Chapter 19: Decision Tree

Exercise 1: Classification Animal

Cho dữ liệu zoo.data.txt.

Thông tin các cột dữ liệu: Data Infromation: Bộ dữ liệu chứa 17 thuộc tính kiểu Boolean. Thuộc tính "type" là class attribute:

Class# -- Set of animals: ======
==========

1. (41) aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sealion, squirrel, vampire, vole, wallaby,wolf
2. (20) chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren
3. (5) pitviper, seasnake, slowworm, tortoise, tuatara
4. (13) bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna
5. (4) frog, frog, newt, toad
6. (8) flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp
7. (10) clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

Thuộc tính:

1. animal name: Unique for each instance
2. hair: Boolean
3. feathers: Boolean
4. eggs: Boolean
5. milk: Boolean
6. airborne: Boolean
7. aquatic: Boolean
8. predator: Boolean
9. toothed: Boolean
10. backbone: Boolean
11. breathes: Boolean
12. venomous: Boolean
13. fins: Boolean
14. legs: Numeric (set of values: {0,2,4,5,6,8})
15. tail: Boolean
16. domestic: Boolean
17. catsize: Boolean
18. type: Numeric (integer values in range [1,7])

Yêu cầu: Hãy áp dụng Decision Tree để dự đoán loại của animal dựa trên các thông tin được cung cấp:

- Đọc dữ liệu và gán cho biến data.
- In thông tin head, tail, str, summary
- Chuẩn hóa dữ liệu nếu cần
- Tạo train:test từ dữ liệu data với tỉ lệ 75:25
- Áp dụng decision tree
- Tìm kết quả
- Tính toán độ chính xác
- Vẽ hình => xem kết quả
- Với các thông tin: c(1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 4, 0, 0, 1), c(1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 6, 0, 0, 0), thì mẫu này thuộc loại nào?

In [1]:
```
library(rpart)
data <- read.csv("zoo.data.txt", header = FALSE)
print(head(data))
print(paste("Is dataframe?",is.data.frame(data)))
```

```
        V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18
1 aardvark  1  0  0  1  0  0  1  1   1   1   0   0   4   0   0   1   1
2 antelope  1  0  0  1  0  0  0  1   1   1   0   0   4   1   0   1   1
3     bass  0  0  1  0  0  1  1  1   1   0   0   1   0   1   0   0   4
4     bear  1  0  0  1  0  0  1  1   1   1   0   0   4   0   0   1   1
5     boar  1  0  0  1  0  0  1  1   1   1   0   0   4   1   0   1   1
6  buffalo  1  0  0  1  0  0  0  1   1   1   0   0   4   1   0   1   1
[1] "Is dataframe? TRUE"
```

In [2]:
```
# tail(data)
```

In [3]:
```
print(paste("cols:", ncol(data)))
print(paste("rows:", nrow(data)))
```

```
[1] "cols: 18"
[1] "rows: 101"
```

In [4]:
```
summary(data)
```

```
       V1           V2              V3              V4        
 frog    : 2   Min.   :0.0000   Min.   :0.000   Min.   :0.0000  
 aardvark: 1   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000  
 antelope: 1   Median :0.0000   Median :0.000   Median :1.0000  
 bass    : 1   Mean   :0.4257   Mean   :0.198   Mean   :0.5842  
 bear    : 1   3rd Qu.:1.0000   3rd Qu.:0.000   3rd Qu.:1.0000  
 boar    : 1   Max.   :1.0000   Max.   :1.000   Max.   :1.0000  
 (Other) :94                                                    
       V5              V6              V7              V8        
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :0.0000   Median :0.0000   Median :0.0000   Median :1.0000  
 Mean   :0.4059   Mean   :0.2376   Mean   :0.3564   Mean   :0.5545  
 3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
       V9              V10             V11             V12         
 Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000  
 1st Qu.:0.000   1st Qu.:1.0000   1st Qu.:1.0000   1st Qu.:0.00000  
 Median :1.000   Median :1.0000   Median :1.0000   Median :0.00000  
 Mean   :0.604   Mean   :0.8218   Mean   :0.7921   Mean   :0.07921  
 3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.00000  
 Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000  
      V13             V14             V15             V16        
 Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :0.0000   Median :4.000   Median :1.0000   Median :0.0000  
 Mean   :0.1683   Mean   :2.842   Mean   :0.7426   Mean   :0.1287  
 3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.:1.0000   3rd Qu.:0.0000  
 Max.   :1.0000   Max.   :8.000   Max.   :1.0000   Max.   :1.0000  
      V17             V18        
 Min.   :0.0000   Min.   :1.000  
 1st Qu.:0.0000   1st Qu.:1.000  
 Median :0.0000   Median :2.000  
 Mean   :0.4356   Mean   :2.832  
 3rd Qu.:1.0000   3rd Qu.:4.000  
 Max.   :1.0000   Max.   :7.000  
```

In [5]:
```
str(data)
```

```
'data.frame':   101 obs. of  18 variables:
 $ V1 : Factor w/ 100 levels "aardvark","antelope",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ V2 : int  1 1 0 1 1 1 1 0 0 1 ...
 $ V3 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ V4 : int  0 0 1 0 0 0 0 1 1 0 ...
 $ V5 : int  1 1 0 1 1 1 1 0 0 1 ...
 $ V6 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ V7 : int  0 0 1 0 0 0 0 1 1 0 ...
 $ V8 : int  1 0 1 1 1 0 0 0 1 0 ...
 $ V9 : int  1 1 1 1 1 1 1 1 1 1 ...
 $ V10: int  1 1 1 1 1 1 1 1 1 1 ...
 $ V11: int  1 1 0 1 1 1 1 0 0 1 ...
 $ V12: int  0 0 0 0 0 0 0 0 0 0 ...
 $ V13: int  0 0 1 0 0 0 0 1 1 0 ...
 $ V14: int  4 4 0 4 4 4 4 0 0 4 ...
 $ V15: int  0 1 1 0 1 1 1 1 1 0 ...
 $ V16: int  0 0 0 0 0 0 1 1 0 1 ...
 $ V17: int  1 1 0 1 1 1 1 0 0 0 ...
 $ V18: int  1 1 4 1 1 1 1 4 4 1 ...
```

In [6]:
```
# Column 18: type
data <- subset(data, select=-V1)
print(head(data))
```

```
  V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18
1  1  0  0  1  0  0  1  1   1   1   0   0   4   0   0   1   1
2  1  0  0  1  0  0  0  1   1   1   0   0   4   1   0   1   1
3  0  0  1  0  0  1  1  1   1   0   0   1   0   1   0   0   4
4  1  0  0  1  0  0  1  1   1   1   0   0   4   0   0   1   1
5  1  0  0  1  0  0  1  1   1   1   0   0   4   1   0   1   1
6  1  0  0  1  0  0  0  1   1   1   0   0   4   1   0   1   1
```

In [7]:
```
# Create the training and test data
set.seed(42)
trainingRowIndex <- sample(1:nrow(data), 0.75*nrow(data))
print("Selected training row indexes:")
print(trainingRowIndex)
trainingData <- data[trainingRowIndex, ]  # training data
testData  <- data[-trainingRowIndex, ]   # test data
print("Rows of training data and test data:")
print(nrow(trainingData))
print(nrow(testData))
```

```
[1] "Selected training row indexes:"
 [1]  93  94  29  82  63  50  70  13  62  65  42  92  84  23  41  81  89  10  4
0
[20]  46  74  12  79  86   7  83  30  68  33  61  53  57  27  47   1  55  67  1
4
[39]  58  38  24  69   3  95  25  54  49  35  52  73  18  51  20 101   2  56  3
1
[58]   8  80  22  28  76  75  91  32  77  85  99  88  44  78   5  36  64   6
[1] "Rows of training data and test data:"
[1] 75
[1] 26
```

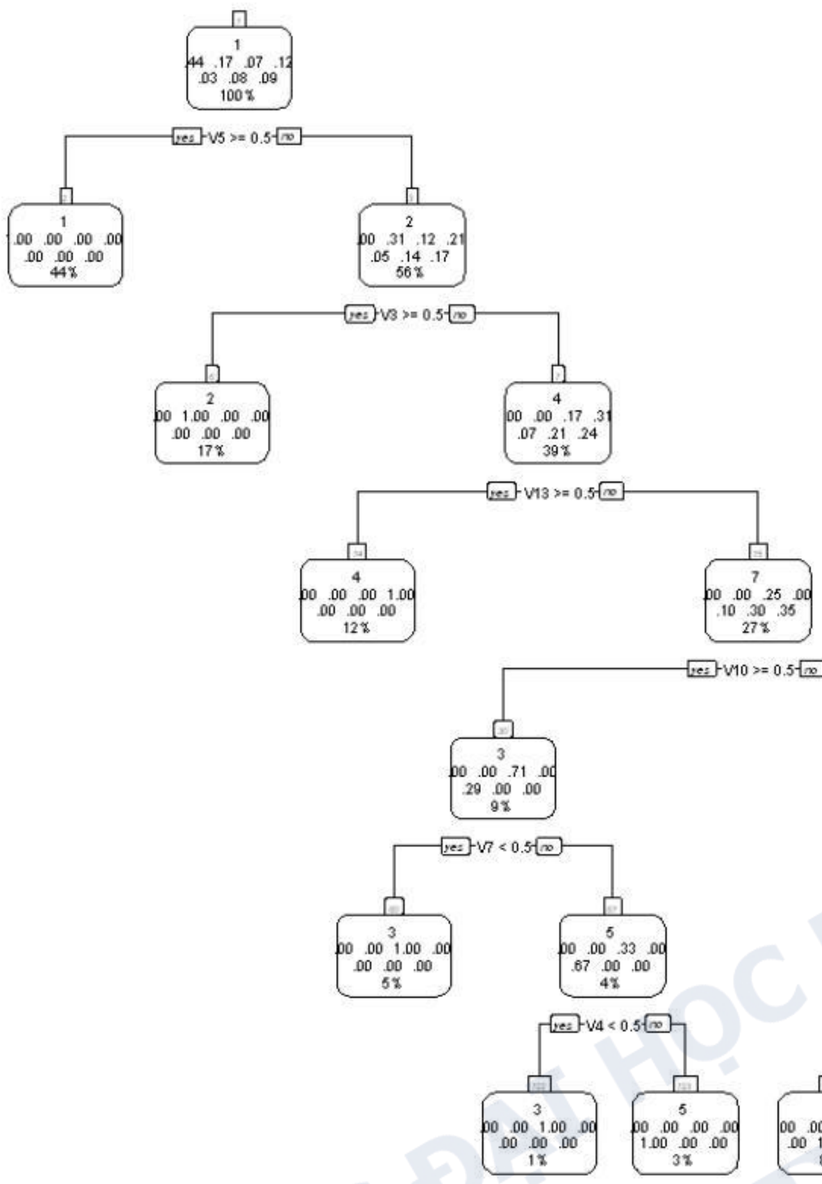
In [8]:
```
# Build model
#use: control = list(maxdepth = 15)
data.tree <- rpart(V18 ~ V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+V17
                   data = trainingData,
                   method="class",
                   minbucket=1)
print(data.tree)
```

```
n= 75

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 75 42 1 (0.44 0.17 0.067 0.12 0.027 0.08 0.093)
    2) V5>=0.5 33  0 1 (1 0 0 0 0 0 0) *
  3) V5< 0.5 42 29 2 (0 0.31 0.12 0.21 0.048 0.14 0.17)
    6) V3>=0.5 13  0 2 (0 1 0 0 0 0 0) *
    7) V3< 0.5 29 20 4 (0 0 0.17 0.31 0.069 0.21 0.24)
     14) V13>=0.5 9  0 4 (0 0 0 1 0 0 0) *
     15) V13< 0.5 20 13 7 (0 0 0.25 0 0.1 0.3 0.35)
       30) V10>=0.5 7  2 3 (0 0 0.71 0 0.29 0 0)
         60) V7< 0.5 4  0 3 (0 0 1 0 0 0 0) *
         61) V7>=0.5 3  1 5 (0 0 0.33 0 0.67 0 0)
          122) V4< 0.5 1  0 3 (0 0 1 0 0 0 0) *
          123) V4>=0.5 2  0 5 (0 0 0 0 1 0 0) *
       31) V10< 0.5 13  6 7 (0 0 0 0 0 0.46 0.54)
         62) V8< 0.5 7  1 6 (0 0 0 0 0 0.86 0.14)
          124) V14>=3 6  0 6 (0 0 0 0 0 1 0) *
          125) V14< 3 1  0 7 (0 0 0 0 0 0 1) *
         63) V8>=0.5 6  0 7 (0 0 0 0 0 0 1) *
```

In [9]:
```
# draw tree
library(rpart.plot)
prp(data.tree,type=2,extra="auto",nn = TRUE,branch=1,varlen=0,yesno=2)
```



In [10]:
```
#test model
pred_new = predict(data.tree, testData, type = "class")
```

In [11]:
```
print("Predict vs Actual:")
result <- data.frame(Predict = pred_new, Actual = testData$V18)
print(result)
```

```
[1] "Predict vs Actual:"
    Predict Actual
4         1      1
9         4      4
11        1      1
15        7      7
16        7      7
17        2      2
19        4      4
21        2      2
26        5      5
34        2      2
37        1      1
39        4      4
43        7      6
45        1      1
48        1      1
59        2      2
60        2      2
66        1      1
71        1      1
72        2      2
87        4      4
90        5      5
96        2      2
97        1      1
98        6      6
100       7      7
```

In [12]:
```
# SOLUTION 2
misClasificError <- mean(pred_new != testData$V18)
print(paste('Accuracy s2: ',1-misClasificError))
```

```
[1] "Accuracy s2:  0.961538461538462"
```

In [13]:
```
# prediction new values
newCase  <- data[c(1,10, 100),]
print(newCase$V18)
newCase$V18 <- NULL
print(newCase)
```

```
[1] 1 1 7
    V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
1    1  0  0  1  0  0  1  1   1   1   0   0   4   0   0   1
10   1  0  0  1  0  0  0  1   1   1   0   0   4   0   1   0
100  0  0  1  0  0  0  0  0   0   1   0   0   0   0   0   0
```

In [14]:
```
print("New predictions:")
pred_new = predict(data.tree, newCase, type = "class")
print(pred_new)
```

```
[1] "New predictions:"
  1  10 100
  1   1   7
Levels: 1 2 3 4 5 6 7
```

In [15]:
```
newdata = data.frame(V2 = c(1, 1),
                     V3 = c(0, 0),
                     V4 = c(0, 1),
                     V5 = c(1, 0),
                     V6 = c(0, 1),
                     V7 = c(0, 0),
                     V8 = c(1, 0),
                     V9 = c(1, 0),
                     V10 = c(1, 0),
                     V11 = c(0, 1),
                     V12 = c(0, 1),
                     V13 = c(0, 0),
                     V14 = c(4, 6),
                     V15 = c(0, 0),
                     V16 = c(0, 0),
                     V17 = c(1, 0))

print("New predictions:")
pred_new = predict(data.tree, newdata, type = "class")
print(pred_new)
```

```
[1] "New predictions:"
1 2
1 6
Levels: 1 2 3 4 5 6 7
```