```
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%
```