UTS I: TAKE HOME

Penambangan Data dalam Sains

Mohammad Rizka Fadhli - 20921004

02 April 2022

SOAL

Anda diberikan klasifikasi `Iris-Setosa` dalam *file* `csv`. Gunakan materi yang telah diberikan dalam kuliah dan praktikum untuk membuat model pembelajaran dengan metode **pilih salah satu**: *Naive Bayes*, *KNN*, *decision tree* dan *neural network*!

JAWAB

Catatan

Pada UTS *take home* ini, saya akan menggunakan metode *decision tree*. Untuk melakukannya, saya menggunakan **R** dengan `library(caret)`.

Decision Tree

Pengertian

Proses pada *decision tree* adalah mengubah bentuk data berupa tabel menjadi bentuk *tree* yang berisi *rules* yang lebih sederhana. Ada berbagai macam algoritma yang bisa digunakan untuk membuatnya, seperti: `CART`, `ID3`, `C4.5`, `CHAID`, dan sebagainya.

Algoritma `C4.5`

Berikut adalah algoritma `C4.5` yang bisa digunakan untuk membuat *decision tree*:

1. Memilih atribut sebagai akar.
2. Membuat cabang dari setiap nilai yang muncul.

3. Membagi kasus per cabang.
4. Proses di atas diulang hingga semua cabang selesai.

Perhitungan *entropy* adalah sebagai berikut:

$$entropy = \sum_{i=1}^{n} -P_i log_2 P_i$$

di mana:

$$S : \text{himpunan kasus}$$
$$n : \text{jumlah partisi } S$$
$$P_i : \text{proporsi dari } S_i terhadap S$$

Perhitungan *gain* untuk masing-masing atribut:

$$gain(S, A) = entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{S} entropy(S_i)$$

di mana:

$$S : \text{himpunan kasus}$$
$$A : \text{atribut}$$
$$n : \text{jumlah partisi } S$$
$$|S_i| : \text{jumlah kasus pada partisi ke } i$$
$$S_i : \text{jumlah kasus dalam} S$$

Jawaban

Berikut adalah proses yang dilakukan untuk membuat *decision tree*.

Import Data

Langkah pertama yang harus dilakukan adalah *import* data `iris.csv`. Berikut adalah *sample* 10 data dari `iris.csv`:

```
data = read.csv("iris.csv")
data %>% head(10) %>% knitr::kable(caption = "Sample 10 Data Teratas")
```

Table 1: Sample 10 Data Teratas

sepallength	sepalwidth	petallength	petalwidth	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5.0	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa

Berikut adalah struktur data yang ada:

```
data %>% str()
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ sepallength: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ sepalwidth : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ petallength: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ petalwidth : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ class      : chr  "Iris-setosa" "Iris-setosa" "Iris-setosa" "Iris-setosa" ...
```

Dari informasi di atas, kita temukan ada `4` atribut numerik yakni:

1. `sepallength`
2. `sepalwidth`
3. `petallength`
4. `petalwidth`

yang akan digunakan untuk memprediksi (klasifikasi) atribut `class`.

Analisa Deskriptif Data Iris

Sebelum kita membuat model *decision tree*, mari kita lihat terlebih dahulu analisa deskriptif dari data yang ada:

```
data %>% summary()
```

```
##   sepallength      sepalwidth     petallength      petalwidth
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##     class
##  Length:150
##  Class :character
##  Mode  :character
##
##
##
```

Ada `150` baris data.

Berikut adalah proporsi dari `class`

Table 2: Proporsi dari class

class	n
Iris-setosa	50
Iris-versicolor	50
Iris-virginica	50

Dapat dilihat bahwa semua `class` tersebar proporsional.

Sebaran Data

Berikut adalah sebaran data dari masing-masing atribut numerik terhadap `class`:

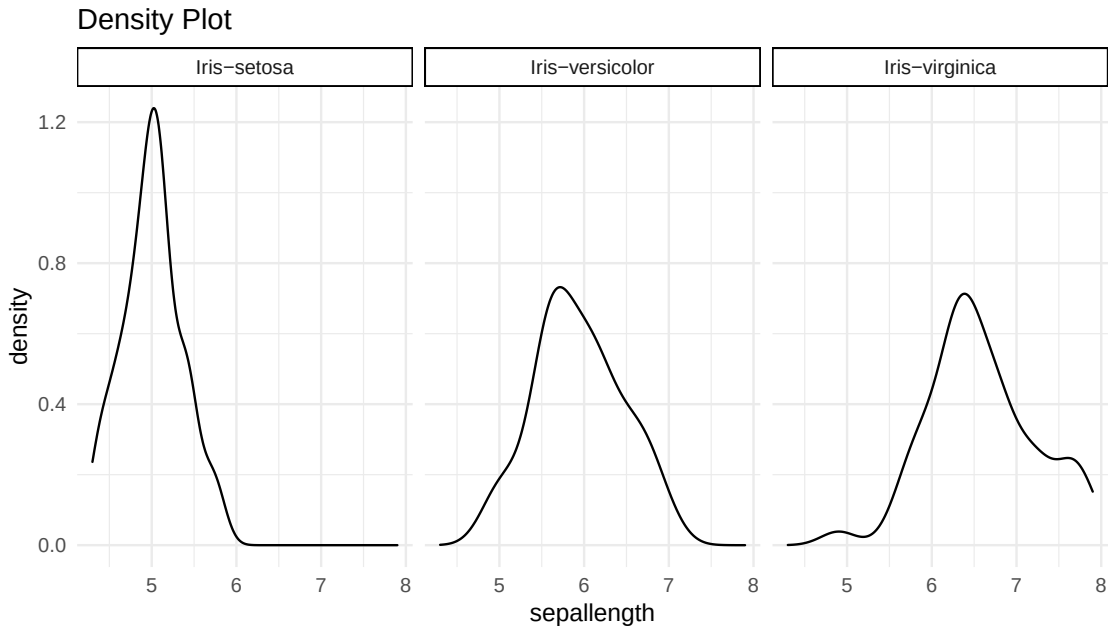


Figure 1: Sebaran Data sepallength Pada Setiap class

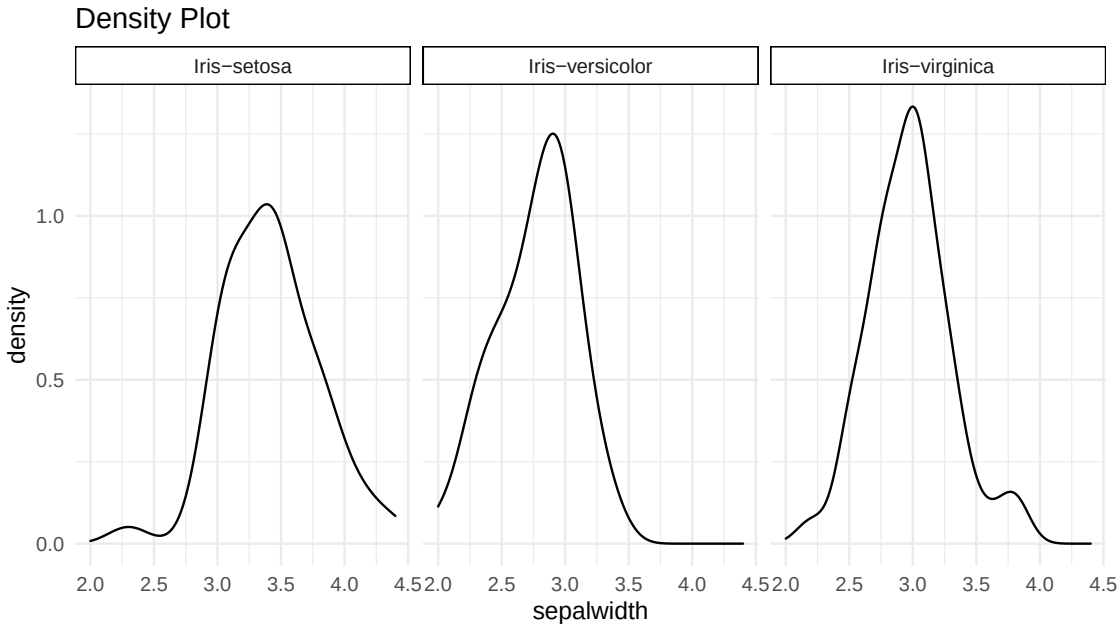


Figure 2: Sebaran Data sepalwidth Pada Setiap class

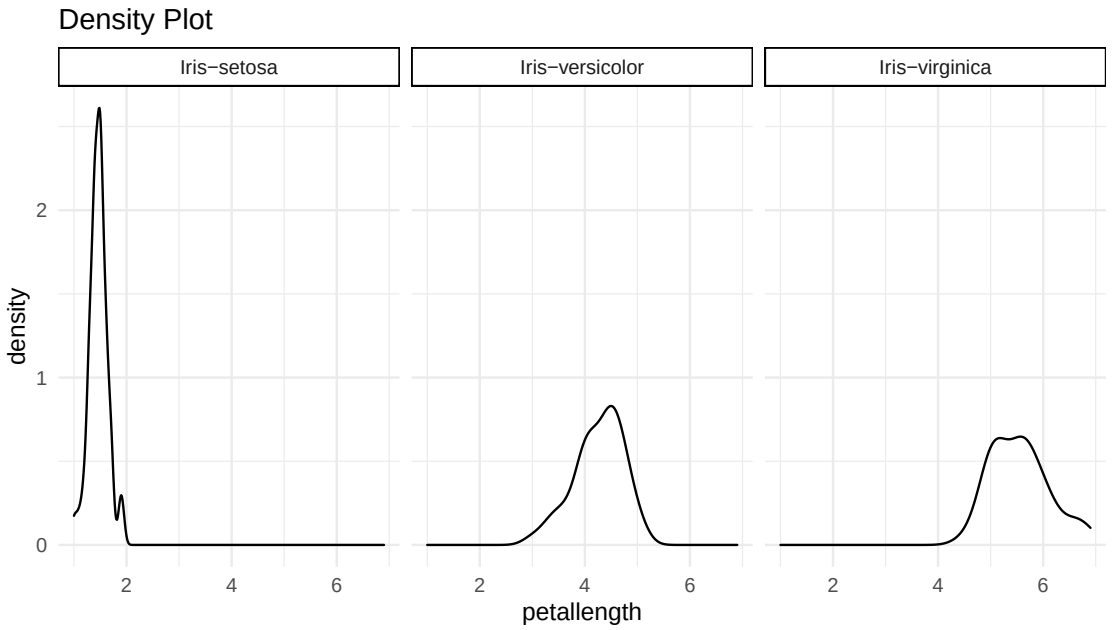


Figure 3: Sebaran Data petallength Pada Setiap class

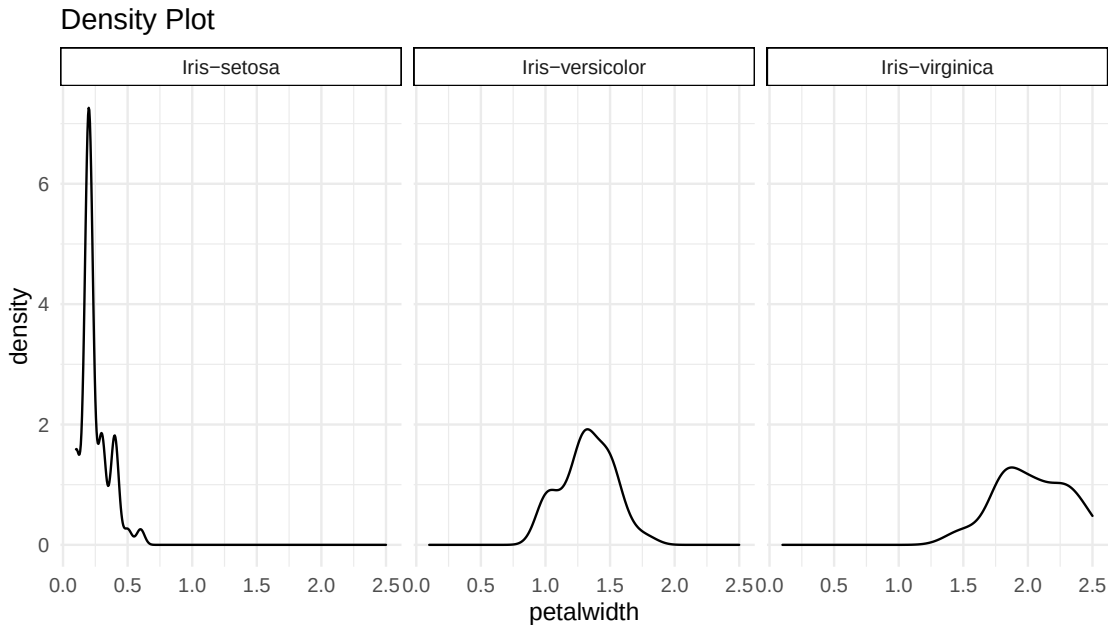


Figure 4: Sebaran Data petalwidth Pada Setiap class

Pre-Processing

Pre-processing yang akan dilakukan adalah memecah data menjadi dua *datasets*, yakni:

1. *Train dataset*
2. *Test dataset*

dengan proporsi `80-20`. Pembagian ini akan dilakukan secara acak:

```
set.seed(20921)
# random id train
id_train = sample(150,120,replace = F)
# membuat train dataset
train_df = data[id_train,]
# membuat test dataset
test_df = data[-id_train,]
```

Berikut adalah proporsi `class` pada *train* dan *test dataset*.

Table 3: Proporsi dari class pada Train Dataset

class	n
Iris-setosa	39

class	n
Iris-versicolor	41
Iris-virginica	40

Table 4: Proporsi dari class pada Test Dataset

class	n
Iris-setosa	11
Iris-versicolor	9
Iris-virginica	10

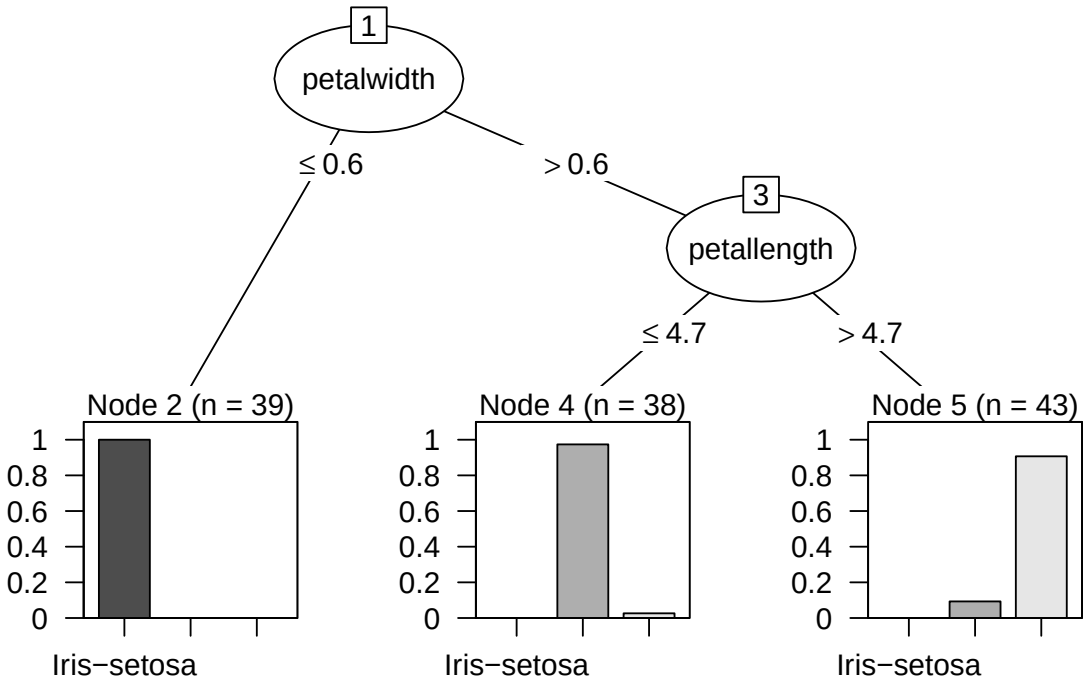
Klasifikasi *Decision Tree*

Berikut adalah proses pembuatan *decision tree*:

```
model_dt = caret::train(factor(class) ~ ., data = train_df, method="J48")
model_dt$finalModel
```

```
## J48 pruned tree
## ------------------
##
## petalwidth <= 0.6: Iris-setosa (39.0)
## petalwidth > 0.6
## |   petallength <= 4.7: Iris-versicolor (38.0/1.0)
## |   petallength > 4.7: Iris-virginica (43.0/4.0)
##
## Number of Leaves  :  3
##
## Size of the tree :   5
```

Berikut adalah *plot* dari modelnya:



Confusion Matrix

Selanjutnya kita akan cek *confusion matrix* menggunakan *test dataset*:

```
# melakukan prediksi dari test dataset
pred = predict(model_dt,newdata = test_df) %>% as.character()

# membuat confusion matrix
table(test_df$class,pred)
```

```
##                  pred
##                   Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              11               0              0
##   Iris-versicolor           0               7              2
##   Iris-virginica            0               0             10
```

Kesimpulan

Terlihat bahwa hanya ada kesalahan misklasifikasi sebanyak 2 kasus saja dari total 30 baris data pada *test dataset*. Sehingga bisa dihitung akurasi dari model *decision tree* ini adalah sebesar:

```
mean(test_df$class == pred)
```

```
## [1] 0.9333333
```