Zadanie 1

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

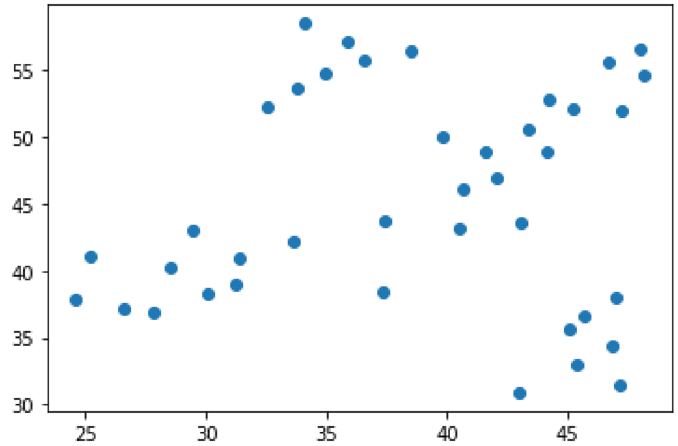
```
dataEx1 = pd.read_csv('k_means_data.csv')
dataEx1
```

	X	Y
0	25.23	41.09
1	45.10	35.69
2	26.59	37.21
3	27.80	36.93
4	28.56	40.21
5	29.49	43.05
6	30.04	38.33
7	31.25	39.03
8	31.35	40.98
9	42.95	30.91
10	37.30	38.42
11	37.39	43.69
12	32.53	52.18
13	33.60	42.20
14	24.60	37.88
15	33.76	53.60
16	34.97	54.72
17	35.84	57.04
18	47.16	31.52
19	36.58	55.76
20	46.82	34.33
21	46.98	38.03
22	45.34	33.02

```
import matplotlib.pyplot as plt
```

24 48.12 54.53

```
plt.scatter(dataEx1.iloc[:,0],dataEx1.iloc[:,1])
plt.show()
```




```
from sklearn.cluster import AgglomerativeClustering
```

```
k = 3
model = AgglomerativeClustering(linkage='single',n_clusters=k,affinity='euclidean', distance_threshold=None)
model.fit(dataEx1)
```

```
AgglomerativeClustering(linkage='single', n_clusters=3)
```

```
model.labels_
```

```
array([0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 2, 2, 2, 1, 2, 1, 1,
       1, 1, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
from sklearn.cluster import KMeans
```

```
k = 3
model = KMeans(n_clusters=3).fit(dataEx1)
```

```
model.labels_
```

```
array([2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 0, 2, 2, 0, 0, 0, 1, 0, 1, 1,
       1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0], dtype=int32)
```

```
K=model.cluster_centers_
K
```

```
array([[40.91210526, 52.77789474],
       [44.84      , 36.31444444],
       [30.26666667, 39.91833333]])
```

Zadanie 2

```
data = pd.read_csv('k_means_data.csv')
print(data)
data = np.array(data)
```

```
        X      Y
0   25.23  41.09
1   45.10  35.69
2   26.59  37.21
3   27.80  36.93
4   28.56  40.21
5   29.49  43.05
6   30.04  38.33
7   31.25  39.03
8   31.35  40.98
9   42.95  30.91
10  37.30  38.42
11  37.39  43.69
12  32.53  52.18
13  33.60  42.20
14  24.60  37.88
15  33.76  53.60
16  34.97  54.72
17  35.84  57.04
18  47.16  31.52
```

```
19  36.58  55.76
20  46.82  34.33
21  46.98  38.03
22  45.34  33.02
23  45.70  36.63
24  48.12  54.53
25  47.25  51.99
26  38.48  56.41
27  34.10  58.45
28  39.81  49.98
29  40.47  43.18
30  40.66  46.02
31  41.59  48.86
32  42.05  46.89
33  43.04  43.52
34  43.34  50.49
35  44.17  48.94
36  44.23  52.81
37  45.19  52.05
38  46.68  55.59
39  47.98  56.47
```

```
Label = np.zeros((len(data),1),dtype=np.float64)
Label
```

```
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.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.]])
```

```
NewData = np.concatenate([data,Label],axis=1)
NewData
```

```
array([[25.23, 41.09,  0.  ],
       [45.1 , 35.69,  0.  ],
       [26.59, 37.21,  0.  ],
       [27.8 , 36.93,  0.  ],
       [28.56, 40.21,  0.  ],
       [29.49, 43.05,  0.  ],
       [30.04, 38.33,  0.  ],
       [31.25, 39.03,  0.  ],
       [31.35, 40.98,  0.  ],
       [42.95, 30.91,  0.  ],
       [37.3 , 38.42,  0.  ],
       [37.39, 43.69,  0.  ],
       [32.53, 52.18,  0.  ],
       [33.6 , 42.2 ,  0.  ],
       [24.6 , 37.88,  0.  ],
       [33.76, 53.6 ,  0.  ],
       [34.97, 54.72,  0.  ],
```

```
       [35.84, 57.04,  0.  ],
       [47.16, 31.52,  0.  ],
       [36.58, 55.76,  0.  ],
       [46.82, 34.33,  0.  ],
       [46.98, 38.03,  0.  ],
       [45.34, 33.02,  0.  ],
       [45.7 , 36.63,  0.  ],
       [48.12, 54.53,  0.  ],
       [47.25, 51.99,  0.  ],
       [38.48, 56.41,  0.  ],
       [34.1 , 58.45,  0.  ],
       [39.81, 49.98,  0.  ],
       [40.47, 43.18,  0.  ],
       [40.66, 46.02,  0.  ],
       [41.59, 48.86,  0.  ],
       [42.05, 46.89,  0.  ],
       [43.04, 43.52,  0.  ],
       [43.34, 50.49,  0.  ],
       [44.17, 48.94,  0.  ],
       [44.23, 52.81,  0.  ],
       [45.19, 52.05,  0.  ],
       [46.68, 55.59,  0.  ],
       [47.98, 56.47,  0.  ]])
```

```
NewData[:,2]
```

```
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., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0.])
```

```
minX = int(np.min(data[:,0]))
maxX = int(np.max(data[:,0]))
minY = int(np.min(data[:,1]))
maxY = int(np.max(data[:,1]))
print(minX," ",maxX," ",minY," ",maxY)
```

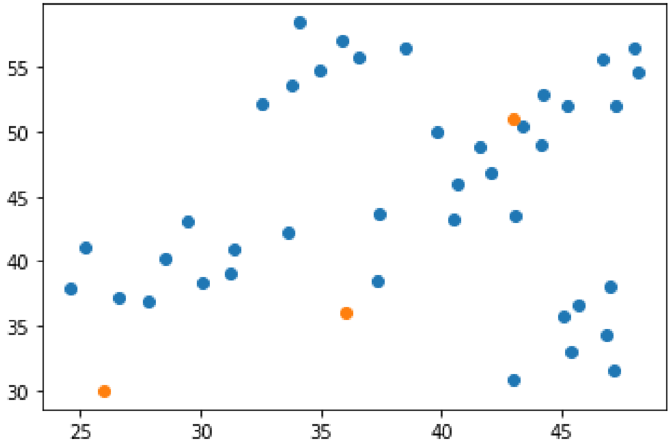
```
24   48   30   58
```

```
import random
CentersX = []
CentersY = []
```

```
for _ in range(0,3):
  CentersX.append(random.randrange(minX,maxX))
  CentersY.append(random.randrange(minY,maxY))

CentersX = np.array(CentersX).reshape(3,1)
CentersY = np.array(CentersY).reshape(3,1)

Centers = np.concatenate([CentersX,CentersY],axis=1).astype(np.float64)

Centers
```

```
array([[36., 36.],
       [43., 51.],
       [26., 30.]])
```

```
import matplotlib.pyplot as plt
plt.scatter(data[:,0],data[:,1])
plt.scatter(Centers[:,0],Centers[:,1])
plt.show()
```




```
import math
```

```
for point in NewData:
  print(point[0],point[1])
```

```
i = -1
j= -1
print("    ")
for cent in Centers:
  print("Cent",cent[0],cent[1])
  i= i+1
  euclidesian=math.sqrt(((cent[1]-point[1])*(cent[1]-point[1]))+((cent[0]-point[0])*(cent[0]-point[0])))
  if(i==0):
    min=euclidesian
    j=i
  if(min>euclidesian):
    min=euclidesian
    j=i

  print(euclidesian)
  print("Max"," i ", min,i,j)
  point[2]=j
```

```
25.23 41.09

Cent 36.0 36.0
11.912220615821385
Max  i  11.912220615821385 0 0
Cent 43.0 51.0
20.34652304449092
Max  i  11.912220615821385 1 0
Cent 26.0 30.0
11.11669915037733
Max  i  11.11669915037733 2 2
45.1 35.69

Cent 36.0 36.0
9.105278688760714
Max  i  9.105278688760714 0 0
Cent 43.0 51.0
15.453352387103585
Max  i  9.105278688760714 1 0
Cent 26.0 30.0
19.929528343641252
Max  i  9.105278688760714 2 0
26.59 37.21

Cent 36.0 36.0
```

```
9.487475955173748
Max  i  9.487475955173748 0 0
Cent 43.0 51.0
21.434836131867208
Max  i  9.487475955173748 1 0
Cent 26.0 30.0
7.234099805780953
Max  i  7.234099805780953 2 2
27.8 36.93

Cent 36.0 36.0
8.252569296891727
Max  i  8.252569296891727 0 0
Cent 43.0 51.0
20.712433463984862
Max  i  8.252569296891727 1 0
Cent 26.0 30.0
7.159951117151569
Max  i  7.159951117151569 2 2
28.56 40.21

Cent 36.0 36.0
8.548549584578662
Max  i  8.548549584578662 0 0
Cent 43.0 51.0
18.02602840339491
Max  i  8.548549584578662 1 0
Cent 26.0 30.0
10.526048641346856
Max  i  8.548549584578662 2 0
29.49 43.05
```

```
NewData
```

```
array([[25.23, 41.09,  2.  ],
       [45.1 , 35.69,  0.  ],
       [26.59, 37.21,  2.  ],
       [27.8 , 36.93,  2.  ],
       [28.56, 40.21,  0.  ],
       [29.49, 43.05,  0.  ],
       [30.04, 38.33,  0.  ],
       [31.25, 39.03,  0.  ],
```

```
       [31.35, 40.98,  0.  ],
       [42.95, 30.91,  0.  ],
       [37.3 , 38.42,  0.  ],
       [37.39, 43.69,  0.  ],
       [32.53, 52.18,  1.  ],
       [33.6 , 42.2 ,  0.  ],
       [24.6 , 37.88,  2.  ],
       [33.76, 53.6 ,  1.  ],
       [34.97, 54.72,  1.  ],
       [35.84, 57.04,  1.  ],
       [47.16, 31.52,  0.  ],
       [36.58, 55.76,  1.  ],
       [46.82, 34.33,  0.  ],
       [46.98, 38.03,  0.  ],
       [45.34, 33.02,  0.  ],
       [45.7 , 36.63,  0.  ],
       [48.12, 54.53,  1.  ],
       [47.25, 51.99,  1.  ],
       [38.48, 56.41,  1.  ],
       [34.1 , 58.45,  1.  ],
       [39.81, 49.98,  1.  ],
       [40.47, 43.18,  1.  ],
       [40.66, 46.02,  1.  ],
       [41.59, 48.86,  1.  ],
       [42.05, 46.89,  1.  ],
       [43.04, 43.52,  1.  ],
       [43.34, 50.49,  1.  ],
       [44.17, 48.94,  1.  ],
       [44.23, 52.81,  1.  ],
       [45.19, 52.05,  1.  ],
       [46.68, 55.59,  1.  ],
       [47.98, 56.47,  1.  ]])
```

```
NewData[:,2]
```

```
array([2., 0., 2., 2., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 2., 1., 1.,
       1., 0., 1., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1.])
```

```
OldCenters = Centers
while True:
  x0=0
```

```
y0=0
i0=0
x1=0
y1=0
i1=0
x2=0
y2=0
i2=0

Cluster1X=[]
Cluster1Y=[]
Cluster2X=[]
Cluster2Y=[]
Cluster3X=[]
Cluster3Y=[]
for point in NewData:
  print(point[0],point[1],point[2])
  if(point[2]==0):
    x0=x0+point[0]
    Cluster1X.append(point[0])
    y0=y0+point[1]
    Cluster1Y.append(point[1])
    i0=i0+1
  if(point[2]==1):
    x1=x1+point[0]
    Cluster2X.append(point[0])
    y1=y1+point[1]
    Cluster2Y.append(point[1])
    i1=i1+1
  if(point[2]==2):
    x2=x2+point[0]
    Cluster3X.append(point[0])
    y2=y2+point[1]
    Cluster3Y.append(point[1])
    i2=i2+1

NewCentersX=[]
NewCentersY=[]

#NewCentersX.append(x0/i0)
```

```
    #NewCentersY.append(y0/i0)
    #NewCentersX.append(x1/i1)
    #NewCentersY.append(y1/i1)
    #NewCentersX.append(x2/i2)
    #NewCentersY.append(y2/i2)
    NewCentersX.append(np.mean(Cluster1X))
    NewCentersX.append(np.mean(Cluster2X))
    NewCentersX.append(np.mean(Cluster3X))
    NewCentersY.append(np.mean(Cluster1Y))
    NewCentersY.append(np.mean(Cluster2Y))
    NewCentersY.append(np.mean(Cluster3Y))
    NewCentersX = np.array(NewCentersX).reshape(3,1)
    NewCentersY = np.array(NewCentersY).reshape(3,1)

    print(np.mean(Cluster1X))
    print(np.mean(Cluster1Y))
    print(np.mean(Cluster2X))
    print(np.mean(Cluster2Y))
    print(np.mean(Cluster3X))
    print(np.mean(Cluster3Y))

    NewCenters = np.concatenate([NewCentersX,NewCentersY],axis=1).astype(np.float64)

    print(x0,y0,i0)
    print(x1,y1,i1)
    print(x2,y2,i2)
    plt.scatter(NewData[:,0],NewData[:,1])
    plt.scatter(NewCenters[:,0],NewCenters[:,1])
    plt.show()

    print(OldCenters)
    print(NewCenters)
    print("Space")

    if np.array_equal(OldCenters,NewCenters)==True:
      break
    OldCenters = NewCenters
    #NewCenters
```

```
for point in NewData:
  #print(point[0],point[1])
  i = -1
  j= -1
  #print("    ")
  for cent in NewCenters:
    print("Cent",cent[0],cent[1])
    i= i+1
    #euclidesian=math.sqrt(((cent[1]-point[1])*(cent[1]-point[1]))+((cent[0]-point[0])*(cent[0]-point[0])))
    euclidesian=math.sqrt(((cent[0]-point[0])*(cent[0]-point[0]))+((cent[1]-point[1])*(cent[1]-point[1])))
    if(i==0):
      min=euclidesian
      j=i
    if(min>euclidesian):
      min=euclidesian
      j=i

    #print(euclidesian)
    #print("Max"," i ", min,i,j)
    point[2]=j
```

```
25.23 41.09 2.0
45.1 35.69 0.0
26.59 37.21 2.0
27.8 36.93 2.0
28.56 40.21 0.0
29.49 43.05 0.0
30.04 38.33 0.0
31.25 39.03 0.0
31.35 40.98 0.0
42.95 30.91 0.0
37.3 38.42 0.0
37.39 43.69 0.0
32.53 52.18 1.0
33.6 42.2 0.0
24.6 37.88 2.0
33.76 53.6 1.0
34.97 54.72 1.0
35.84 57.04 1.0
47.16 31.52 0.0
36.58 55.76 1.0
46.82 34.33 0.0
46.98 38.03 0.0
45.34 33.02 0.0
45.7 36.63 0.0
48.12 54.53 1.0
47.25 51.99 1.0
38.48 56.41 1.0
34.1 58.45 1.0
39.81 49.98 1.0
40.47 43.18 1.0
40.66 46.02 1.0
41.59 48.86 1.0
42.05 46.89 1.0
43.04 43.52 1.0
43.34 50.49 1.0
44.17 48.94 1.0
44.23 52.81 1.0
45.19 52.05 1.0
46.68 55.59 1.0
47.98 56.47 1.0
38.602
37.736
40.99238095238094
51.87999999999999
```

```
26.055
```

```
NewData
```

```
array([[25.23, 41.09,  2.  ],
       [45.1 , 35.69,  0.  ],
       [26.59, 37.21,  2.  ],
       [27.8 , 36.93,  2.  ],
       [28.56, 40.21,  2.  ],
       [29.49, 43.05,  2.  ],
       [30.04, 38.33,  2.  ],
       [31.25, 39.03,  2.  ],
       [31.35, 40.98,  2.  ],
       [42.95, 30.91,  0.  ],
       [37.3 , 38.42,  0.  ],
       [37.39, 43.69,  0.  ],
       [32.53, 52.18,  1.  ],
       [33.6 , 42.2 ,  2.  ],
       [24.6 , 37.88,  2.  ],
       [33.76, 53.6 ,  1.  ],
       [34.97, 54.72,  1.  ],
       [35.84, 57.04,  1.  ],
       [47.16, 31.52,  0.  ],
       [36.58, 55.76,  1.  ],
       [46.82, 34.33,  0.  ],
       [46.98, 38.03,  0.  ],
       [45.34, 33.02,  0.  ],
       [45.7 , 36.63,  0.  ],
       [48.12, 54.53,  1.  ],
       [47.25, 51.99,  1.  ],
       [38.48, 56.41,  1.  ],
       [34.1 , 58.45,  1.  ],
       [39.81, 49.98,  1.  ],
       [40.47, 43.18,  0.  ],
       [40.66, 46.02,  1.  ],
       [41.59, 48.86,  1.  ],
       [42.05, 46.89,  1.  ],
       [43.04, 43.52,  0.  ],
       [43.34, 50.49,  1.  ],
       [44.17, 48.94,  1.  ],
       [44.23, 52.81,  1.  ],
       [45.19, 52.05,  1.  ],
       [46.68, 55.59,  1.  ],
       [47.98, 56.47,  1.  ]])
```

Cent 38.602 37.736

```
test=NewData[:,2]
```

Cent 38.602 37.736

```
X=np.asarray(TEST[0])
#print(X)
a=np.asarray(X)
print(a)
```

```
[1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 2. 1. 1. 2. 2. 2. 0. 2. 0. 0. 0. 0.
 2. 2. 2. 2. 2. 0. 2. 2. 2. 0. 2. 2. 2. 2. 2. 2.]
```

Cent 38.602 37.736

```
df = pd.DataFrame(NewData, columns = ['X','Y','L'])
```

Cent 38.602 37.736

```
df.head()
```

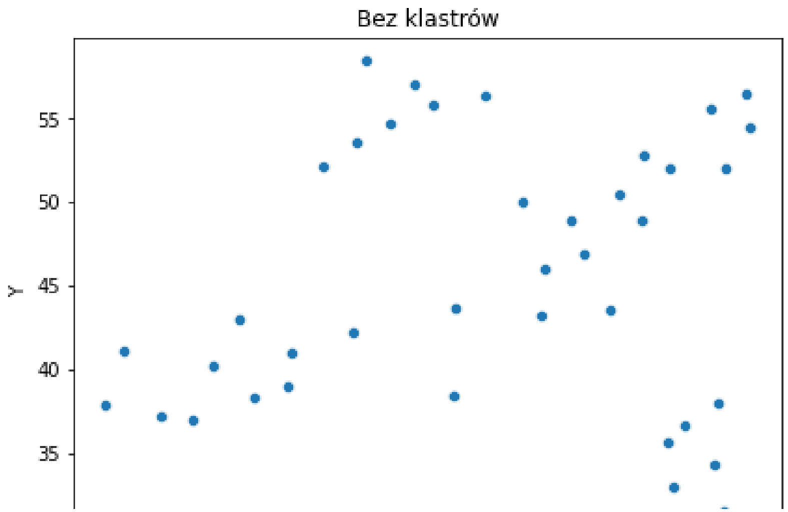
	X	Y	L
0	25.23	41.09	2.0
1	45.10	35.69	0.0
2	26.59	37.21	2.0
3	27.80	36.93	2.0
4	28.56	40.21	2.0

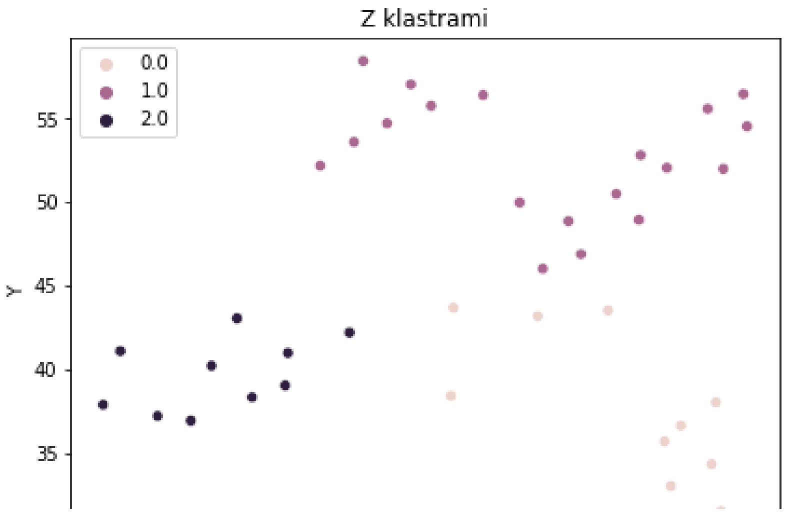
Cent 38.602 37.736

```
dataset = df[["X", "Y"]]
```

Cent 38.602 37.736

```
import seaborn as sns
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(15,5))
sns.scatterplot(ax=axes[0], data=df, x='X', y='Y').set_title('Bez klastrów')
sns.scatterplot(ax=axes[1], data=df, x='X', y='Y', hue=test).set_title('Z klastrami');
```






```
NewData
```

```
array([[25.23, 41.09,  2.  ],
       [45.1 , 35.69,  0.  ],
       [26.59, 37.21,  2.  ],
       [27.8 , 36.93,  2.  ],
       [28.56, 40.21,  2.  ],
       [29.49, 43.05,  2.  ],
       [30.04, 38.33,  2.  ],
       [31.25, 39.03,  2.  ],
       [31.35, 40.98,  2.  ],
       [42.95, 30.91,  0.  ],
       [37.3 , 38.42,  0.  ],
       [37.39, 43.69,  0.  ],
       [32.53, 52.18,  1.  ],
       [33.6 , 42.2 ,  2.  ],
       [24.6 , 37.88,  2.  ],
       [33.76, 53.6 ,  1.  ],
       [34.97, 54.72,  1.  ],
       [35.84, 57.04,  1.  ],
       [47.16, 31.52,  0.  ],
       [36.58, 55.76,  1.  ],
       [46.82, 34.33,  0.  ],
       [46.98, 38.03,  0.  ],
       [45.34, 33.02,  0.  ],
       [45.7 , 36.63,  0.  ],
       [48.12, 54.53,  1.  ],
```

```
       [47.25, 51.99,  1.  ],
       [38.48, 56.41,  1.  ],
       [34.1 , 58.45,  1.  ],
       [39.81, 49.98,  1.  ],
       [40.47, 43.18,  0.  ],
       [40.66, 46.02,  1.  ],
       [41.59, 48.86,  1.  ],
       [42.05, 46.89,  1.  ],
       [43.04, 43.52,  0.  ],
       [43.34, 50.49,  1.  ],
       [44.17, 48.94,  1.  ],
       [44.23, 52.81,  1.  ],
       [45.19, 52.05,  1.  ],
       [46.68, 55.59,  1.  ],
       [47.98, 56.47,  1.  ]])
```

```
27.8 36.93 2.0
```

Aby edytować zawartość komórki, kliknij ją dwukrotnie (lub naciśnij klawisz Enter)

```
30.04 38.33 2.0
```

Zadanie 3

```
42.95 30.91 0.0
```

```
def lcs(X, Y, m, n):

    if m == 0 or n == 0:
       return 0;
    elif X[m-1] == Y[n-1]:
       return 1 + lcs(X, Y, m-1, n-1);
    else:
       return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));


X = "abcde"
Y = "bcdxy"
print ("Length of LCS is ", lcs(X, Y, len(X), len(Y)))
```

```
Length of LCS is  3
34.1 58.45 1.0
```

Zadanie 4

```
40.66 46.02 1.0
```

```
dataEx4 = pd.read_csv('iris.csv')
dataEx4
```

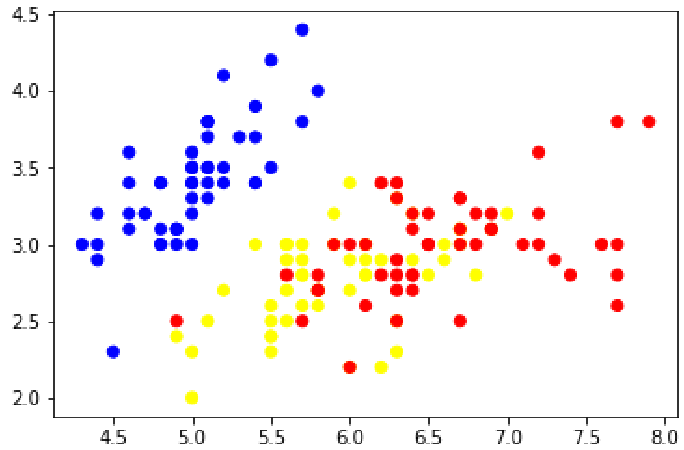
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
dataEx4.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000

```
colors = {'virginica':'red', 'setosa':'blue','versicolor':'yellow'}
plt.scatter(dataEx4.iloc[:,0],dataEx4.iloc[:,1],c=dataEx4.iloc[:,4].map(colors))

plt.show()
```




```
Cent 40.01210526315 79 52.777894736842 11
```

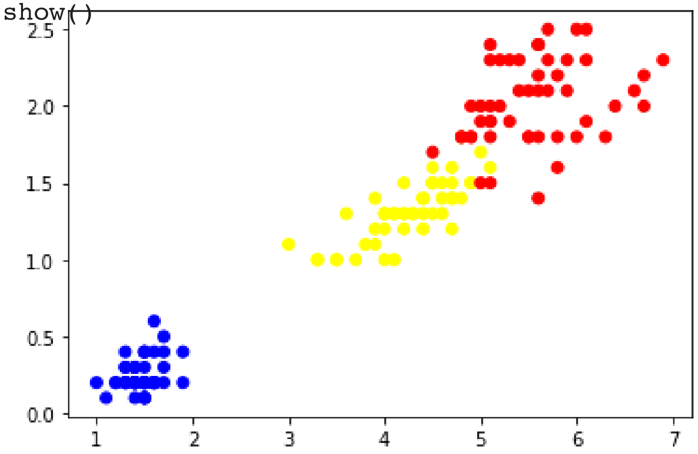
```
cor = dataEx4.iloc[:,:].corr()
cor.style.background_gradient(cmap='coolwarm')
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000

```
Cent 42.654166666666 37.594999999999
```

```
colors = {'virginica':'red', 'setosa':'blue','versicolor':'yellow'}
plt.scatter(dataEx4.iloc[:,2],dataEx4.iloc[:,3],c=dataEx4.iloc[:,4].map(colors))
```

```
plt.show()
```



Cent 42.6541666666666 37.594999999999

5.TAK

```
test=dataEx4.copy()
```

Cent 42.6541666666666 37.594999999999

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Cent 40.912105263157 52.777894736842

```
trainingData=pd.DataFrame(data=dataEx4,columns=['sepal_length','sepal_width',],copy=True)
```

Cent 40.912105263157 52.777894736842

```
k = 3
model2 = AgglomerativeClustering(linkage='average',n_clusters=k,affinity='euclidean', distance_threshold=None)
model2.fit(trainingData)
```

```
AgglomerativeClustering(linkage='average', n_clusters=3)
```

```
model2.labels_
```

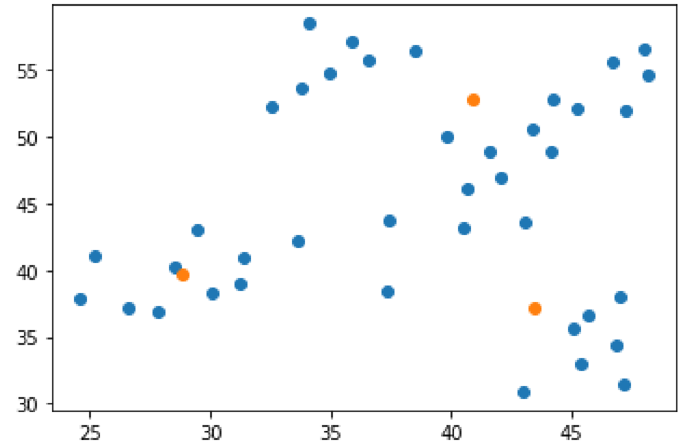
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 1, 2, 2, 1, 0, 1, 2, 1,
       2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 1, 1, 1,
       2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
k = 3
model2 = KMeans(n_clusters=3).fit(trainingData)
model2.labels_
```

```
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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1,
       2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1,
       1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1,
       1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2], dtype=int32)
```

```
29.49 43.05 2.0
30.04 38.33 2.0
31.25 39.03 2.0
31.35 40.98 2.0
42.95 30.91 0.0
37.3 38.42 0.0
37.39 43.69 0.0
32.53 52.18 1.0
33.6 42.2 2.0
24.6 37.88 2.0
33.76 53.6 1.0
34.97 54.72 1.0
35.84 57.04 1.0
47.16 31.52 0.0
```

```
36.58 55.76 1.0
46.82 34.33 0.0
46.98 38.03 0.0
45.34 33.02 0.0
45.7 36.63 0.0
48.12 54.53 1.0
47.25 51.99 1.0
38.48 56.41 1.0
34.1 58.45 1.0
39.81 49.98 1.0
40.47 43.18 0.0
40.66 46.02 1.0
41.59 48.86 1.0
42.05 46.89 1.0
43.04 43.52 0.0
43.34 50.49 1.0
44.17 48.94 1.0
44.23 52.81 1.0
45.19 52.05 1.0
46.68 55.59 1.0
47.98 56.47 1.0
43.47727272727272
37.17636363636364
40.9121052631579
52.77789473684211
28.851000000000006
39.690999999999995
478.25000000000006 408.94 11
777.33 1002.7799999999999 19
288.51000000000005 396.91 10
```



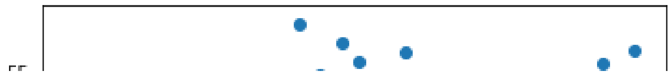

```
[[42.65416667 37.595 ]
```

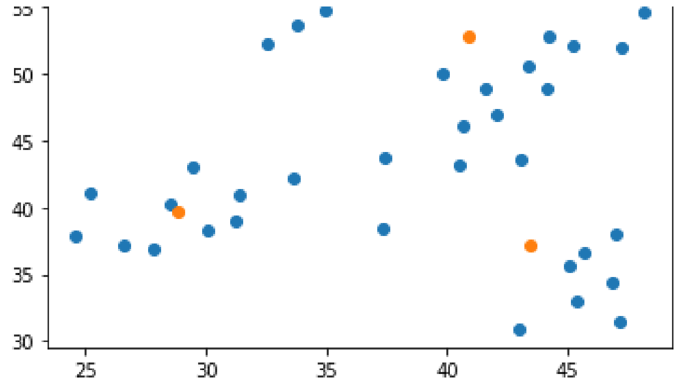
```
[[42.65416667 37.595     ]
 [40.91210526 52.77789474]
 [28.32333333 39.41222222]]
[[43.47727273 37.17636364]
 [40.91210526 52.77789474]
 [28.851      39.691     ]]
Space
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
```

```
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
```

```
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
Cent 43.47727272727272 37.17636363636364
Cent 40.9121052631579 52.77789473684211
Cent 28.851000000000006 39.690999999999995
25.23 41.09 2.0
45.1 35.69 0.0
26.59 37.21 2.0
27.8 36.93 2.0
28.56 40.21 2.0
29.49 43.05 2.0
30.04 38.33 2.0
```

```
31.25 39.03 2.0
31.35 40.98 2.0
42.95 30.91 0.0
37.3 38.42 0.0
37.39 43.69 0.0
32.53 52.18 1.0
33.6 42.2 2.0
24.6 37.88 2.0
33.76 53.6 1.0
34.97 54.72 1.0
35.84 57.04 1.0
47.16 31.52 0.0
36.58 55.76 1.0
46.82 34.33 0.0
46.98 38.03 0.0
45.34 33.02 0.0
45.7 36.63 0.0
48.12 54.53 1.0
47.25 51.99 1.0
38.48 56.41 1.0
34.1 58.45 1.0
39.81 49.98 1.0
40.47 43.18 0.0
40.66 46.02 1.0
41.59 48.86 1.0
42.05 46.89 1.0
43.04 43.52 0.0
43.34 50.49 1.0
44.17 48.94 1.0
44.23 52.81 1.0
45.19 52.05 1.0
46.68 55.59 1.0
47.98 56.47 1.0
43.47727272727272
37.17636363636364
40.9121052631579
52.77789473684211
28.851000000000006
39.690999999999995
478.25000000000006 408.94 11
777.33 1002.7799999999999 19
288.51000000000005 396.91 10
```





```
[[43.47727273 37.17636364]
 [40.91210526 52.77789474]
 [28.851      39.691     ]]
[[43.47727273 37.17636364]
 [40.91210526 52.77789474]
 [28.851      39.691     ]]
Space
```