Assignment 7

Josh Barber

barb4630@mylaurier.ca

215854630

1. implement a **mydist** function as in Code Listing 7.04. Hint: use one common distance metric in this [Wikipedia page](https://en.wikipedia.org/wiki/Metric_(mathematics)) (1 point)

I used the Manhattan Distance algorithm:

# Manhattan Distance metric

def mydist(x, y):

distance = 0

for x1, y1 in zip(x, y):

difference = y1 - x1

absolute\_difference = abs(difference)

distance += absolute\_difference

return distance

1. use the **mydist** function for a k-NN classifier with k=10 and train it with 80% of the breast cancer training set (we can reuse Code Listing 7.05) (1 point)

from sklearn.datasets import load\_breast\_cancer

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

def warn(\*args, \*\*kwargs):

pass

import warnings

warnings.warn = warn

data\_bunch = load\_breast\_cancer()

bc\_X, bc\_y = data\_bunch.data, data\_bunch.target

x\_train, x\_test, y\_train, y\_test = train\_test\_split(bc\_X, bc\_y, test\_size=0.2)

# Manhattan Distance metric

def mydist(x, y):

distance = 0

for x1, y1 in zip(x, y):

difference = y1 - x1

absolute\_difference = abs(difference)

distance += absolute\_difference

return distance

knn\_regr = KNeighborsRegressor(n\_neighbors=10, metric=mydist)

knn\_regr.fit(x\_train, y\_train)

print("Performance of kNN regressor with n = 10 with Manhattan Distance:", knn\_regr.score(x\_test, y\_test))

bc\_y\_pred = knn\_regr.predict(x\_test)

print("Mean squared error: %.2f" % mean\_squared\_error(y\_test, bc\_y\_pred))

print("Coefficient of determination: %.2f" % r2\_score(y\_test, bc\_y\_pred))

print("\n")

knn\_regr\_default = KNeighborsRegressor(n\_neighbors=10)

knn\_regr\_default.fit(x\_train, y\_train)

print("Performance of kNN regressor with n = 10 with default metric:", knn\_regr\_default.score(x\_test, y\_test))

bc\_y\_pred\_default = knn\_regr\_default.predict(x\_test)

print("Mean squared error: %.2f" % mean\_squared\_error(y\_test, bc\_y\_pred\_default))

print("Coefficient of determination: %.2f" % r2\_score(y\_test, bc\_y\_pred\_default))

1. evaluate the performance of the k-NN classifier by the average accuracy score on the breast cancer test set and compare its performance with a k-NN classifier with k=10 and it uses the default distance metric. Write down the comparison result. (1 point)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Run 1 Score | Run 2 Score | Run 3 Score | Avg. Score |
| Default Metric | 0.8066871794871 | 0.7867023053792 | 0.8701666666666 | 0.8211853838443 |
| Manhattan Dist. | 0.8429333333333 | 0.8293618443033 | 0.8955 | 0.8559317258788 |

It seems that the Manhattan Distance outperforms the default metric, which is I believe the Minkowski distance, with p=2 (Euclidean Distance).