機器學習< Assignment #4 －SVM>

姓名：翁佳煌　　學號：409430030

首先先安裝 livsvm，並在 colab 上引入相關函式庫。

```
!make install
!pip install scipy
!pip install -U libsvm

import numpy as np
from libsvm.svmutil import *
```

▲圖 1

要求 1.

生成了 500 個數據點作為原始資料，其中 x 值為-100 到 100 之間均勻分佈的 500 個點，y 值為 2 * x 加上平均值為 0、標準差為 1 的正態分佈中隨機生成的噪聲。並且將原始資料轉換為 SVM 的輸入格式，包括特徵資料 data（由 x 和 y 組成的二維陣列）和標籤 labels（根據 y 和 2 * x 的關係生成的二元分類標籤，大於則為 1，小於則為-1）。

```
# Generate data
np.random.seed(0)
x = np.linspace(-100, 100, 500)
y = 2 * x + np.random.normal(0, 1, size=500)

# Transform data for SVM
data = np.array([x, y]).T
labels = np.where(y > 2*x, 1, -1)
```

▲圖 2

要求 2 和 3.

使用迴圈來進行 RBF SVM 模型的參數選擇，其中外層迴圈遍歷不同的 C 值（代表錯誤項的懲罰因子），內層迴圈遍歷不同的 gamma 值（代表 RBF 核函數的寬度參數）。在每一次參數組合下，使用 svm_train 函式進行 5 折交叉驗證，並計算準確度 accuracy。

在所有參數組合中選擇準確度最高的模型，並記錄其參數值 best_params。

使用選擇出的最佳參數值 best_params 再次使用 svm_train 函式來訓練 RBF SVM 模型，並得到訓練好的模型 model。

生成了 500 個測試資料點 test_data，並根據原始資料生成了對應的測試標籤 test_labels。

```
# RBF SVM with 5-fold cross validation
c_values = [0.01, 0.1, 1, 10, 100]
gamma_values = [0.0001, 0.001, 0.01, 0.1, 1]
best_accuracy = 0
for c in c_values:
  for gamma in gamma_values:
    params = '-t 2 -c {} -g {}'.format(c, gamma)
    accuracy = svm_train(labels, data, params + ' -v 5')
    if accuracy > best_accuracy:
      best_accuracy = accuracy
      best_params = params

# Train RBF SVM with the best parameters
model = svm_train(labels, data, best_params)

# Test RBF SVM with and without scaling
test_data = np.random.uniform(-100, 100, size=(500, 2))
test_labels = np.where(test_data[:, 1] > 2*test_data[:, 0], 1, -1)
```

▲圖 3

要求 4.

使用未進行標準化的測試資料 test_data 進行預測，並計算準確度 accuracy1。

使用原始資料的均值和標準差來進行測試資料的標準化，得到標準化後的測試資料 scaled_test_data。

使用標準化後的測試資料 scaled_test_data 進行預測，並計算準確度 accuracy2。

最後，將未進行標準化的測試資料的準確度 accuracy1 和標準化後的測試資料的準確度 accuracy2 輸出，分別顯示在不進行標準化和進行標準化的情況下，RBF SVM 模型的預測準確度。

```
# Without scaling
_, accuracy1, _ = svm_predict(test_labels, test_data, model)

# With scaling
means = np.mean(data, axis=0)
stds = np.std(data, axis=0)
scaled_test_data = (test_data - means) / stds
_, accuracy2, _ = svm_predict(test_labels, scaled_test_data, model)

print('Accuracy without scaling:', accuracy1[0])
print('Accuracy with scaling:', accuracy2[0])
```

▲圖 4

問題與討論

下圖 5 為印出結果，不同的交叉驗證準確度（Cross Validation Accuracy）在不同的情況下有所變化，從最低的 47.8%到最高的 87.6%不等。這可能表示模型在不同的交叉驗證折數或數據集上有不同的表現。對應的測試數據集上的分類準確度約為 94.4%。

此外，結果中還包含了兩種不同的準確度，分別是「Accuracy without scaling」和「Accuracy with scaling」。這是在對數據進行了特徵縮放（scaling）處理，並且對縮放前和縮放後的數據進行了準確度測試，結果顯示經過特徵縮放後，模型的準確度更高。

當使用特徵縮放後，模型可能會有更好的準確度表現，因為特徵縮放可以幫助模型更好地捕捉特徵之間的關係，提供更一致的特徵值範圍，從而使模型的預測更加穩定。而如果不進行特徵縮放，模型可能會在不同特徵間存在偏差，導致模型難以適應數據的不同尺度和範圍，進而影響準確度。

因此，特徵縮放可以對機器學習模型的準確度產生影響，並且可能導致使用特徵縮放後的準確度比未經特徵縮放的準確度更好。

```
Cross Validation Accuracy = 51.6%
Cross Validation Accuracy = 51.6%
Cross Validation Accuracy = 51.6%
Cross Validation Accuracy = 51.6%
Cross Validation Accuracy = 51.6%
Cross Validation Accuracy = 47.8%
Cross Validation Accuracy = 54.6%
Cross Validation Accuracy = 50.4%
Cross Validation Accuracy = 51.6%
Cross Validation Accuracy = 51.6%
Cross Validation Accuracy = 52.2%
Cross Validation Accuracy = 55.2%
Cross Validation Accuracy = 53.6%
Cross Validation Accuracy = 52.2%
Cross Validation Accuracy = 53.4%
Cross Validation Accuracy = 51.2%
Cross Validation Accuracy = 53.2%
Cross Validation Accuracy = 61.2%
Cross Validation Accuracy = 68.8%
Cross Validation Accuracy = 61%
Cross Validation Accuracy = 65.2%
Cross Validation Accuracy = 81.6%
Cross Validation Accuracy = 87.6%
Cross Validation Accuracy = 66.6%
Cross Validation Accuracy = 58.8%
Accuracy = 67.4% (337/500) (classification)
Accuracy = 94.4% (472/500) (classification)
Accuracy without scaling: 67.4
Accuracy with scaling: 94.39999999999999
```

▲圖 5