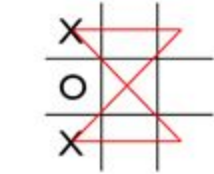
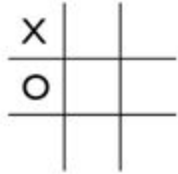
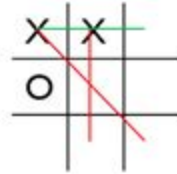


# Errata

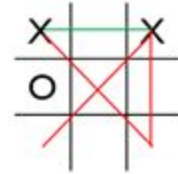
- หน้า 2, "หุ่นยนต์ (Robotics) ... สร้างร่างกาย**ใน**กับระบบปัญญาประดิษฐ์" แก้เป็น "หุ่นยนต์ (Robotics) ... สร้างร่างกาย**ให้**กับระบบปัญญาประดิษฐ์"
- หน้า 18, แก่สมการ 2.8 เป็น  $V(b) = X(b) - O(b) + 1$
- หน้า 19, แก่รูป 2.14 เป็น



$$V = (1+1+1) - 1 + 1 = 4$$



$$V = (2+1+1) - 1 + 1 = 4$$



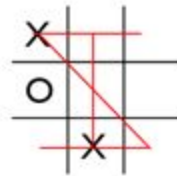
$$V = (2+1+1) - 1 + 1 = 5$$



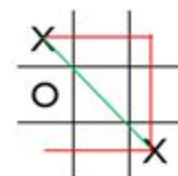
$$V = (2+1+1) - 0 + 1 = 6$$



$$V = (1+1+1) - 0 + 1 = 4$$



$$V = (1+1+1) - 1 + 1 = 4$$



$$V = (2+1+1) - 1 + 1 = 5$$

- หน้า 19, แก่รูป 2.15 เป็น

X		
O	X	
		O

X	X	
O	X	
		O

$$V = (2+2+1) - 2+1 = 4$$

X		X
O	X	
		O

$$V = (2+2+1) - 1+1 = 5$$

X		
O	X	X
		O

$$V = (1+1+1) - 1+1 = 3$$

X		
O	X	
X		O

$$V = (2+1+1) - 1+1 = 4$$

- หน้า 20, แก่รูป 2.16 เป็น

X	O	X
O	X	
		O

X	O	X
O	X	X
		O

$$V = 2 - 1 + 1 = 2$$

X	O	X
O	X	
X		O

$$V = 3 - 0 + 1 = 4$$

X	O	X
O	X	
	X	O

$$V = 2 - 0 + 1 = 3$$

**WIN**

- หน้า 21, เพิ่มย่อหน้าให้บรรทัด for w in win:

```
def calSOX(O,X):
    SO = SX = 0
    criticalmove = []
    for w in win:
        o = [i-1 in O for i in w]
        x = [i-1 in X for i in w]
        if not any(x):
            nO = o.count(True)
            SO += nO
            if nO == 2:
                print('critical',w)
                criticalmove = w
```

```
def calSOX(O,X):
    SO = SX = 0
    criticalmove = []
    for w in win:
        o = [i-1 in O for i in w]
        x = [i-1 in X for i in w]
        if not any(x):
            nO = o.count(True)
            SO += nO
            if nO == 2:
                print('critical',w)
                criticalmove = w
        if not any(o):
            SX += x.count(True)
    return SO,SX,criticalmove
```

21

- หน้า 41 แก้ในไฟล์ kNN.py โดยเพิ่ม k=1 ใน def kNN(Xtrain, Ytrain, Xtest, k=1):
- หน้า 42 ไฟล์ iris\_dataset.py ให้แก้ไขเป็น (อย่าลืมสร้างโฟลเดอร์ชื่อ dataset ไว้ที่เดียวกับ code ด้วยนะ ครับถ้าไม่เปลี่ยน path)

```
import pandas as pd
import os
import numpy as np

def load(path='./dataset/iris.csv', split_train_test=None):
    if os.path.isfile(path):
        iris = pd.read_csv(path)
    else:
        url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
        iris = pd.read_csv(url, header=None)
        iris.to_csv(path, index=False)
    X = iris.iloc[:, :4].values
    Y = iris.iloc[:, -1].values
    if split_train_test:
        classes = np.unique(Y)
        itrain = np.empty((0,)) , dtype=np.int)
        itest = np.empty((0,)) , dtype=np.int)
        for i in classes:
            idx = np.where(Y == i)[0]
            split = int(len(idx) * split_train_test)
            itrain = np.concatenate((itrain, idx[:split]))
            itest = np.concatenate((itest, idx[split:]))
        return X[itrain], Y[itrain], X[itest], Y[itest]
    return X, Y

if __name__ == '__main__':
    irisInputs, irisTargets = load()
    print(irisInputs)
    print(irisTargets)
```

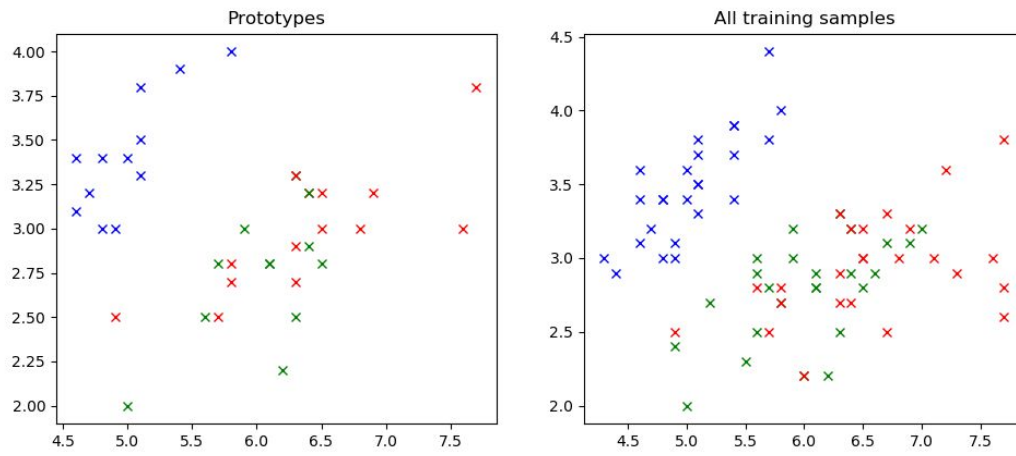
- หน้า 46, แก่ฟังก์ชัน CNN เป็นดังนี้

```
def CNN(Xtrain, Ytrain):
    nsample = Xtrain.shape[0]
    # Distance
    Dtrain = []
    for x in Xtrain:
        Dtrain.append(np.sqrt(np.sum((Xtrain - x)**2, axis=1)))
    Dtrain = np.array(Dtrain)

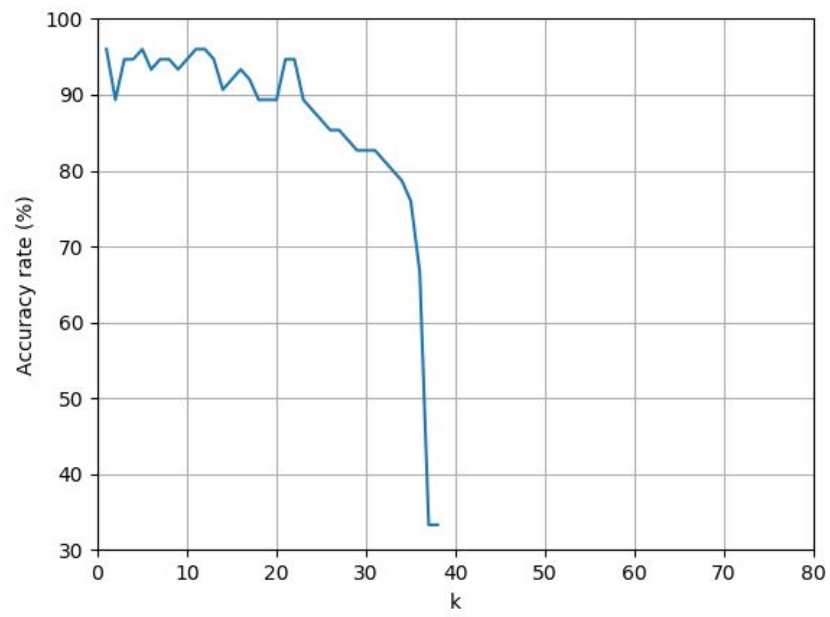
    # Border ratio
    a = []
    for i in range(nsample):
        # y = the closest sample from x with different class from x
        yidx = np.where(Ytrain != Ytrain[i])[0]
        yidx = yidx[np.argmin(Dtrain[i, yidx])]
        # xd = the closest sample from y with same class as x
        xdidx = np.where(Ytrain == Ytrain[i])[0]
        xdidx = np.delete(xdidx, np.where(xdidx == np.array(i)))
        xdidx = xdidx[np.argmin(Dtrain[yidx, xdidx])]
        a.append(Dtrain[yidx, xdidx] / Dtrain[i, yidx])
    # Scan order
    order = np.argsort(-np.array(a))
    # Prototypes
    Prototypes = order[0]
    i = 0
    while len(order) > 0 and i < len(order):
        idx = np.argsort(Dtrain[order[i], Prototypes])
        for j in idx:
            if Ytrain[order[i]] != Ytrain[order[j]]:
                Prototypes = np.append(Prototypes, order[i])
                order = np.delete(order, i)
                break
        i += 1

    return Prototypes
```

- หน้า 48 แก่รูปที่ 4.6 และ 4.7 เป็นดังนี้

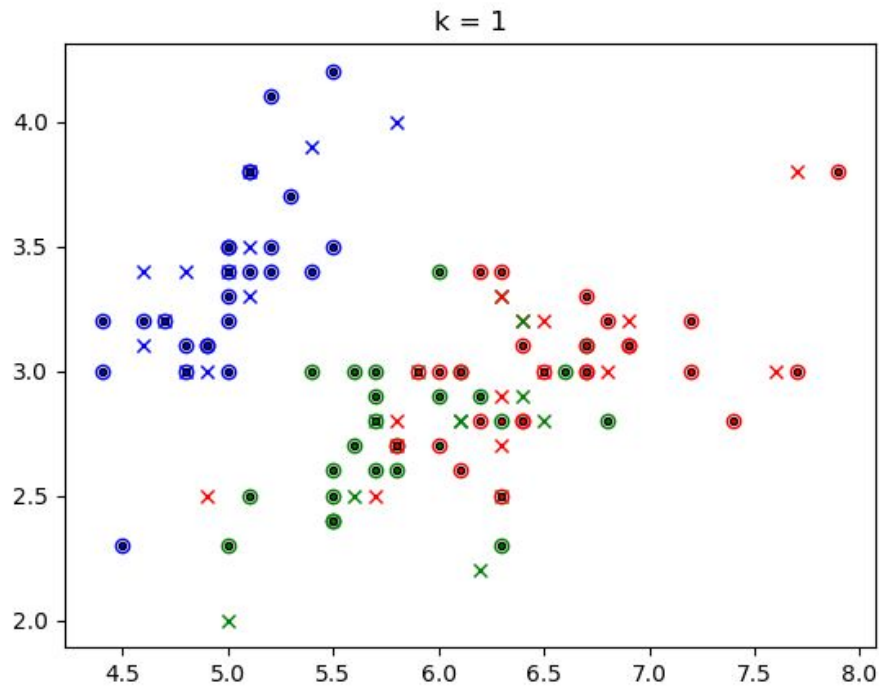


รูปที่ 4.6



รูปที่ 4.7

- หน้า 49 แก่รูปที่ 4.8 เป็นดังนี้



รูปที่ 4.8

- หน้า 87 แก่ในฟังก์ชัน predict จาก tree เป็น self ดังนี้

```
def predict(self, X):
    X = np.array(X)
    Y = []
    for i, x in enumerate(X):
        node = 0
        while True:
            # Find children
            children = [j for j, p in enumerate(self.parent) if p == node + 1]
            if len(children) == 0:
                Y.append(self.node[node])
                break
            not_found = True
            for c in children:
                if self.branch[c - 1] == x[self.attr_id[node]]:
                    node = c
                    not_found = False
                    break
            if not_found:
                Y.append(None) # Out of values
                break
    return np.array(Y)
```

- หน้า 204 ไฟล์ SVM.py ให้แก้ def predict อันแรกเป็น def \_\_predict ดังนี้

```
def __predict(self, X):
    return (self.y * self.alpha) @ Kernel(self.sv, X, kernel=self.kernel).K - self.bias

def predict(self, X):
    K = len(self.classes)
    if K == 2:
```

- หน้า 216 สมการ 8.21 บรรทัดสุดท้ายแก้จาก 0.21 เป็น 0.0376
- หน้า 224 ตัด code ตั้งแต่บรรทัด import spam\_dataset ออก เพราะส่วนนี้ถูกใช้ในหน้า 229
- หน้า 262 แก้ Y เป็น L ตรงตัว Highlight สีเหลือง

```
from PCA1 import PCA
import iris_dataset
X, L = iris_dataset.load()

from matplotlib import pyplot as plt
color = {'Iris-setosa': 'r',
         'Iris-versicolor': 'g',
         'Iris-virginica': 'b', }
marker = {'Iris-setosa': 'o',
          'Iris-versicolor': 'x',
          'Iris-virginica': 's', }
k = 1
for m in range(X.shape[1]-1):
    for n in range(m+1, X.shape[1]):
        plt.subplot(3, 2, k)
        k += 1
        for i in range(len(X)):
            plt.plot(X[i][m], X[i][n], marker[L[i]], c=color[L[i]], markersize=3)
            plt.xlabel('Attribute ' + str(m+1))
            plt.ylabel('Attribute ' + str(n+1))
plt.subplots_adjust(wspace=0.3)
plt.show()
```

- หน้า 268 ลิงค์ [http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att\\_faces.zip](http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip) ไม่สามารถใช้งานได้แล้วสามารถใช้ลิงค์ <https://www.kaggle.com/kasikrit/att-database-of-faces>
- หน้า 306 แก้ from SOM import SOM เป็น from SOM import SOM, plot