

## ดำเนินการและเขียนรายงานผลการดำเนินการ ประกอบด้วยหัวข้อดังนี้

### 2.1. การเตรียมชุดข้อมูล (Data acquisition)

อธิบายว่าน.ศ.อ่านข้อมูลเข้ามาอย่างไร ทำอะไรบ้างเพื่อศึกษาให้เข้าใจข้อมูล และเข้าใจข้อมูลว่าอย่างไรบ้าง แนบ code ประกอบ

เริ่มจากดาวน์โหลด ไฟล์ จากแหล่งข้อมูลนำไปไว้ที่ **path** ของโปรแกรม **R** และนำข้อมูลเก็บไว้ในตัวแปร **mushroom** โดยใช้คำสั่ง `read_csv("mushrooms.csv")` เป็นข้อมูลเกี่ยวกับลักษณะต่างๆของเห็ดว่ามีรูปทรง สี กลิ่น ฯลฯ แบบใดถึงเป็นเห็ดพิษ หรือเห็ดที่สามารถรับประทานได้

```
9 library( rpart )
10 mushroom <- read_csv("mushrooms.csv")
11 mushroom
12 <

10:1 (Top Level) ↕ R Sc

Console Terminal x
~/
> .default = col_character()
> )
See spec(...) for full column specifications.
> mushroom
# A tibble: 8,124 x 23
  class capshape capsurface capcolor bruises odor gillattachment gillspacing gillsize
  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
1 p x s n t p f c n
2 e x s y t a f c b
3 e b s w t l f c b
4 p x y w t p f c n
5 e x s g f n f w b
6 e x y y t a f c b
7 e b s w t a f c b
8 e b y w t l f c b
9 p x y w t p f c n
10 e b s y t a f c b
# ... with 8,114 more rows, and 14 more variables: gillcolor <chr>, stalkshape <chr>,
# stalkroot <chr>, stalksurfaceabovering <chr>, stalksurfacebelowring <chr>,
# stalkcolorabovering <chr>, stalkcolorbelowring <chr>, veiltype <chr>, veilcolor <chr>.
```

### 2.2. การแบ่งข้อมูลเพื่อ Train และ Test แบบจำลอง (Data partitioning) อธิบายหลักในการแบ่งข้อมูลเป็นชุดสำหรับ train และ test แบบจำลอง แนบ code ประกอบ

ใช้ฟังก์ชัน `partitionData` จากแหล่งข้อมูล <https://www.kaggle.com/mokosan/mushroom-classification> เพื่อเป็นการจัดระเบียบ data set เพื่อให้งานต้องการทำขั้นตอนต่อไปฟังก์ชันการแบ่งข้อมูลใช้กรอบข้อมูลและจะมีค่าทศนิยมระหว่าง 0 ถึง 1 ซึ่งแสดงถึงส่วนของข้อมูลที่เหมาะสมกับข้อมูลที่จะกลายเป็นพารามิเตอร์ที่ใช้ในการคำนวณต่อไป

โดย Train จะแบ่งไว้จำลองโมเดลและ Test จะถูกแบ่งไว้เพื่อใช้ทดสอบโมเดล

```

partitionData <- function( data, fractionOfDataForTrainingData = 0.6 )
{
  numberOfRows <- nrow( data )
  randomRows <- runif( numberOfRows )
  flag <- randomRows <= fractionOfDataForTrainingData
  trainingData <- data[ flag, ]
  testingData <- data[ !flag, ]
  datasetsplit <- list( trainingData = trainingData, testingData = testingData )
  datasetsplit
}
partitionedData <- partitionData( mushroom )
trainingData <- partitionedData$trainingData
testingData <- partitionedData$testingData

```

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

อธิบายกระบวนการในการเลือก attribute ที่มีความสำคัญในการสร้างแบบจำลอง ใช้หลักการอะไร ดาเนินการอย่างไร แบบ code ประกอบ ได้ผลอย่างไร

เป็นการเลือกตัวแทนที่มีอัตราส่วนที่มีผลต่อส่วนใหญ่และมีคุณลักษณะเฉพาะจากข้อมูลที่เป็นส่วนหนึ่งของสคีมาของเรา กับโมเดล โดยจากการใช้ ฟังก์ชัน GainRatioAttributeEval แล้วก็สามารถรู้ได้ว่า odor หรือกลิ่นมีค่าที่ใกล้เคียงที่สุด (0.3878530) ทำให้ค่านี้มีอิทธิพลต่อการสร้างที่สุุดจึงเลือก attribute นี้ในการสร้างโมเดล

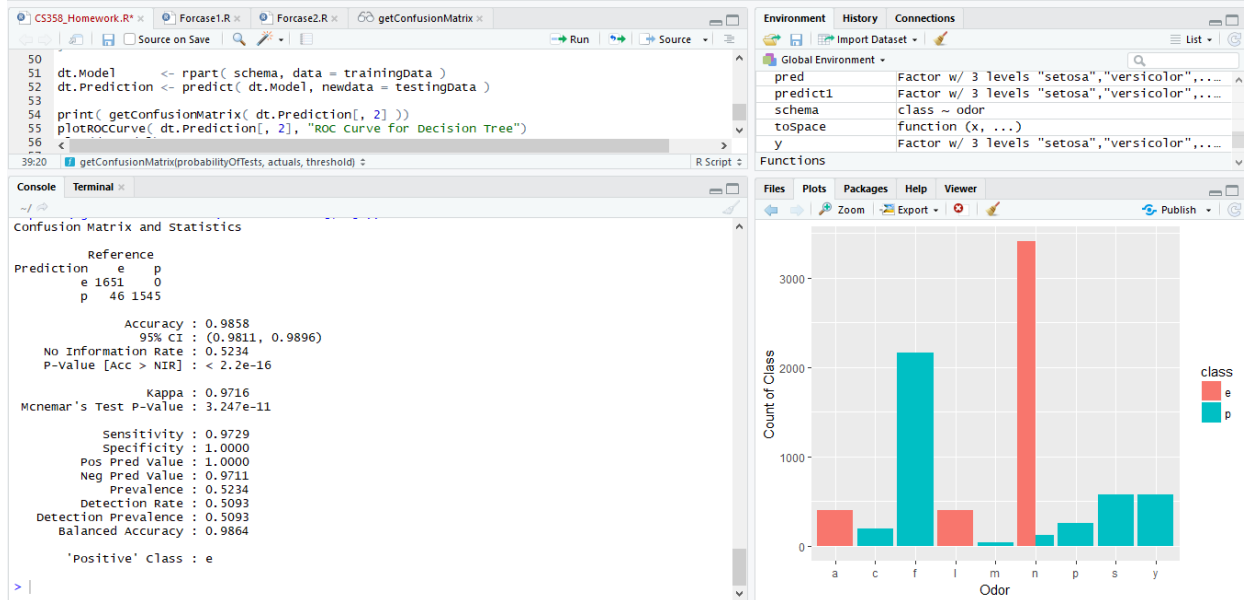
```
gainRatioResult <- GainRatioAttributeEval( class ~ . , data = mushroom )
```

```
print( sort( gainRatioResult, decreasing = TRUE ))
```

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

อธิบายสิ่งที่ทำเพื่อให้เข้าใจความสัมพันธ์ของ attribute ที่ได้จากข้อ 2.3 กับค่าที่ต้องการพยากรณ์ให้มากขึ้น แบบ code ประกอบ และระบุผลที่ได้

จากข้อ 2.3 จำได้นำกลิ่นกับ ชนิดของเห็ด(class) มาคำนวณหา ConfusionMatrix เพื่อหาค่าประมาณที่จะใช้ในการสร้างโมเดลดังนี้



ซึ่งทำให้เห็นว่ากลิ่นกับชนิดของเห็ดมีความสัมพันธ์กัน แสดงค่าความแม่นยำถึง 98.58 เปอร์เซ็นต์ (kappa = 0.9716)

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

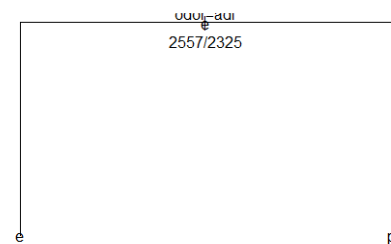
### 2.5.2. ใช้วิธีการเรียกไลบรารีสำเร็จรูปในการสร้าง decision tree เช่น rpart package

อธิบายขั้นตอนการสร้างแบบจำลองด้วยการเรียกใช้ฟังก์ชันในไลบรารีสำเร็จรูป แบบ code ประกอบ อธิบายที่มาของค่าพารามิเตอร์ต่างๆ ที่ใช้ และแสดง decision tree ที่ได้ รวมทั้งผลการทดสอบด้วย

```
plotROCcurve <- function( predictionResults, title, color = "red" )
{
  plot( roc( testingData$class, predictionResults, direction="<" ),
        print.auc=TRUE, col = color, lwd = 3, main = title )
}

dt.Model <- rpart( schema, data = trainingData )
dt.Prediction <- predict( dt.Model, newdata = testingData )

print( getConfusionMatrix( dt.Prediction[, 2] ) )
plotROCcurve( dt.Prediction[, 2], "ROC Curve for Decision Tree" )
plot( dt.Model )
text( dt.Model, use.n = TRUE, all = TRUE, cex = 1 )
```



## 2.6. สรุปองค์ความรู้ที่ได้จากการใช้แบบจำลองในการแก้ปัญหา และสิ่งที่ได้เรียนรู้เกี่ยวกับกระบวนการในการใช้ข้อมูลแก้ปัญหาจากการบ้านนี้

เป็นการจัดประเภท mushrooms กินได้หรือไม่ นอกจากนี้ยังตอบคำถามได้ว่า อะไรคือลักษณะสำคัญของเห็ดที่กินได้ และแบบจำลองต่างๆ สามารถจัดการ tree ได้อย่างถูกต้องสามารถแบ่งแยกการกินของเห็ดได้โดยมีพารามิเตอร์ 2 ตัวในการทำ confusion matrix การทำ Random Forest และ SVM