ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
-------oOo-------

TIỂU LUẬN
MÔN: NHẬP MÔN PHÂN TÍCH DỮ LIỆU

ĐỀ TÀI:
XÂY DỰNG MÔ HÌNH DỰ ĐOÁN LOÀI HOA DỰA TRÊN SỐ ĐO CHIỀU DÀI VÀ CHIỀU RỘNG CÁNH HOA VÀ ĐÀI HOA CỦA BỘ DỮ LIỆU IRIS

NHÓM THỰC HIỆN : **NHÓM 1**

GIẢNG VIÊN HƯỚNG DẪN : **VŨ NGỌC BÌNH**

Hà Nội, năm 2022

MỤC LỤC

I. LỜI MỞ ĐẦU

Trong quá trình hoạt động, con người tạo ra nhiều dữ liệu nghiệp vụ. Các tập dữ liệu được tích luỹ có kích thước ngày càng lớn, và có thể chứa nhiều thông tin ẩn dựng những quy luật chưa được khám phá. Chính vì vậy, một nhu cầu đặt ra là cần tìm cách trích rút từ tập dữ liệu đó các luật về phân lớp dữ liệu hay dự đoán những xu hướng dữ liệu tương lai. Những quy tắc nghiệp vụ thông minh được tạo ra sẽ phục vụ đắc lực cho các hoạt động thực tiễn, cũng như phục vụ đắc lực cho quá trình nghiên cứu khoa học. Công nghệ phân lớp và dự đoán dữ liệu ra đời để đáp ứng mong muốn đó.

Công nghệ phân lớp dữ liệu đã, đang và sẽ phát triển mạnh mẽ trước những khao khát tri thức của con người. Trong những năm qua, phân lớp dữ liệu đã thu hút sự quan tâm của các nhà nghiên cứu trong nhiều lĩnh vực như học máy (machine learning), hệ chuyên gia (expert system), thống kê (statistics)... Công nghệ này cũng có tính ứng dụng cao trong nhiều lĩnh vực thực tế như: thương mại, nhà băng, marketing, nghiên cứu thị trường, bảo hiểm, y tế, giáo dục,...

Nhiều kỹ thuật phân lớp đã được đề xuất như: Phân lớp cây quyết định (Decision Trees), phân cụm K-means, mạnh nơron, phân tích thống kê,...

Trong các kỹ thuật đó, decision trees được coi là công cụ mạnh, phổ biến và đặc biệt thích hợp cho data mining. Trong các mô hình phân lớp, thuật toán phân lớp là chủ đạo. Do vậy, cần xây dựng những thuật toán có độ chính xác cao, thực thi nhanh, đi kèm với khả năng mở rộng được để có thể thao tác với những tập dữ liệu ngày càng lớn.

Trong bài tiểu luận này, chúng em sẽ sử dụng mô hình phân lớp Decision Trees và K-means để giải quyết bài toán liên quan đến tập dữ liệu Iris nổi tiếng.

II. NỘI DUNG TIỂU LUẬN

1. Giới thiệu chủ đề

Giới thiệu Iris Dataset

Sepal: Đài hoa

Petal: Cánh hoa

Hình 1

Tập dữ liệu hoa Iris (hoa Diên Vĩ) hoặc tập dữ liệu của Fisher là tập dữ liệu đa biến được giới thiệu bởi nhà thống kê và nhà sinh vật học người Anh Ronald Fisher trong bài báo năm 1936.

Bộ dữ liệu bao gồm 150 mẫu (bản ghi) từ 3 loài Iris (Iris Setosa, Iris Virginica, Iris Versicolor), được thu thập từ kho dữ liệu học máy UCI. Bốn đặc điểm được đo từ mỗi mẫu gồm: chiều dài và chiều rộng của đài hoa, chiều dài và chiều rộng của cánh hoa, tính bằng centimet.

Bộ dữ liệu khi được rút gọn bao gồm 5 thuộc tính: Tên của loài hoa Iris (Iris Setosa, Iris Virginica, Iris Versicolor), chiều dài đài hoa, chiều rộng đài hoa, chiều dài cánh hoa, chiều rộng cánh hoa (*Hình 1*).

2. Đặt vấn đề (bài toán)

Iris (hoa Diên Vĩ) là một loài hoa được rất nhiều người yêu thích hiện nay. Trong văn hoá châu Âu, Diên Vĩ được xem là loài hoa đại diện cho lòng dũng cảm, trung thành và sự khôn ngoan. Vì vậy, loài hoa này được chọn làm biểu tượng của nhiều gia đình

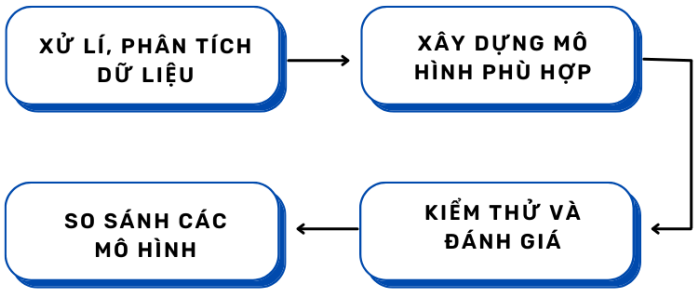
hoàng tộc tại châu Âu. Không chỉ vậy, hoa Diên Vĩ còn được xem là loài hoa của sự may mắn và tình yêu. Do có giá trị cao về mặt truyền thống và kinh tế nên việc phân lớp, dự đoán chính xác loài hoa Iris mang lại nhiều ý nghĩa quan trọng trong thực tiễn.

Trong những năm gần đây, có rất nhiều nhóm nghiên cứu về bài toán phân lớp, dự đoán. Đến nay, có rất nhiều công trình nghiên cứu sử dụng thuật toán học máy, trí tuệ nhân tạo được áp dụng thành công cho bài toán phân lớp, dự đoán.

Trong bài tiểu luận này sẽ trình bày về hai mô hình có thể sử dụng để dự đoán các loài hoa Iris là mô hình Decision Trees và mô hình K-means.

Bài toán: *Xây dựng mô hình dự đoán loài hoa dựa trên số đo chiều dài và chiều rộng cánh hoa và đài hoa của bộ dữ liệu Iris.*

3. Giải quyết bài toán



3.1. Phân tích, xử lí dữ liệu - Exploratory Data Analysis

3.1.1. Làm sạch dữ liệu - Data Cleaning

Việc đảm bảo dữ liệu đã được làm sạch là vô cùng quan trọng. Tuy nhiên, tập dữ liệu Iris là một tập dữ liệu tương đối đơn giản và hoàn chỉnh nên chúng ta có thể bỏ qua bước này.

3.1.2. Thống kê tổng quan dữ liệu

Chúng ta cũng có thể có một số thống kê cơ bản về các trường dữ liệu bằng hàm `summary()`. Kết quả trả về sẽ gồm trung bình, min, max, median, các giá trị tới hạn của phân vị thứ nhất và thứ ba.

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length   Petal.Width
##  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   :3.057   Mean   :3.758   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
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
```

	Chiều dài đài hoa	**Chiều rộng đài hoa**	**Chiều dài cánh hoa**	**Chiều rộng cánh hoa**
Giá trị nhỏ nhất	4,300	2,000	1,000	0,100
Tứ phân vị thứ nhất (Là giá trị mà 25% số liệu nhỏ hơn giá trị đó)	5,100	2,800	1,600	0,300
Trung vị	5,800	3,000	4,350	1,300
Trung bình	5,843	3,057	3,758	1,199
Tứ phân vị thứ ba (Là giá trị mà 75% số liệu nhỏ hơn giá trị đó)	6,400	3,300	5,100	1,800
Giá trị lớn nhất	7,900	4,400	6,900	2,500

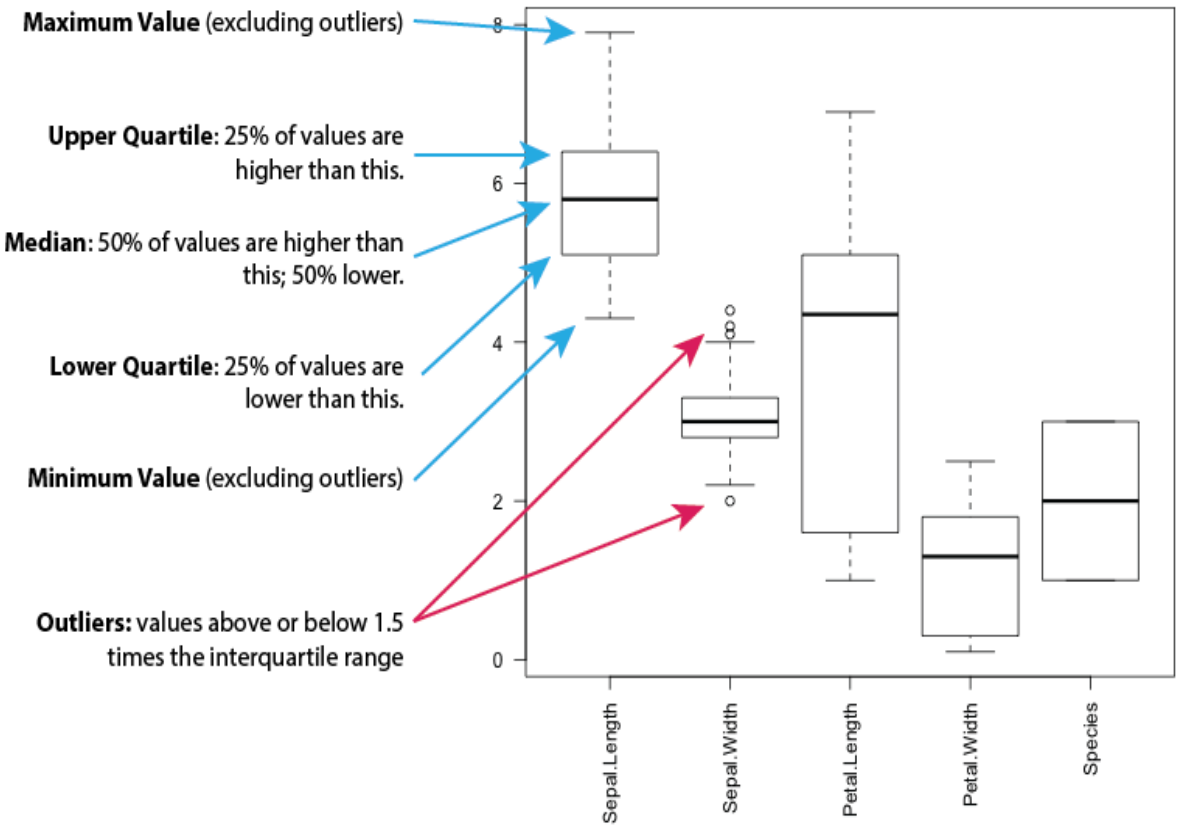
3.1.3. Trực quan hoá dữ liệu và xác định các đặc điểm của dữ liệu

Để có thể trình bày biểu đồ, ta cần ánh xạ tập các thuộc tính vào không gian biểu diễn. Ta thực hiện hai bước:

- Nhận diện kiểu dữ liệu: Kiểu dữ liệu phân loại (Category Data), kiểu dữ liệu số liệu (Metric data).
- Chọn biểu đồ phù hợp với kiểu dữ liệu hiện tại.

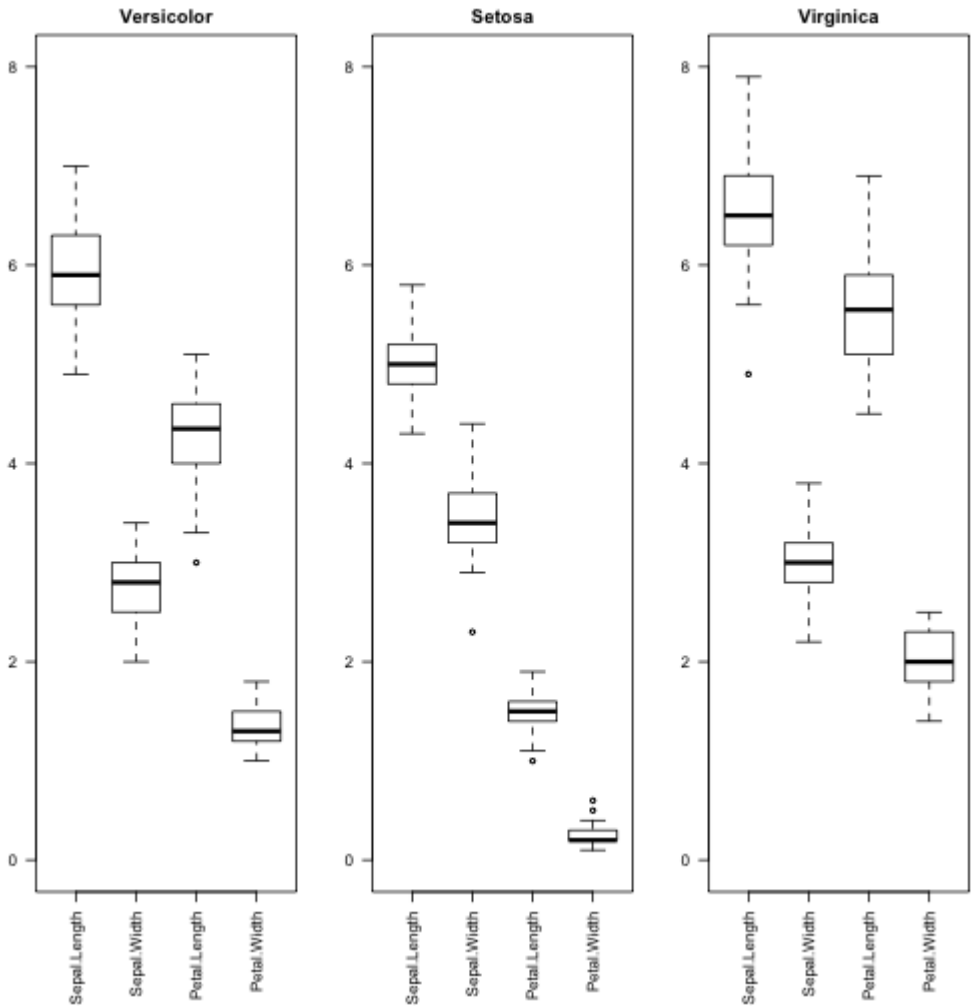
3.1.3.1. Boxplot

```
par(mar=c(7,5,1,1))
boxplot(iris,las=2)
```



Boxplot này cung cấp cho chúng ta những số liệu tổng quan về sự phân phối của các giá trị cho từng thuộc tính. Nhưng có lẽ sẽ có ý nghĩa hơn khi xem sự phân bố của các giá trị đang xem xét từng lớp bằng cách vẽ biểu đồ boxplot cho từng loài hoa.

```
irisVer <- subset(iris, Species == "versicolor")
irisSet <- subset(iris, Species == "setosa")
irisVir <- subset(iris, Species == "virginica")
par(mfrow=c(1,3),mar=c(6,3,2,1))
boxplot(irisVer[,1:4], main="Versicolor",ylim = c(0,8),las=2)
boxplot(irisSet[,1:4], main="Setosa",ylim = c(0,8),las=2)
boxplot(irisVir[,1:4], main="Virginica",ylim = c(0,8),las=2)
```



Nhận xét biểu đồ:

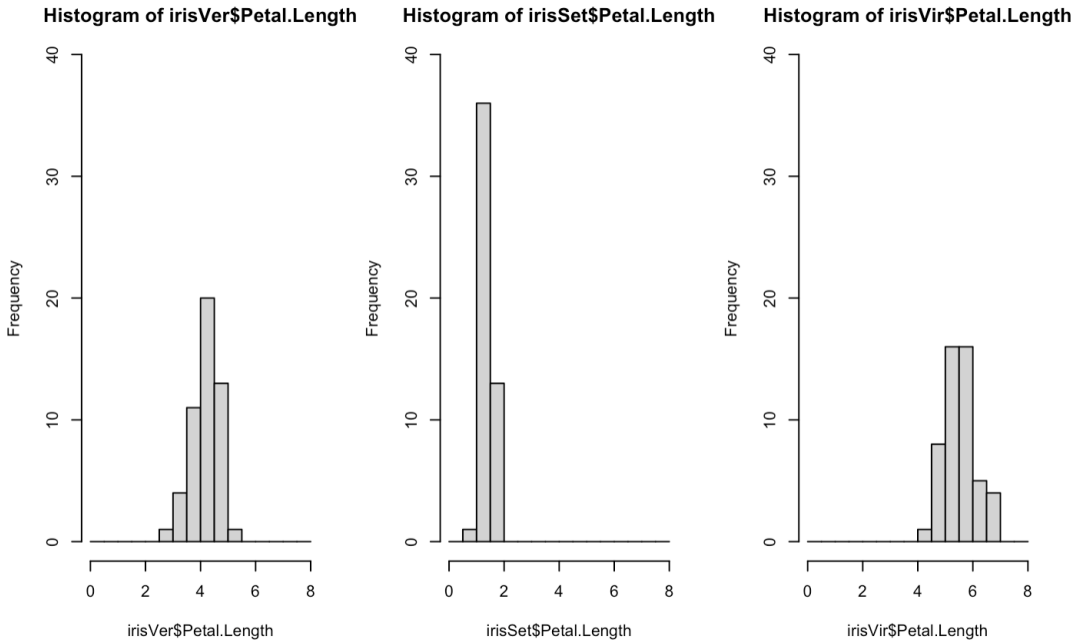
Từ biểu đồ Boxplot trên chúng ta có thể dễ dàng quan sát được sự tương quan của các thuộc tính ở cả ba loài hoa.

3.1.3.2. Histogram

Sử dụng biểu đồ Histogram để mô tả số liệu của mỗi đặc tính (chiều dài và chiều rộng của đài hoa và cánh hoa) cho mỗi loài và rút ra sự tương quan giữa chúng.

Histogram mô tả chiều dài cánh hoa cho mỗi loài (Petal.Length):

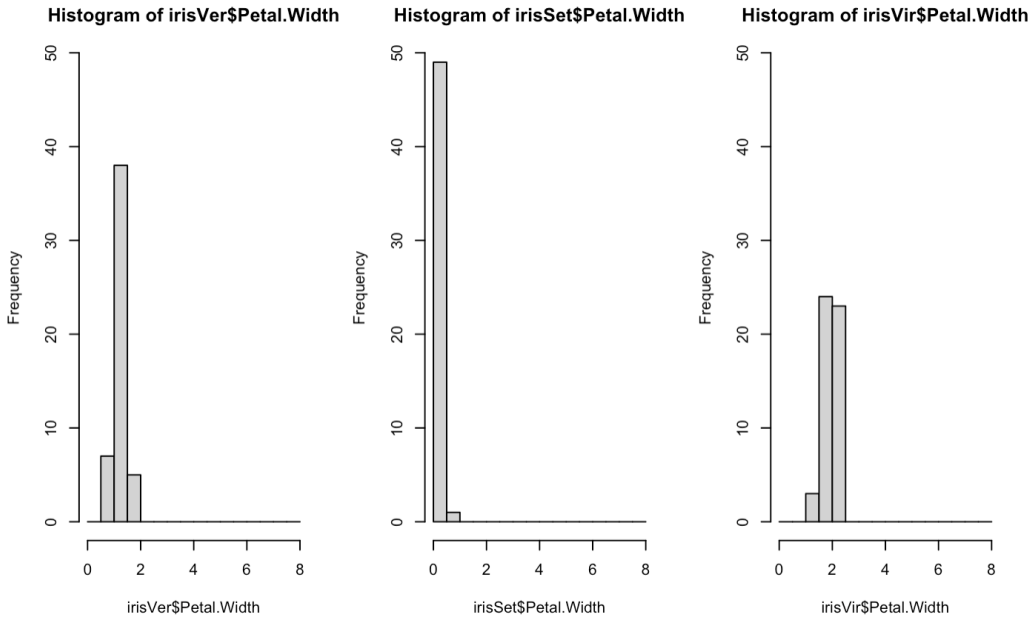
```
par(mfrow=c(1,3))
hist(irisVer$Petal.Length,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,40))
hist(irisSet$Petal.Length,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,40))
hist(irisVir$Petal.Length,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,40))
```



Nhận xét: Những biểu đồ trên cho thấy rằng sự phân bố của các giá trị Petal.Length (chiều dài cánh hoa) là khác nhau đối với mỗi lớp.

Histogram mô tả chiều rộng cánh hoa cho mỗi loài (Petal.Width):

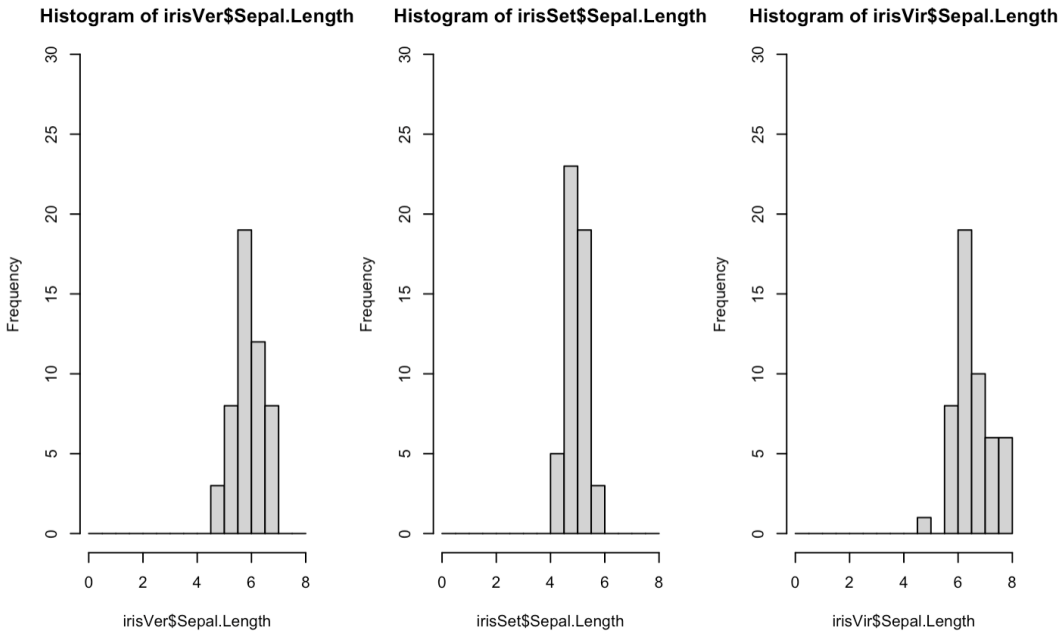
```
par(mfrow=c(1,3))
hist(irisVer$Petal.Width,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,50))
hist(irisSet$Petal.Width,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,50))
hist(irisVir$Petal.Width,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,50))
```



Nhận xét: Những biểu đồ trên cho thấy rằng sự phân bố của các giá trị Petal.Width (chiều rộng cánh hoa) là khác nhau đối với mỗi lớp.

Histogram mô tả chiều dài đài hoa cho mỗi loài (Sepal.Length):

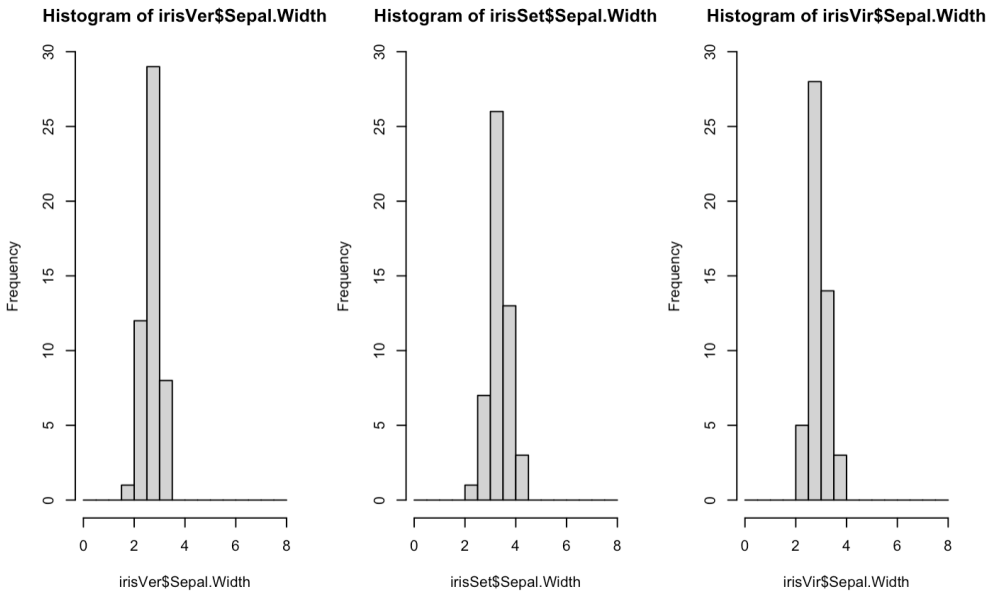
```
par(mfrow=c(1,3))
hist(irisVer$Sepal.Length,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,30))
hist(irisSet$Sepal.Length,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,30))
hist(irisVir$Sepal.Length,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,30))
```



Nhận xét: Những biểu đồ trên cho thấy rằng sự phân bố của các giá trị Sepal.Length (chiều dài đài hoa) là khá tương đồng nhau đối với mỗi lớp. Tập trung chủ yếu ở khoảng 4 - 6.

Histogram mô tả chiều rộng đài hoa cho mỗi loài (Sepal.Width):

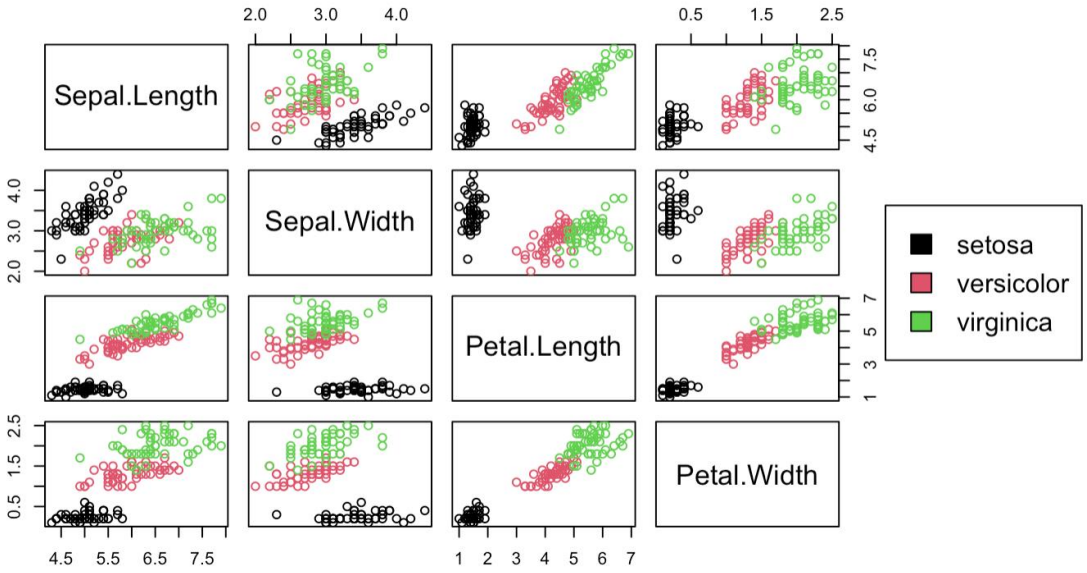
```
par(mfrow=c(1,3))
hist(irisVer$Sepal.Width,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,30))
hist(irisSet$Sepal.Width,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,30))
hist(irisVir$Sepal.Width,breaks=seq(0,8,l=17),xlim=c(0,8),ylim=c(0,30))
```



Nhận xét: Những biểu đồ trên cho thấy rằng sự phân bố của các giá trị Sepal.Width (chiều rộng đài hoa) ở cả ba lớp là khá tương đồng với nhau. Tập trung chủ yếu ở khoảng từ 2 - 4.

3.1.3.3. Scatterplot

```
pairs(iris[,1:4],col=iris[,5],oma=c(4,4,6,12))
par(xpd=TRUE)
legend(0.85,0.6, as.vector(unique(iris$Species)),fill=c(1,2,3))
```



Đồ thị Sepal.Length và Sepal.Width

- Đối với Setosa: có độ dài chủ yếu tập trung từ 4.5 - 6.0, chiều rộng từ 3.0 - 4.5
- Đối với Versicolor: có độ dài từ 5.0 - 6.0, chiều rộng từ 2.0 - 3.0
- Đối với Virginica: có độ dài từ 5.5 - 7.5, chiều rộng từ 2.5 - 3.5

→ Setosa sẽ có chiều dài thấp hơn so với 2 loại còn lại trong đó chiều rộng lại lớn hơn so với 2 loài còn lại nên dễ dàng phân loại được Setosa.

→ Versicolor sẽ thường có độ dài ngắn hơn so với Virginica có độ dài tương ứng.

Đồ thị Sepal.Length và Petal.Length

- Đối với Setosa: có độ dài đài hoa chủ yếu 4.5 - 6.0, chiều dài cánh hoa 1.0 - 2.0
- Đối với Versicolor: có độ dài đài hoa từ 5.0 - 6.0, chiều dài cánh hoa 3.0 - 5.0
- Đối với Virginica: có độ dài đài hoa từ 5.5 - 7.5, chiều dài cánh hoa 5.0 - 7.0

→ Setosa có chiều dài đài hoa và cánh hoa ngắn hơn so với 2 loại còn lại → dễ dàng phân biệt được Setosa.

→ Độ dài cánh hoa của versicolor thường ngắn hơn so với Virginica.

Đồ thị Sepal.Length và Petal.Width

Có thể dễ dàng nhìn thấy 2 thuộc tính này tỉ lệ thuận với nhau khá rõ ràng:

- Đối với Setosa: có độ dài đài hoa chủ yếu 4.5 - 6.0, chiều rộng cánh hoa 0 - 0.5
- Đối với Versicolor: có độ dài đài hoa từ 5.0 - 6.0, chiều rộng cánh hoa 1.0 - 1.5
- Đối với Virginica: có độ dài đài hoa từ 5.5 - 7.5, chiều rộng cánh hoa 1.5 - 2.5

→ Setosa có chiều dài đài hoa ngắn hơn và chiều rộng cũng ngắn hơn hẳn so với 2 loài hoa còn lại.

→ Độ dài đài hoa của Versicolor ngắn hơn so với Virginica và chiều rộng tập trung và ngắn hơn so với Virginica.

Đồ thị Sepal.Width và Petal.Length

- Đối với Setosa: là một dải dài quanh độ dài cánh hoa 1 - 2, trải từ 3 - 4
- Đối với Versicolor: có chiều rộng từ 2.0 - 3.0, độ dài cánh hoa từ 3.0 - 5.0
- Đối với Virginica: có chiều rộng từ 2.5 - 3.5, độ dài cánh hoa 5.0 -7.0

→ Setosa có cánh hoa ngắn và đài hoa rộng hơn 2 loại còn lại.

→ Độ dài cánh hoa của Versicolor thường ngắn hơn so với Virginica.

Tương tự với 2 đồ thị còn lại. Ta cũng có thể thấy rõ được sự phân vùng của các thuộc tính của mỗi loài.

Kết luận:

- Setosa thường tập trung ở một vùng riêng biệt nên có thể xác định được Setosa qua cách xem các thuộc tính của nó có nằm trong vùng đó hay không.
- Versicolor và Virginica sẽ có một vùng là giao của hai vùng mà hai loài tập trung. Tại vùng đó nó có thuộc tính nằm giữa của hai loài.

  → Mỗi loài sẽ có một vùng tập trung cụ thể.

  → Có thể tạo ra được phương pháp để xác định thông qua xem thuộc tính nó thuộc vùng nào (mô hình K-means).

3.2. Quy trình huấn luyện và kiểm thử (train và test)

- Dữ liệu để xây dựng mô hình là dữ liệu gốc (orginal data), dữ liệu này cần phải có thuộc tính phân lớp (categorial attribute).
- Dữ liệu gốc sẽ được chia làm hai phần là Training set (Để xây dựng mô hình) và Testing set (Để kiểm định mô hình).
- Cuối cùng là đánh giá mô hình.

3.3. Xây dựng mô hình

3.3.1. Mô hình Decision Trees

3.3.1.1. Thuật toán

Decision Tree là một đồ thị của các quyết định và hệ quả có thể xảy ra, được dùng để phân lớp các đối tượng dựa trên dãy các điều kiện (quy luật), là một cấu trúc phân cấp gồm các nút và các nhánh.

Trong đó có 3 loại nút (node):

- Nút điều kiện biểu diễn một đặc trưng (câu hỏi)
- Nút gốc (root) là đặc trưng (câu hỏi) đầu tiên
- Nút lá biểu diễn kết quả

Ngoài ra, mỗi nhánh biểu diễn một quy luật.

Ý tưởng của thuật toán này là khi cho trước bộ dữ liệu về các đối tượng cùng các thuộc tính và lớp của nó, Decision Tree sẽ sinh ra các quy luật để dự đoán lớp của các dữ liệu chưa biết. Decision Tree được dùng trong phân lớp bằng cách duyệt từ nút gốc qua các nhánh cho đến nút lá cuối cùng.

Xây dựng Decision Tree trong R sử dụng thuật toán C4.5, là thuật toán cải tiến của ID3 (Iterative Dichotomiser 3), sử dụng các chỉ số Entropy và Information gain để tính toán và đánh giá. Thuật toán về cơ bản chạy một vòng lặp sau.

Bước 1: Tính toán entropy cho tập dữ liệu.

Bước 2: Với tất cả đặc trưng:

- Tính toán entropy của tất cả giá trị.
- Tính entropy trung bình cho thuộc tính đang thực hiện.

Bước 3: Chọn đặc trưng có IG cao nhất làm gốc.

Bước 4: Lặp lại cho đến khi thu được cây như mong muốn.

3.3.1.2. Xây dựng model

Bước 1: Cài đặt và gọi đến thư viện cần thiết

```
install.packages(C50)
library(C50)
```

Bước 2: Load dataset để xây dựng model

```
# Tạo input (gồm 4 cột số đo các đặc tính của hoa, bỏ đi cột Species) để
tính toán trên đó các giá trị entropy, information gain và quyết định các
quy luật để phân loại
input <- iris[,1:4]

# Tạo output (cột Species - đặc trưng về giống hoa) là nhãn dán cho các lớp
(ở đây là tên giống hoa)
output <- iris[,5]
```

Bước 3: Thực hiện quá trình xây dựng Decision Tree

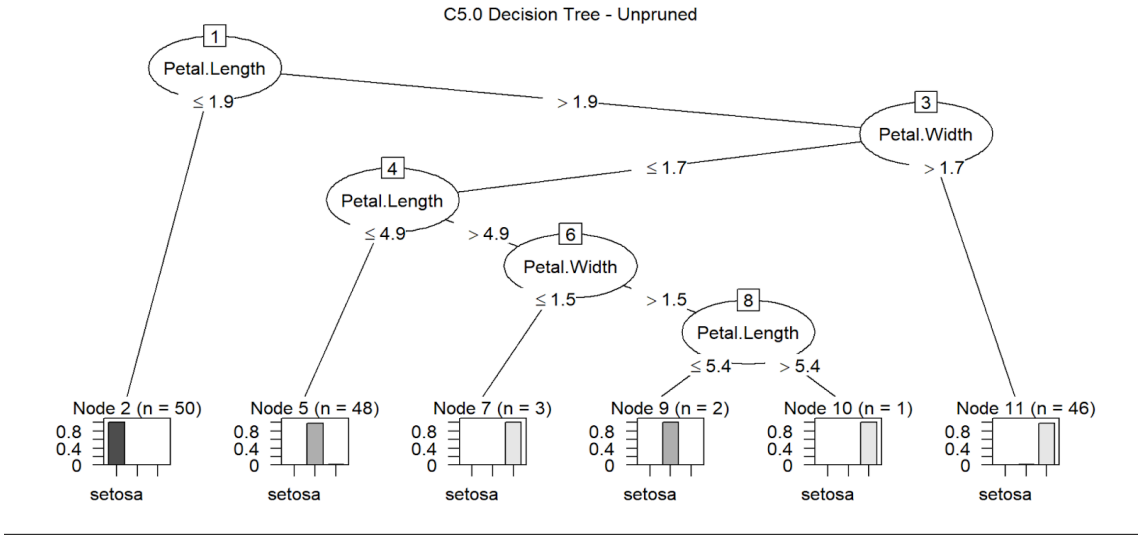
```
model1 <- C5.0(input, output, control = C5.0Control(noGlobalPruning = TRUE,
minCases = 1))
model2 <- C5.0(input, output, control = C5.0Control(noGlobalPruning = FALSE))
```

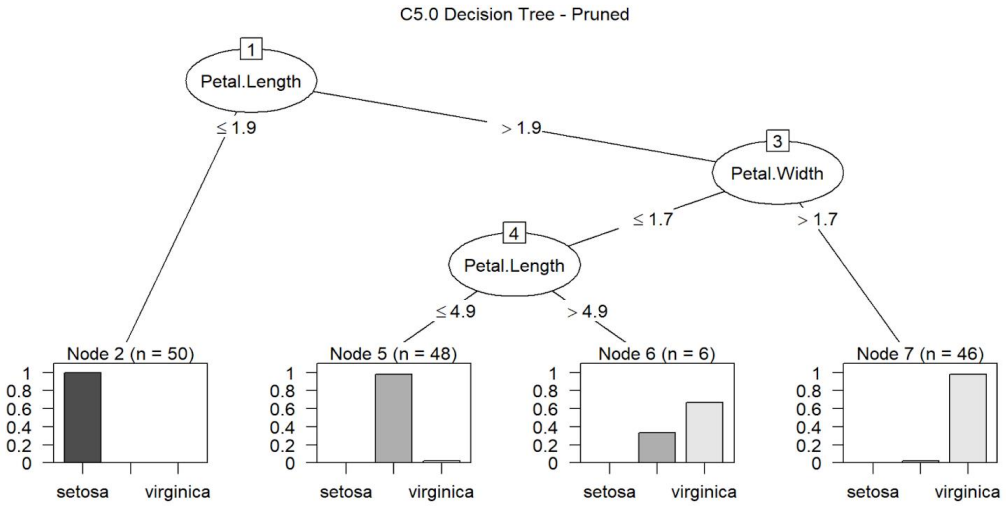
Để tạo Decision Tree ta sử dụng hàm `C5.0()` với các đối số dưới đây:

- x = input
- y = output
- control = `C5.0Control()` - được tách thành một hàm riêng với các đối số:
  + noGlobalPruning (default = FALSE): biến logic, quyết định việc Pruning (cắt tỉa, ngừng phân nhánh) để đơn giản hóa cây có được thực hiện hay không. Đối với model1, noGlobalPruning = TRUE, nghĩa là không Pruning. Ngược lại ở model2, noGlobalPruning = FALSE, nghĩa là có Pruning.
  + minCases (default = 2): số lượng mẫu nhỏ nhất được phân vào các node lá (tức minCases càng nhỏ thì Decision Tree càng nhiều nhánh).

Bước 4: Vẽ Decision Tree

```
plot(model1, main = "C5.0 Decision Tree - Unpruned")
plot(model2, main = "C5.0 Decision Tree - Pruned")
```





Bước 5: Xem kết quả model

```
summary(model2)
```

```
1 ##
2 ## Call:
3 ## C5.0.default(x = input, y = output, control =
4 ##  C5.0Control(noGlobalPruning = FALSE))
5 ##
6 ##
7 ## C5.0 [Release 2.07 GPL Edition]      Mon Jul 29 09:20:13 2019
8 ## -------------------------------
9 ##
10 ## Class specified by attribute `outcome'
```

```
11 ##
12 ## Read 150 cases (5 attributes) from undefined.data
13 ##
14 ## Decision tree:
15 ##
16 ## Petal.Length <= 1.9: setosa (50)
17 ## Petal.Length > 1.9:
18 ## :...Petal.Width > 1.7: virginica (46/1)
19 ##     Petal.Width <= 1.7:
20 ##     :...Petal.Length <= 4.9: versicolor (48/1)
21 ##         Petal.Length > 4.9: virginica (6/2)
22 ##
23 ##
24 ## Evaluation on training data (150 cases):
25 ##
26 ##           Decision Tree
27 ##         ----------------
28 ##         Size      Errors
29 ##
30 ##            4    4( 2.7%)   <<
31 ##
32 ##
33 ##          (a)   (b)   (c)    <-classified as
34 ##         ----  ----  ----
35 ##           50                (a): class setosa
36 ##                 47     3    (b): class versicolor
37 ##                  1    49    (c): class virginica
38 ##
39 ##
40 ##       Attribute usage:
41 ##
42 ##       100.00%     Petal.Length
43 ##        66.67%     Petal.Width
44 ##
45 ##
56 ## Time: 0.0 secs
```

Phần 1 - Thông tin khái quát: Nhắc lại hàm (dòng 3 - 4), cho biết tên thư viện được sử dụng, thời gian sử dụng (dòng 7), nội dung quá trình phân loại (dòng 10), số

mẫu được dùng trong training set, ở đây là tập input với 150 mẫu (dòng 12). Quá trình phân loại chỉ sử dụng 2 đặc tính Petal.Length và Petal.Width (dòng 40 - 43).

Phần 2 - Quá trình phân loại: Decision Tree bắt đầu phân loại từ thuộc tính Petal.Length. Có thể thấy toàn bộ 50 mẫu setosa đều có chiều dài cánh hoa nhỏ hơn 1.9. Đối với Petal.Length lớn hơn hoặc bằng 1.9, thuật toán sử dụng tiếp đến thuộc tính Petal.Width để phân loại. Với Petal.Width nhỏ hơn 1.7 ta thu được 46 cây là virginica và 1 cây không phải. Ngược lại với Petal.Width lớn hơn hoặc bằng 1.7, dựa vào Petal.Length ta phân loại được 48 versicolor và 1 virginica có chiều dài cánh hoa nhỏ hơn 4.9, còn lại là 6 virginica và 2 versicolor có chiều dài cánh hoa lớn hơn 4.9.

Phần 3 - Đánh giá chất lượng quá trình phân loại: Decision Tree có 4 node lá và kết quả có 4 mẫu bị phân loại sai. Một ma trận được đưa ra mô tả tỉ lệ phân loại của Decision Tree, trong đó 50 mẫu setosa đều được phân loại đúng, 47 versicolor phân loại đúng và 3 mẫu bị nhầm thành virginica, 49 virginica phân loại đúng và 1 mẫu bị nhầm thành versicolor.

3.3.2. Mô hình K-means

3.3.2.1. Thuật toán

Mô hình K-Means được sử dụng để giải quyết các bài toán phân cụm (các cỗ máy tìm kiếm, phân loại khách hàng, thống kê dữ liệu…). Ý tưởng của thuật toán là quá trình phân chia 1 bộ dữ liệu X không có nhãn thành các cụm khác nhau với số lượng cụm được cho trước là K.

Trong đó, mỗi cụm dữ liệu có một điểm trung tâm (center). Với các điểm còn lại, nếu một điểm gần với center nào nhất thì thuộc cùng nhóm với center đó. Các điểm dữ liệu trong một cụm thì có cùng một số tính chất nhất định và hai cụm dữ liệu phân biệt với nhau.

Các bước thực hiện:

- Đầu vào: Bộ dữ liệu X và số lượng cụm cần tìm K
- Đầu ra: Các điểm trung tâm M và dán nhãn cho từng điểm dữ liệu Y
- Vòng lặp:

Bước 1: Khởi tạo K điểm dữ liệu trong bộ dữ liệu và tạm thời coi nó là tâm của các cụm dữ liệu.

Bước 2: Phân mỗi điểm dữ liệu vào cụm có khoảng cách đến điểm trung tâm của cụm đó là nhỏ nhất.

Bước 3: Sau khi tất cả các điểm dữ liệu đã có cụm, tính toán lại vị trí của tâm cụm để đảm bảo tâm của cụm nằm ở chính giữa cụm.

Bước 4: Lặp lại bước 2 và bước 3 cho đến khi thuật toán hội tụ (giữa hai lần cập nhật tâm cụm thì vị trí của tâm cụm không thay đổi hoặc chênh lệch vị trí giữa tâm cũ và mới nhỏ hơn một số delta cho phép nào đó) thì kết thúc.

3.3.2.2. Xây dựng mô hình

Bước 1: Cài đặt và gọi đến các thư viện cần thiết

```
install.packages("dplyr")
install.packages("ggplot2")
install.packages("corrplot")

library(dplyr)
library(ggplot2)
library(corrplot)
```

Bước 2: Load dataset để xây dựng model

```
# Tạo bản sao của dataset iris
iris2 <- iris

# Loại bỏ nhãn dán của các lớp dữ liệu (cột Species)
iris2$Species <- NULL
```

Bước 3: Xác định số lượng cụm K

Đối với dataset iris, ta đã biết trước số lượng giống hoa là 3, tương ứng với số lượng cụm K = 3, vì vậy ta có thể bỏ qua bước này.

Tuy nhiên, trong thực tế, không phải lúc nào ta cũng biết trước số lượng nhóm hợp lý để phân loại dataset, khi đó ta sẽ cần đến một số phương pháp để xác định số lượng cụm cần chọn, một trong số đó là phương pháp Elbow point (bài viết chỉ đề cập đến và sẽ không đi vào giải thích cụ thể về phương pháp này)

Bước 4: Thực hiện quá trình phân cụm K-means

```r
# Hàm set.seed() đảm bảo bộ sinh số ngẫu nhiên có thể cho ra cùng một kết quả (ở đây là khởi tạo bộ 3 số ngẫu nhiên cho centers)
set.seed(8593)

# Sử dụng hàm kmeans() với số lượng cụm K = 3
kmeans.result <- kmeans(iris2, 3)

# In ra kết quả quá trình phân cụm
kmeans.result
```

```
## K-means clustering with 3 clusters of sizes 38, 62, 50
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     6.850000    3.073684     5.742105    2.071053
## 2     5.901613    2.748387     4.393548    1.433871
## 3     5.006000    3.428000     1.462000    0.246000
##
## Clustering vector:
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [71] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1
## [106] 1 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1
## [141] 1 1 2 1 1 1 2 1 1 2
##
## Within cluster sum of squares by cluster:
## [1] 23.87947 39.82097 15.15100
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

Số lượng các điểm được phân vào các cụm lần lượt là 38, 62, 50. Cluster means cho biết vị trí của 3 điểm trung tâm. Ví dụ cụm thứ nhất có giá trị trung bình của

Sepal.Length = 6.85, Sepal.Width = 3.07, Petal.Length = 5.74 and Petal.Width = 2.07. Clustering vector là một vector 150 thành phần chỉ gồm các giá trị là 1, 2, 3, miêu tả điểm dữ liệu nào được phân loại về cụm nào trong 3 cụm.

Bước 5: Kết quả phân loại

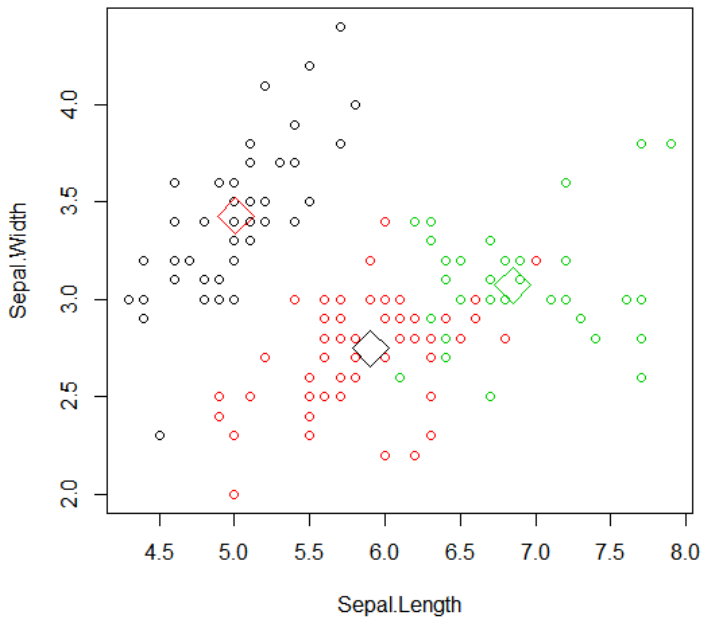
```
# Lập bảng so sánh kết quả phân loại với giống hoa vốn có
table(iris$Species, kmeans.result$cluster)
```

```
##
##                1  2  3
##    setosa      0  0 50
##    versicolor  2 48  0
##    virginica  36 14  0
```

Bước 6: Vẽ biểu đồ biểu diễn các cụm

```
# Vẽ biểu đồ biểu diễn các cụm
plot(iris2[c("Sepal.Length", "Sepal.Width")], col = kmeans.result$cluster)

# Để phân biệt các cụm dễ hơn, ta đánh dấu các tâm cụm
points(kmeans.result$centers[c("Sepal.Length", "Sepal.Width")], col = 1:3,
pch = 8, cex = 2)
```



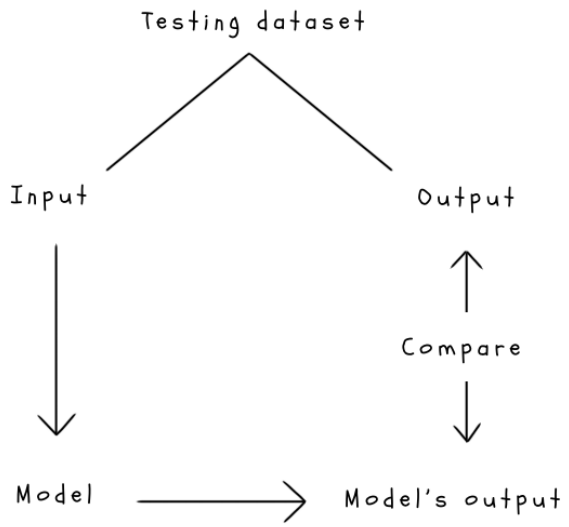
3.4. So sánh hai mô hình Decision Trees và K-means

3.4.1. Quy trình test mô hình

Đối với bài toán này, input là các số đo của hoa, output là giống hoa.

Quá trình test mô hình:

- Đưa dữ liệu (các kích thước của hoa) qua model để xử lý
- Nhận output (giống hoa) từ model
- So sánh với output đã biết (giống hoa thực tế) để xác định độ chính xác



Nhận xét:

a. Decision Tree

- Model được tạo nên từ việc training dữ liệu trước đó. Tức là sẽ phải dùng 2 lệnh, một để tạo model (C5.0) và một để test (predict).
- Tác giả sử dụng 2 lần test:
  + Lần đầu tiên, tác giả trích ra testing data là 3 hàng đầu của mỗi giống hoa, sau đó dùng lệnh `predict()` với các tham số lần lượt là model = model2, data = newcases, type = "class":

```
newcases <- iris[c(1:3,51:53,101:103),]

predicted <- predict(model2, newcases, type="class")
predicted
```

Kết quả:

```
## [1] setosa     setosa     setosa     versicolor versicolor versicolor
## [7] virginica  virginica  virginica
## Levels: setosa versicolor virginica
```

+ Lần thứ 2, testing data là iris dataset, rồi dùng lệnh `predict()` như trên:

```
predicted <- predict(model2, iris, type="class")
```

➢ Cả 2 lần tác giả đều sử dụng model2.

- Sau đó, tác giả đưa kết quả test của lần test thứ hai thành một thuộc tính predictedC501 của iris, rồi trích ra dữ liệu những cá thể có giống ban đầu khác với giống dự đoán để so sánh:

```
iris$predictedC501 <- predicted
iris[iris$Species != iris$predictedC501,]
```

Kết quả:

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 71           5.9         3.2          4.8         1.8 versicolor
## 78           6.7         3.0          5.0         1.7 versicolor
## 84           6.0         2.7          5.1         1.6 versicolor
## 107          4.9         2.5          4.5         1.7  virginica
##     predictedC501
## 71      virginica
## 78      virginica
## 84      virginica
## 107    versicolor
```

➢ Kết quả này sẽ được phân tích ở phần sau.

b. K-means

- Ta sử dụng lệnh có sẵn, tức là thuật toán đã có sẵn. Chỉ cần dùng 1 lệnh `kmeans()` (với tham số là dataset và số cụm), các lệnh phía sau là để xây dựng biểu đồ trực quan hóa.

```
kmeans.result <- kmeans(iris2, 3)
```

- Ở đây, output không phải là tên hoa mà là các số, tuy nhiên, vẫn có thể suy ra tên giống hoa từ đó.
- Tác giả sử dụng bảng phân bố tần số (table) để so sánh kết quả test. Ta sẽ theo dõi ở phần dưới.

3.4.2. Kết quả test của Decision Tree và Kmeans trong bài viết

Ở đây, ta so sánh độ chính xác giữa 2 model Decision Tree và 1 model Kmeans sử dụng trên iris dataset. Sử dụng lệnh table với kết quả test, ta được:

```
[1] "Using unpruned tree"   (model1)

                  setosa       versicolor       virginica

  setosa              50                0               0

  versicolor           0               49               1

  virginica            0                1              49
```

➢ Có 2 cá thể sai khác. Cụ thể hơn, có 2 cá thể bị đảo ngược là 2 cá thể của versicolor và virginica.

```
[1] "Using pruned tree"   (model2)

                  setosa       versicolor       virginica

  setosa              50                0               0

  versicolor           0               47               3

  virginica            0                1              49
```

➢ Có 4 cá thể sai khác. Có 3 cá thể versicolor nhưng kết quả test là virginica. Ngược lại với cá thể còn lại.

```
[1] "Using kmeans"
                    setosa       versicolor       virginica
 setosa                 50                0               0
 versicolor              0               48               2
 virginica               0               14              36
```

➢ Có 16 cá thể sai khác. Có 2 cá thể versicolor nhưng kết quả test là virginica. Ngược lại với 14 cá thể còn lại.

➢ Ta có thể thấy mức độ chính xác đang giảm dần.

3.4.3. So sánh hai mô hình Decision Trees và Kmeans

	Decision Tree	Kmeans
Đặc điểm	- Biểu diễn dạng sơ đồ cây - Thuật toán rẽ nhánh, chủ yếu so sánh giữa các giá trị mà không cần tính toán nhiều	- Biểu diễn dạng biểu đồ (sơ đồ Venn, biểu diễn điểm) - Thuật toán phân cụm, tính toán giá trị trung bình nhiều lần
Ưu điểm	- Dễ hiểu, rõ ràng, giúp nhìn thấy logic từ dữ liệu - Không yêu cầu về chuẩn bị dữ liệu (chuẩn hóa dữ liệu, loại bỏ outlier…) - Có thể xử lý một lượng dữ liệu lớn trong thời gian ngắn - Có khả năng xử lý dữ liệu bị thiếu, lỗi, outliers	- Dễ cài đặt, trực quan, phù hợp với tập dữ liệu lớn - Đảm bảo luôn hội tụ (thuật toán luôn kết thúc) - Linh hoạt, tự điều chỉnh khi có thay đổi trong tập dữ liệu ban đầu

Nhược điểm	- Overfitting (khi model quá khớp với một tập dữ liệu cụ thể, dẫn đến mất tính tổng quát và dễ xảy ra sai sót khi có thay đổi trong tập dữ liệu) - Bất ổn, thay đổi nhỏ trong dữ liệu ban đầu có thể dẫn đến thay đổi kết quả đáng kể - Khó khăn khi tính toán và dễ phân điểm dữ liệu vào nhầm nhánh do có sự liên quan hoặc tương đồng với nhau - Không phù hợp cho tập dữ liệu lớn	- Cần biết số lượng cụm - Outliers dễ làm sai thuật toán - Phụ thuộc vào các tâm cụm ban đầu mà cách phân nhóm sẽ thay đổi → cần lặp lại thuật toán nhiều lần để có kết quả chính xác - Chỉ phù hợp với các nhóm phân biệt (một đối tượng không ở trong nhiều nhóm) và tập dữ liệu có dạng lồi - Chỉ áp dụng được khi xác định được giá trị trung bình - Các cụm cần có kích thước tương đương nhau

III. LỜI KẾT

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành nhất đến thầy Vũ Ngọc Bình, phụ trách bộ môn Nhập môn phân tích dữ liệu. Trong quá trình học tập và tìm hiểu bộ môn trên, chúng em đã nhận được sự giúp đỡ, hướng dẫn rất tận tình, tâm huyết của thầy. Thầy đã giúp chúng em tích lũy thêm nhiều kiến thức để có thể xây dựng nền tảng kiến thức quan trọng đối với các sinh viên ngành Khoa học dữ liệu nói riêng và các bạn sinh viên đến từ hầu hết các ngành nói chung.

Thông qua bài luận này, chúng em đã áp dụng những kiến thức được truyền đạt để tiến hành phân tích và tìm hiểu về quy trình giải quyết một bài toán, vấn đề thực tế thông qua các kỹ năng phân tích dữ liệu.

Trong quá trình hoàn thành bài tiểu luận chắc chắn không thể tránh khỏi những thiếu sót, chúng em rất mong nhận được những góp ý đến từ thầy để bài tiểu luận của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn thầy. Chúc thầy luôn luôn mạnh khoẻ, hạnh phúc và gặt hái được nhiều thành công hơn nữa trên con đường sự nghiệp giảng dạy.

Bảng đánh giá mức độ hoàn thành công việc của các thành viên trong nhóm (thang điểm 10)

STT	Họ và tên	Mức độ hoàn thành công việc (trên thang điểm 10)
1	Vương Thị Diễm Quỳnh	10
2	Hoàng Thảo Nguyên	10
3	Nguyễn Khánh Huyền	10
4	Phan Diệu Linh	10
5	Tạ Khánh Ly	9
6	Nguyễn Nam Anh	9,5
7	Nguyễn Bá Mạnh	8,5
8	Nguyễn Hà Nam	8,5
9	Nguyễn Đức Nam	8,5
10	Võ Ngọc Hiếu	7,5