商業分析 HW3
105305072 企管四 許惠甄

1. 辨認忠誠與不忠誠客戶

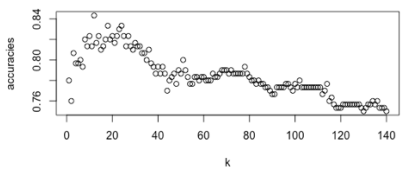
   a. 利用 GLM 配適模型的預測正確率：0.8933333

```
Call:  glm(formula = is_loyal ~ depart_on_time + register_method + seat_rate +
    meal_rate + flight_rate + tv_ad + dm_message + credit_card_vendor +
    credit_card_bonus + coupon, family = binomial(link = "logit"),
    data = train)

Coefficients:
               (Intercept)              depart_on_time       register_methodothers
                 -16.38488                     4.55456                    -4.90690
     register_methodphone      register_methodwebsite                   seat_rate
                  -0.27805                     2.83622                     2.17601
                meal_rate                  flight_rate                       tv_ad
                   0.79792                     0.79501                     5.99622
               dm_message  credit_card_vendorVendor B  credit_card_vendorVendor C
                  -9.39785                     1.19880                     1.44958
        credit_card_bonus                      coupon
                   2.21745                     0.02927

Degrees of Freedom: 699 Total (i.e. Null);  686 Residual
Null Deviance:      921.5
Residual Deviance: 267.5        AIC: 295.5
```

```
     0   1
0 96  17
1 15 172
```

   b. 監督式配適模型：

      i. KNN(k=12)模型下預測正確率：0.8266667




```
     predicted_knn
real   0   1
   0  72  39
   1  13 176
```

      ii. 決策樹模型下預測正確率：0.6833333

```
             Predict
Real          Satisfied Unsatisfied
  Satisfied         181           8
  Unsatisfied        87          24
```

      iii. 隨機森林模型下預測正確率：0.8866667

```
                Predict
Real             Satisfied Unsatisfied
  Satisfied            178           11
  Unsatisfied           23           88
```

      iv. SVM 模型下預測正確率：0.86

	Predict	
Real	Satisfied	Unsatisfied
Satisfied	166	23
Unsatisfied	19	92

v. 參數調整後的 SVM 模型預測準確率：100%

	Predict	
Real	Satisfied	Unsatisfied
Satisfied	189	0
Unsatisfied	0	111

➢ 結論：雖然調整後 GLM 的預測準確率高達百分之百，但過於複雜的模型容易產生過度配適問題，因此未來可以使用準確率近 90%的 GLM 模型進行預測。

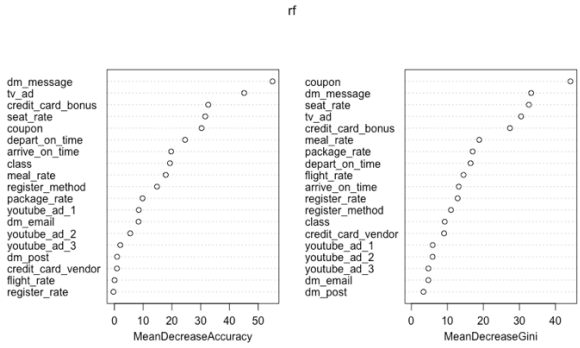
2. 找出重要變數

a. 利用 GLM 配適模型的 AIC 找：

	Df	Deviance	AIC
<none>		267.53	295.53
- credit_card_vendor	2	278.20	302.20
- meal_rate	1	277.47	303.47
- flight_rate	1	294.57	320.57
- coupon	1	328.95	354.95
- seat_rate	1	349.65	375.65
- register_method	3	356.79	378.79
- credit_card_bonus	1	376.47	402.47
- depart_on_time	1	400.10	426.10
- tv_ad	1	474.34	500.34
- dm_message	1	539.69	565.69

b. 利用 Random Forest 模型的 Mean Decrease Gini：

	Satisfied	Unsatisfied	MeanDecreaseAccuracy	MeanDecreaseGini
depart_on_time	17.0443543	20.9003546	24.56758588	16.434826
arrive_on_time	13.7948206	15.2435827	19.72401393	13.144120
register_method	11.4729539	8.4963778	14.79402296	10.996387
register_rate	0.9651832	-1.5132337	-0.34705709	12.844799
class	15.6864810	12.5724014	19.31621954	9.263218
seat_rate	15.2604780	28.5327283	31.58220692	32.619815
meal_rate	16.6856361	8.1225303	17.85489954	18.833297
flight_rate	0.8742011	-1.0578274	0.02887707	14.477509
package_rate	2.3995891	12.4654221	9.77746206	17.012378
tv_ad	32.7190961	40.3003400	45.12300134	30.455362
youtube_ad_1	6.4500005	5.1124869	8.45475789	5.928575
youtube_ad_2	4.0375172	3.2469018	5.48910869	5.898858
youtube_ad_3	2.1008378	0.5102290	2.02854205	4.725285
dm_message	41.9312441	50.2472297	54.99439043	33.266187
dm_post	1.6771665	-0.5262497	0.91690328	3.358847
dm_email	5.4975449	4.4103839	8.36276364	4.710656
credit_card_vendor	1.8640823	-0.8508070	0.87762514	9.083469
credit_card_bonus	23.3778812	25.4410150	32.63527627	27.362085
coupon	29.3113452	8.4729395	30.32116437	44.162199



- 結論：兩種評估方式交叉參照比對出以下五個重要參數

  dm_message、tv_ad、seat_rate、coupon、credit_card_bonus

3. 提出建議：

公司可以把行銷廣告集中投放在簡訊及電視渠道，並藉由提供折扣給信用卡不同紅利等級的客戶增加來客量，在實際服務上則是可以優化座位的舒適程度，藉此提高客戶的忠誠度．

```
airline <- read.csv("airline.csv")

#1.
#a.
airline$is_loyal <- ifelse(airline$is_loyal == "Satisfied",1,0) #把忠誠度轉換
成binary

library(tidyverse)
airline$index =c(1:nrow(airline))
set.seed(200)
train <- airline %>% group_by(is_loyal) %>% sample_frac(0.7)
test <- anti_join(airline, train, by ='index')

fit_model <- glm(is_loyal ~ depart_on_time + register_method + seat_rate +
meal_rate + flight_rate + tv_ad + dm_message + credit_card_vendor +
credit_card_bonus + coupon, data = train, family = binomial(link =
"logit"))
step(fit_model) #用stepwise盡量減少變數：但仍剩下10個變數
summary(fit_model)

predicted <- predict(fit_model, test, type="response") #用模型預測測試集


library(InformationValue)
thres = optimalCutoff(test$is_loyal, predicted) #找到最適cut point

CFMatrix = confusionMatrix(test$is_loyal, prediCFcted, threshold = thres) #
做出混淆矩陣
GLM_matrix <- as.matrix(CFMatrix)

sum(diag(GLM_matrix))/sum(GLM_matrix) #預測正確率=0.8933333
misClassError(test$is_loyal, predicted, threshold = thres) #預測錯誤率
=0.1067

plotROC(test$is_loyal, predicted)

sensitivity(test$is_loyal, predicted, threshold = thres)
specificity(test$is_loyal, predicted, threshold = thres)


##GLM預測正確率：0.8933333##


#1.
#b.

#b-1.KNN
airline$register_method <- as.numeric(airline$register_method)
airline$credit_card_vendor <- as.numeric(airline$credit_card_vendor)

#標準化參數
stand.features <- scale(airline[3:21])
```

```
var(stand.features[,])
KNN_data <- cbind(airline[2], stand.features)

#分訓練組及測試組
library(class)
KNN_data$index =c(1:nrow(airline))
set.seed(200)
train_KNN <- KNN_data %>% group_by(is_loyal) %>% sample_frac(0.7)
test_KNN <- anti_join(KNN_data, train, by ='index')

#選k值
range <- 1:round(0.2 * nrow(train_KNN)) #k 上限為訓練樣本數的 20%(140)
accuracies <- rep(NA, length(range))

for (i in range) {
  test_predicted <- knn(train_KNN[,2:20], test_KNN[,2:20], cl =
train$is_loyal, k = i)
  conf_mat <- table(test_KNN$is_loyal, test_predicted)
  accuracies[i] <- sum(diag(conf_mat))/sum(conf_mat)
}

##視覺化選K的結果
plot(range, accuracies, xlab = "k")
which.max(accuracies) #K值=12

#建立KNN模型
library(class)
predicted_knn <- knn(train_KNN[,2:20], test_KNN[,2:20], cl =
train$is_loyal, k=12)
KNN_matrix <- table(real=test_KNN[,1], predicted_knn) #confusion table
sum(diag(KNN_matrix))/sum(KNN_matrix) #預測正確率=0.8266667


##KNN預測正確率：0.8266667##


#b-2.Decision Tree
airline_DT <- read.csv("airline.csv") #重新讀原始資料

#拆測試組及訓練組
library(tidyverse)
airline_DT$index =c(1:nrow(airline_DT))
set.seed(200)
train_DT <- airline_DT %>% group_by(is_loyal) %>% sample_frac(0.7)
test_DT <- anti_join(airline_DT, train_DT, by ='index')

#決策樹模型
library(rpart)
tree <- rpart(is_loyal ~. ,data=train_DT, method="class")
predicted_DT <- predict(tree, newdata=test_DT, type="class")
DT_matrix <- table(Real = test_DT$is_loyal, Predict = predicted_DT)
#confusion table
```

```
sum(diag(DT_matrix))/sum(DT_matrix) #預測正確率=0.6833333


##DT預測正確率：0.6833333##


#b-3.Random Forests

library(randomForest)
train_RF <- train_DT[2:21]
test_RF <- test_DT[2:21]

rf <- randomForest(is_loyal ~ ., data = train_RF, importance=TRUE)
rf

#看需要幾棵樹：約接近100棵即可
plot(rf)
legend("topright", colnames(rf$err.rate),col=1:4,cex=0.8,fill=1:4)

#看變數重要性
importance(rf)
varImpPlot(rf)

#預測
predicted_RF=predict(rf, newdata = test_RF)
RF_matrix <- table(Real = test_RF$is_loyal, Predict = predicted_RF)
sum(diag(RF_matrix))/sum(RF_matrix) #預測正確率=0.8866667

##RF預測正確率：0.8866667##


#b-4.SVM
airline <- read.csv("airline.csv")

library(tidyverse)
airline$index =c(1:nrow(airline))
set.seed(200)
train <- airline %>% group_by(is_loyal) %>% sample_frac(0.7)
test <- anti_join(airline, train, by ='index')


library(e1071)
s <- svm(is_loyal ~ ., data = train, probability = TRUE)
predicted_SVM <- predict(s, test, probability = TRUE)
SVM_matrix <- table(Real = test$is_loyal, Predict = predicted_SVM)
sum(diag(SVM_matrix))/sum(SVM_matrix) #預測正確率=0.86

##SVM預測正確率：0.86##


svm_tune <- tune(svm, is_loyal ~ .,data=train,
                 kernel="radial", ranges=list(cost=10^(-1:2), gamma=c(.
5,1,2)))
```

```
svm_tune$best.model
plot(svm_tune)

after_tune <- svm(is_loyal ~ ., data=airline, kernel="radial", cost=1,
gamma=0.5)
summary(after_tune)
pred <- predict(after_tune,test)
table(Real = test$is_loyal, Predict = pred)

#Tuned SVM 預測準確率達100%
```