

Data Science Exercise 2 Report

Class: CSIE 4A

ID: 105502506

Name: 黃靖筌

Result1:

Train set: 72
Test set: 78
Accuracy(k=1): 94.87179487179486%
Accuracy(k=2): 94.87179487179486%
Accuracy(k=3): 96.15384615384616%
Accuracy(k=4): 96.15384615384616%
Accuracy(k=5): 97.43589743589743%
Accuracy(k=6): 97.43589743589743%
Accuracy(k=7): 96.15384615384616%
Accuracy(k=8): 97.43589743589743%
Accuracy(k=9): 97.43589743589743%
Accuracy(k=10): 97.43589743589743%
Accuracy(k=11): 97.43589743589743%
Accuracy(k=12): 96.15384615384616%
Accuracy(k=13): 98.71794871794873%
Accuracy(k=14): 96.15384615384616%
Accuracy(k=15): 94.87179487179486%
Accuracy(k=16): 94.87179487179486%
Accuracy(k=17): 94.87179487179486%
Accuracy(k=18): 97.43589743589743%
Accuracy(k=19): 96.15384615384616%
Accuracy(k=20): 96.15384615384616%

Result2:

Train set: 80
Test set: 70
Accuracy(k=1): 95.71428571428572%
Accuracy(k=2): 95.71428571428572%
Accuracy(k=3): 95.71428571428572%
Accuracy(k=4): 95.71428571428572%
Accuracy(k=5): 95.71428571428572%
Accuracy(k=6): 95.71428571428572%
Accuracy(k=7): 95.71428571428572%
Accuracy(k=8): 95.71428571428572%
Accuracy(k=9): 98.57142857142858%
Accuracy(k=10): 97.14285714285714%
Accuracy(k=11): 97.14285714285714%
Accuracy(k=12): 95.71428571428572%
Accuracy(k=13): 95.71428571428572%
Accuracy(k=14): 94.28571428571428%
Accuracy(k=15): 92.85714285714286%
Accuracy(k=16): 94.28571428571428%
Accuracy(k=17): 92.85714285714286%
Accuracy(k=18): 92.85714285714286%
Accuracy(k=19): 92.85714285714286%
Accuracy(k=20): 94.28571428571428%

We learned in the theory that we usually choose $k = \sqrt{n}$ for the best choice of k .

For Result 1, we have 72 training data, 78 test data. We should choose 9 as k for it is closer to $\sqrt{72}$ and it is an odd number. However when $k = 13$, we have the highest accuracy. I consider it a sweet coincident.

For Result 2, we have 80 training data, 70 test data. We choose 9 as k and it is obviously the highest accuracy among all.

Ref: <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/>