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)**

**Code:**

# calculate the Euclidean distance between x and y

def mydist(x, y):

distance = 0.0

for i in range(len(x)-1):

distance += (x[i] - y[i])\*\*2

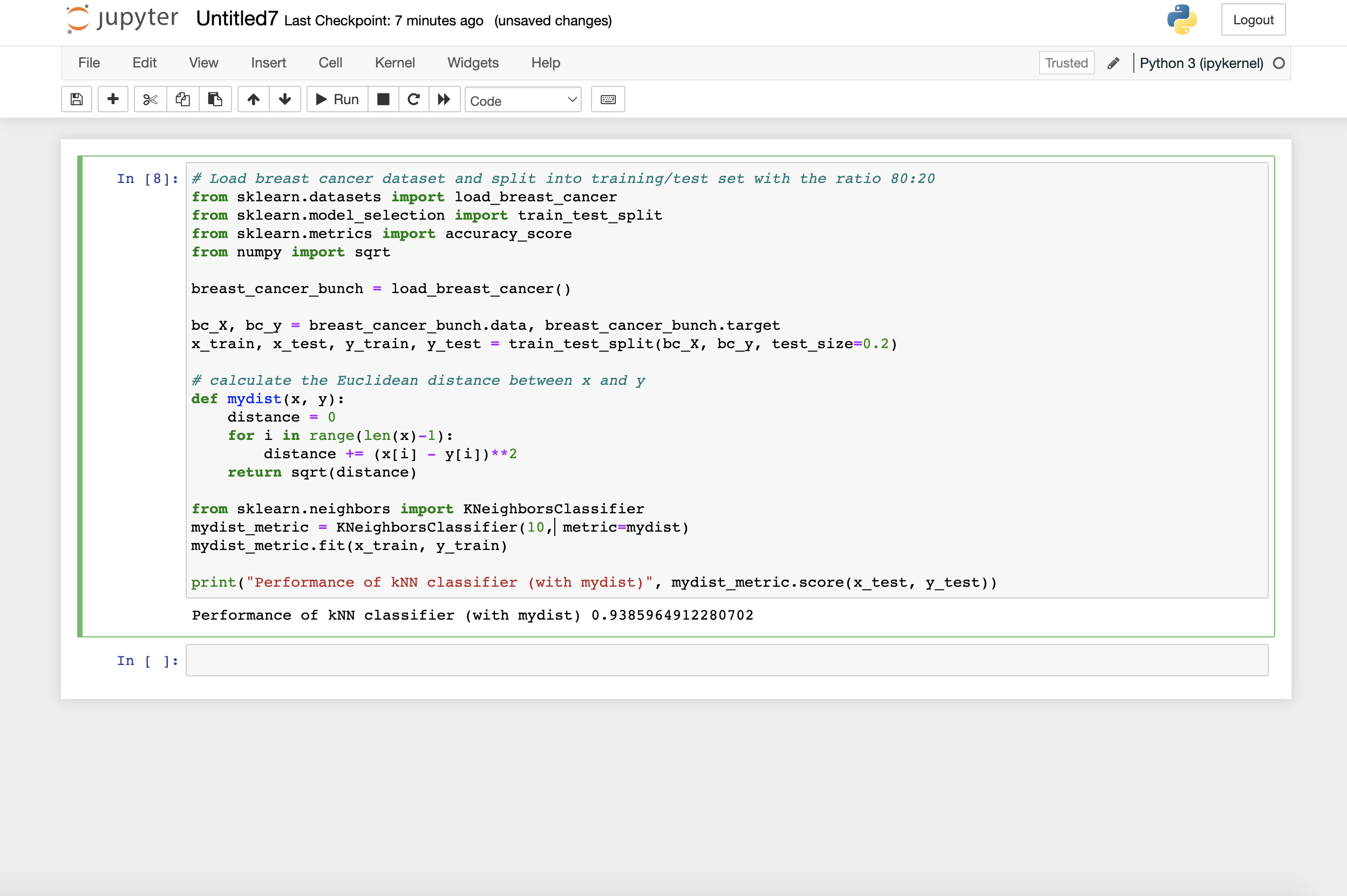
return sqrt(distance)

I used the Euclidean metric from the Wikipedia page:

<https://en.wikipedia.org/wiki/Euclidean_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)**

**Output:**



Performance of kNN classifier (with mydist) 0.9385964912280702

**Code:**

# Load breast cancer dataset and split into training/test set with the ratio 80:20

from sklearn.datasets import load\_breast\_cancer

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

from numpy import sqrt

breast\_cancer\_bunch = load\_breast\_cancer()

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

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

# calculate the Euclidean distance between x and y

def mydist(x, y):

distance = 0

for i in range(len(x)-1):

distance += (x[i] - y[i])\*\*2

return sqrt(distance)

from sklearn.neighbors import KNeighborsClassifier

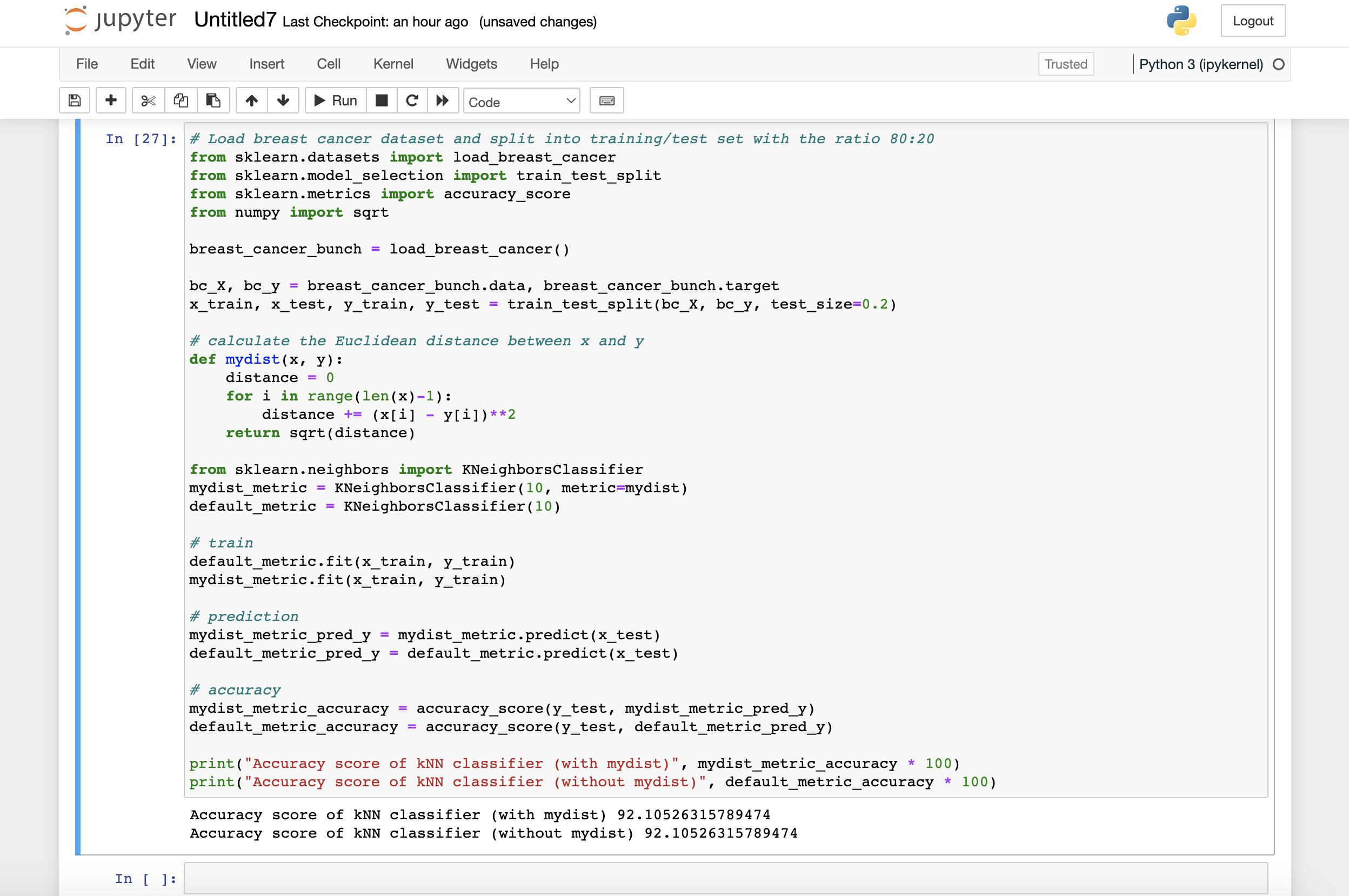
mydist\_metric = KNeighborsClassifier(10, metric=mydist)

mydist\_metric.fit(x\_train, y\_train)

print("Performance of kNN classifier (with mydist)", mydist\_metric.score(x\_test, y\_test))

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)**

**Output:**



Run 1:

Accuracy score of kNN classifier (with mydist) 92.10526315789474  
Accuracy score of kNN classifier (without mydist) 92.10526315789474

Run 2:

Accuracy score of kNN classifier (with mydist) 90.35087719298247  
Accuracy score of kNN classifier (without mydist) 90.35087719298247

Run 3:

Accuracy score of kNN classifier (with mydist) 92.98245614035088  
Accuracy score of kNN classifier (without mydist) 92.98245614035088

The default distance metric (which is, according to <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>, the Minkowski metric) performs identically to the Euclidean metric I’ve implemented in the mydist() function. At first I thought this was an error, after repeated runs they are always identical. However, if I change the metric of the default model to be cosine (metric=‘cosine’), the accuracy score does change, so I do not believe this is an error. It appears that the performance of the default metric and mydist() custom Euclidean metric are identical.

**Code:**

# Load breast cancer dataset and split into training/test set with the ratio 80:20

from sklearn.datasets import load\_breast\_cancer

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

from sklearn.metrics import accuracy\_score

from numpy import sqrt

breast\_cancer\_bunch = load\_breast\_cancer()

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

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

# calculate the Euclidean distance between x and y

def mydist(x, y):

distance = 0

for i in range(len(x)-1):

distance += (x[i] - y[i])\*\*2

return sqrt(distance)

from sklearn.neighbors import KNeighborsClassifier

mydist\_metric = KNeighborsClassifier(10, metric=mydist)

default\_metric = KNeighborsClassifier(10)

# train

default\_metric.fit(x\_train, y\_train)

mydist\_metric.fit(x\_train, y\_train)

# prediction

mydist\_metric\_pred\_y = mydist\_metric.predict(x\_test)

default\_metric\_pred\_y = default\_metric.predict(x\_test)

# accuracy

mydist\_metric\_accuracy = accuracy\_score(y\_test, mydist\_metric\_pred\_y)

default\_metric\_accuracy = accuracy\_score(y\_test, default\_metric\_pred\_y)

print("Accuracy score of kNN classifier (with mydist)", mydist\_metric\_accuracy \* 100)

print("Accuracy score of kNN classifier (without mydist)", default\_metric\_accuracy \* 100)