Psychoinformatics - Week 9 (Exercises)

by 徐舒庭 (b11705018@ntu.edu.tw)

In [ ]:
```
import numpy as np
from sklearn import *
from sklearn import model_selection
from matplotlib.pyplot import *
%matplotlib inline
from sklearn import datasets
from sklearn import neighbors
```

1 檢查 machine learning pipeline (8 points)

1.1 請打亂原本的Y觀察正確率是否和chance level (0.33)有差異? 若有, why? (4 points)

In [ ]:
```
# 本題在研究打亂X和打亂Y有差別嗎?
iris = datasets.load_iris()
X=iris.data
Y=iris.target
Y2=np.random.permutation(Y)
print("Y=", Y)
print("Y2=", Y2)
clf=neighbors.KNeighborsClassifier(1)
clf.fit(X,Y2)
orig_accuracy=np.mean(clf.predict(X)==Y)
new_accuracy=np.mean(clf.predict(X)==Y2)
ac=np.mean(clf.predict(X)==Y)
print("原正確率:", orig_accuracy)
print("新正確率:", new_accuracy)
```

```
Y= [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
Y2= [1 1 1 2 2 1 0 1 2 0 2 0 2 0 1 0 1 0 2 0 0 2 2 1 1 1 2 1 2 0 0 2 1 0 0 2 1
 1 2 0 2 1 1 1 1 1 0 2 2 0 2 1 2 2 1 1 1 0 0 0 1 1 2 1 2 0 1 0 0 0 0 0 1 2
 2 0 0 0 1 2 1 1 2 0 0 2 0 2 0 2 0 0 1 2 2 2 2 1 2 2 1 1 0 0 1 0 1 1 0 2 0
 0 2 0 1 2 1 2 2 2 0 0 1 0 2 2 2 2 0 1 1 1 0 0 0 2 0 1 1 2 1 2 1 2 1 0 1 0
 2 2]
原正確率: 0.3
新正確率: 1.0
```

有差異。在原本的Y中，類別標籤的分佈是有順序的，而KNeighborsClassifier會選擇K個最近鄰中出現最多次的類別作為待分類樣本的類別，此時，因有序排列，同一種類別會聚集。然而，在打亂Y之後，類別標籤的分佈較為均勻。這可能有助於分類器更好地泛化到未見過的數據，提高分類的正確率。

1.2 請用母數或無母數統計檢定以下accuracies中的結果是否和chance level (0.5)有差異? 若有, why? (4 points)

In [ ]:
```
Y=np.remainder(range(200),2)
print(Y) #Y的0和1個數一樣多
```

```
[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1]
```

In [ ]:
```
# 跑一百次測試:
clf=svm.SVC()
accuracies=[]
for i in range(100):
 X=np.random.rand(200,2) # X取亂數
 kf=model_selection.KFold(len(Y),shuffle=True) # Leave-one-out cross-validation
 sc=model_selection.cross_val_score(clf,X,Y,cv=kf)
 accuracies.append(sc.mean())
```

In [ ]:
```
# Please do your statistical tests here:
from scipy.stats import ttest_1samp

t_statistic, p_value = ttest_1samp(accuracies, 0.5)

print("t statistic:", t_statistic)
print("p value:", p_value)

# 根據p_value判斷是否有差異
alpha = 0.05
if p_value < alpha:
    print("accuracies與0.5有顯著差異")
else:
    print("accuracies與0.5沒有顯著差異")
```

```
t statistic: -1.4713413130333677
p value: 0.14437043182525255
accuracies與0.5沒有顯著差異
```

有差異

t statistic為負數，代表樣本均值小於期望值，正確率在100次實驗中小於0.5。

差異原因可能有以下三個：

1.隨機數據：np.random.rand()會生成一個均勻分布的隨機数据，可能不容易被SVM有效地分離，因此無法很好地分類这些數據。

2.樣本量和特徵：dataset包含200個數據，每個數據有2個特徵。由於數據和特徵數量較少，對於SVM，通常需要大量的數據和更多的特徵以獲得更好的模型。

3.數據分佈：由于數據是隨機生成，可能導致類別之間的邊界不明顯，或兩個類別的分布重疊多，使分類困難。因為SVM是一種用于處理線性可分或近似線性可分數據的演算法，當數據不滿足，性能可能會下降。

Please submit your notebook in PDF to NTU Cool by next Friday (11/10).