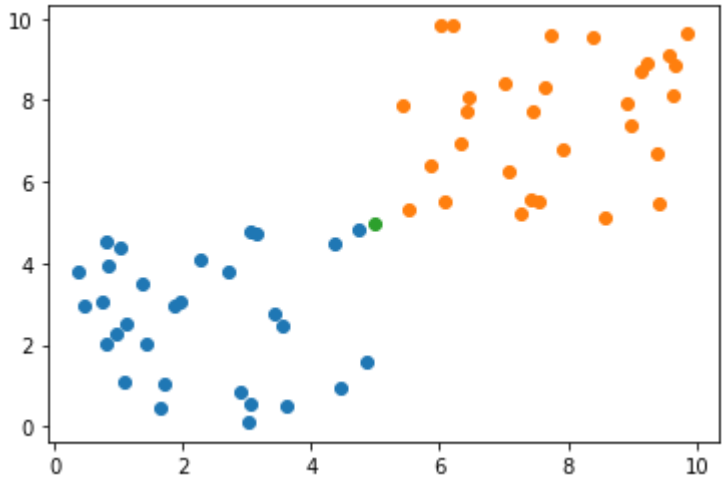
```
import pandas as pd
import numpy as np
import random
```

Generujemy zbiór danych:

```
X_A,X_B,Y_A,Y_B = [],[],[],[]
for j in range(30):
  X_A.append(random.random()*4.9)
  X_B.append(random.random()*5.1+5)
  Y_A.append(random.random()*5.1)
  Y_B.append(random.random()*4.9+5)
```

Punkty podzielone są na dwie klasy. Interesuje nas przynależność punktu (5,5) - punkt zielony.

```
import matplotlib.pyplot as plt
plt.scatter(X_A,Y_A)
plt.scatter(X_B,Y_B)
plt.scatter(5,5)
plt.show()
```



Łączymy współrzędne w krotki i zapisujemy je w tablicy. Etykiety umieszczamy w tablicy.

```
X_A = np.array(X_A)
X_B = np.array(X_B)
Y_A = np.array(Y_A)
Y_B = np.array(Y_B)
X = np.concatenate([X_A,X_B])
Y = np.concatenate([Y_A,Y_B])

data = list(zip(X,Y))
L_0 = np.full((30,),0)
L_1 = np.full((30,),1)
L = np.concatenate([L_0,L_1])
data,L
```

```
 (4.459233919235428, 0.9629914464592356),
 (3.041118979100316, 0.5441236766451497),
 (1.9703797677356307, 3.0561516823583066),
 (4.864913982433372, 1.5793828842514333),
 (4.733835918593861, 4.8262896266309685),
 (1.1109784794943462, 2.511524337525316),
 (1.4481497250663267, 2.042563664110412),
 (2.2784001029825465, 4.082563287603871),
 (3.5677644856977144, 2.483048323247082),
 (4.362948109841908, 4.466492921227131),
 (3.0330459955090734, 0.11597616626244223),
 (0.8543233386074509, 3.960220640666344),
 (1.8555545813876895, 2.9458846747718215),
 (1.3641319988938372, 3.4931405646913323),
 (0.8130990796821035, 4.532618376359794),
 (3.063254643519773, 4.783749102756461),
 (1.6992989920019572, 1.0310619532487744),
 (2.718775797549668, 3.8145562742992123),
 (0.9599115050325665, 2.2702690187042047),
 (3.1424076231709277, 4.742520761713112),
 (0.7346524930682816, 3.0715291731065677),
 (2.9145256299908406, 0.849254945880444),
 (1.034609512918968, 4.405086550560783),
 (1.1056699256850997, 1.1017493731105126),
 (0.8117658813314549, 2.0180764607592456),
 (3.4238991946963604, 2.757068117339592),
 (6.441095747995938, 8.057364353218357),
 (7.932387341773877, 6.81212086769558),
 (8.387717735676143, 9.53119038707946),
 (7.532013233707925, 5.519184025437082),
 (5.523738536028185, 5.341923993453809),
 (7.441610000824111, 7.712133213594188),
 (7.638548956480964, 8.309098692999953),
 (9.675559797356655, 8.845141877830422),
 (9.62572998981317, 8.144438340522342),
 (8.569796174173586, 5.1480701858739275),
 (6.21371392361088, 9.828932880232463),
 (5.863137081131592, 6.412905895622561),
 (9.866155642193302, 9.63931324588986),
 (6.093737171526932, 5.518372226660665),
 (6.419672994114615, 7.714685364230203),
 (9.559201240494444, 9.119787270542789),
 (9.376168109388212, 6.702692128301865),
 (7.719514058581455, 9.600685149818098),
 (7.2747050861691385, 5.202374321232242),
 (8.975105462601634, 7.406621967075468),
 (7.09021644506945, 6.238385323665208),
 (9.237370527559289, 8.919932327196099),
 (9.407206010234907, 5.480170542320798),
 (7.426492089748249, 5.547415673319502),
 (9.148392269336519, 8.698826216856565),
 (6.320927090673742, 6.934105880083319),
 (7.031249408817753, 8.403047299516157),
 (5.421262906817128, 7.89787219295391),
 (6.0135177262058015, 9.816446662581878),
 (8.929833996881953, 7.914726371132247)],
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]))
```

Klasyfikacja dla k=3:

```
from sklearn.neighbors import KNeighborsClassifier

neigh = KNeighborsClassifier(n_neighbors=6)
neigh.fit(data, L)

point = np.array([[5,5]])

print(neigh.predict(point))
```

```
     [0]
```

mój kNN:

```
import math

k = 5;
data_point = np.array([[5, 5]])
new_arr = np.array([[0, 0, 0, 0]])

for i in range(len(data)):
    d = math.sqrt( ((float(data[i][0]) - data_point[0][0]) **2) + (float(data[i][1]) - dat
    new_arr = np.append(new_arr, [[d, L[i], data[i][0], data[i][1]]], axis = 0)

new_arr = np.delete(new_arr, 0, axis = 0)
sorted_arr = new_arr[np.argsort(new_arr[:,0])]
nearest_tab = []

for i in range(k):
  nearest_tab.append([NewData[i, 0], NewData[i, 1], NewData[i, 2], NewData[i, 3]])

#print(np.sum(nearest_tab[:,1] == 1.0))
#print(np.sum(nearest_tab[:,1] == 0.0))

if (sum(row.count(1.0) for row in nearest_tab) > sum(row.count(0.0) for row in nearest_tab
  print("Wynik: ", 1);
else:
  print("Wynik: ", 0);
```

```
     Wynik:  1
```

```
import matplotlib.patheffects as path_effects

plt.scatter(X_A,Y_A)
plt.scatter(X_B,Y_B)

for i in range(k):
  if (sorted_arr[i][1] == 0.0):
    plt.scatter(sorted_arr[i][2], sorted_arr[i][3], c = "#00008B");
  else:
```

```
    plt.scatter(sorted_arr[i][2], sorted_arr[i][3], c = "#ff3700");

plt.scatter(data_point[0][0], data_point[0][1], path_effects=[path_effects.SimpleLineShado

plt.title("Zona pomaranczowa") if sum(row.count(1.0) for row in nearest_tab) > sum(row.cou
```

```
plt.show()
```