally leads to a loss*
- *of information. Minimizing such a possible loss is the mark*
- *of **good** discretization method.*

- Diskretisasi
  - Merging
    - Supervised
      - ChiMerge
  - Splitting
    - Un-Supervised
      - Equal Width
      - Equal Freq
    - Supervised
      - Tree/MDLP

UNSUPERVISED DISCRETIZATION

Equal Width dan Equal Frequency

In equal width, the continuous range of a feature is divided into intervals that have an equal width and each interval represents a bin. The arity can be calculated by the relationship between the chosen width for each interval and the total length of the attribute range.

In equal frequency, an equal number of continuous values are placed in each bin. Thus, the width of each interval is computed by dividing the length of the attribute range by the desired arity.

```
x <- c(15, 4, 21, 11, 16, 18, 24, 26, 28)
library(classInt)

#equal width
eqwid <- classIntervals(x, 4, style = 'equal')
eqwid$brks
```

```
> eqwid$brks
[1]  4 10 16 22 28
```

```
x.eqwid <- cut(x, breaks=eqwid$brks, include.lowest=TRUE)
cbind(x, x.eqwid)
```

```
> cbind(x, x.eqwid)
      x x.eqwid
[1,] 15       2
[2,]  4       1
[3,] 21       3
[4,] 11       2
[5,] 16       2
[6,] 18       3
[7,] 24       4
[8,] 26       4
[9,] 28       4
```

```
#equal freq
eqfreq <- classIntervals(x, 4, style = 'quantile')
eqfreq$brks
```

```
> eqfreq$brks
[1]  4 15 18 24 28
```

```
x.eqfreq <- cut(x, breaks=eqfreq$brks, include.lowest=TRUE)
cbind(x, x.eqwid, x.eqfreq)
```

```
> cbind(x, x.eqwid, x.eqfreq)
      x x.eqwid x.eqfreq
[1,] 15       2        1
[2,]  4       1        1
[3,] 21       3        3
[4,] 11       2        1
[5,] 16       2        2
[6,] 18       3        2
[7,] 24       4        3
[8,] 26       4        4
[9,] 28       4        4
```

Ilustrasi efek diskretisasi terhadap kualitas model prediktif

Akan dipaparkan situasi dimana diskretisasi mampu memberikan peningkatan akurasi prediksi pada model regresi logistik

Akan dibandingkan akurasi dua model dengan data yang sama

- Model pertama menggunakan variabel prediktor asli
- Model kedua menggunakan variabel prediktor yang telah didiskretkan

```
data <- read.csv("D:/disk01.csv", header=TRUE)
head(data)
```

Ilustrasi efek diskretisasi terhadap kualitas model prediktif

Pemodelan Regresi Logistik dengan X asli

- model.asli <- glm(class ~ x, data=data, family="binomial")
- maudiprediksi <- data.frame(data$x)
- colnames(maudiprediksi) <- c("x")
- prediksi.prob.asli <- predict(model.asli, newdata=maudiprediksi, type="response")
- prediksi.asli <- ifelse(prediksi.prob.asli > 0.5, 1, 0)
- table(data$class, prediksi.asli)
- mean(data$class == prediksi.asli)

Ilustrasi efek diskretisasi terhadap kualitas model prediktif

- eqwid <- classIntervals(data$x, 10, style = 'equal')
- x.eqwid <- cut(data$x, breaks=eqwid$brks, include.lowest=TRUE)

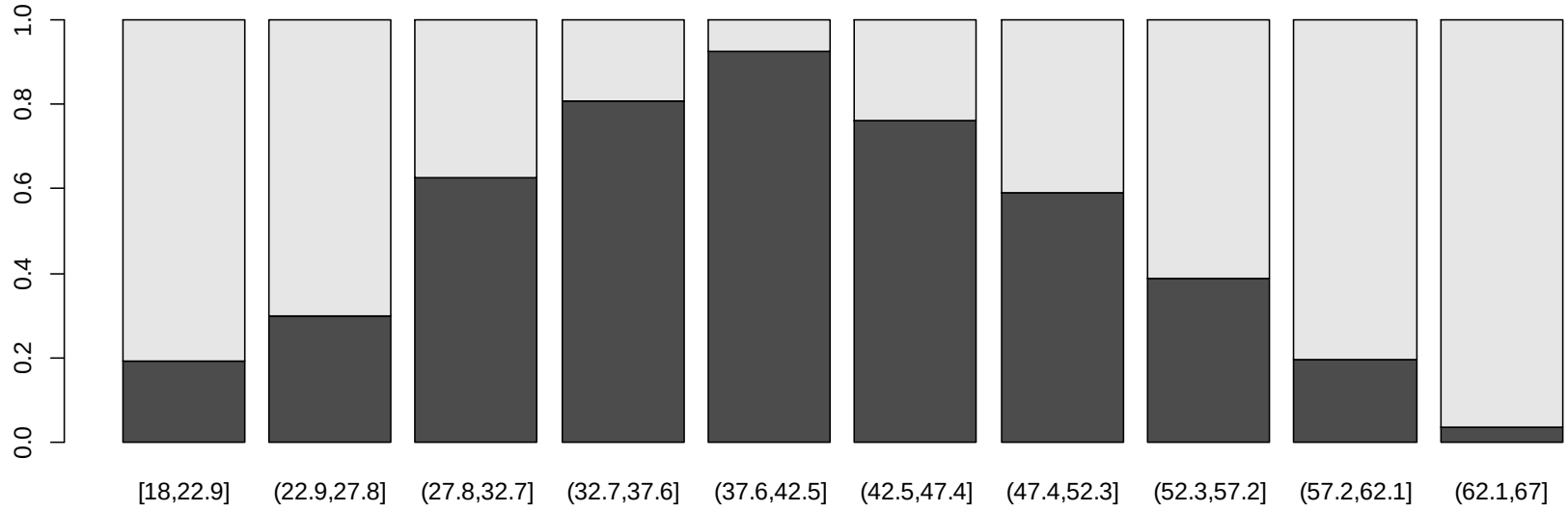
Pemodelan Regresi Logistik dengan X yang sudah didiskretkan

- model.disk <- glm(data$class ~ x.eqwid, family="binomial")
- prediksi.prob.disk <- predict(model.disk, newdata=x.eqwid, type="response")
- prediksi.disk <- ifelse(prediksi.prob.disk> 0.5, 1, 0)
- table(data$class, prediksi.disk)
- mean(data$class == prediksi.disk)

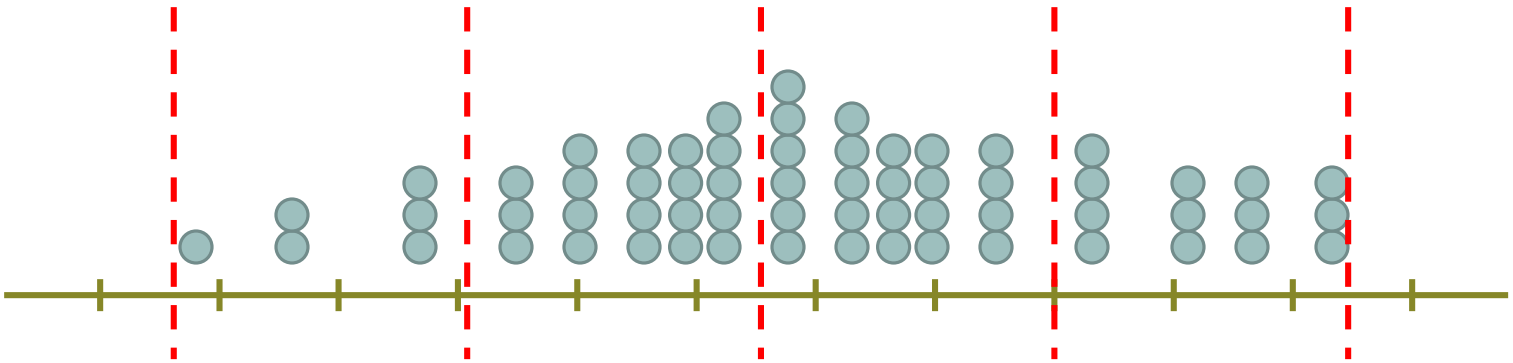
```
> table(data$class, prediksi.asli)
   prediksi.asli
      0   1
  0 350 386
  1 312 481
> mean(data$class == prediksi.asli)
[1] 0.5434925
```

```
> table(data$class, prediksi.disk)
   prediksi.disk
      0   1
  0 569 167
  1 210 583
> mean(data$class == prediksi.disk)
[1] 0.7534336
```

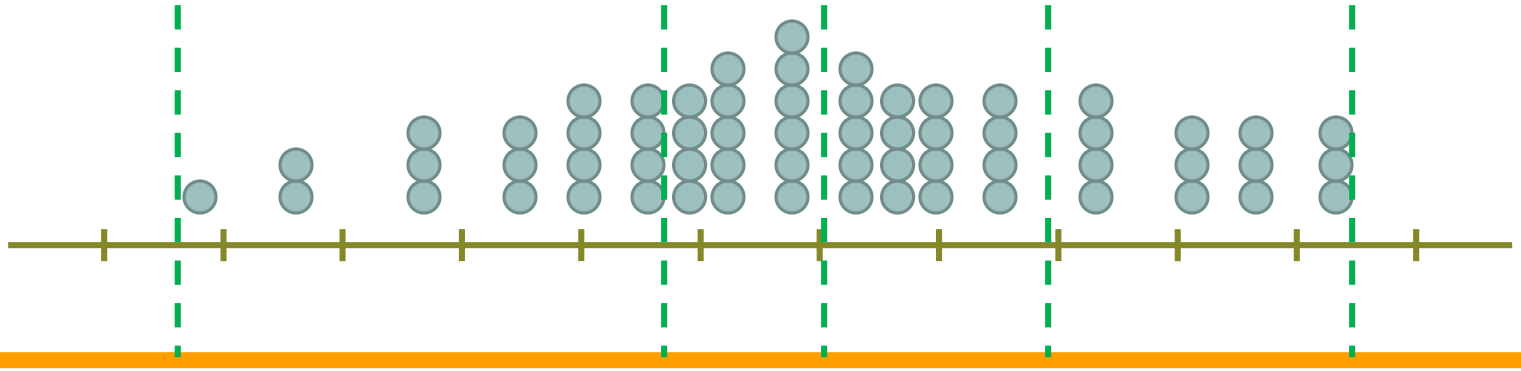
```
table(x.eqwid, data$class)
prop.table(table(x.eqwid, data$class), margin=1)

proporsi <- prop.table(table(x.eqwid, data$class), margin=1)
barplot(t(proporsi))
```



Unsupervised Discretization: Equal Width Discretization



Unsupervised Discretization: Equal Freq Discretization



SUPERVISED DISCRETIZATION

Description of Representative Methods

Splitting

Algorithm 1 Splitting Algorithm

```
Require: S = Sorted values of attribute A
  procedure SPLITTING(S)
      if StoppingCriterion() == true then
          Return
      end if
      T = GetBestSplitPoint(S)
      S1 = GetLeftPart(S,T)
      S2 = GetRightPart(S,T)
      Splitting(S1)
      Splitting(S2)
  end procedure
```

Splitting

Description of Representative Methods

MDLP — This discretizer uses the entropy measure to evaluate candidate cut points. Entropy is one of the most commonly used discretization measures in the literature. The entropy of a sample variable *X* is

$$H(X) = -\sum_{x} p_x \log p_x$$

where x represents a value of *X* and p_x its estimated probability of occurring.

Splitting

Description of Representative Methods

MDLP — Information is high for lower probable events and low otherwise. This discretizer uses the *Information Gain* of a cut point, which is defined as

$$G(A, T; S) = H(S) - H(A, T; S) = H(S) - \frac{|S_1|}{N} H(S_1) - \frac{|S_2|}{N} H(S_2)$$

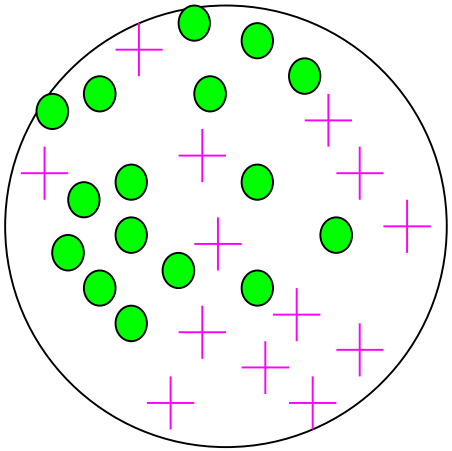
where *A* is the attribute in question, *T* is a candidate cut point and *S* is the set of *N* examples. So, S_i is a partitioned subset of examples produced by *T*. The MDLP discretizer applies the *Minimum Description Length Principle* to decide the acceptation or rejection for each cut point and to govern the stopping criterion.

$$G(A, T; S) > \frac{\log_2(N-1)}{N} + \frac{\delta(A, T; S)}{N}$$

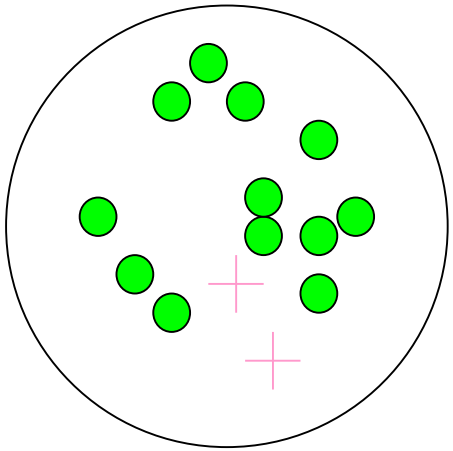
where $\delta(A, T; S) = \log_2(3^c - 2) - [c \cdot H(S) - c_1 \cdot H(S_1) - c_2 \cdot H(S_2)]$

Entropy

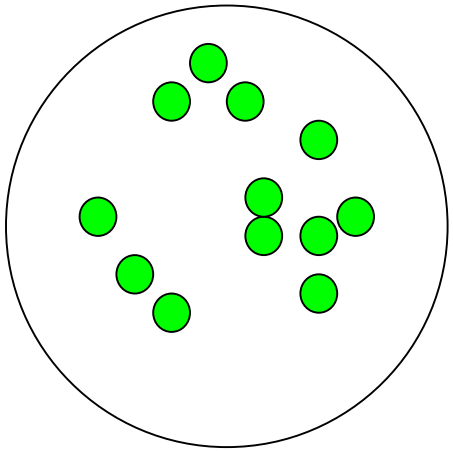
Very impure group



Less impure



Minimum impurity



Entropy

Entropy is used in information theory to measure the amount of information stored in a given number of bits.

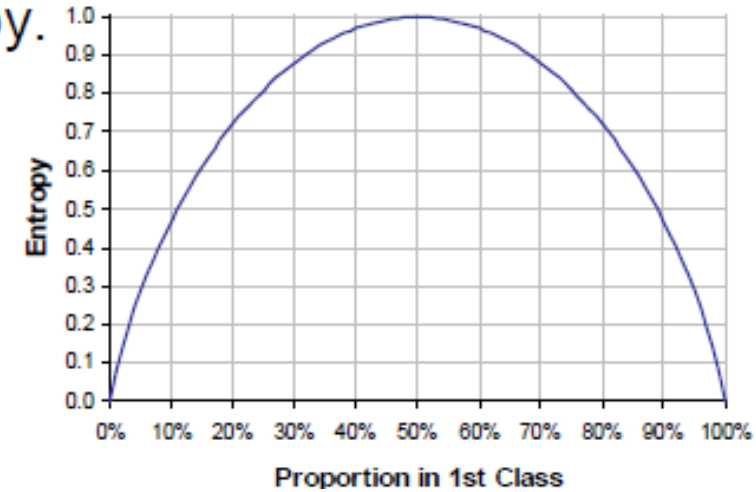
A pure population has an entropy of 0.

If there are two groups equally represented, then the entropy is 1.

The calculation for entropy is shown here:

$$-1*(p_1\log_2(p_1) + p_2\log_2(p_2)).$$

The goal is to minimize entropy.



Entropy

```
> proporsi <- prop.table(table(data$class))
> entropy = -proporsi[1] * log(proporsi[1]) - proporsi[2] * log(proporsi[2])
> entropy
       0
0.6924521
```

```
> library(discretization)
> ent(data$class)
[1] 0.6924521
```

Information Gain

Mana yang lebih baik?

- $X < 35$ vs $X \geq 35$

atau

- $X < 40$ vs $X \geq 40$

Information Gain

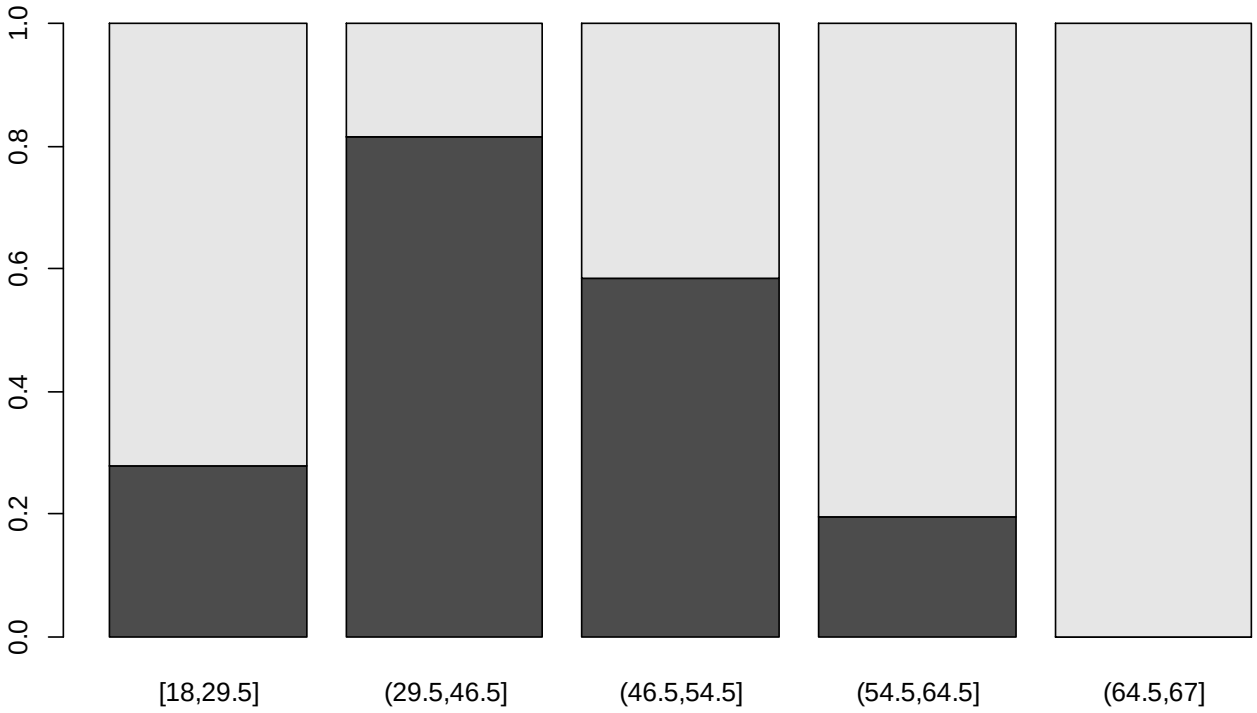
```
ent.total <- ent(data$class)

ent.35.1 <- ent(data$class[data$x < 35])
ent.35.2 <- ent(data$class[data$x >= 35])

ent.40.1 <- ent(data$class[data$x < 40])
ent.40.2 <- ent(data$class[data$x >= 40])

ig.35 <- ent.total -
          ( length(data$class[data$x < 35])/length(data$class) * ent.35.1 +
            length(data$class[data$x >= 35])/length(data$class) * ent.35.2)
ig.40 <- ent.total -
          ( length(data$class[data$x < 40])/length(data$class) * ent.40.1 +
            length(data$class[data$x >= 40])/length(data$class) * ent.40.2)
```

```
> ig.35
[1] 0.002782724
> ig.40
[1] 0.003429511
```

```
> data <- read.csv("D:/disk01.csv", header=TRUE)
> head(data)
  x class
1 51     0
2 19     1
3 66     1
4 35     0
5 64     1
6 48     1
>
> library(discretization)
> ##---- MDLP discretization ----
> mdlp <- mdlp(data)
> mdlp$cutp
[[1]]
[1] 29.5 54.5 46.5 64.5
```

```
> x.mdlp <- mdlp$Disc.data$x
> table(x.mdlp)
x.mdlp
  1   2   3   4   5
344 507 280 315  83
```

```
> proporsi <- prop.table(table(x.mdlp, data$class), margin=1)
> barplot(t(proporsi))
>
```



Hasil Reg Logistik setelah Diskretisasi MDLP

```
> model.mdlp <- glm(data$class ~ x.mdlp, family="binomial")
> prediksi.prob.mdlp <- predict(model.mdlp, newdata=x.mdlp, type="response")
> prediksi.mdlp <- ifelse(prediksi.prob.mdlp > 0.5, 1, 0)
> table(data$class, prediksi.mdlp)
   prediksi.mdlp
      0   1
  0 578 158
  1 209 584
> mean(data$class == prediksi.mdlp)
[1] 0.7599738
```

Recall… ini akurasi tanpa diskretisasi

0.5434925

Description of Representative Methods

Merging

Algorithm 2 Merging Algorithm

```
Require: S = Sorted values of attribute A
  procedure MERGING(S)
      if StoppingCriterion() == true then
          Return
      end if
      T = GetBestAdjacentIntervals(S)
      S = MergeAdjacentIntervals(S,T)
      Merging(S)
  end procedure
```

Description of Representative Methods

Merging

ChiMerge — χ^2 is a statistical measure that conducts a significance test on the relationship between the values of an attribute and the class. This statistic determines the similarity of adjacent intervals based on some significance level. Actually, it tests the hypothesis that two adjacent intervals of an attribute are independent of the class. χ^2 is computed as:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{c} \frac{(N_{ij} - E_{ij})^2}{E_{ij}}$$

Description of Representative Methods

Merging

ChiMerge —

$$c = \text{ number of classes}$$

$$N_{ij} = \text{ number of distinct values in the } i\text{th interval, } j\text{th class}$$

$$R_i = \text{ number of examples in } i\text{th interval } = \sum_{j=1}^{c} N_{ij}$$

$$C_j = \text{ number of examples in } j\text{th class } = \sum_{i=1}^{m} N_{ij}$$

$$N = \text{ total number of examples } = \sum_{j=1}^{c} C_j$$

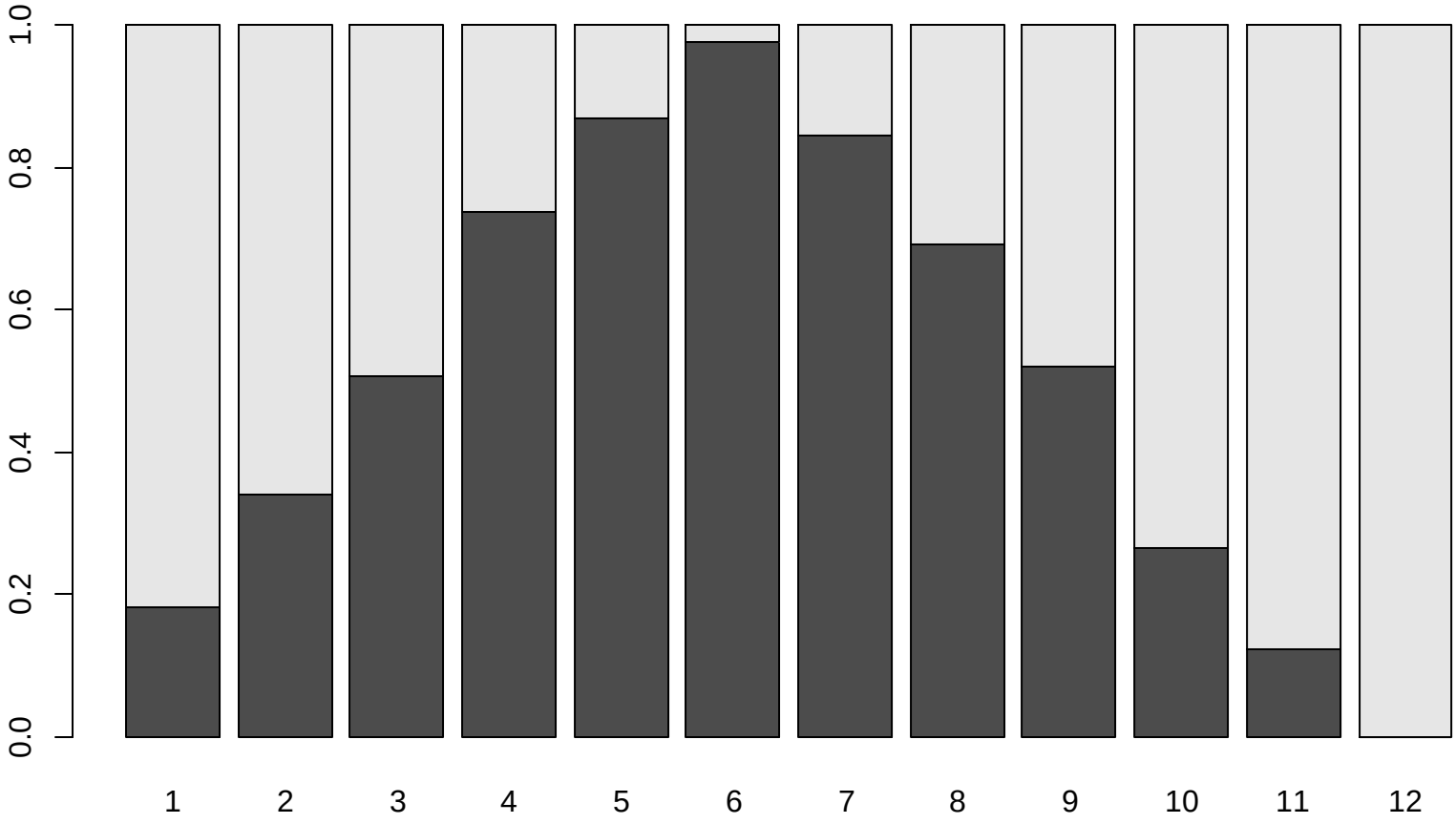
$$E_{ij} = \text{ expected frequency of } N_{ij} = (R_i \times C_j)/N$$

It is a supervised, bottom-up discretizer. At the beginning, each distinct value of the attribute is considered to be one interval. **χ^2** tests are performed for every pair of adjacent intervals. Those adjacent intervals with the least **χ^2** value are merged until the chosen stopping criterion is satisfied.

```
> data <- read.csv("D:/disk01.csv", header=TRUE)
> head(data)
   x class
1 51     0
2 19     1
3 66     1
4 35     0
5 64     1
6 48     1
> disk.chi <- chiM(data,0.05)
> disk.chi$cutp
[[1]]
 [1] 23.5 27.5 30.5 33.5 38.5 40.5 45.5 49.5 54.5
59.5 64.5
```

```
> x.chim <- disk.chi$Disc.data$x
> table(x.chim)
x.chim
  1   2   3   4   5   6   7   8   9  10  11  12
175 115  87 110 159  45 123 159 158 162 153  83
```

```
> proporsi <- prop.table(table(x.chim, data$class), margin=1)
> barplot(t(proporsi))
```



```
> model.chi <- glm(data$class ~ as.factor(x.chim), family="binomial")
> prediksi.prob.chi <- predict(model.chi, newdata= as.factor(x.chim), type="response")
> prediksi.chi <- ifelse(prediksi.prob.chi > 0.5, 1, 0)
> table(data$class, prediksi.chi)
   prediksi.chi
      0   1
  0 603 133
  1 238 555
> mean(data$class == prediksi.chi)
[1] 0.7573578
```

Recall… ini akurasi tanpa diskretisasi

0.5434925

terima kasih