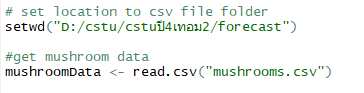
5709611635 นายธนกร สว่างโลก

**รายงานผลการดำเนินการ**

2.1. การเตรียมชุดข้อมูล (Data acquisition)

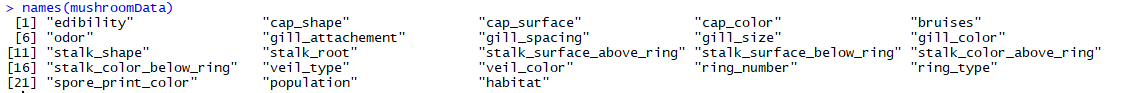
อธิบายว่าน.ศ.อ่านข้อมูลเข้ามาอย่างไร ทำอะไรบ้างเพื่อศึกษาให้เข้าใจข้อมูล และเข้าใจข้อมูลว่าอย่างไรบ้าง แนบ code ประกอบ

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

รูปภาพ 1 code คำสั่งการนำเข้าข้อมูล mushroom.csv

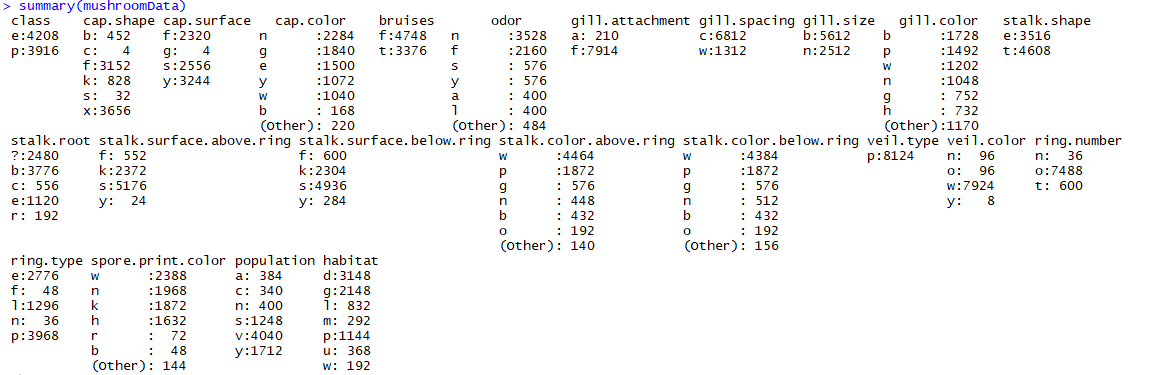
2.1.2 ข้อมูล mushroom มีทั้งหมด 8124 sample 23 attribute ใช้คำสั่ง dim(“mushroom data”) ตรวจสอบ

รูปภาพ 2 code คำสั่ง dim

จากคำสั่ง names(mushroomData) ทำให้ทราบว่า attribute แต่ละตัวมีชื่อว่าอะไรบ้าง

รูปภาพ 3 ผลลัพธ์จากคำสั่ง names

จากคำสั่ง summary(mushroomData) ทำให้ทราบว่า ข้อมูลมีการใช้ตัวอักษรย่อแทนคำใน 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)   
<- c(“ชื่อที่ต้องการ”,”…”,….) สร้างชื่อเรียก attribute value ตามที่ต้องการ

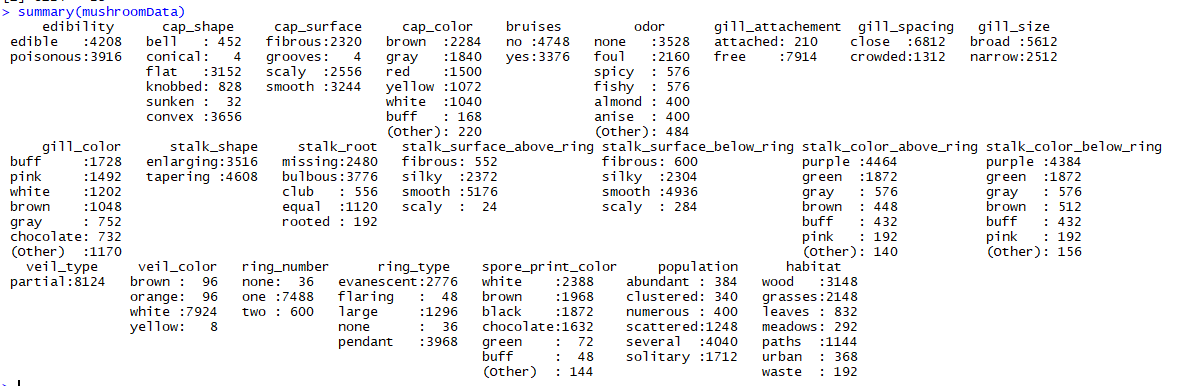
รูปภาพ 4 code คำสั่ง summary และผลลัพธ์ก่อนแก้ไข name



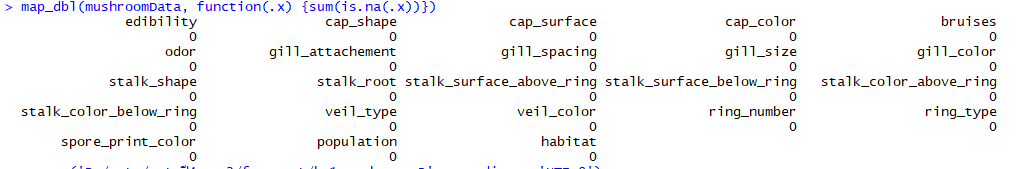
รูปภาพ 5 code คำสั่ง แก้ไขชื่อ attribute name,attribute value

จากคำสั่ง summary บอกจำนวนของ attribute value มาให้ว่ามีจำนวนกี่ตัว

เนื่องจาก ข้อมูลเป็นชนิด factor เลยไม่มีการคิดค่า max,min หรือ basic statistic ตัวอื่นมาให้ ทำให้ไม่ทราบว่ามี missing value หรือไม่ ตรวจสอบโดยใช้คำสั่ง map\_dbl(mushroomData, function(.x) {sum(is.na(.x))}) จากผลลัพธ์ไม่พบ missing value



รูปภาพ 6 code คำสั่ง summaryและ ผลลัพธ์หลังแก้ไข name



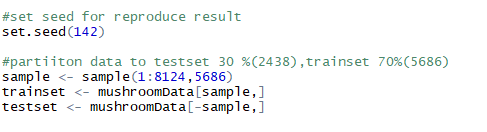
รูปภาพ 7 ผลลัพธ์คำสั่ง map\_dbl(mushroomData, function(.x) {sum(is.na(.x))})

2.2. การแบ่งข้อมูลเพื่อ Train และ Test แบบจำลอง (Data partitioning)

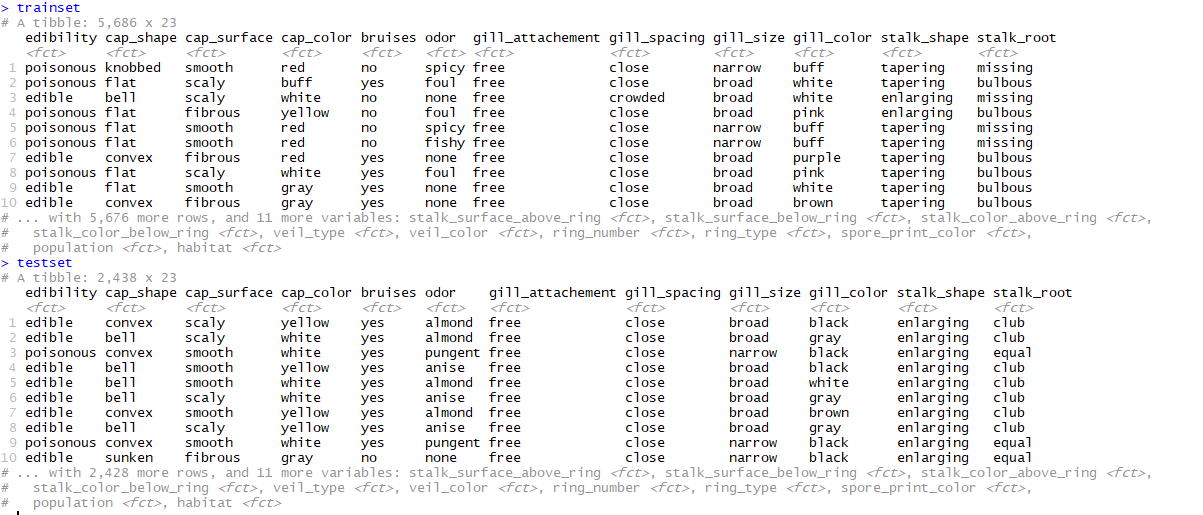
อธิบายหลักในการแบ่งข้อมูลเป็นชุดสำหรับ train และ test แบบจำลอง แนบ code ประกอบ

แบ่งเป็น traindata 70%(5686 sample) testdata 30 % (2438 ที่ไม่ซ้ำกับ traindata) โดย  
ใช้คำสั่ง 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

รูปภาพ 8 คำสั่ง setseed,sample ในการแบ่ง partition



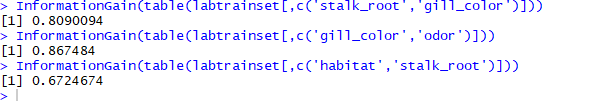
รูปภาพ 9 ผลลัพธ์ที่ได้จากการแบ่ง partition ของ train,test data

2.3. การเลือก Attribute เพื่อสร้างแบบจำลอง (Attribute selection)

อธิบายกระบวนการในการเลือก attribute ที่มีความสำคัญในการสร้างแบบจำลอง ใช้หลักการอะไร ดำเนินการอย่างไร แนบ code ประกอบ ได้ผลอย่างไร

ใช้หลักการหา informationGain ดูข้อมูลแต่ละคู่ของ attribute ประกอบกัน

โดยถ้าคู่ไหนให้ค่าที่สูงที่สุด จะนำ attribute คู่นั้นไปสร้าง tree

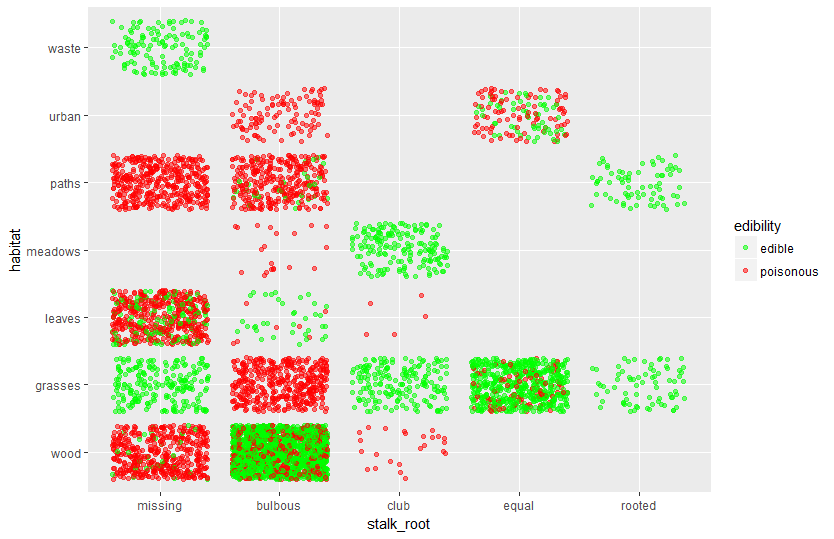
จากรูปภาพที่ 10 เปรียบเทียบผล คู่ gill\_color และ odor ให้ค่าสูงสุด จะนำคู่นั้นไปสร้าง tree

รูปภาพ 10 ผลลัพธ์ที่ได้จากการหา informationGain

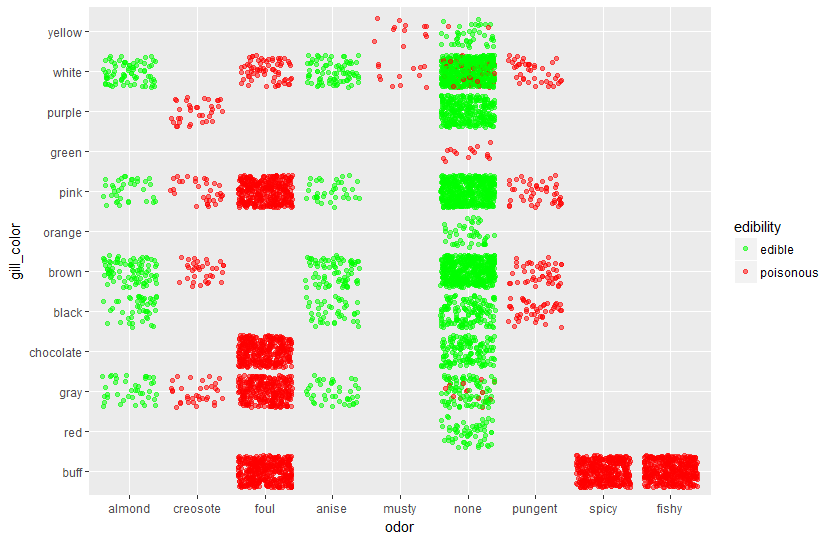
2.4. การแสดงภาพเกี่ยวกับ Attribute ที่เลือก (Attribute visualization)

อธิบายสิ่งที่ทำเพื่อให้เข้าใจความสัมพันธ์ของ attribute ที่ได้จากข้อ 2.3 กับค่าที่ต้องการพยากรณ์ให้มากขึ้น แนบ 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

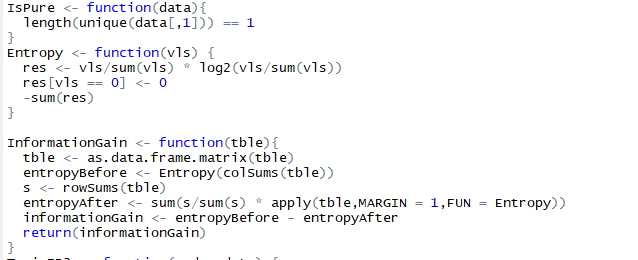
2.5.1. ใช้วิธีการเขียนฟังก์ชันต่างๆ เพื่อสร้าง decision tree เอง อาศัย data.tree package (อ้างอิงจาก Lab)

อธิบายขั้นตอนการสร้างแบบจำลองด้วยวิธีการแบบที่ทำใน lab แนบ code ประกอบ และแสดง decision tree ที่ได้ รวมทั้งผลการทดสอบด้วย

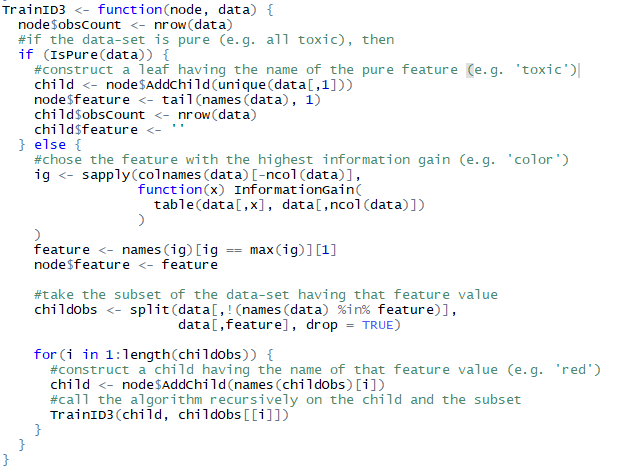
ใช้ library data.tree ในการสร้าง tree model

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

ผลลัพธ์จากการสร้าง tree แสดงดังรูปภาพที่ 13

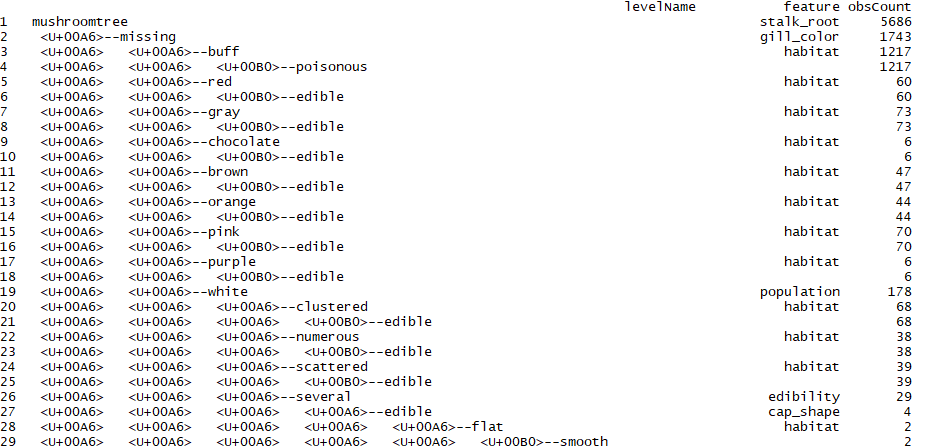
 ทดสอบโมเดลโดยส่ง record ที่ 1 ของ testset ไปให้ โดยผลการทำนายสามารถทำนายได้ถูกจ้อง

รูปภาพ 13 code คำสั่ง Ispure,Entropy,InformationGain

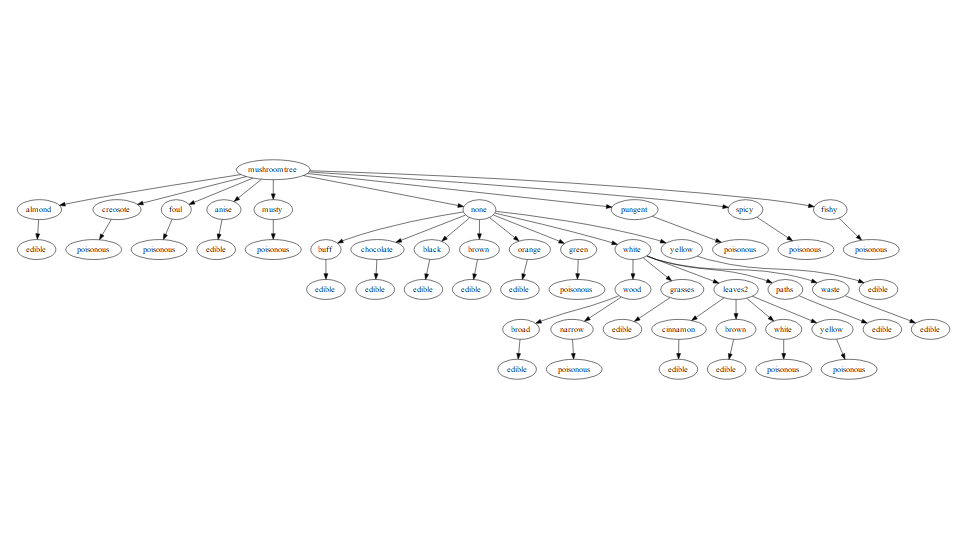
รูปภาพ 14 code คำสั่ง TrainID3 สำหรับสร้าง tree



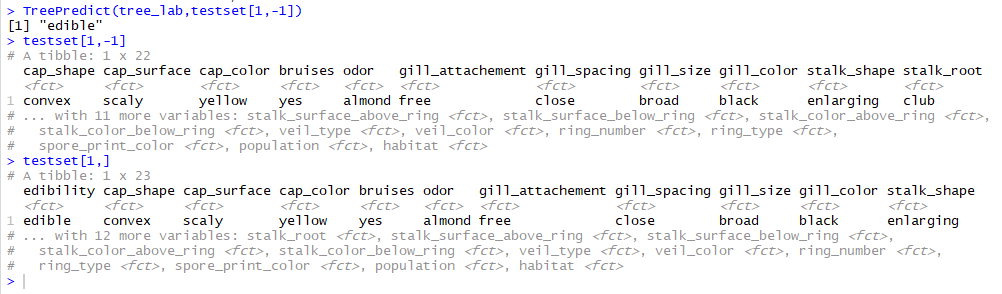
รูปภาพ 15 code คำสั่ง TreePredict สำหรับทดสอบ tree



รูปภาพ 16 ผลลัพธ์ที่ได้จากการ print tree



รูปภาพ 17 ผลลัพธ์ที่ได้จากการ plot tree



รูปภาพ 18 ผลลัพธ์ที่ได้จากการทดสอบ tree ด้วย record ที่ 1 ของ testdata

2.5.2. ใช้วิธีการเรียกไลบราลีสำเร็จรูปในการสร้าง decision tree เช่น rpart package

อธิบายขั้นตอนการสร้างแบบจำลองด้วยการเรียกใช้ฟังก์ชันในไลบราลีสำเร็จรูป แนบ code ประกอบ อธิบายที่มาของค่าพารามิเตอร์ต่างๆ ที่ใช้ และแสดง decision tree ที่ได้ รวมทั้งผลการทดสอบด้วย

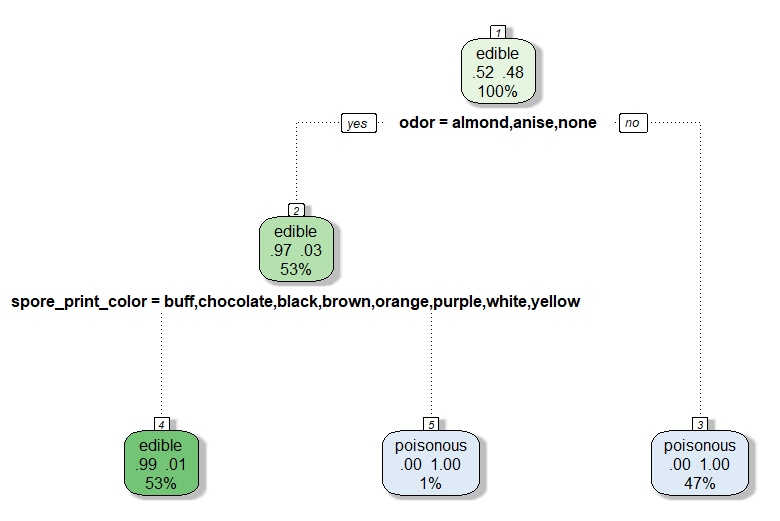
ใช้ library(rpart) ในการสร้างแบบจำลอง decision tree และ ใช้ library(rpart.plot) ในการ plot result graph

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

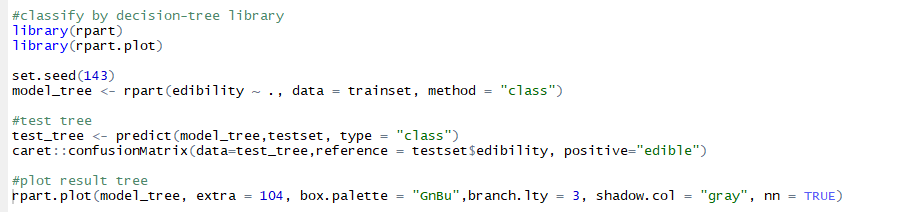
ทดสอบ model ด้วยคำสั่ง predict(model\_tree,testset, type = "class") พารามิเตอร์ที่ใช้ model\_tree คือ model ที่ได้จากการ train model, testset คือ ข้อมูลที่ใช้ทดสอบ ,type = “class” คือใช้ข้อมูลแบบ factor

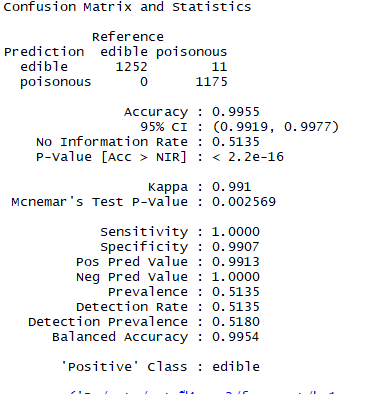
ดูผลการทดสอบ model ด้วยคำสั่ง caret::confusionMatrix(data=test\_tree,reference = testset$edibility, positive="edible") caret คือ library สำหรับสร้าง confusionmatrix ,data= test\_tree คือ ผลการทดสอบ model, reference = testset$edibility คือข้อมูลจริงที่ใช้เปรียบเทียบ , positive="edible" คือ ข้อมูลที่ใช้เป็นค่า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

ผลการทดสอบพบว่า มีความถูกต้อง 99.55%

 ภาพผล decision tree แสดงรูปภาพที่ 19

รูปภาพ 19 ผลลัพธ์ที่ได้จากการสร้าง tree



รูปภาพ 20 คำสั่งการสร้าง decision tree ทดสอบ plot tree

รูปภาพ 21 ผลลัพธ์ที่ได้จาก confusion matrix

2.6. สรุปองค์ความรู้ที่ได้จากการใช้แบบจำลองในการแก้ปัญหา และสิ่งที่ได้เรียนรู้เกี่ยวกับกระบวนการในการใช้ข้อมูลแก้ปัญหาจากการบ้านนี้

ใช้องค์ความรู้ด้านการทำ machine learning, data mining เช่น การแบ่ง partitionของdata 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