HW1 report

Qiong Wang
5906740674

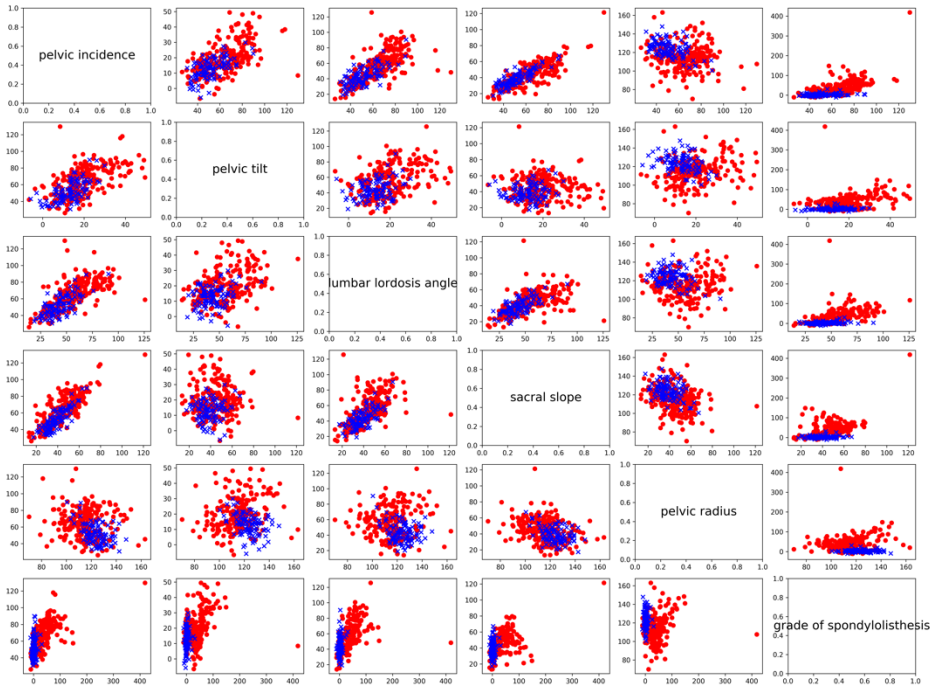
(a) Download the data set: column_2C_weka.arff

(b) 1. Scatterplot:

Red: Abnormal

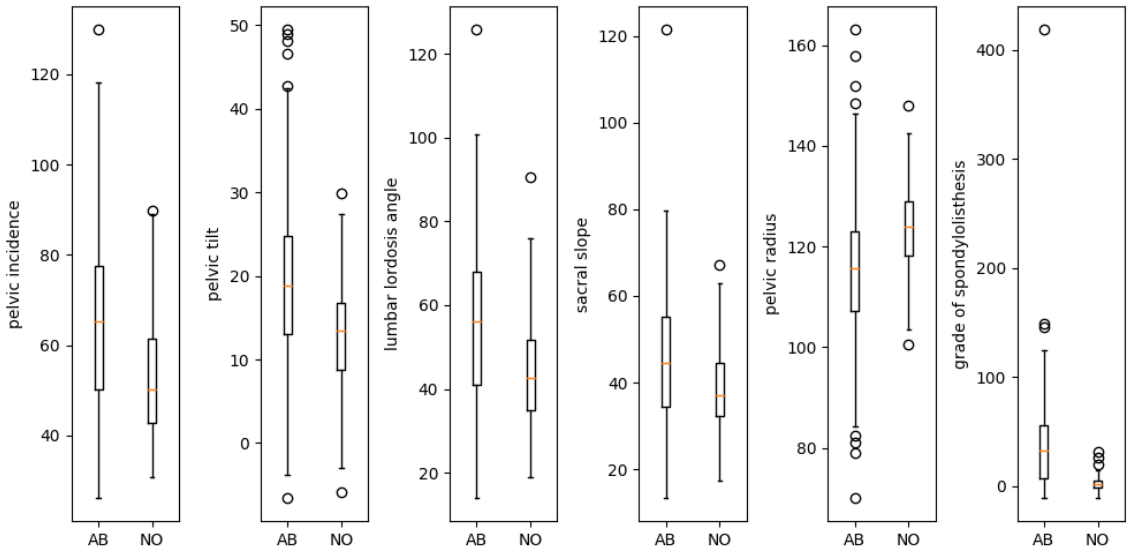
Blue: Normal

The code is B1.py



2. Boxplot:

The code is B2.py

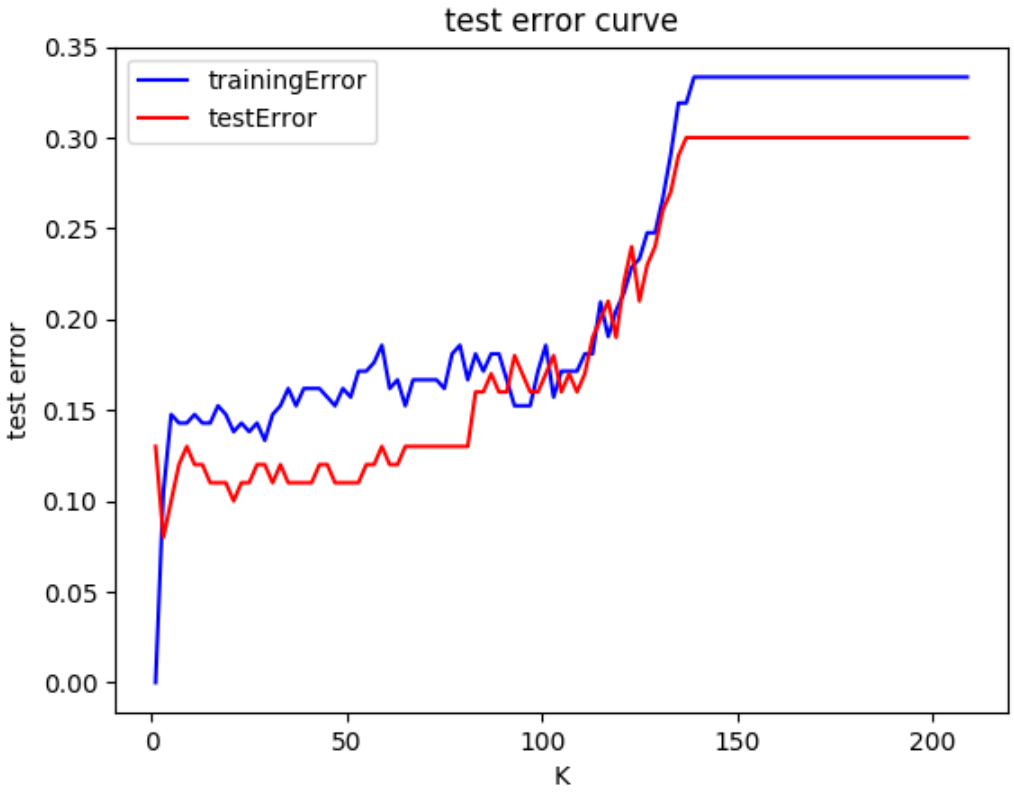


3. first 70 rows of class 0 and the first 140 rows of class 1 are training set and the rest of data are test set

(c) 1. I use sklearn package

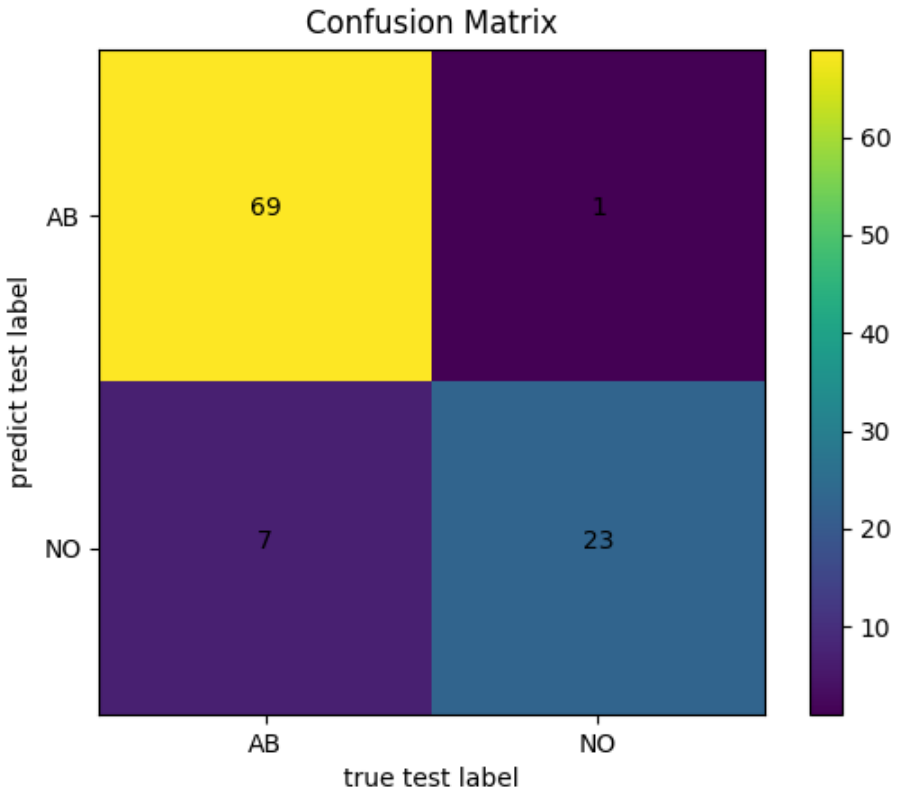
2. k = {1, 3, 5……,207}

The code is C2.py



Best k = 3

Best test error rate = 0.07999999999999996



When k = 3

TPR = TP/(TP+FN) = 0.9079

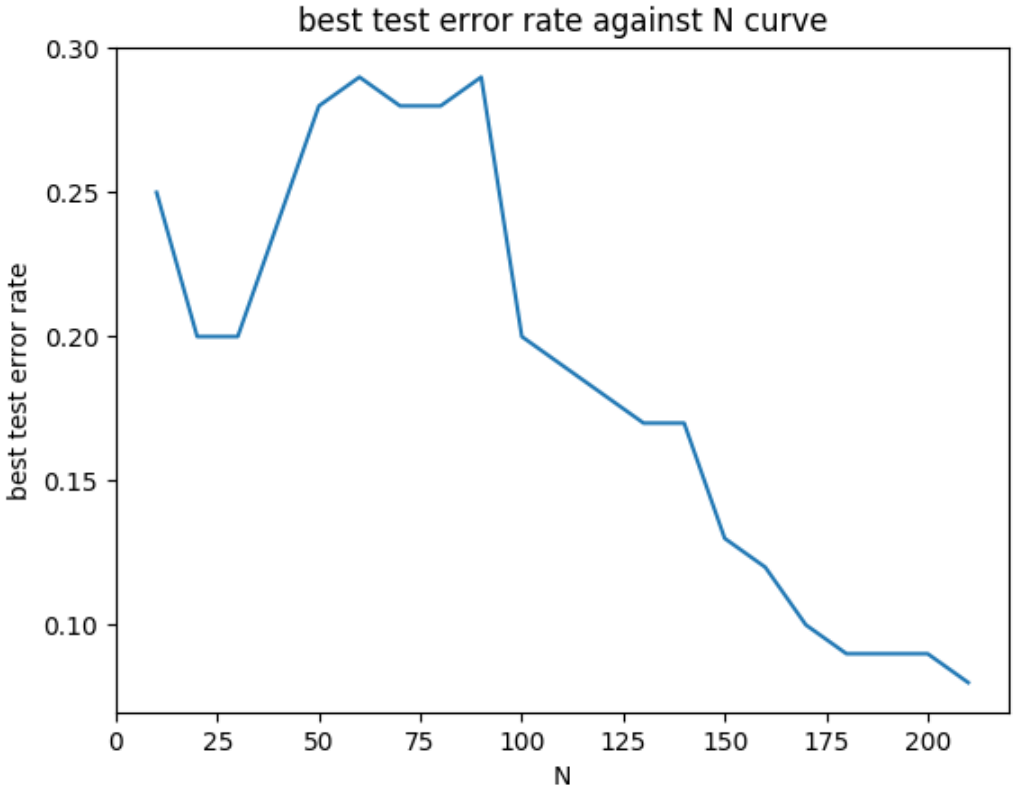
TNR = TN/(TN+FP) = 0.9583

Precision = TP/(TP+FP) = 0.9857

F-score = 0.9452

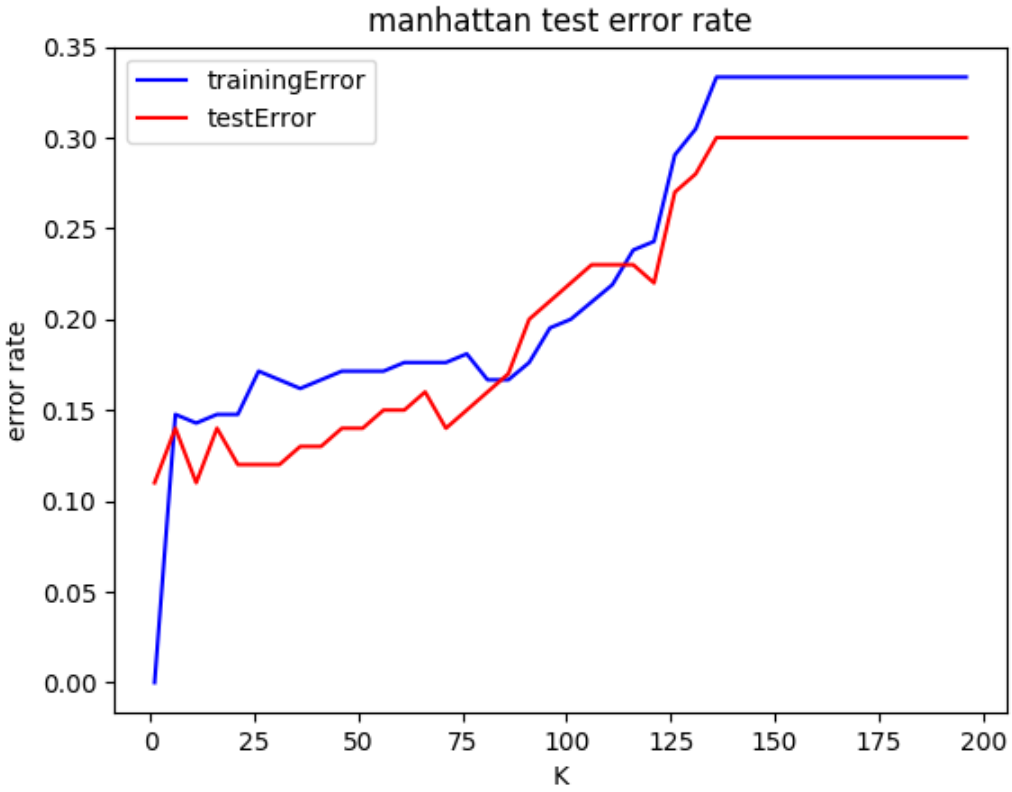
3. learning curve

The code is C3.py



(d) Replace the Euclidean metric

The code is D.py

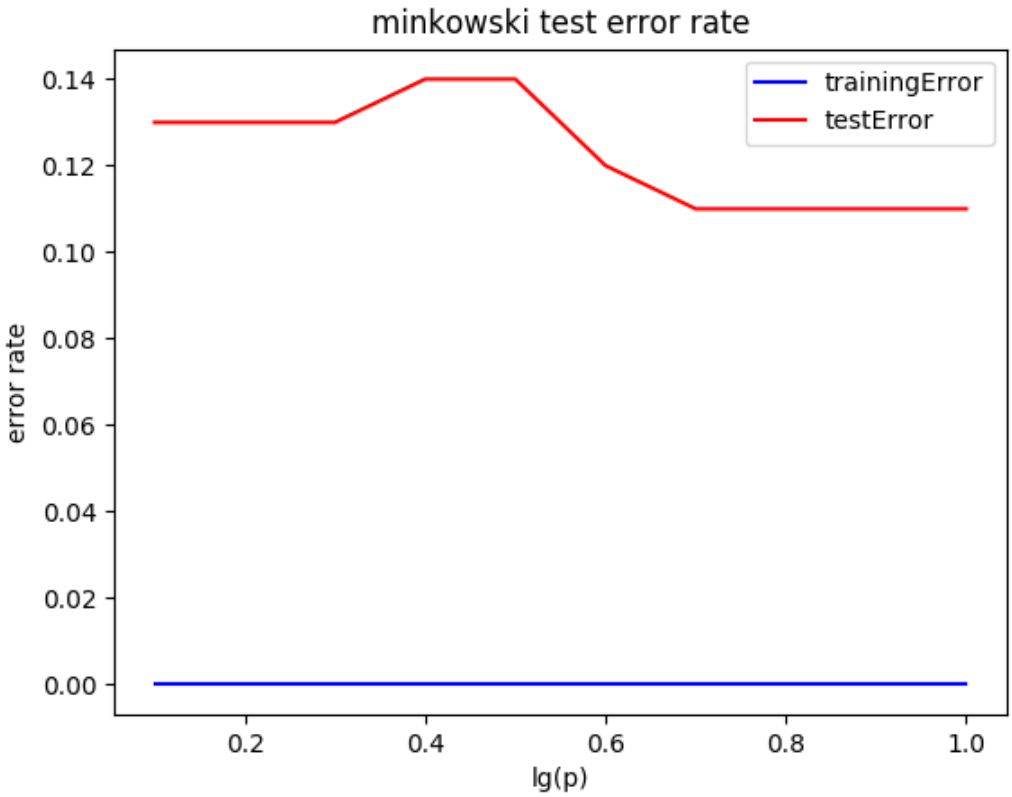


When p = 1

Manhattan metric:

Best k = 1

Best error rate = 0.10999999999999999



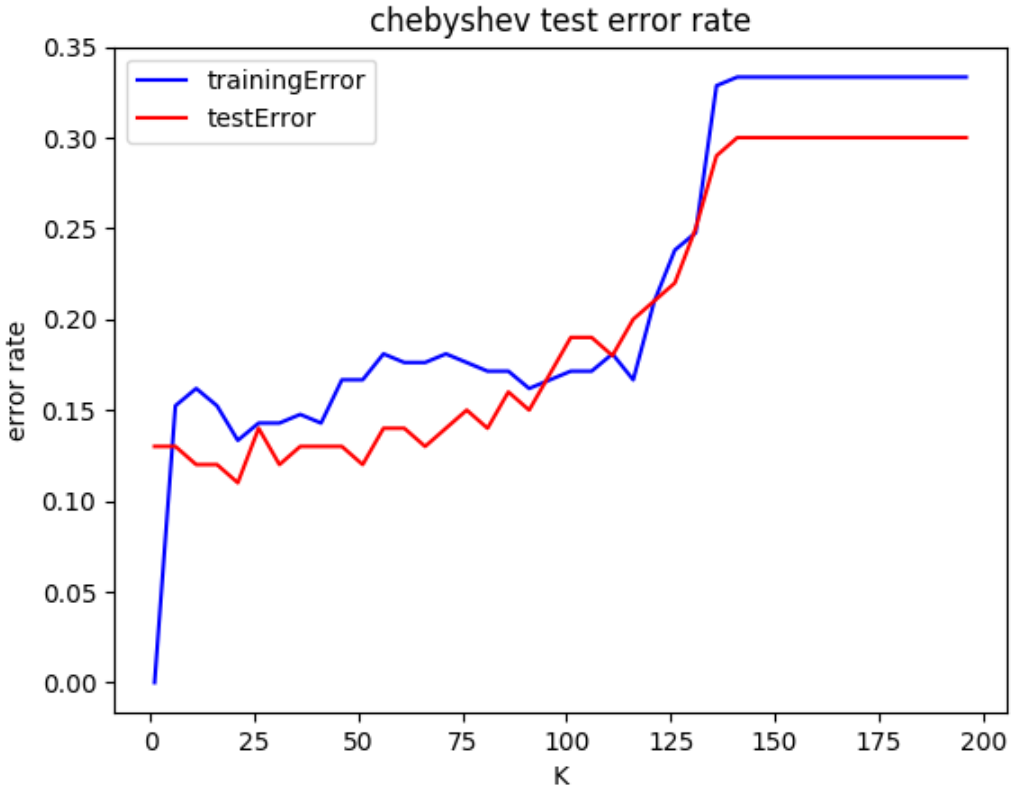
Here k = 1, so the training error is always 0

When log(p) = [0.1, 0.2, ……, 1]

Minkowski metric:

Best lg(p) = 0.7

Best test error = 0.10999999999999999

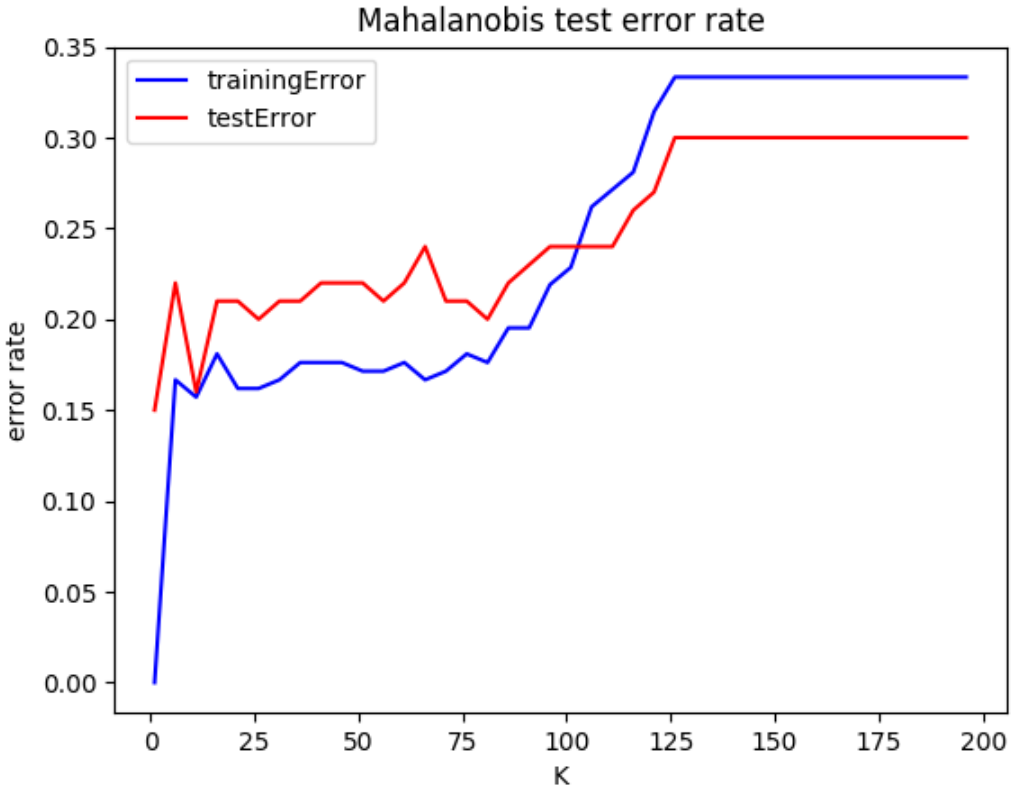


When p is infinite

Chebyshev metric:

Best k = 21

Best test error = 0.10999999999999999



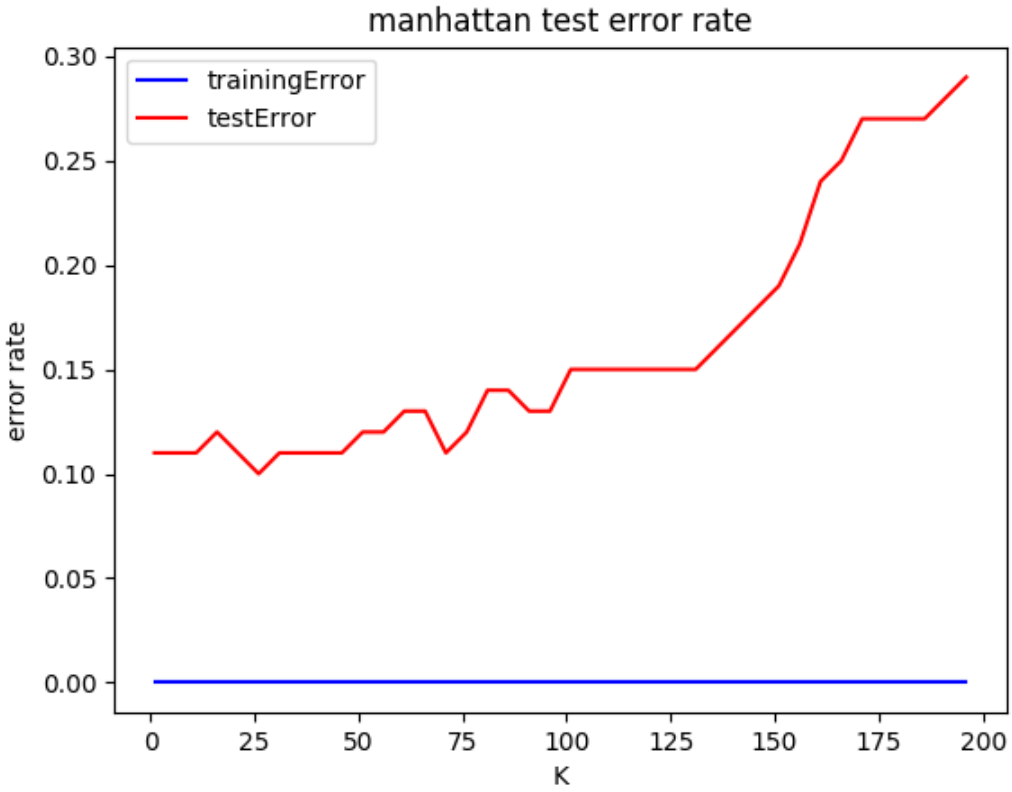
Mahalanobis metric:

Best k = 1

Best error rate = 0.15000000000000002

(e) In weighted decision situation

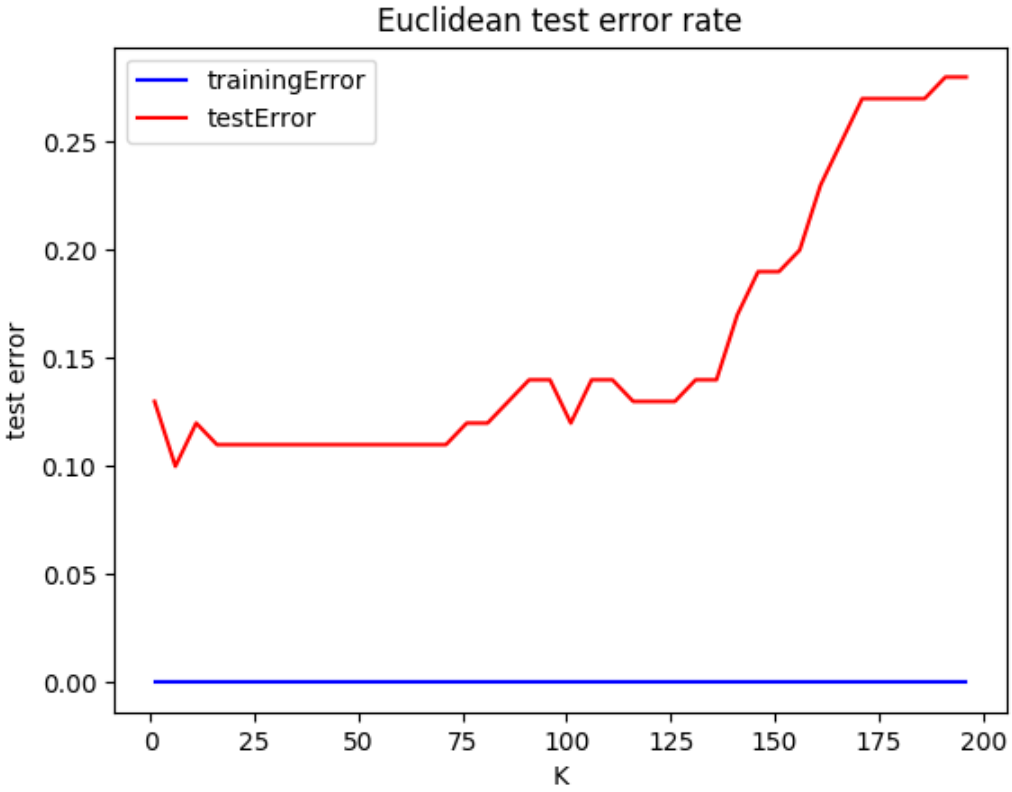
The code is E.py



Manhattan metric:

best k = 26

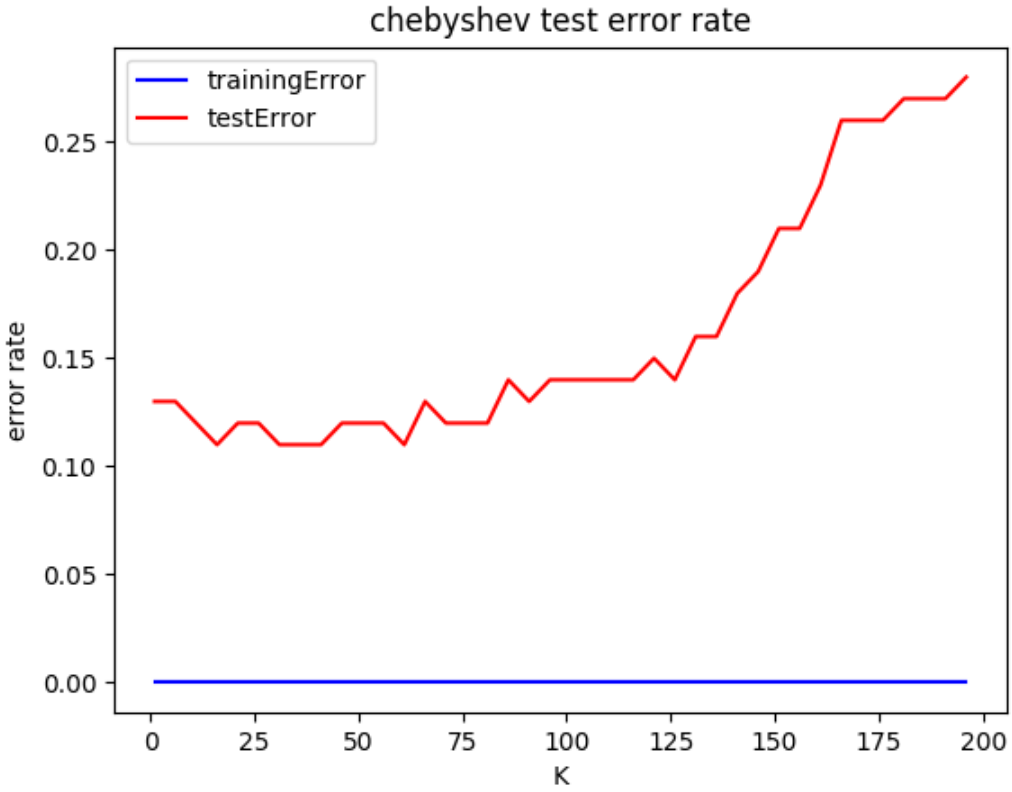
error rate = 0.09999999999999998



Euclidean metric:

Best k = 6

Test error = 0.09999999999999998



Chebyshev metric:
Best k = 16
Test error = 0.10999999999999999

(f) When K=1 or weighted decision (inversely proportional to its distance), the training error rate = 0, which is the lowest training error.