R軟體分類法整理

大綱

- R軟體分類方法簡介
- R軟體分類法預測結果
- R軟體分類法預測程式碼

R軟體分類方法簡介

R軟體分類方法整理

方法		套件	函數
決策樹	決策樹	{tree}	tree()
	C50分類樹	{C50}	C5.0()
	分類迴歸樹	{rpart}	rpart()
	條件推論樹	{party}	ctree()
	隨機森林	{randomForest}	randomForest()
類神經網路		{class}	knn()
支持向量機		{e1071}	svm()
貝氏分類		{e1071}	naiveBayes()

R軟體分類方法簡介(1/2)

決策樹	• 每個事件都可能引出兩個或多個事件，導致不同的結果。
C50分類樹	• 與決策樹分類法相似，差別在於選取特徵值不同。
分類迴歸樹	• 每個切分特徵與切分點共同決定了一個集合應該以怎樣的方式來切分子點。
條件推論樹	• 選擇分類變量時的依據是顯著性測量的結果，而不是採用訊息最大化法。

R軟體分類方法簡介(2/2)

隨機森林	• 是由多棵CART構成的，在訓練每棵樹的節點時，使用的特徵是從所有特徵中按照一定比例隨機地無放回的抽取。
類神經網路	• 模擬生物大腦神經的人工智慧系統，利用樣本去對神經網路做訓練，找出最接近的輸出值。
支持向量機	• 找出一個超平面(hyperplane)，使之將兩個不同的集合分開。
貝氏分類	• 透過機率的計算，用以判斷未知類別的資料應該屬於那一個類別。

R軟體分類法預測結果

預測資料欄位說明

104-105年第一階段指考成績

- 預測目的：錄取後學生未來註冊
- 資料欄位：國文、英文、數學、社會、自然、PR值、是否錄取
- 測試/訓練資料筆數：7560/7315
- 測試/訓練資料年度：104/105

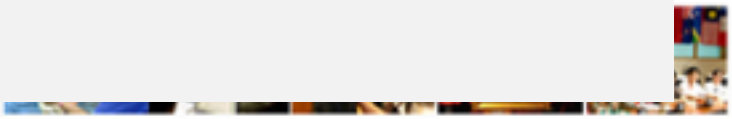
R軟體分類方法結果

方法		Recall rate	Precision rate	Accuracy	F1
決策樹	決策樹	61.6	0	61.6	0
	C50分類樹	41.7	62.1	59.5	49.9
	分類迴歸樹	64.6	0	64.6	0
	條件推論樹	61.6	0	61.6	0
	隨機森林	65.3	36.9	57.2	47.1
類神經網路		64.7	35.5	59.3	45.8
支持向量機		65.1	40	62.8	49.6
貝氏分類		70	43.5	59.5	53.6

R軟體分類法預測程式碼

資料精整程式碼

```
library(C50)
library(readxl)
a1026 <- read.csv("/Users/sharon/Documents/銘傳大學/專案研究/學生報到預測
/data/app1026完整.csv")
#全校為單位
z<-subset(a1026,a1026$年度==104, select =c(9:14,19))
x<-subset(a1026,a1026$年度==105, select =c(9:14,19))
a <- z[,-7]
b <- x[,-7]
#z分佈(常態分佈) ####
mina1<-apply(a,2,min)
maxa1<-apply(a,2,max)
a1 <- maxa1- mina1
minb1<-apply(b,2,min)
maxb1<-apply(b,2,max)
b2 <- maxb1-minb1
a <- cbind(a,z$註冊)
b <- cbind(b,x$註冊)
```

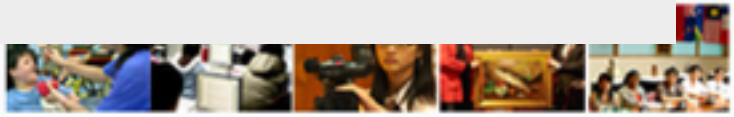


決策樹tree程式碼

```
a1 <- a
b1 <- b
library(tree)
a1$`z$註冊` <- as.factor(a1$`z$註冊`)
b1$`x$註冊` <- as.factor(b1$`x$註冊`)
wdbc.tree=tree(a1$`z$註冊`~.,data=a1)
train.pred=predict(wdbc.tree,newdata=b1, type='class')
(table.train=table(b1$`x$註冊`,train.pred))
#準確度####
tp <- length(which(all01$come=='未到'&all01$Spec.Pred=='未到'))#未＝未
tn <- length(which(all01$come=='報到'&all01$Spec.Pred=='報到'))#來＝來
fp <- length(which(all01$come=='未到'&all01$Spec.Pred=='報到'))#未＝來
fn <- length(which(all01$come=='報到'&all01$Spec.Pred=='未到'))#來＝未
accuracy <- ((tp + tn)/(tp + tn + fp + fn))*100 #準確率
recall <- (tp/(tp+fn))*100 #recall
precision <- (tn/(tn + fp))*100 #precision
f1 <- ((2*recall*precision)/(recall+precision)) ##F1-measure綜合評價指標
```

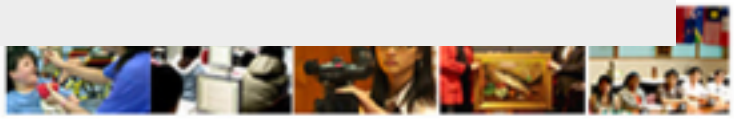
C50分類樹程式碼(1/2)

```
a1 <- a
b1 <- b
a1$`z$註冊` <- gsub('報到','Y',a1$`z$註冊`)
a1$`z$註冊` <- gsub('未到','N',a1$`z$註冊`)
b1$`x$註冊` <- gsub('報到','Y',b1$`x$註冊`)
b1$`x$註冊` <- gsub('未到','N',b1$`x$註冊`)
a1$`z$註冊` <- as.factor(a1$`z$註冊`)
b1$`x$註冊` <- as.factor(b1$`x$註冊`)
colnames(a1)= c("C","E","M","S","N","PR","come")
colnames(b1)= c("C","E","M","S","N","PR","come”)
```



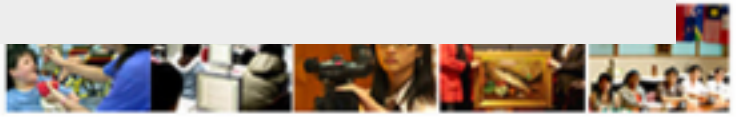
C50分類樹程式碼(2/2)

```
a.C5 <- C5.0(a1$come~ ., data = a1, rules = TRUE)
summary(a.C5)
plot(a.C5)p <- predict( a.C5, b1, type="class" )
(confus.matrix <- table(real=a1$come, predict=p))
(confus.matrix[2,2]/(confus.matrix[2,1]+confus.matrix[2,2])*100)
#準確度####
tp <- confus.matrix[2, 2]
tn <- confus.matrix[1, 1]
fp <- confus.matrix[2, 1]
fn <- confus.matrix[1, 2]
accuracy <- ((tp + tn)/(tp + tn + fp + fn))*100 #準確率
recall <- (tp/(tp+fn))*100 #recall
precision <- (tn/(tn + fp))*100 #precision
f1 <- ((2*recall*precision)/(recall+precision)) ##F1-measure綜合評價指標
```



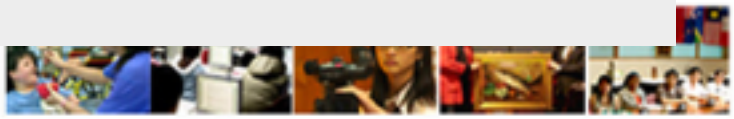
分類迴歸樹程式碼

```
library(rpart)
a1 <- a
b1 <- b
# CART的模型
Xcart.model<- rpart(a1$`z$註冊`~. , data=a1,model = F)
pred <- predict(cart.model, newdata=b1, type="class")# 用table看預測的情況
all04 =data.frame(b1,Spec.Pred=pred)#可查看結果
#準確度####
tp <- length(which(all04$x.註冊=='未到'&all04$Spec.Pred=='未到'))#未＝未
tn <- length(which(all04$x.註冊=='報到'&all04$Spec.Pred=='報到'))#來＝來
fp <- length(which(all04$x.註冊=='未到'&all04$Spec.Pred=='報到'))#未＝來
fn <- length(which(all04$x.註冊=='報到'&all04$Spec.Pred=='未到'))#來＝未
accuracy <- ((tp + tn)/(tp + tn + fp + fn))*100 #準確率
recall <- (tp/(tp+fn))*100 #recall
precision <- (tn/(tn + fp))*100 #precision
f1 <- ((2*recall*precision)/(recall+precision)) ##F1-measure綜合評價指標
```



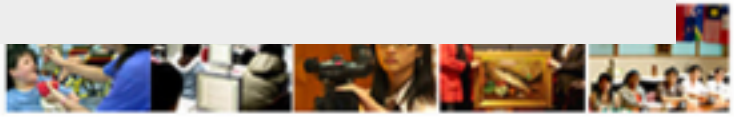
條件推論樹程式碼(1/2)

```
library(party)
a1 <- a
b1 <- b
a1$`z$註冊` <- gsub('報到','Y',a1$`z$註冊`)
a1$`z$註冊` <- gsub('未到','N',a1$`z$註冊`)
b1$`x$註冊` <- gsub('報到','Y',b1$`x$註冊`)
b1$`x$註冊` <- gsub('未到','N',b1$`x$註冊`)
a1$`z$註冊` <- as.factor(a1$`z$註冊`)
b1$`x$註冊` <- as.factor(b1$`x$註冊`)
colnames(a1)= c("C","E","M","S","N","PR","come")
colnames(b1)= c("C","E","M","S","N","PR","come")
```



條件推論樹程式碼(2/2)

```
fit <- ctree(a1$come~. , data=a1)
par(family = "STKaiti")
plot(fit, main="Conditional Inference Tree")
table(predict(fit), b1$come)
fi_t <- table(predict(fit), b1$come)
all05 = data.frame(a1,Spec.Pred=predict(fit)) #可查看結果
#準確度####
tp <- length(which(all05$come=='N'&all05$Spec.Pred=='N'))#未＝未
tn <- length(which(all05$come=='Y'&all05$Spec.Pred=='Y'))#來＝來
fp <- length(which(all05$come=='N'&all05$Spec.Pred=='Y'))#未＝來
fn <- length(which(all05$come=='Y'&all05$Spec.Pred=='N'))#來＝未
accuracy <- ((tp + tn)/(tp + tn + fp + fn))*100 #準確率
recall <- (tp/(tp+fn))*100 #recall
precision <- (tn/(tn + fp))*100 #precision
f1 <- ((2*recall*precision)/(recall+precision)) ##F1-measure綜合評價指標
```

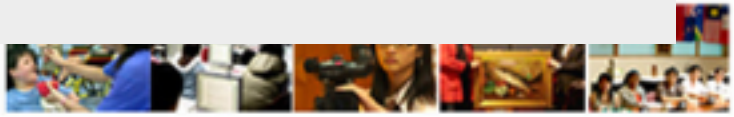


隨機森林程式碼

```
library(randomForest)
set.seed(1117)
a1 <- a
b1 <- b
randomforestM <- randomForest(factor(a1$`z$註冊`)~ ., data = a1, importane
= T, proximity = T, do.trace = 100)
pred<-predict(randomforestM,newdata=b1)
all06 = data.frame(b1,Spec.Pred=pred)#可查看結果
#準確度####
tp <- length(which(all06$x.註冊==‘未到’&all06$Spec.Pred==‘未到’))#未＝未
tn <- length(which(all06$x.註冊==‘報到’&all06$Spec.Pred==‘報到’))#來＝來
fp <- length(which(all06$x.註冊==‘未到’&all06$Spec.Pred==‘報到’))#未＝來
fn <- length(which(all06$x.註冊==‘報到’&all06$Spec.Pred==‘未到’))#來＝未
accuracy <- ((tp + tn)/(tp + tn + fp + fn))*100 #準確率
recall <- (tp/(tp+fn))*100 #recall
precision <- (tn/(tn + fp))*100 #precision
f1 <- ((2*recall*precision)/(recall+precision)) ##F1-measure綜合評價指標
```

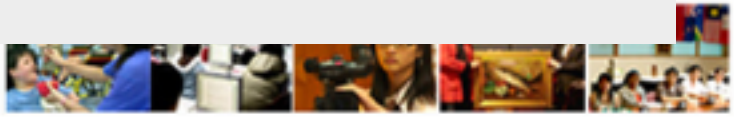
類神經網路程式碼(1/2)

```
a1 <- a
b1 <- b
library(class)
library(dplyr)
trainLabels <- b1$`x$註冊` #(參數1)準備訓練樣本組答案
#(參數2)(參數3)去除兩個樣本組答案
knnTrain <- a1[, - c(7)]
knnTest <- b1[, - c(7)]
#計算k值(幾個鄰居)通常可以用資料數的平方根
kv <- round(sqrt(10))
#(4)建立模型
prediction <- knn(train = knnTrain, test = knnTest, cl = trainLabels, k = kv)
cm <- table(x = a1$`z$註冊`, y = prediction)#, dnn = c("實際", "預測"))
all07 = data.frame(b1,Spec.Pred=prediction) #可查看結果
```



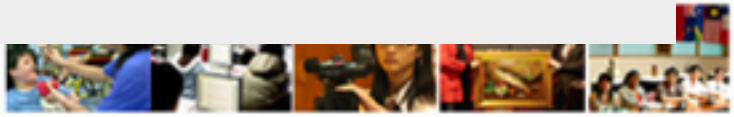
類神經網路程式碼(2/2)

```
#準確度####
tp <- length(which(all07$x.註冊=='未到'&all07$Spec.Pred=='未到'))#未＝未
tn <- length(which(all07$x.註冊=='報到'&all07$Spec.Pred=='報到'))#來＝來
fp <- length(which(all07$x.註冊=='未到'&all07$Spec.Pred=='報到'))#未＝來
fn <- length(which(all07$x.註冊=='報到'&all07$Spec.Pred=='未到'))#來＝未
accuracy <- ((tp + tn)/(tp + tn + fp + fn))*100 #準確率
recall <- (tp/(tp+fn))*100 #recall
precision <- (tn/(tn + fp))*100 #precision
f1 <- ((2*recall*precision)/(recall+precision)) ##F1-measure綜合評價指標
```



支持向量機程式碼

```
library(e1071)
a1 <- a
b1 <- b
svmM <- svm(a1$`z$註冊` ~ ., data = a1, probability = TRUE)
results <- predict(svmM, b1, probability = TRUE)
cm <- table(x = b1$`x$註冊`, y = results)
#準確度####
tp <- length(which(all08$x.註冊=='未到'&all08$Spec.Pred=='未到'))#未＝未
tn <- length(which(all08$x.註冊=='報到'&all08$Spec.Pred=='報到'))#來＝來
fp <- length(which(all08$x.註冊=='未到'&all08$Spec.Pred=='報到'))#未＝來
fn <- length(which(all08$x.註冊=='報到'&all08$Spec.Pred=='未到'))#來＝未
accuracy <- ((tp + tn)/(tp + tn + fp + fn))*100 #準確率
recall <- (tp/(tp+fn))*100 #recall
precision <- (tn/(tn + fp))*100 #precision
f1 <- ((2*recall*precision)/(recall+precision)) ##F1-measure綜合評價指標
```



貝氏分類程式碼

```
library(e1071)
a1 <- a
b1 <- b
nbcm <- naiveBayes(a1$`z$註冊` ~ ., data = a1)
results <- predict(nbcm, b1)
all09 = data.frame(b1,Spec.Pred=results)#可查看結果
#準確度####
tp <- length(which(all09$x.註冊=='未到'&all09$Spec.Pred=='未到'))#未＝未
tn <- length(which(all09$x.註冊=='報到'&all09$Spec.Pred=='報到'))#來＝來
fp <- length(which(all09$x.註冊=='未到'&all09$Spec.Pred=='報到'))#未＝來
fn <- length(which(all09$x.註冊=='報到'&all09$Spec.Pred=='未到'))#來＝未
accuracy <- ((tp + tn)/(tp + tn + fp + fn))*100 #準確率
recall <- (tp/(tp+fn))*100 #recall
precision <- (tn/(tn + fp))*100 #precision
f1 <- ((2*recall*precision)/(recall+precision)) ##F1-measure綜合評價指標
```