Report for HW1 6(2)

Instructions:

Environment: MacOS, Python 3.8

Package: numpy, pandas, sklearn, matplotlib

Sample: http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data

Dataset split: 70% training data, 30% test data

Test:

Test 1

Rate: 0.05

Max loop: 300

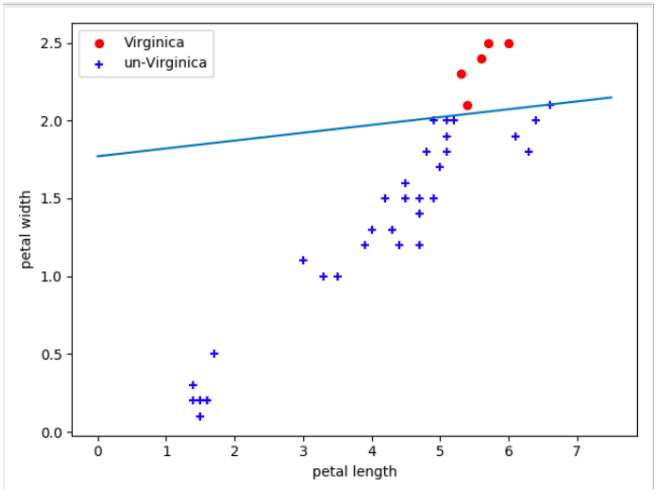
Theta: [[-55.7696507] [-1.58694847] [31.49766668]]

Right: 35

Wrong: 10

Accuracy: 77.8%

Plot:



Test 2

Rate: 0.05

Max loop: 1000

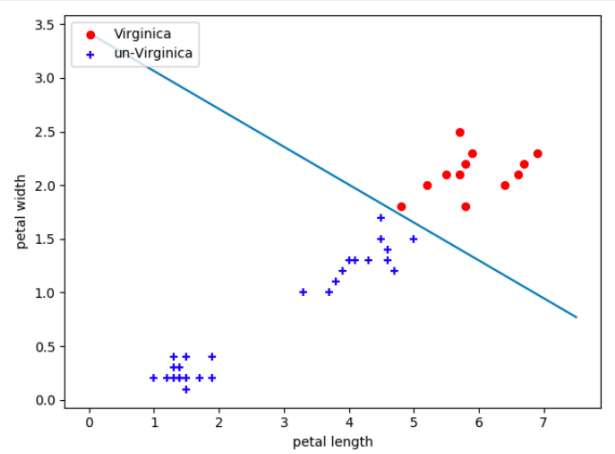
Theta: [[-77.18910843] [7.97248751] [22.5859258]]

Right: 43

Wrong: 2

Accuracy: 95.6%

Plot:



Test 3

Rate: 0.01

Max loop: 300

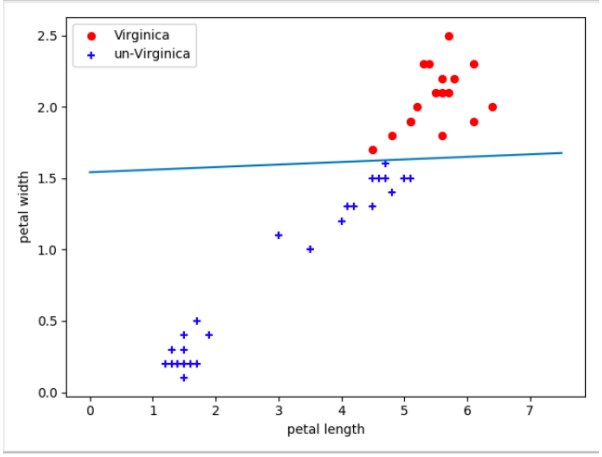
Theta: [[-11.22340884] [-0.1307119] [7.27943171]]

Right: 43

Wrong: 2

Accuracy: 96.7%

Plot:



Test 4

Rate: 0.01

Max loop: 300

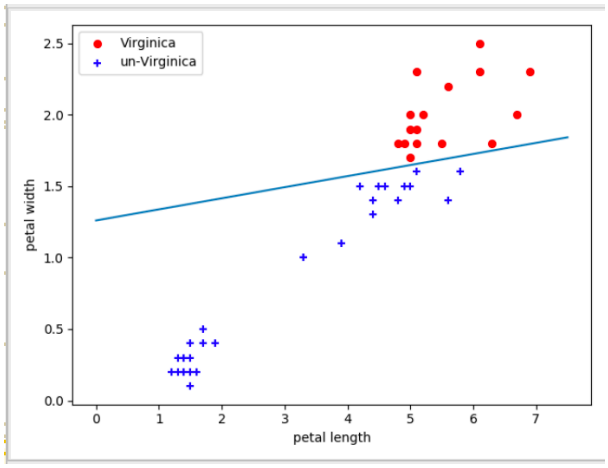
Theta: [[-10.01127119] [-0.61628711] [7.94608333]]

Right: 40

Wrong: 5

Accuracy: 88.9%

Plot:



Test 5

Rate: 0.01

Max loop: 600

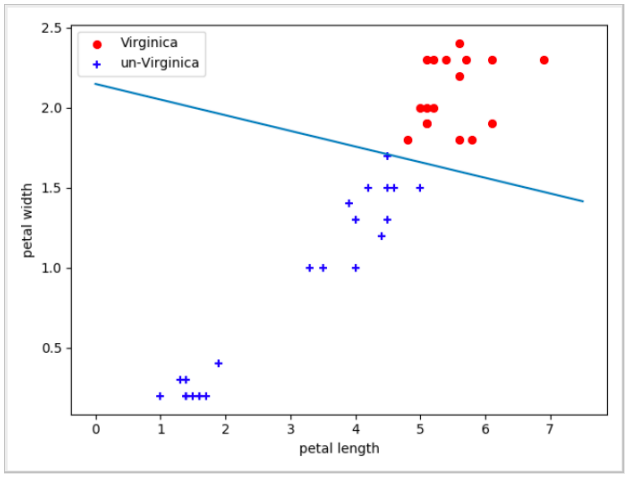
Theta: [[-14.05622584] [0.63966893] [6.54384881]]

Right: 43

Wrong: 2

Accuracy: 96%

Plot:



Code:

```
##
# NAME: Yiqun Pengs
##

import sys
from numpy import *
import pandas
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as pyplot


def dataProcessing(data):
    dataMat = []
    labelMat = []
    for i in range(data.iloc[:, 0].size):
        dataMat.append([1.0, float(data.iloc[i, 2]), float(data.iloc[i, 3])])  #
theta1 + theta12 * X1 + theta13 * X2
        if data.iloc[i, 4] == 'Iris-setosa' or data.iloc[i, 4] == 'Iris-
versicolor':
            labelMat.append(0)
        else:
            labelMat.append(1)
    return dataMat, labelMat


def getTheta(x, y):
    dataMatrix = mat(x)  # X:105x3
    labelMatrix = mat(y).transpose()  # Y:1x105
    m, n = shape(dataMatrix)  # 105x3
    weights = ones((n, 1))  # 3x1
    rate = 0.01
    maxLoop = 600
    for i in range(maxLoop):
        predict = sigmoid(dataMatrix * weights)  # 105x1
        error = labelMatrix - predict  # 105x1
        minLoss = -dataMatrix.transpose() * error  # 3x1
        weights = weights - rate * minLoss
    return weights
```

```
def sigmoid(z):
    return 1.0 / (1 + exp(-z))


if __name__ == "__main__":

    iris = pandas.read_csv('http://archive.ics.uci.edu/ml/machine-learning-
databases/iris/iris.data', header=None)
    training, test = train_test_split(iris, test_size=0.3)

    ## training
    x, y = dataProcessing(training)
    theta = getTheta(x, y)
    print('X:', x)
    print('Y:', y)
    print('Theta:', theta)
    print('\n')

    ## predict
    test_x, test_y = dataProcessing(test)
    result = sigmoid(mat(test_x) * theta)
    predict = (sign(result - 0.5) + 1) / 2  # 0 to 1 -> -0.5 to 0.5 -> -1 or 1
-> 0 or 2 -> 0 or 1
    # for j in range(len(result)):
    #   if result[j,0] > 0.5:
    #      result[j,0] = 1
    #  else:
    #      result[j,0] = 0
    # print(result)
    print('predict', predict)

    ## score
    right = 0
    wrong = 0
    for i in range(len(test_y)):
        if predict[i] == test_y[i]:
            right = right + 1
        else:
            wrong = wrong + 1
    accuracy = right / (right + wrong)
    print('Right:', right)
```

```
print('Wrong', wrong)
print('Accuracy', accuracy)

## plot the classifier
resultPlot = array(test_x)
len = shape(resultPlot)[0]
x1 = [];
y1 = []
x2 = [];
y2 = []
for k in range(len):
    if int(predict[k]) == 1:
        x1.append(resultPlot[k, 1]);
        y1.append(resultPlot[k, 2])
    else:
        x2.append(resultPlot[k, 1]);
        y2.append(resultPlot[k, 2])
pyplot.scatter(x1, y1, s=30, c='red', marker='o', label='Virginica')
pyplot.scatter(x2, y2, s=30, c='blue', marker='+', label='un-Virginica')
xl = arange(0, 8, 0.5)
yl = (-theta[0] - theta[1] * xl) / theta[2]
yl = transpose(yl)
pyplot.plot(xl, yl)
pyplot.legend(loc=2)
pyplot.xlabel('petal length')
pyplot.ylabel('petal width')
pyplot.show()
```