

## รายงานผลการดำเนินการ

### 2.1. การเตรียมชุดข้อมูล (Data acquisition)

อธิบายว่าน.ศ.อ่านข้อมูลเข้ามาอย่างไร ทำอะไรบ้างเพื่อศึกษาให้เข้าใจข้อมูล และเข้าใจข้อมูลว่าอย่างไรบ้าง แขนง code ประกอบ

2.1.1 อ่านข้อมูล mushrooms เข้ามาด้วยการใช้คำสั่ง `read.csv("mushrooms.csv")` โดยต้อง set path ไปที่ folder ที่เก็บข้อมูล mushrooms.csv ไว้ด้วยคำสั่ง `setwd("mushroom folder path")`

```
# set location to csv file folder
setwd("D:/cstu/cstu141nam2/forecast")

#get mushroom data
mushroomData <- read.csv("mushrooms.csv")
```

รูปภาพ 1 code คำสั่งการนำเข้าข้อมูล mushroom.csv

2.1.2 ข้อมูล mushroom มีทั้งหมด 8124 sample 23 attribute ใช้คำสั่ง `dim("mushroom data")` ตรวจสอบ

```
> dim(mushroomData)
[1] 8124 23
```

รูปภาพ 2 code คำสั่ง dim

จากคำสั่ง `names(mushroomData)` ทำให้ทราบว่า attribute แต่ละตัวมีชื่อว่าอะไรบ้าง

```
> names(mushroomData)
[1] "edibility"      "cap_shape"      "cap_surface"    "cap_color"      "bruises"
[6] "odor"           "gill_attachment" "gill_spacing"    "gill_size"       "gill_color"
[11] "stalk_shape"    "stalk_root"     "stalk_surface_above_ring" "stalk_surface_below_ring" "stalk_color_above_ring"
[16] "stalk_color_below_ring" "veil_type"      "veil_color"      "ring_number"     "ring_type"
[21] "spore_print_color" "population"     "habitat"
```

รูปภาพ 3 ผลลัพธ์จากคำสั่ง names

จากคำสั่ง `summary(mushroomData)` ทำให้ทราบว่า ข้อมูลมีการใช้ตัวอักษรย่อแทนค่าใน attribute ซึ่งมีซ้ำกัน เช่น attribute `cap.color` และ `odor` มี attribute value คือ n เหมือนกัน ทำให้สับสนในการ analysis ข้อมูลได้ จึงได้ทำการปรับเปลี่ยนข้อมูลเพื่อให้แต่ละ attribute value อ่านง่ายขึ้น โดย rename (อ้างอิงชื่อจาก <https://www.kaggle.com/uciml/mushroom-classification/data>)

ใช้คำสั่ง `colnames(mushroomData)` เปลี่ยนชื่อ attribute column name

ใช้คำสั่ง `mushroomData <- mushroomData %>% map_df(function(x) as.factor(x))`

สร้าง dataframe แบบ factor และใช้คำสั่ง `levels(mushroomdata$attributename)`

```
> summary(mushroomData)
class      cap.shape cap.surface  cap.color  bruises  odor  gill.attachment gill.spacing gill.size  gill.color  stalk.shape
?:4208    b: 452    f:2320      n          f:4748  n      :3528    a: 210    c:6812    b:5612    b          :1728    e:3516
e:3916    c:  4     g:  4       g          t:3376  f      :2160    f:7914    w:1312    n:2512    p          :1492    t:4608
          f:3152    s:2556      e          :1500      s          : 576      w          :1202
          k: 828    y:3244      y          :1072      y          : 576      n          :1048
          s:  32      w          :1040      a          : 400      g          : 752
          x:3656      b          : 168      l          : 400      h          : 732
                                (other): 220      (other): 484                                (other):1170

stalk.root stalk.surface.above.ring stalk.surface.below.ring stalk.color.above.ring stalk.color.below.ring veil.type veil.color ring.number
?:2480    f: 552      f: 600      w          :4464    w          :4384    p:8124    n: 96    n: 36
b:3776    k:2372      k:2304      p          :1872    p          :1872      o: 96    o:7488
c: 556    s:5176      s:4936      g          : 576    g          : 576      w:7924    t: 600
e:1120    y:  24      y: 284      n          : 448    n          : 512      y:  8
r: 192      o          : 432    b          : 432
                                o          : 192      (other): 140      (other): 156

ring.type spore.print.color population habitat
e:2776    w          :2388    a: 384    d:3148
f:  48    n          :1968    c: 340    g:2148
l:1296    k          :1872    n: 400    l: 832
n:  36    h          :1632    s:1248    m: 292
p:3968    r          :  72    v:4040    p:1144
          b          :  48    y:1712    u: 368
          (other): 144      w: 192
```

`<- c("ชื่อที่ต้องการ", "...", "...")` สร้างชื่อเรียก attribute value ตามที่ต้องการ

รูปภาพ 4 code คำสั่ง summary และผลลัพธ์ก่อนแก้ไข name

```

# Rename the variables
colnames(mushroomData) <- c("edibility", "cap_shape", "cap_surface",
                             "cap_color", "bruises", "odor",
                             "gill_attachment", "gill_spacing", "gill_size",
                             "gill_color", "stalk_shape", "stalk_root",
                             "stalk_surface_above_ring", "stalk_surface_below_ring", "stalk_color_above_ring",
                             "stalk_color_below_ring", "veil_type", "veil_color",
                             "ring_number", "ring_type", "spore_print_color",
                             "population", "habitat")

# Defining the levels for the categorical variables
## We make each variable as a factor
mushroomData <- mushroomData %>% map_df(function(.x) as.factor(.x))

## We redefine each of the category for each of the variables
levels(mushroomData$edibility) <- c("edible", "poisonous")
levels(mushroomData$cap_shape) <- c("bell", "conical", "flat", "knobbed", "sunken", "convex")
levels(mushroomData$cap_color) <- c("buff", "cinnamon", "red", "gray", "brown", "pink",
                                     "green", "purple", "white", "yellow")
levels(mushroomData$cap_surface) <- c("fibrous", "grooves", "scaly", "smooth")
levels(mushroomData$bruises) <- c("no", "yes")
levels(mushroomData$odor) <- c("almond", "creosote", "foul", "anise", "musty", "none", "pungent", "spicy", "fishy")
levels(mushroomData$gill_attachment) <- c("attached", "free")
levels(mushroomData$gill_spacing) <- c("close", "crowded")
levels(mushroomData$gill_size) <- c("broad", "narrow")
levels(mushroomData$gill_color) <- c("buff", "red", "gray", "chocolate", "black", "brown", "orange",
                                     "pink", "green", "purple", "white", "yellow")
levels(mushroomData$stalk_shape) <- c("enlarging", "tapering")
levels(mushroomData$stalk_root) <- c("missing", "bulbous", "club", "equal", "rooted")
levels(mushroomData$stalk_surface_above_ring) <- c("fibrous", "silky", "smooth", "scaly")
levels(mushroomData$stalk_surface_below_ring) <- c("fibrous", "silky", "smooth", "scaly")
levels(mushroomData$stalk_color_above_ring) <- c("buff", "cinnamon", "red", "gray", "brown", "pink",
                                                  "green", "purple", "white", "yellow")
levels(mushroomData$stalk_color_below_ring) <- c("buff", "cinnamon", "red", "gray", "brown", "pink",
                                                  "green", "purple", "white", "yellow")
levels(mushroomData$veil_type) <- "partial"
levels(mushroomData$veil_color) <- c("brown", "orange", "white", "yellow")
levels(mushroomData$ring_number) <- c("none", "one", "two")
levels(mushroomData$ring_type) <- c("evanescent", "flaring", "large", "none", "pendant")
levels(mushroomData$spore_print_color) <- c("buff", "chocolate", "black", "brown", "orange",
                                             "green", "purple", "white", "yellow")
levels(mushroomData$population) <- c("abundant", "clustered", "numerous", "scattered", "several", "solitary")
levels(mushroomData$habitat) <- c("wood", "grasses", "leaves", "meadows", "paths", "urban", "waste")

```

รูปภาพ 5 code คำสั่ง แก้ไขชื่อ attribute name, attribute value

จากคำสั่ง summary บอกจำนวนของ attribute value มาให้ว่ามีจำนวนกี่ตัว

เนื่องจาก ข้อมูลเป็นชนิด factor เลยไม่มีการคิดค่า max,min หรือ basic statistic ตัวอื่นมาให้ ทำให้ไม่

ทราบว่า มี missing value หรือไม่ ตรวจสอบโดยใช้คำสั่ง map\_dbl(mushroomData, function(x)

{sum(is.na(x))}) จากผลลัพธ์ไม่พบ missing value

```
-- summary(mushroomData)
edibility      cap_shape      cap_surface      cap_color      bruises      odor      gill_attachment      gill_spacing      gill_size
edible :4208    bell : 452    fibrous:2320    brown :2284    no :4748    none :3528    attached: 210    close :6812    broad :5612
poisonous:3916  conical: 4    grooves: 4    gray :1840    yes:3376    foul :2160    free :7914    crowded:1312  narrow:2512
               flat :3152    scaly :2556    red :1500           spicy : 576
               knobby: 828    smooth:3244    yellow:1072        fishy : 576
               sunken : 32           white :1040        almond : 400
               convex :3656           buff : 168         anise : 400
                               (other): 220        (other): 484

gill_color      stalk_shape      stalk_root      stalk_surface_above_ring      stalk_surface_below_ring      stalk_color_above_ring      stalk_color_below_ring
buff :1728      enlarging:3516    missing:2480    fibrous: 552    fibrous: 600    purple :4464    purple :4384
pink :1492      tapering :4608    bulbous:3776    silky :2372    silky :2304    green :1872    green :1872
white :1202           club : 556    smooth:5176    smooth :4936    gray : 576    gray : 576
brown :1048           equal :1120    scaly : 24    scaly : 284    brown : 448    brown : 512
gray : 752           rooted : 192           buff : 432    buff : 432
chocolate: 732           pink : 192    pink : 192
(other) :1170           (other): 140    (other): 156

veil_type      veil_color      ring_number      ring_type      spore_print_color      population      habitat
partial:8124    brown : 96    none: 36    evanescent:2776    white :2388    abundant : 384    wood :3148
               orange: 96    one :7488    flaring : 48    brown :1968    clustered: 340    grasses:2148
               white :7924    two : 600    large :1296    black :1872    numerous : 400    leaves : 832
               yellow: 8           none : 36    chocolate:1632    scattered:1248    meadows: 292
                               pendant :3968    green : 72    several :4040    paths :1144
                               buff : 48    solitary :1712    urban : 368
                               (other) : 144    waste : 192
```

รูปภาพ 6 code คำสั่ง summary และ ผลลัพธ์หลังแก้ไข name

```
> map_dbl(mushroomData, function(x) {sum(is.na(x))})
edibility      cap_shape      cap_surface      cap_color      bruises
0              0              0              0              0
odor           gill_attachment      gill_spacing      gill_size      gill_color
0              0              0              0              0
stalk_shape     stalk_root      stalk_surface_above_ring      stalk_surface_below_ring      stalk_color_above_ring
0              0              0              0              0
stalk_color_below_ring      veil_type      veil_color      ring_number      ring_type
0              0              0              0              0
spore_print_color      population      habitat
0              0              0
```

รูปภาพ 7 ผลลัพธ์คำสั่ง map\_dbl(mushroomData, function(x) {sum(is.na(x))})

## 2.2. การแบ่งข้อมูลเพื่อ Train และ Test แบบจำลอง (Data partitioning)

อธิบายหลักในการแบ่งข้อมูลเป็นชุดสำหรับ train และ test แบบจำลอง แนว code ประกอบ

แบ่งเป็น traindata 70%(5686 sample) testdata 30 % (2438 ที่ไม่ซ้ำกับ traindata) โดยใช้คำสั่ง set.seed เพื่อให้ค่า random เท่าเดิม และ recheck ได้ง่ายขึ้น

ใช้คำสั่ง sample(start:end,number ที่จะใช้) ในการสร้างเลขสุ่มขึ้นมา

ใช้คำสั่ง mushroomData[sample,] เพื่อสร้าง traindata และ mushroomData[-sample,] เพื่อสร้าง test data

```
#set seed for reproduce result
set.seed(142)

#partiiton data to testset 30 %(2438),trainset 70%(5686)
sample <- sample(1:8124,5686)
trainset <- mushroomData[sample,]
testset <- mushroomData[-sample,]
```

รูปภาพ 8 คำสั่ง setseed,sample ในการแบ่ง partition

```
> trainset
# A tibble: 5,686 x 23
  edibility cap_shape cap_surface cap_color bruises odor gill_attachment gill_spacing gill_size gill_color stalk_shape stalk_root
  <fct>      <fct>      <fct>      <fct>      <fct> <fct> <fct>      <fct>      <fct>      <fct>      <fct>
1 poisonous knobbed smooth red no spicy free close narrow buff tapering missing
2 poisonous flat scaly buff yes foul free close broad white tapering bulbous
3 edible bell scaly white no none free crowded broad white enlarging missing
4 poisonous flat fibrous yellow no foul free close broad pink enlarging bulbous
5 poisonous flat smooth red no spicy free close narrow buff tapering missing
6 poisonous flat smooth red no fishy free close narrow buff tapering missing
7 edible convex fibrous red yes none free close broad purple tapering bulbous
8 poisonous flat scaly white yes foul free close broad pink tapering bulbous
9 edible flat smooth gray yes none free close broad white tapering bulbous
10 edible convex fibrous gray yes none free close broad brown tapering bulbous
# ... with 5,676 more rows, and 11 more variables: stalk_surface_above_ring <fct>, stalk_surface_below_ring <fct>, stalk_color_above_ring <fct>,
# stalk_color_below_ring <fct>, veil_type <fct>, veil_color <fct>, ring_number <fct>, ring_type <fct>, spore_print_color <fct>,
# population <fct>, habitat <fct>

> testset
# A tibble: 2,438 x 23
  edibility cap_shape cap_surface cap_color bruises odor gill_attachment gill_spacing gill_size gill_color stalk_shape stalk_root
  <fct>      <fct>      <fct>      <fct>      <fct> <fct> <fct>      <fct>      <fct>      <fct>      <fct>
1 edible convex scaly yellow yes almond free close broad black enlarging club
2 edible bell scaly white yes almond free close broad gray enlarging club
3 poisonous convex smooth white yes pungent free close narrow black enlarging equal
4 edible bell smooth yellow yes anise free close broad black enlarging club
5 edible bell smooth white yes almond free close broad white enlarging club
6 edible bell scaly white yes anise free close broad gray enlarging club
7 edible convex smooth yellow yes almond free close broad brown enlarging club
8 edible bell scaly yellow yes anise free close broad gray enlarging club
9 poisonous convex smooth white yes pungent free close narrow black enlarging equal
10 edible sunken fibrous gray no none free close narrow black enlarging equal
# ... with 2,428 more rows, and 11 more variables: stalk_surface_above_ring <fct>, stalk_surface_below_ring <fct>, stalk_color_above_ring <fct>,
# stalk_color_below_ring <fct>, veil_type <fct>, veil_color <fct>, ring_number <fct>, ring_type <fct>, spore_print_color <fct>,
# population <fct>, habitat <fct>
```

รูปภาพ 9 ผลลัพธ์ที่ได้จากการแบ่ง partition ของ train,test data

### 2.3. การเลือก Attribute เพื่อสร้างแบบจำลอง (Attribute selection)

อธิบายกระบวนการในการเลือก attribute ที่มีความสำคัญในการสร้างแบบจำลอง ใช้หลักการอะไร ดำเนินการอย่างไร แขนง code ประกอบ ได้ผลอย่างไร

ใช้หลักการหา informationGain คู่อข้อมูลแต่ละคู่ของ attribute ประกอบกัน

โดยถ้าคู่ไหนให้ค่าที่สูงที่สุด จะนำ attribute คู่ นั้นไปสร้าง tree

จากรูปภาพที่ 10 เปรียบเทียบผล คู่ gill\_color และ odor ให้ค่าสูงสุด จะนำคู่ นั้นไปสร้าง tree

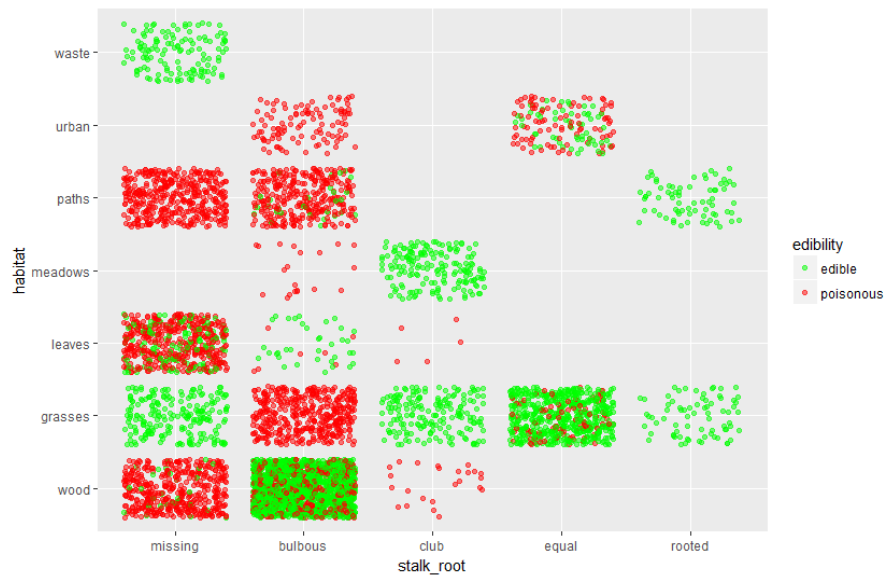
```
> InformationGain(table(labtrainset[,c('stalk_root','gill_color'))))  
[1] 0.8090094  
> InformationGain(table(labtrainset[,c('gill_color','odor'))))  
[1] 0.867484  
> InformationGain(table(labtrainset[,c('habitat','stalk_root'))))  
[1] 0.6724674  
> |
```

*รูปภาพ 10 ผลลัพธ์ที่ได้จากการหา informationGain*

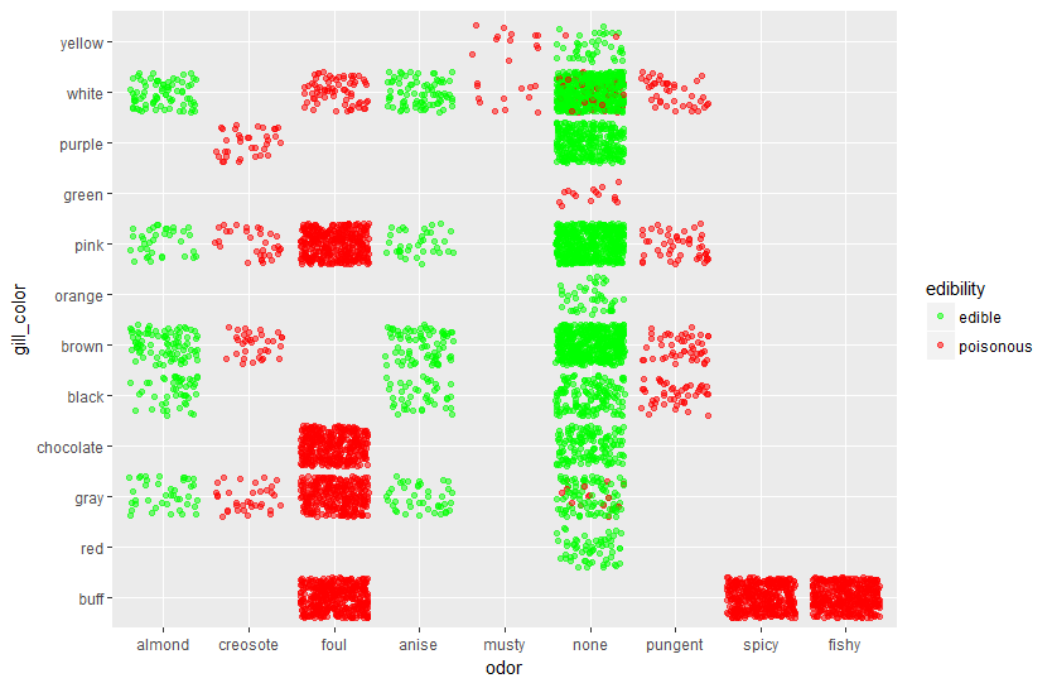
### 2.4. การแสดงภาพเกี่ยวกับ Attribute ที่เลือก (Attribute visualization)

อธิบายสิ่งที่ทำให้เข้าใจความสัมพันธ์ของ attribute ที่ได้จากข้อ 2.3 กับค่าที่ต้องการพยากรณ์ให้มากขึ้น แขนง code ประกอบ และระบุผลที่ได้

จากภาพที่ 11 และ 12 จะเห็นได้ว่าภาพที่ 12 มีการแบ่งกลุ่มข้อมูลได้ชัดเจนกว่าภาพที่ 11 (จากการสังเกตด้วยตาเปล่า) ซึ่งตรงกับค่า informationGain ในข้อ 2.3 คู่ odor และ gill\_color ให้ค่า 0.867 ซึ่งสูงกว่า คู่ stalk\_root และ habitat ให้ค่า 0.672



รูปภาพ 11 ผลลัพธ์ที่ได้จากการ plot ค่า attribute คู่ Stalk\_root และ habitat



รูปภาพ 12 ผลลัพธ์ที่ได้จากการ plot ค่า attribute คู่ gill\_color และ odor

2.5. Classification ด้วย Decision Tree (Classification with Decision Tree) แบบที่ได้เรียนมาในชม. บรรยาย

2.5.1. ใช้วิธีการเขียนฟังก์ชันต่างๆ เพื่อสร้าง decision tree เอง อาศัย data.tree package (อ้างอิงจาก Lab)

อธิบายขั้นตอนการสร้างแบบจำลองด้วยวิธีการแบบที่ทำใน lab แขนง code ประกอบ และแสดง decision tree ที่ได้ รวมทั้งผลการทดสอบด้วย

ใช้ library data.tree ในการสร้าง tree model

สร้าง tree model ด้วย function TrainID3 โดยใช้ function information gain, Entropy, isPure ประกอบ ทดสอบ tree model ด้วย TreePredict

ผลลัพธ์จากการสร้าง tree แสดงดังรูปภาพที่ 13

ทดสอบโมเดลโดยส่ง record ที่ 1 ของ testset ไปให้ โดยผลการทำนายสามารถทำนายได้ถูกต้อง

```
IsPure <- function(data){
  length(unique(data[,1])) == 1
}
Entropy <- function(vls) {
  res <- vls/sum(vls) * log2(vls/sum(vls))
  res[vls == 0] <- 0
  -sum(res)
}
InformationGain <- function(tbl){
  tbl <- as.data.frame.matrix(tbl)
  entropyBefore <- Entropy(colSums(tbl))
  s <- rowSums(tbl)
  entropyAfter <- sum(s/sum(s) * apply(tbl,MARGIN = 1,FUN = Entropy))
  informationGain <- entropyBefore - entropyAfter
  return(informationGain)
}
```

รูปภาพ 13 code คำสั่ง Ispure,Entropy,InformationGain



```

TrainID3 <- function(node, data) {
  node$obsCount <- nrow(data)
  #if the data-set is pure (e.g. all toxic), then
  if (IsPure(data)) {
    #construct a leaf having the name of the pure feature (e.g. 'toxic')
    child <- node$AddChild(unique(data[,1]))
    node$feature <- tail(names(data), 1)
    child$obsCount <- nrow(data)
    child$feature <- ''
  } else {
    #chose the feature with the highest information gain (e.g. 'color')
    ig <- sapply(colnames(data)[-ncol(data)],
      function(x) InformationGain(
        table(data[,x], data[,ncol(data)])
      )
    )
    feature <- names(ig)[ig == max(ig)][1]
    node$feature <- feature

    #take the subset of the data-set having that feature value
    childobs <- split(data[,!(names(data) %in% feature)],
      data[,feature], drop = TRUE)

    for(i in 1:length(childobs)) {
      #construct a child having the name of that feature value (e.g. 'red')
      child <- node$AddChild(names(childobs)[i])
      #call the algorithm recursively on the child and the subset
      TrainID3(child, childobs[[i]])
    }
  }
}

```

รูปภาพ 14 code คำสั่ง TrainID3 สำหรับสร้าง tree

```

TreePredict <- function(tree, features){
  if(tree$children[[1]]$isLeaf)
    return (tree$children[[1]]$name)
  child <- tree$children[[features[[tree$feature]]]]
  return (TreePredict(child, features))
}

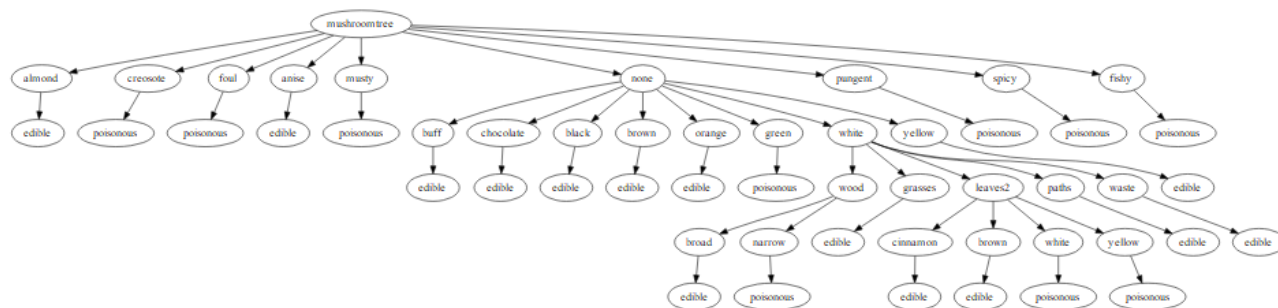
tree_lab <- Node$new("mushroomtree")
#mushsample <- caret::createDataPartition(y = mushroomData$edibility, times = 1, p = 0.8, list = FALSE)
labtrainset <- as.data.frame(trainset)
TrainID3(tree_lab, labtrainset)
print(tree_lab, "feature", "obsCount")
TreePredict(tree_lab, testset[1, -1])

```

รูปภาพ 15 code คำสั่ง TreePredict สำหรับทดสอบ tree

	levelName	feature	obsCount
1	mushroomtree	stalk_root	5686
2	<U+00A6>--missing	gill_color	1743
3	<U+00A6> <U+00A6>--buff	habitat	1217
4	<U+00A6> <U+00A6> <U+00B0>--poisonous		
5	<U+00A6> <U+00A6>--red	habitat	60
6	<U+00A6> <U+00A6> <U+00B0>--edible		60
7	<U+00A6> <U+00A6>--gray	habitat	73
8	<U+00A6> <U+00A6> <U+00B0>--edible		73
9	<U+00A6> <U+00A6>--chocolate	habitat	6
10	<U+00A6> <U+00A6> <U+00B0>--edible		6
11	<U+00A6> <U+00A6>--brown	habitat	47
12	<U+00A6> <U+00A6> <U+00B0>--edible		47
13	<U+00A6> <U+00A6>--orange	habitat	44
14	<U+00A6> <U+00A6> <U+00B0>--edible		44
15	<U+00A6> <U+00A6>--pink	habitat	70
16	<U+00A6> <U+00A6> <U+00B0>--edible		70
17	<U+00A6> <U+00A6>--purple	habitat	6
18	<U+00A6> <U+00A6> <U+00B0>--edible		6
19	<U+00A6> <U+00A6>--white	population	178
20	<U+00A6> <U+00A6> <U+00A6>--clustered	habitat	68
21	<U+00A6> <U+00A6> <U+00A6> <U+00B0>--edible		68
22	<U+00A6> <U+00A6> <U+00A6>--numerous	habitat	38
23	<U+00A6> <U+00A6> <U+00A6> <U+00B0>--edible		38
24	<U+00A6> <U+00A6> <U+00A6>--scattered	habitat	39
25	<U+00A6> <U+00A6> <U+00A6> <U+00B0>--edible		39
26	<U+00A6> <U+00A6> <U+00A6>--several	edibility	29
27	<U+00A6> <U+00A6> <U+00A6> <U+00A6>--edible	cap_shape	4
28	<U+00A6> <U+00A6> <U+00A6> <U+00A6> <U+00A6>--flat	habitat	2
29	<U+00A6> <U+00A6> <U+00A6> <U+00A6> <U+00A6> <U+00B0>--smooth		2

รูปภาพ 16 ผลลัพธ์ที่ได้จากการ print tree



รูปภาพ 17 ผลลัพธ์ที่ได้จากการ plot tree

```

> TreePredict(tree_lab,testset[1,-1])
[1] "edible"
> testset[1,-1]
# A tibble: 1 x 22
  cap_shape cap_surface cap_color bruises odor  gill_attachment gill_spacing gill_size gill_color stalk_shape stalk_root
  <fct>      <fct>      <fct>      <fct> <fct>  <fct>      <fct>      <fct>      <fct>      <fct>      <fct>
1 convex    scaly      yellow    yes  almond free    close      broad    black    enlarging club
# ... with 11 more variables: stalk_surface_above_ring <fct>, stalk_surface_below_ring <fct>, stalk_color_above_ring <fct>,
# stalk_color_below_ring <fct>, veil_type <fct>, veil_color <fct>, ring_number <fct>, ring_type <fct>,
# spore_print_color <fct>, population <fct>, habitat <fct>
> testset[1,]
# A tibble: 1 x 23
  edibility cap_shape cap_surface cap_color bruises odor  gill_attachment gill_spacing gill_size gill_color stalk_shape
  <fct>      <fct>      <fct>      <fct> <fct> <fct>  <fct>      <fct>      <fct>      <fct>      <fct>
1 edible    convex    scaly      yellow    yes  almond free    close      broad    black    enlarging
# ... with 12 more variables: stalk_root <fct>, stalk_surface_above_ring <fct>, stalk_surface_below_ring <fct>,
# stalk_color_above_ring <fct>, stalk_color_below_ring <fct>, veil_type <fct>, veil_color <fct>, ring_number <fct>,
# ring_type <fct>, spore_print_color <fct>, population <fct>, habitat <fct>
> |

```

รูปภาพ 18 ผลลัพธ์ที่ได้จากการทดสอบ *tree* ด้วย *record* ที่ 1 ของ *testdata*

## 2.5.2. ใช้วิธีการเรียกไลบรารีสำเร็จรูปในการสร้าง decision tree เช่น rpart package

อธิบายขั้นตอนการสร้างแบบจำลองด้วยการเรียกใช้ฟังก์ชันในไลบรารีสำเร็จรูป แบบ code ประกอบ อธิบายที่มาของค่าพารามิเตอร์ต่างๆ ที่ใช้ และแสดง decision tree ที่ได้ รวมทั้งผลการทดสอบด้วย

ใช้ library(rpart) ในการสร้างแบบจำลอง decision tree และ ใช้ library(rpart.plot) ในการ plot result graph

สร้าง model ด้วยคำสั่ง rpart(edibility ~ ., data = trainset, method = "class") พารามิเตอร์ที่ใช้ formula คือ target : edibility ~ all attribute except edibility,

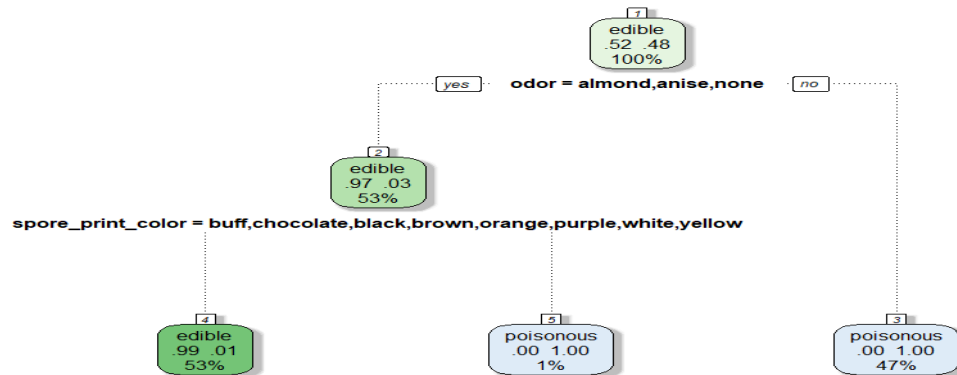
data คือ trainset ที่ใช้ฝึกฝน model, method = "class" คือ ข้อมูลที่ใช้เป็น factor

ทดสอบ model ด้วยคำสั่ง predict(model\_tree, testset, type = "class") พารามิเตอร์ที่ใช้ model\_tree คือ model ที่ได้จากการ train model, testset คือ ข้อมูลที่ใช้ทดสอบ , type = "class" คือใช้ข้อมูลแบบ factor

ดูผลการทดสอบ model ด้วยคำสั่ง caret::confusionMatrix(data=test\_tree, reference = testset\$edibility, positive="edible") caret คือ library สำหรับสร้าง confusionmatrix , data= test\_tree คือ ผลการทดสอบ model, reference = testset\$edibility คือข้อมูลจริงที่ใช้เปรียบเทียบ , positive="edible" คือ ข้อมูลที่ใช้เป็นค่า positive ใน confusionmatrix

ผลการทดสอบพบว่า มีความถูกต้อง 99.55%

ภาพผล decision tree แสดงรูปภาพที่ 19



รูปภาพ 19 ผลลัพธ์ที่ได้จากการสร้าง tree

```
#classify by decision-tree library
library(rpart)
library(rpart.plot)

set.seed(143)
model_tree <- rpart(edibility ~ ., data = trainset, method = "class")

#test tree
test_tree <- predict(model_tree, testset, type = "class")
caret::confusionMatrix(data=test_tree, reference = testset$edibility, positive="edible")

#plot result tree
rpart.plot(model_tree, extra = 104, box.palette = "GnBu", branch.lty = 3, shadow.col = "gray", nn = TRUE)
```

รูปภาพ 20 คำสั่งการสร้าง decision tree ทดสอบ plot tree

```
Confusion Matrix and Statistics

              Reference
Prediction edible poisonous
edible      1252         11
poisonous     0        1175

      Accuracy : 0.9955
      95% CI   : (0.9919, 0.9977)
No Information Rate : 0.5135
P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.991
McNemar's Test P-Value : 0.002569

      Sensitivity : 1.0000
      Specificity : 0.9907
      Pos Pred Value : 0.9913
      Neg Pred Value : 1.0000
      Prevalence : 0.5135
      Detection Rate : 0.5135
      Detection Prevalence : 0.5180
      Balanced Accuracy : 0.9954

      'Positive' Class : edible
```

รูปภาพ 21 ผลลัพธ์ที่ได้จาก confusion matrix

2.6. สรุปองค์ความรู้ที่ได้จากการใช้แบบจำลองในการแก้ปัญหา และสิ่งที่ได้เรียนรู้เกี่ยวกับกระบวนการในการใช้ข้อมูลแก้ปัญหาจากการบ้านนี้

ใช้องค์ความรู้ด้านการทำ machine learning, data mining เช่น การแบ่ง partition ของ data การสร้าง model ทั้งสร้างจาก package r part หรือ code จาก ตัวอย่างใน lab ได้เรียนรู้วิธีการแก้ปัญหาเกี่ยวกับ syntax ของ r ได้ลองใช้ function, package ใหม่ๆ ของ r ได้ฝึกภาษาอังกฤษในการ debug code และอ่าน document English