三軍總醫院北投分院統計及實驗設計課程之七

2021/7/9

ytai1123@gmail.com

使用方法:

1. 使用gmail帳號登入
2. 按"執行階段" -->"全部執行" 以執行全部內容, 若要個別執行可點選每格程式左方箭頭或按 Control + Enter 鍵執行。

```
##0-1
!git  clone  https://github.com/YuehMintTai/RPython.git
```

```
Cloning into 'RPython'...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (93/93), done.
remote: Total 95 (delta 49), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (95/95), done.
```

```
##0-2
!pip  install  rpy2
```

```
Requirement already satisfied: rpy2 in /usr/local/lib/python3.7/dist-packages (3.4.5)
Requirement already satisfied: tzlocal in /usr/local/lib/python3.7/dist-packages (from rpy2)
Requirement already satisfied: cffi>=1.10.0 in /usr/local/lib/python3.7/dist-packages (from r
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from rpy2) (20
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from rpy2) (
Requirement already satisfied: pycparser in /usr/local/lib/python3.7/dist-packages (from cffi
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (fr
```

```
##0-3
%load_ext  rpy2.ipython
```

```
##6-1
%%R
myData<-read.csv('RPython/samples.csv')
tail(myData,1)
```

```
    SID 性別 年齡 入伍前職業 教育程度 婚姻狀況 皆無過去病史01 早產兒01
188   4    1   25         商        4        1              1        0
    頭部曾受傷01 發展遲緩01 注意力不足過動症01 癲癇01 癲癇服藥治療 癲癇服藥期間
```

```
188            0          0                  0      0            0            0
    軍種 軍階 役別 入伍至今_年 聽過自殺課程_次 求助心輔_次 求助精神科_次
188    1    1    2         0.5               1           0             2
    使用1995_次 使用24h專線_次 特殊狀況 父母婚姻狀態 自殺意念_bsrs6 B型肝炎01
188           0              0        4            4              4         0
    C型肝炎01 氣喘史01 過敏史01 心臟病史01 高血壓01 醣尿病01 甲狀腺01 類風濕01
188         0        1        1          0        0        0        1        0
    重大意外01 自殺意念01 透露父母 透露手足 透露好友 透露同儕 透露長官 透露心輔
188          1          1        0        0        0        0        0        0
    透露醫師 拒告父母 拒告手足 拒告好友 拒告同儕 拒告長官 拒告心輔 拒告醫師
188        0        1        1        1        1        1        1        1
    BSRS總分 BSRSR總分 過動症總分 Inattention Impulsivity opposition depression
188       20         5         18           9           9          8         57
    anxiety burdensome belonging 家庭滿意度apgar 網路成癮症01 網路成癮分數YDQ
188 29.0294         42        12               0            0               0
    existeness meaning control seeking death suicidea 睡眠困擾_bsrs1
188         28      10      22      16    15        7              4
    睡眠困擾_bsrsr1 睡眠困擾_bdi16 易怒_bsrs3 易怒_bsrsr3 depress impuls
188               1              3          4           1      57      9
    Internet ADHD
188        0   18
```

```
##7-1-1  繪出預測值(predicted_value)和實際值的關係圖
%%R
formula1<-'網路成癮分數YDQ~家庭滿意度apgar'
model1<-glm(formula1,myData,family='gaussian')
predicted_value1<-predict(model1,myData)
predicted_value1
plot(myData$家庭滿意度apgar,predicted_value1,col='red',
          xlab='APGAR',ylab='YDQ',
          xlim=range(c(0,11)),ylim=range(c(0,10)))
points(myData$家庭滿意度apgar,  myData$網路成癮分數YDQ,col='blue')
```

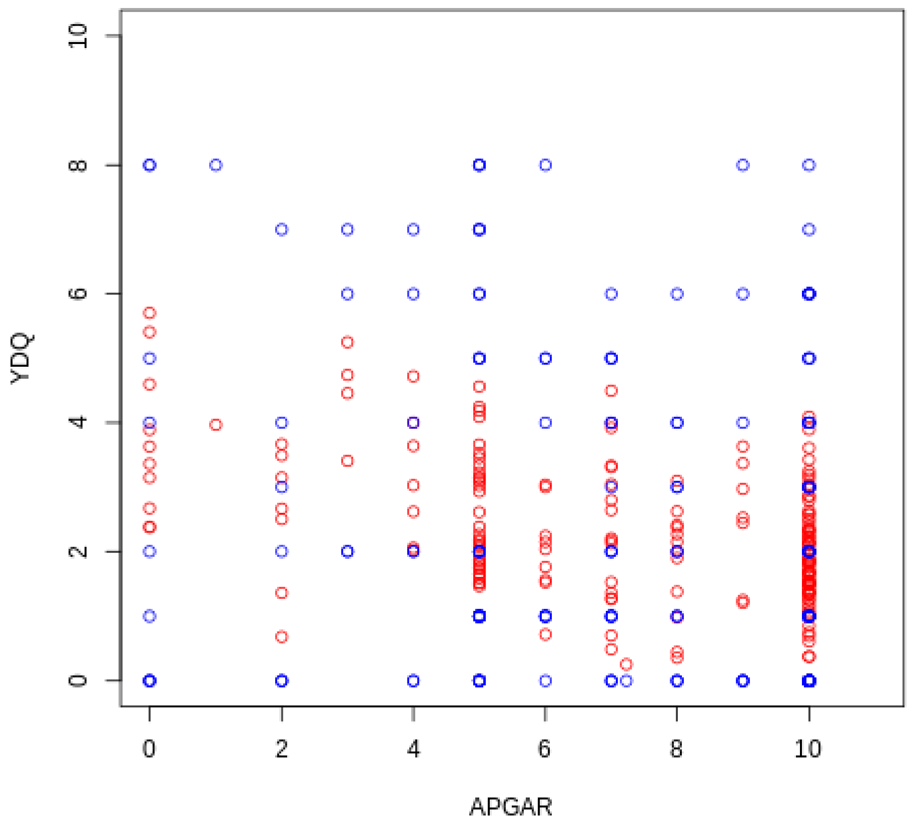
```
%%R  ##下載  rsq  package
rm(list  =  ls())
install.packages('rsq')
```

```
R[write to console]: Installing package into  ‘/usr/local/lib/R/site-library’
(as  ‘lib’  is unspecified)

R[write to console]: also installing the dependencies  ‘minqa’ ,  ‘nloptr’ ,  ‘RcppEigen


R[write to console]: trying URL 'https://cran.rstudio.com/src/contrib/minqa_1.2.4.tar.gz

R[write to console]: Content type 'application/x-gzip'
R[write to console]:  length 53548 bytes (52 KB)

R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
```

```
##7-1-2  ##計算R-square
%%R
library(rsq)
print(rsq(model1))
print(rsq(model1,adj=TRUE))
with(summary(model1),1-deviance/null.deviance)
```

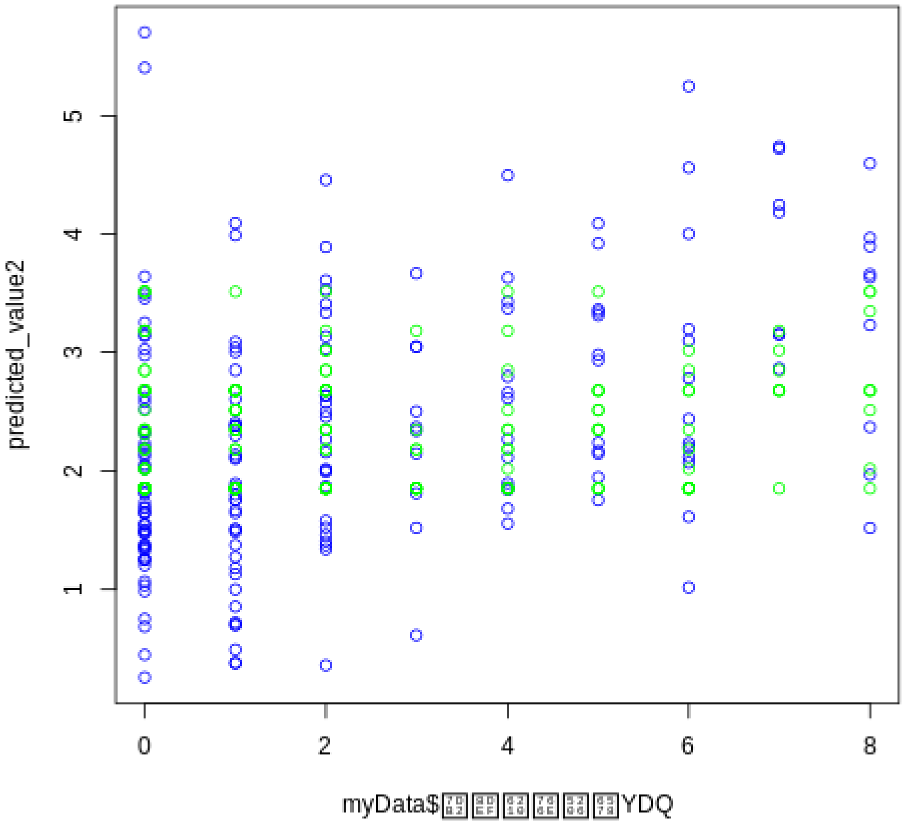
```
[1] 0.04158183
[1] 0.03642905
[1] 0.04158183
```

```
##7-2-2  ##使用較多X的model...
%%R
formula2<-'網路成癮分數YDQ~as.factor(性別)+家庭滿意度apgar+年齡+BSRS總分+anxiety+depression+burdens
model2<-glm(formula2,  myData,  family='gaussian')
predicted_value2<-predict(model2,myData)
plot(myData$家庭滿意度apgar,predicted_value2,col='red',
          xlab='APGAR',ylab='YDQ',
          xlim=range(c(0,11)),ylim=range(c(0,10)))
points(myData$家庭滿意度apgar,  myData$網路成癮分數YDQ,col='blue')
```



```
%%R
```

```
%%R
plot(myData$網路成癮分數YDQ,predicted_value2,col='blue')
points(myData$網路成癮分數YDQ,predicted_value1,col='green')
```



```
##7-2-2  ##計算R-square
%%R
library(rsq)
print(rsq(model2))
print(rsq(model2,adj=TRUE))
with(summary(model2),1-deviance/null.deviance)
```

```
    [1] 0.1879277
    [1] 0.1516339
    [1] 0.1879277
```

```
##7-3 Python statsmodels predicting & R^2
import pandas as pd
import statsmodels.formula.api as smf
formula='網路成癮分數YDQ~家庭滿意度apgar'
df=pd.read_csv('RPython/samples.csv')
model3=smf.ols(formula,df).fit()
#model3=smf.glm(formula,df).fit()
#predicted_value=model3.predict(df.家庭滿意度apgar)
#predicted_value
model3.summary()
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarnir
  import pandas.util.testing as tm
                        OLS Regression Results
     Dep. Variable:    網路成癮分數YDQ      R-squared:         0.042
           Model:      OLS              Adj. R-squared:    0.036
          Method:      Least Squares      F-statistic:     8.070
            Date:      Mon, 12 Jul 2021 Prob (F-statistic): 0.00500
            Time:      07:44:56          Log-Likelihood:   -431.90
  No. Observations:    188                     AIC:        867.8
     Df Residuals:     186                     BIC:        874.3
        Df Model:      1
  Covariance Type:     nonrobust
                         coef   std err    t    P>|t|  [0.025 0.975]
      Intercept        3.5125  0.457   7.679  0.000  2.610  4.415
  家庭滿意度apgar     -0.1658  0.058  -2.841  0.005 -0.281 -0.051
      Omnibus:        19.880  Durbin-Watson:   1.968
  Prob(Omnibus):      0.000  Jarque-Bera (JB): 21.495
        Skew:         0.786      Prob(JB):      2.15e-05
      Kurtosis:       2.480      Cond. No.      20.6


  Warnings:
  [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

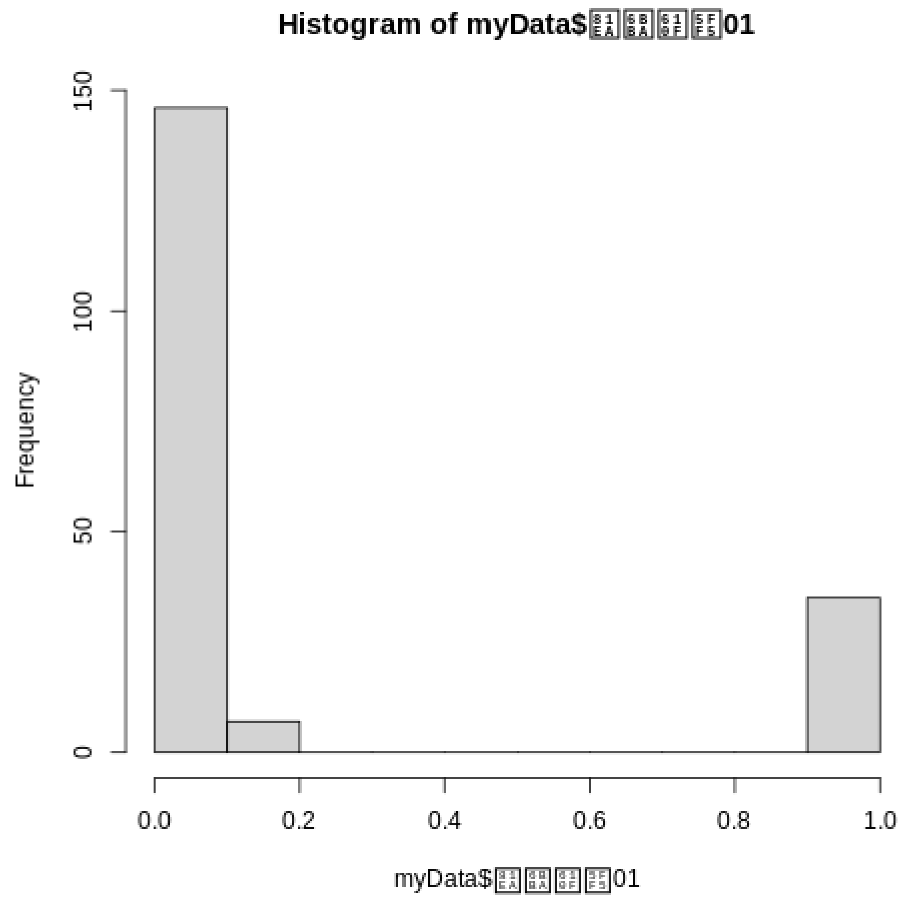
```
##7-4-1 Python sklearn predicting and R^2

from sklearn.linear_model import LinearRegression
model4=LinearRegression()
x=df['家庭滿意度apgar']
model4.fit(x.values.reshape(-1,1),df['網路成癮分數YDQ'].values.tolist())  ##fit(x,y)
predicted_value=model4.predict(df['家庭滿意度apgar'].values.reshape(-1,1))
R2=model4.score(df['家庭滿意度apgar'].values.reshape(-1,1),df['網路成癮分數YDQ'].values.tolist())
N_y=len(df['網路成癮分數YDQ'])
AdjR2=1-(1-R2)*(N_y-1)/(N_y-x.values.reshape(-1,1).shape[1]-1)
print(R2)
AdjR2
```

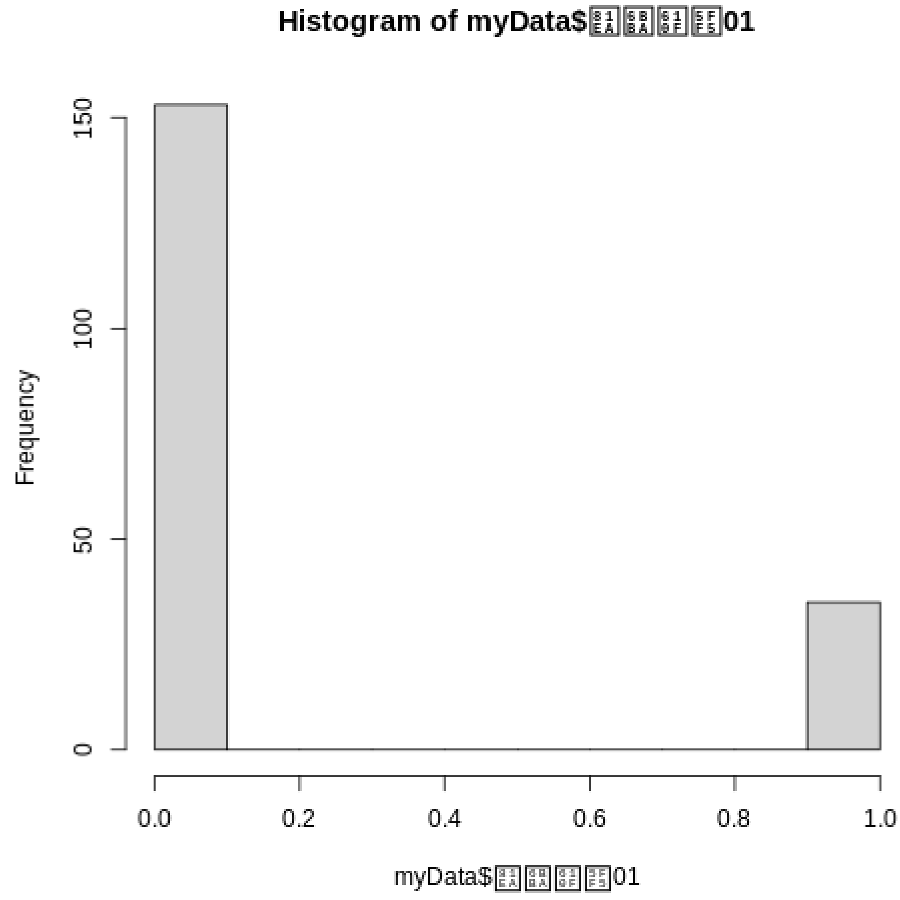
```
0.04158183160640083
0.03642904575482231
```

```
##7-4-2 Calculate adjusted R^2 in sklearn...
from sklearn.metrics import r2_score
print(r2_score(df['網路成癮分數YDQ'], predicted_value))
r2_score(df['網路成癮分數YDQ'],predicted_value, multioutput='raw_values' )
```

```
0.04158183160640083
array([0.04158183])
```

```
%%R
hist(myData$自殺意念01)
myData$自殺意念01<-as.integer(myData$自殺意念01)
hist(myData$自殺意念01)
```





```
##7-5-1  Predicting  probability  from  logistic  regression  model
%%R
formula<-'自殺意念01~as.factor(性別)+網路成癮分數YDQ+家庭滿意度apgar'
```

```
model5<-glm(formula,myData,  family='binomial')
summary(model5)
預測機率1<-predict(model5,type="response")
預測機率1[1:20]
```

```
          1          2          3          4          5          6          7 
0.05973068 0.06517669 0.08090854 0.05973068 0.05531099 0.05531099 0.23784212 
          8          9         10         11         12         13         14 
0.05120053 0.05120053 0.05120053 0.05120053 0.12122299 0.05176214 0.16949776 
         15         16         17         18         19         20 
0.05120053 0.22191810 0.25294191 0.07012439 0.12837035 0.05120053 
```

```
%%R
install.packages('pROC')
```

```
##7-5-2  Another  way  to  predicting  probability
%%R
library(pROC)
pROC_obj<-roc(myData$自殺意念01,預測機率1,  smoothed=TRUE,  print.auc=TRUE,  ci=TRUE,ci.alpha=0.9,p
myROC.ci<-ci.se(pROC_obj)
plot(myROC.ci,  type='shape',col='lightblue')
```

```
     R[write to console]: Type 'citation("pROC")' for a citation.

     R[write to console]: 
     Attaching package: ‘pROC’


     R[write to console]: The following objects are masked from ‘package:stats’:

         cov, smooth, var
```

```
##7-5-3  Comparing  two  roc  curves
%%R
formula<-'自殺意念01~as.factor(性別)+網路成癮分數YDQ+家庭滿意度apgar+depression+anxiety+belonging+b
model6<-glm(formula,myData,  family='binomial')
summary(model6)
預測機率2<-predict(model6,type="response")
預測機率2[1:20]
roc1<-roc(myData$自殺意念01,預測機率1)
roc2<-roc(myData$自殺意念01,預測機率2)
roc.test(roc1,roc2)
```

```
     R[write to console]: Setting levels: control = 0, case = 1

     R[write to console]: Setting direction: controls < cases

     R[write to console]: Setting levels: control = 0, case = 1

     R[write to console]: Setting direction: controls < cases


             DeLong's test for two correlated ROC curves

     data:  roc1 and roc2
     Z = -3.558, p-value = 0.0003737
     alternative hypothesis: true difference in AUC is not equal to 0
     sample estimates:
     AUC of roc1 AUC of roc2 
       0.7872082   0.8976657 
```

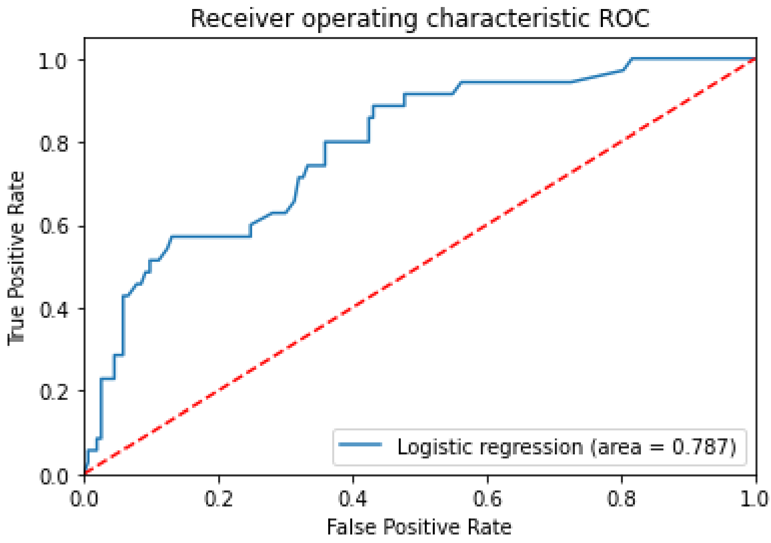
```
%%R
install.packages('InformationValue')
```

```
##7-6-1  Confusion  Table,  sensitivity  and  specificity
%%R
library(InformationValue)
預測機率1<-predict(model5,myData,type='response')
optimal<-optimalCutoff(myData$自殺意念01,預測機率1)[1]
confusionMatrix(myData$自殺意念01,預測機率1)                   ##       0    1
                                                               ##   0  147   27
                                                               ##   1    6    8
#confusionMatrix(myData$自殺意念01,預測機率2,threshold=optimal)##       0    1
                                                               ##   0    140   9
                                                               ##   1    13   26
#sensitivity(myData$自殺意念01,預測機率2,threshold=optimal)      ##0.7428571
#specificity(myData$自殺意念01,預測機率2,threshold=optimal)      ##0.9150327
#specificity(myData$自殺意念01,預測機率2,threshold=0.5)        ##0.9411765
```

```
#specificity(myData$自殺意念01,預測機率2)                          ##0.9411765
#optimal                                                          ##0.366444
```

```
     0   1
0  147  27
1    6   8
```

```
##7-7-1 Statsmodels with ROC and AUC
from matplotlib import pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
df['sex']='男'
df.loc[df['性別']==2,'sex']='女'
x=df[['sex','網路成癮分數YDQ','家庭滿意度apgar']]
x=pd.get_dummies(data=x,drop_first=True)
y=df['自殺意念01'].astype(int)
model6=LogisticRegression()
result=model6.fit(x,y)
##ROC曲線
預測機率4=result.predict_proba(x)
AUC面積=roc_auc_score(y,預測機率4[:,1])
fpr, tpr, thresholds = roc_curve(y,預測機率4[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic regression (area = %0.3f)' % AUC面積)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic ROC')
plt.legend(loc="lower right")
plt.show()
```



```
##7-7-2 accuracy and other rates..
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

```
##列聯表
預測類別4=result.predict(x)
confusion_matrix(y,預測類別4)       ##array([[148,      5],
                                  ##       [ 27,      8]])
tn, fp, fn, tp = confusion_matrix(y,預測類別4).ravel()  ##tn=true negative, fp=false positi
accuracy_score(y,預測類別4)          ##準確率=0.8297872340425532  預設是以0.5為threshold
sensitivity=tp/(tp+fn)            ##sensitivity=0.22857142857142856  預設是以0.5為threshold
specificity=tn/(tn+fp)            ##specificity=0.9673202614379085  預設是以0.5為threshold
accuracy=(tp+tn)/(tp+tn+fp+fn)    ##accuracy=0.8297872340425532  預設是以0.5為threshold


#7-8-1Training set and Testing set by Sklearn
##Accuracy without any validation 不使用任何validation方式...
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
import numpy as np
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=100)
model9=LogisticRegression()
model9_trained=model9.fit(x_train,y_train)
預測類別9=model9_trained.predict(x_test)
預測機率9=model9_trained.predict_proba(x_test)
fpr,tpr,thresholds=roc_curve(y_test,預測機率9[:,1])
optimal_index=np.argmax(tpr-fpr)
optimal=thresholds[optimal_index]                                    ##計算最佳切分點...
accuracy_score(y_test,預測類別9)                                      ##0.8157894736842105
#accuracy_score(y_test,[m>optimal for m in 預測機率9[:,1]])           ##0.8421052631578947
roc_auc_score(y_test,預測機率9[:,1])                                  ##0.780357142857143
```

```
0.780357142857143
```

```
##7-8-2 Accuracy with K-fold Cross-Validation,使用K-fold validation
from sklearn.model_selection import KFold,cross_val_predict
from sklearn.metrics import accuracy_score
import numpy as np
kfold=KFold(n_splits=3, random_state=100)
model5=LogisticRegression()
model5.fit(x_train,y_train)
預測機率5=cross_val_predict(model5,x_test,y_test,cv=kfold, method='predict_proba')
##計算最佳切分點 optimal
fpr,tpr,thresholds=roc_curve(y_test,預測機率5[:,1])
optimal_index=np.argmax(tpr-fpr)
optimal=thresholds[optimal_index]
預測類別5=cross_val_predict(model5,x_test,y_test,cv=kfold,method='predict')
accuracy_score(y_test,預測類別5)                                ###0.8421052631578947
accuracy_score(y_test,[m>optimal for m in 預測機率5[:,1]])      ##0.8157894736842105
roc_auc_score(y_test,預測機率5[:,1])                            ###0.7214285714285714
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_split.py:296: FutureWarning:
  FutureWarning
0.7214285714285714
```

```
##7-8-3 Accuracy with StraitfiedK-fold Cross-Validation 使用stratified K-fold
from sklearn.model_selection import StratifiedKFold, cross_val_score,cross_validate,cross_val_p
from sklearn.metrics import accuracy_score
kfold=StratifiedKFold(n_splits=3, random_state=100)
model5=LogisticRegression()
model5.fit(x_train,y_train)
預測機率5=cross_val_predict(model5,x_test,y_test,cv=kfold, method='predict_proba')
###計算最佳切分點..
fpr, tpr, thresholds=roc_curve(y_test,預測機率5[:,1])
opitmal=thresholds[np.argmax(tpr-fpr)]

預測類別5=cross_val_predict(model5,x_test,y_test,cv=kfold,method='predict')
accuracy_score(y_test,預測類別5)                          ###0.8421052631578947
accuracy_score(y_test,[m>optimal for m in 預測機率5[:,1]])  ##0.8421052631578947
#roc_auc_score(y_test,預測機率5[:,1])                       ###0.7464285714285714
```

```
    /usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_split.py:296: FutureWarning:
      FutureWarning
    0.8421052631578947
```

```
##7-8-4 Accuracy with Leave-One-Out cross validation (LOOCV)
from sklearn.model_selection import LeaveOneOut
from sklearn.metrics import accuracy_score
kfold=LeaveOneOut()
model5=LogisticRegression()
model5.fit(x_train,y_train)
預測機率5=cross_val_predict(model5,x_test,y_test,cv=kfold, method='predict_proba')
###計算最佳切分點=optimal..
fpr, tpr, thresholds=roc_curve(y_test,預測機率5[:,1])
opitmal=thresholds[np.argmax(tpr-fpr)]
預測類別5=cross_val_predict(model5,x_test,y_test,cv=kfold,method='predict')
accuracy_score(y_test,預測類別5)                          ###0.8421052631578947
accuracy_score(y_test,[m>optimal for m in 預測機率5[:,1]])  ##0.8421052631578947
roc_auc_score(y_test,預測機率5[:,1])                        ###0.7035714285714285
```

```
    0.7035714285714285
```

以下的R程式碼有問題,請自動忽略....

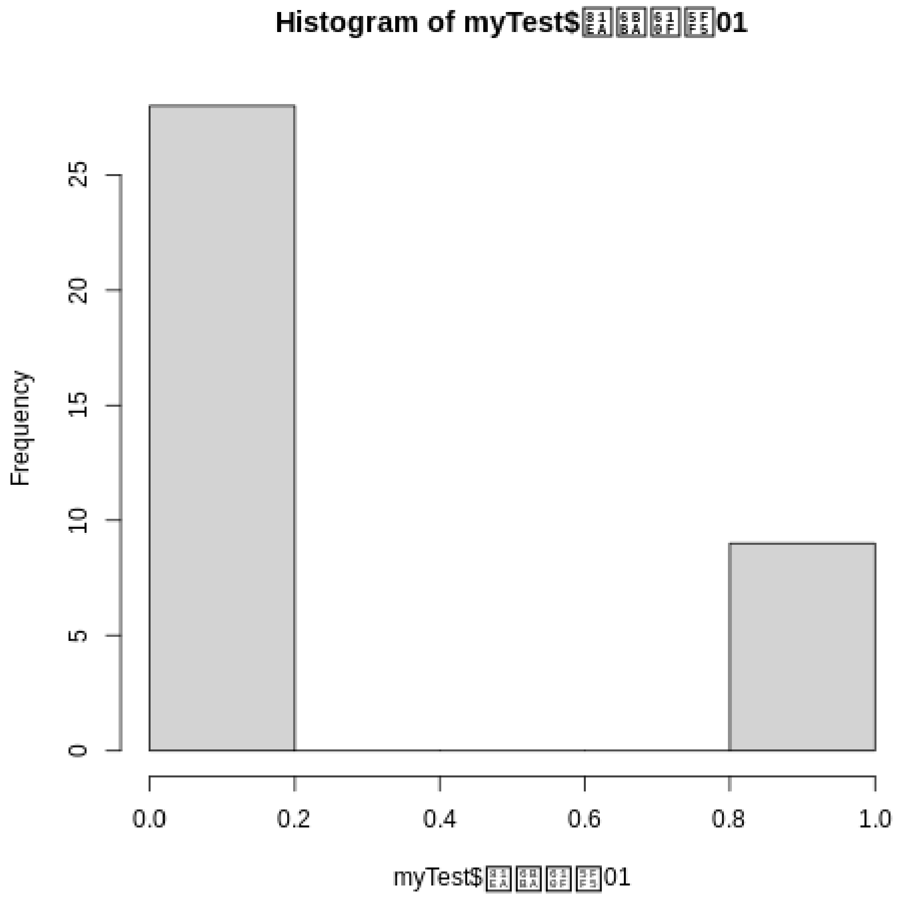
```
%%R
install.packages('caret')
```

```
%%R
library(caret)
```

```
    R[write to console]: Loading required package: lattice

    R[write to console]: Loading required package: ggplot2

    R[write to console]:
    Attaching package: ‘caret’
    R[write to console]: The following objects are masked from ‘package:InformationValue’:

        confusionMatrix, precision, sensitivity, specificity
```

```
##7-8 cross validation in R
%%R
split<-0.80
trainIndex<-createDataPartition(myData$自殺意念01,p=split,list=FALSE)
myTrain<<-myData[trainIndex,]
myTest<-myData[-trainIndex,]
```

```
%%R
hist(myTest$自殺意念01)
```



```
%%R
library(caret)
formula<-’自殺意念01~as.factor(性別)+網路成癮分數YDQ+家庭滿意度apgar’
model8<-glm(formula,myTrain, family=’binomial’)
train_contro<-trainControl(method=’boot’,number=100)
預測機率8<-predict(model8,myTest[,c(’性別’,’網路成癮分數YDQ’,’家庭滿意度apgar’)],type=’response’)
optimal<-optimalCutoff(myTest$自殺意念01,預測機率8)[1]
confusionMatrix(myTest$自殺意念01,預測機率8)
length(myTest$自殺意念01)
length(預測機率8)
```

● ×