Машинно обучение с метода на к най-близки съседи (Machine Learning with k Nearest Neighbours)

Д-р инж. Огнян Кабранов
Дипл. инж. Даниела Цветкова

2019 година

Днес ще се занимаем с малко теория

- Теорията е страшна само, ако не е свързана с проблема, който решаваме :-)
- Теорията е **методът на к-най-близки съседи за класификация.**

Нека започнем с два вестника

- Имаме две хипотетични списания: “**Дунавски Компютър**” (ДК) и “**Северозападен фермер**” (СФ).
  - “Дунавски Компютър” (ДК) пише на компютърни теми и статиите са специализирани за ИТ специалисти от Видин и региона.
  - “Северозападен фермер” пише на селскостопански и земеделски теми за фермерската аудитория от Северозапада.

Проблем - към кой вестник принадлежи неизвестен текст

- Попаднали сме на текст от списание, но не знам дали текстът е от “Дунавски компютър” или “Северозападен фермер”.
- Искаме да създадем метод, който автоматизирано да установи, откъде е текстът.

Откъде да започнем?

- Забелязваме, че в текста, който искаме да анализираме, думите “**компютър**” и “**хакер**” се срещат съответно 18 и 13 пъти.
- Очевидно, в специализирани компютърни списания ще се пише много по-често за компютри и хакери.
- Преди това нека видим в миналото колко пъти тези думи се срещат в различните статии на “Дунавски Компютър” и “Северозападен Фермер”.

списание ДК	"компютър"	"хакер"
статия 1	12	10
статия 2	21	9
статия 3	15	18
статия 4	9	23
статия 5	19	16

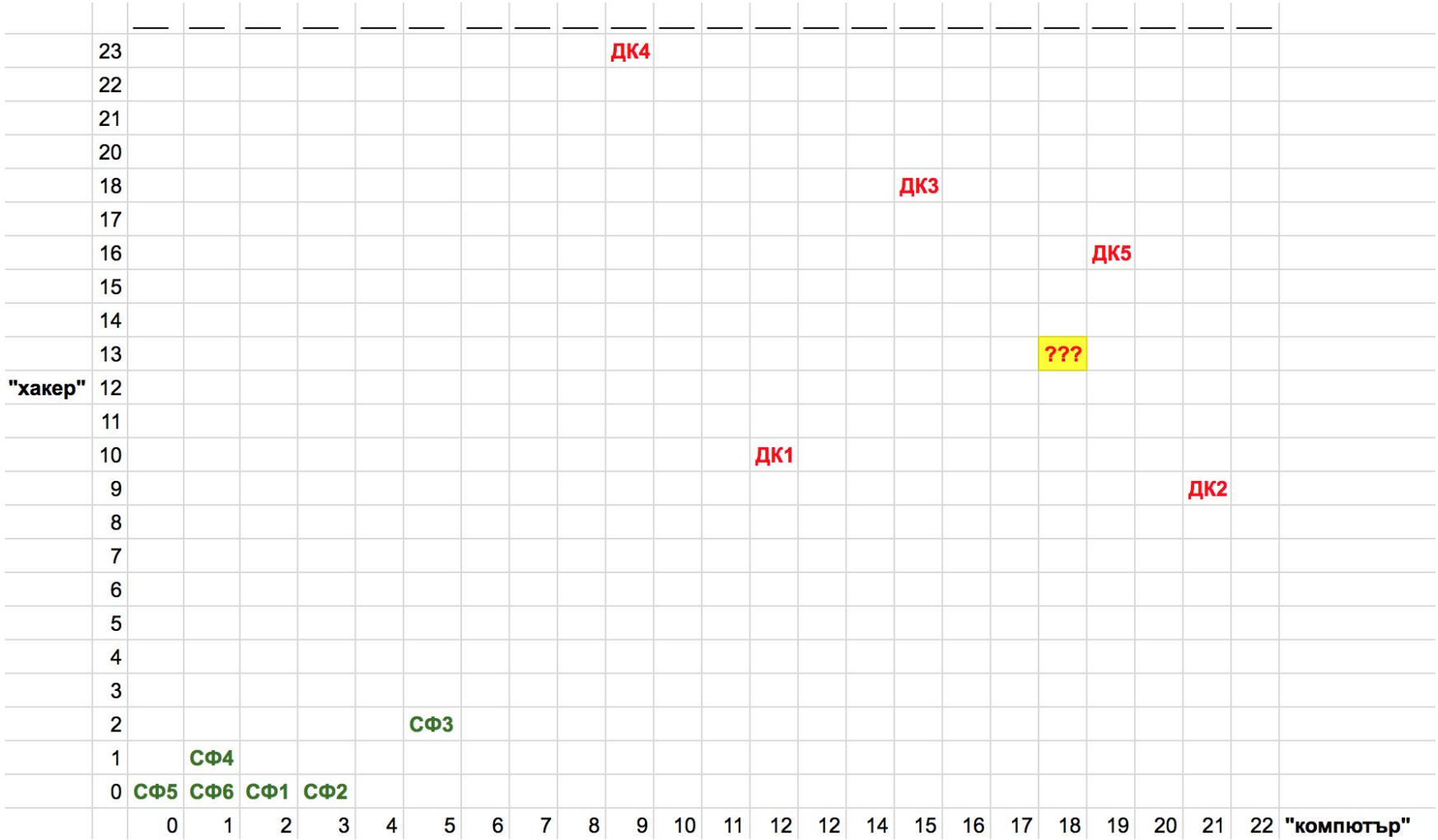
списание СФ	"компютър"	"хакер"
статия 1	2	0
статия 2	3	0
статия 3	5	2
статия 4	1	0
статия 5	0	0
статия 6	1	0

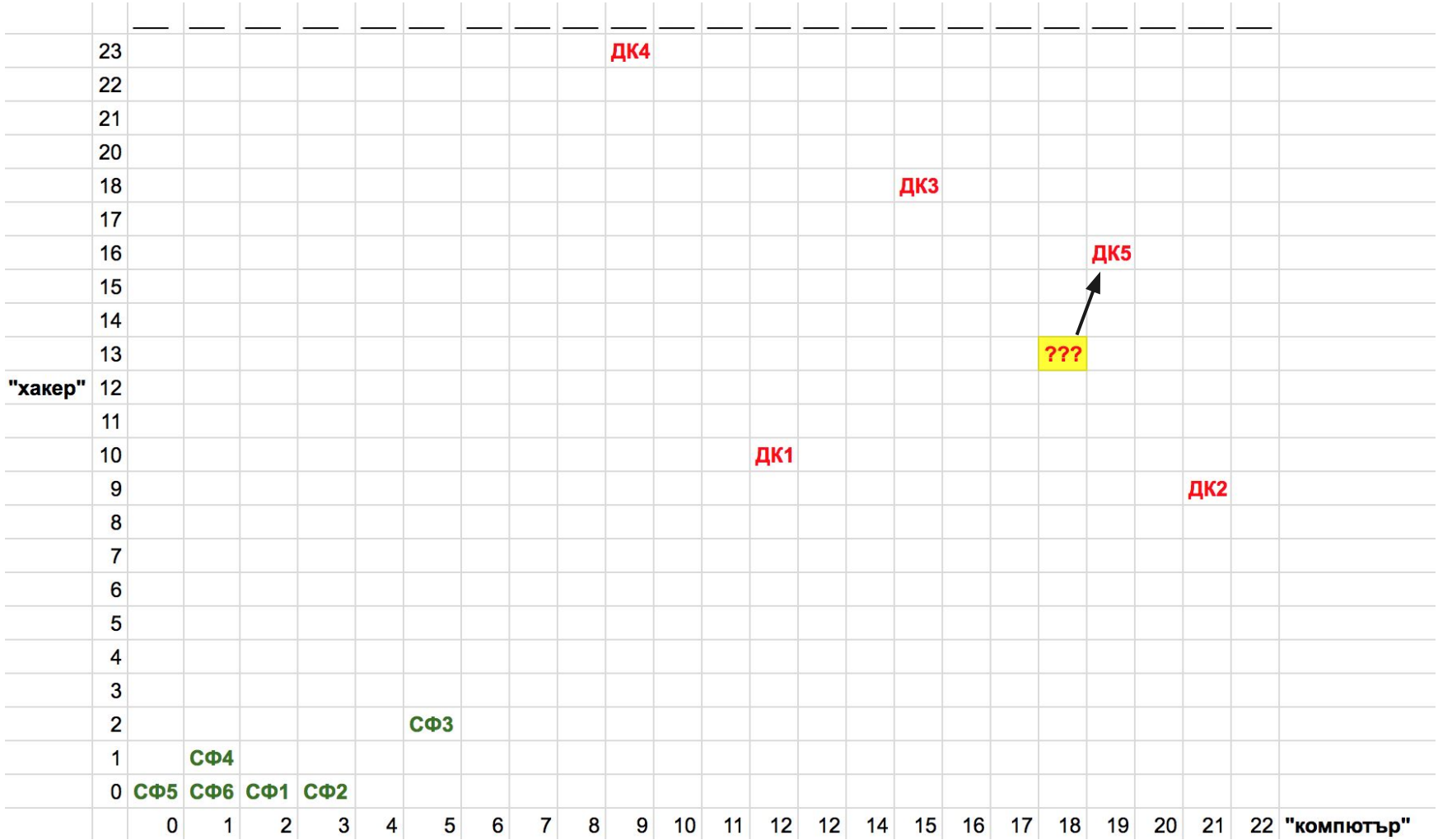
От кое списание е статията ???

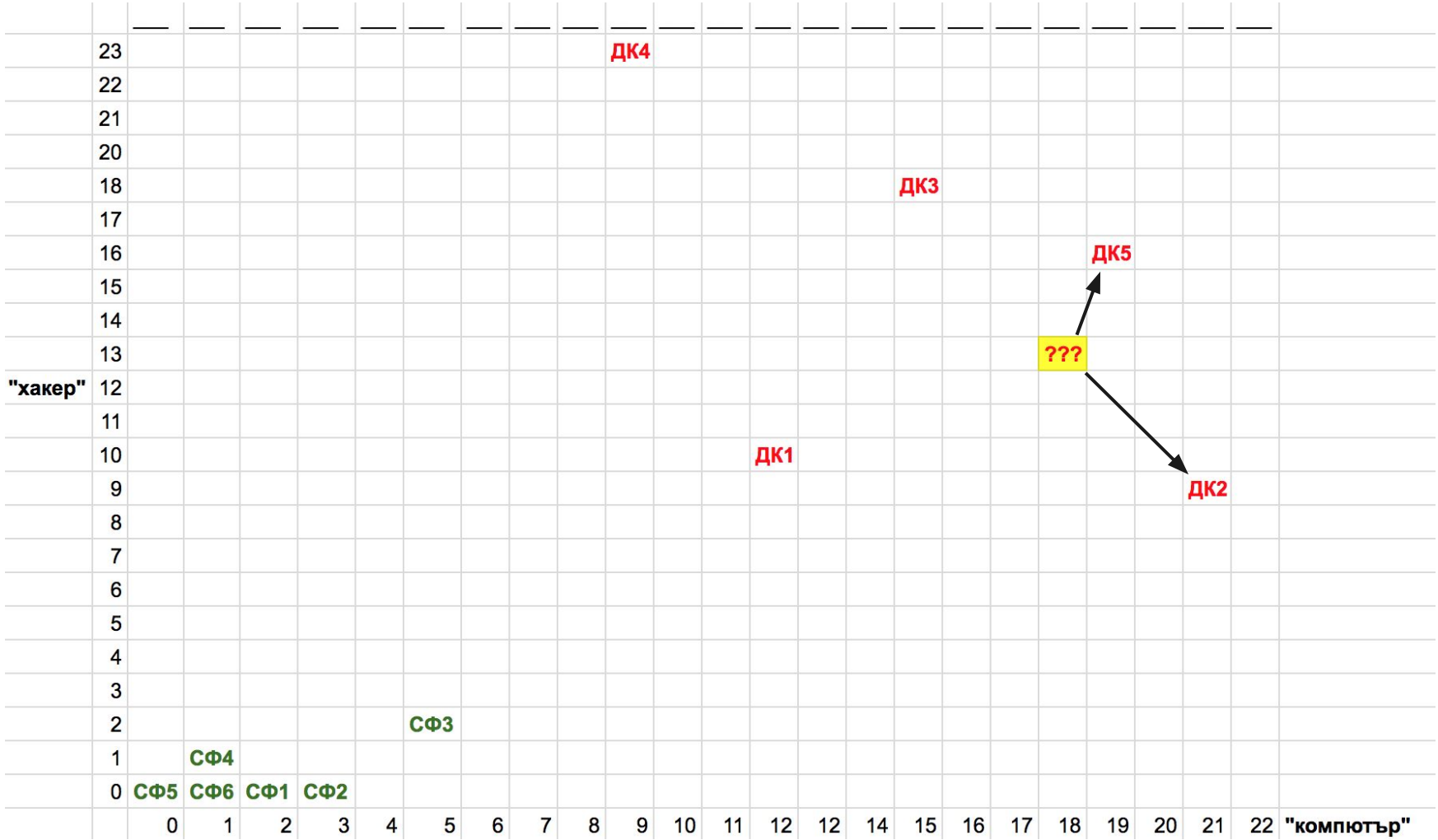
? списание	"компютър"	"хакер"
статия 1	18	13

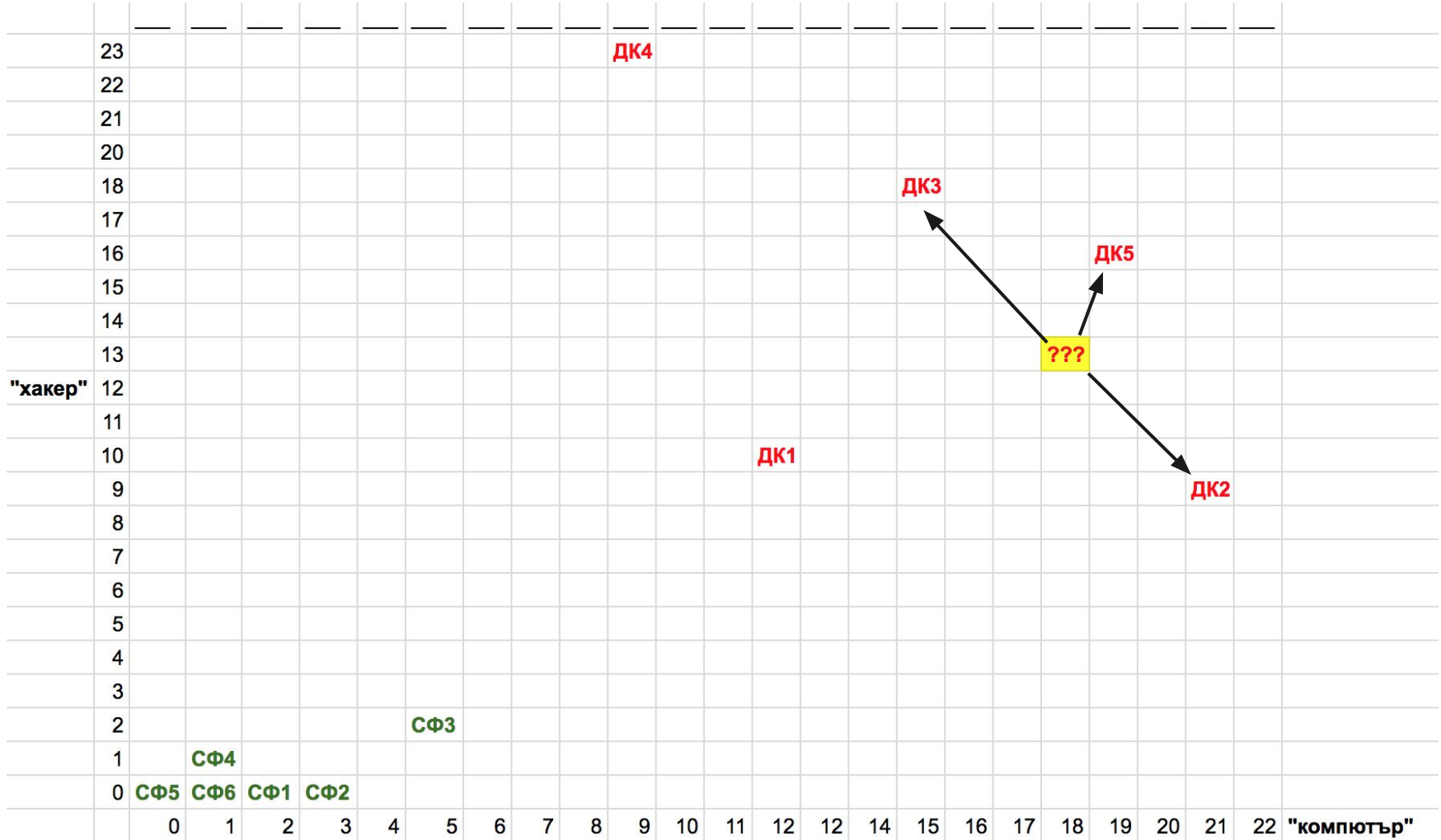
Изглежда е от “Дунавски компютър”, защото повече се говори за “компютри” и “хакери”.

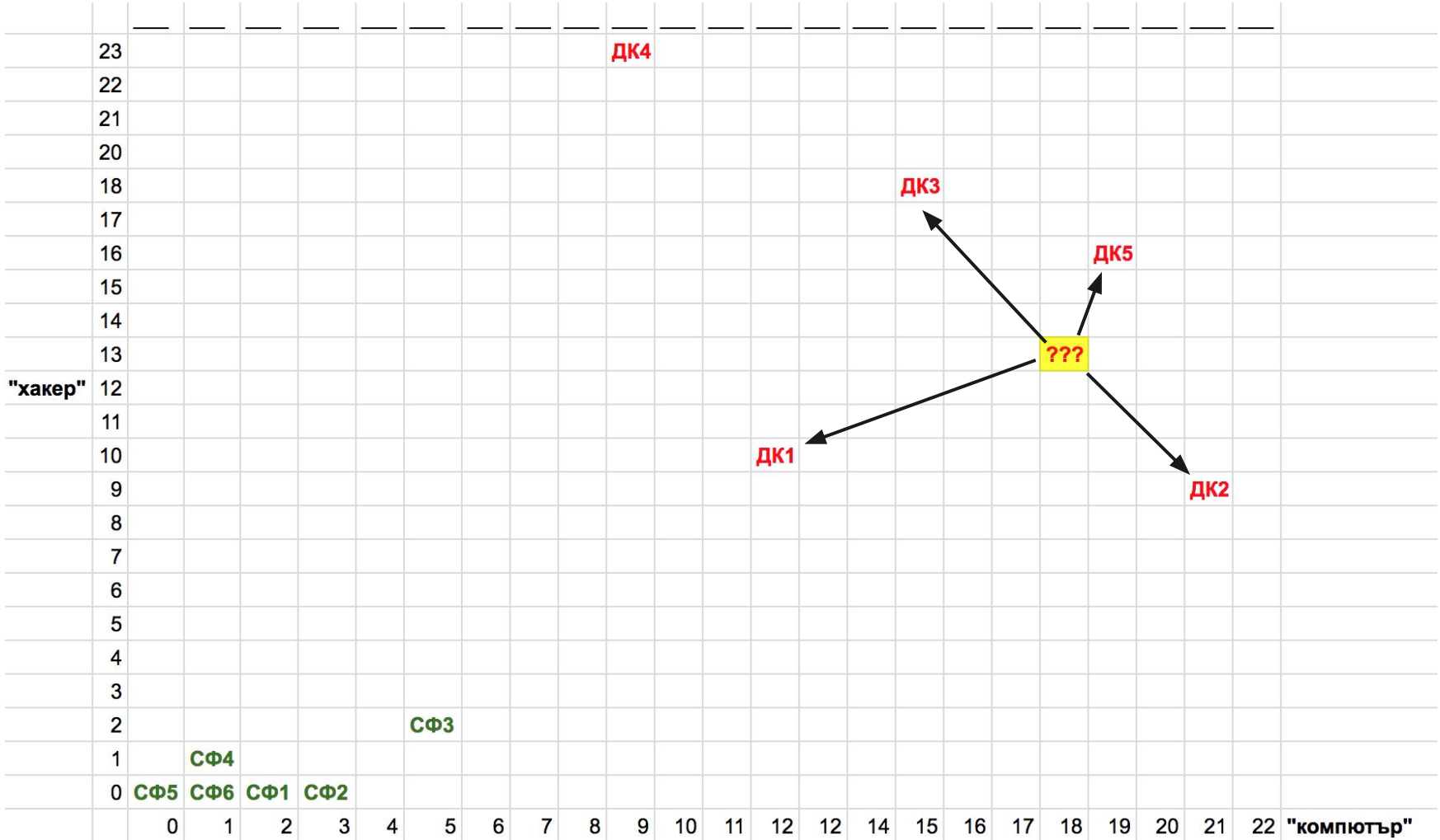
Сега ще го представим като графика.

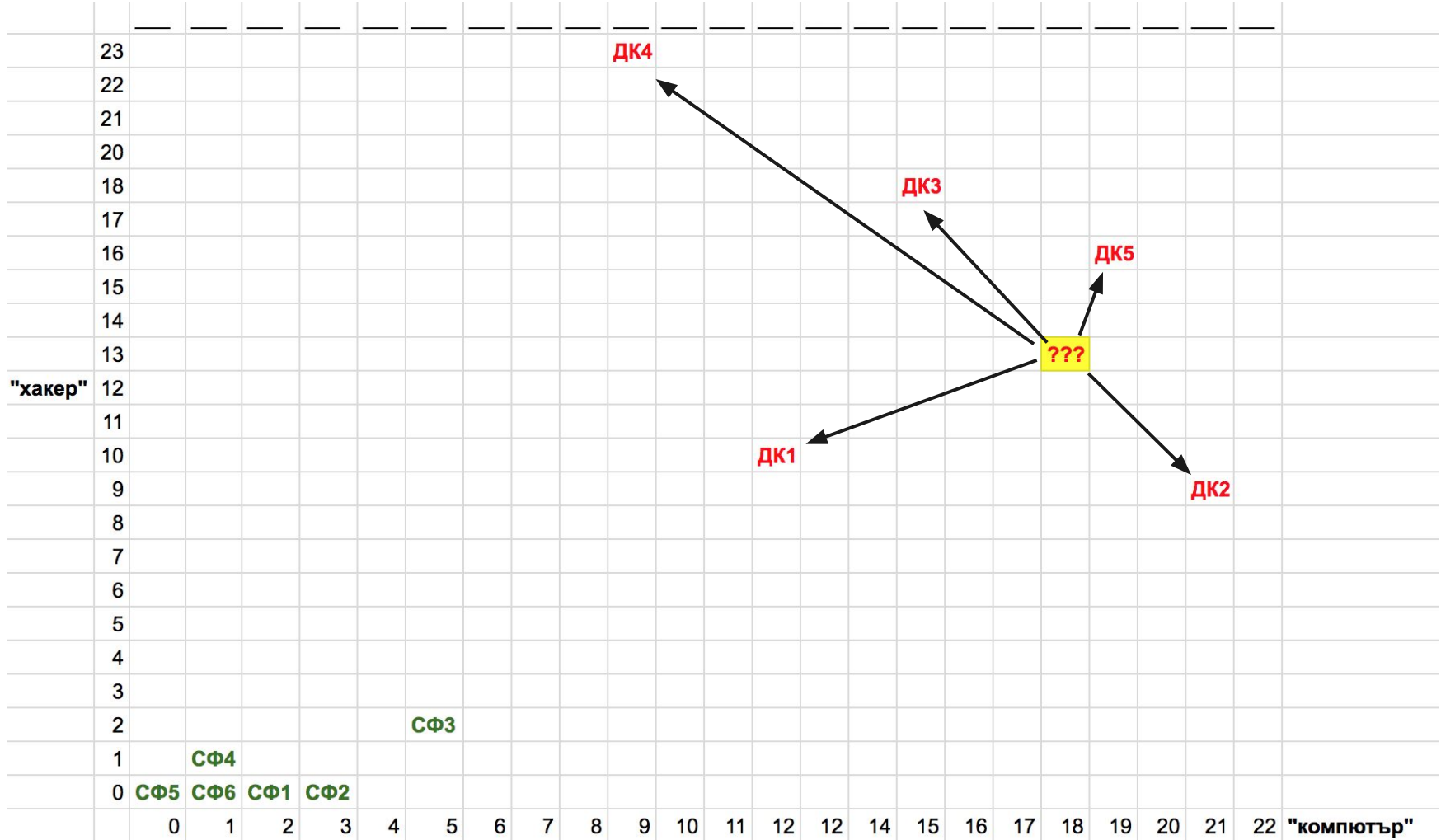

ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"

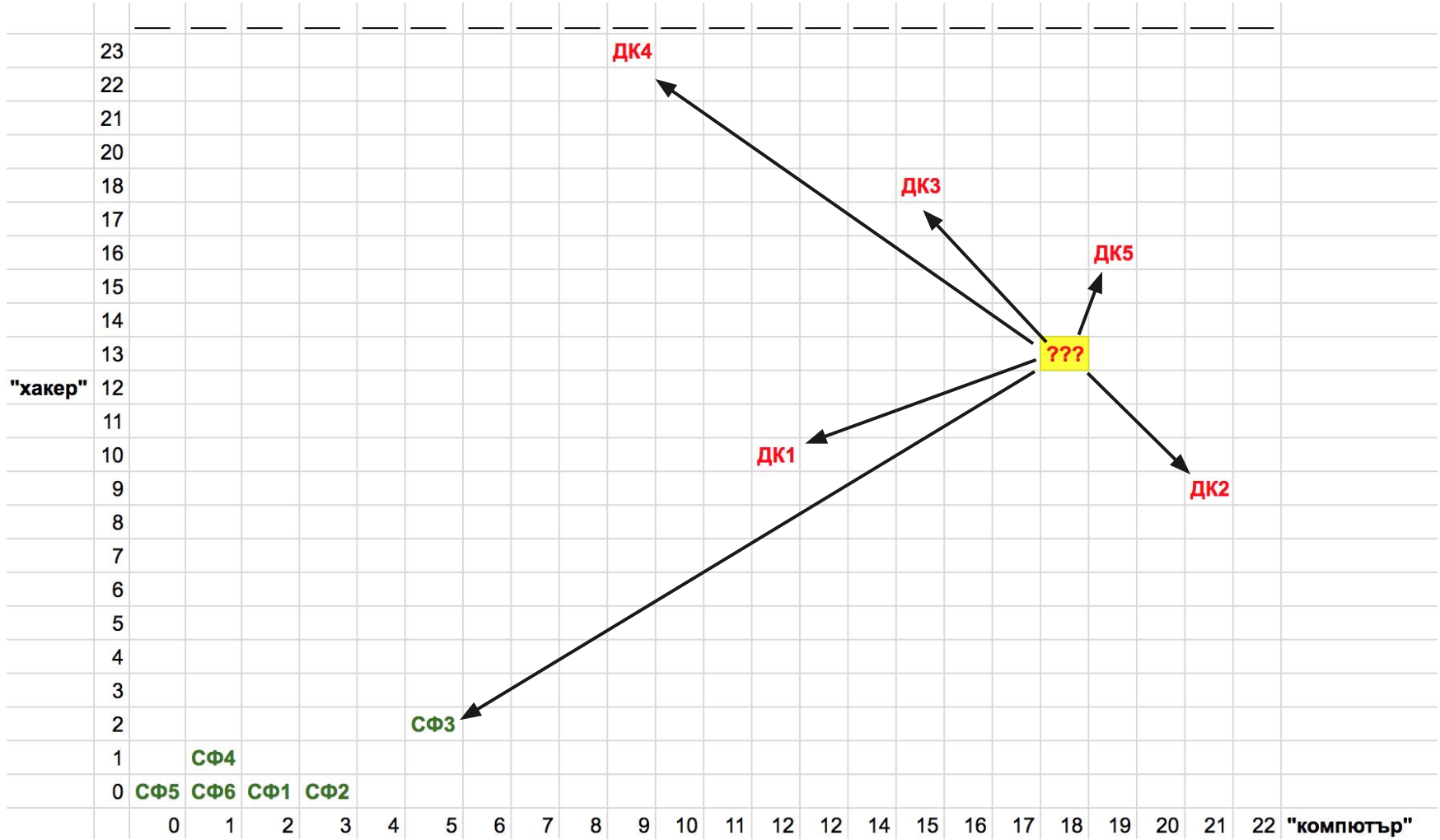

ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"

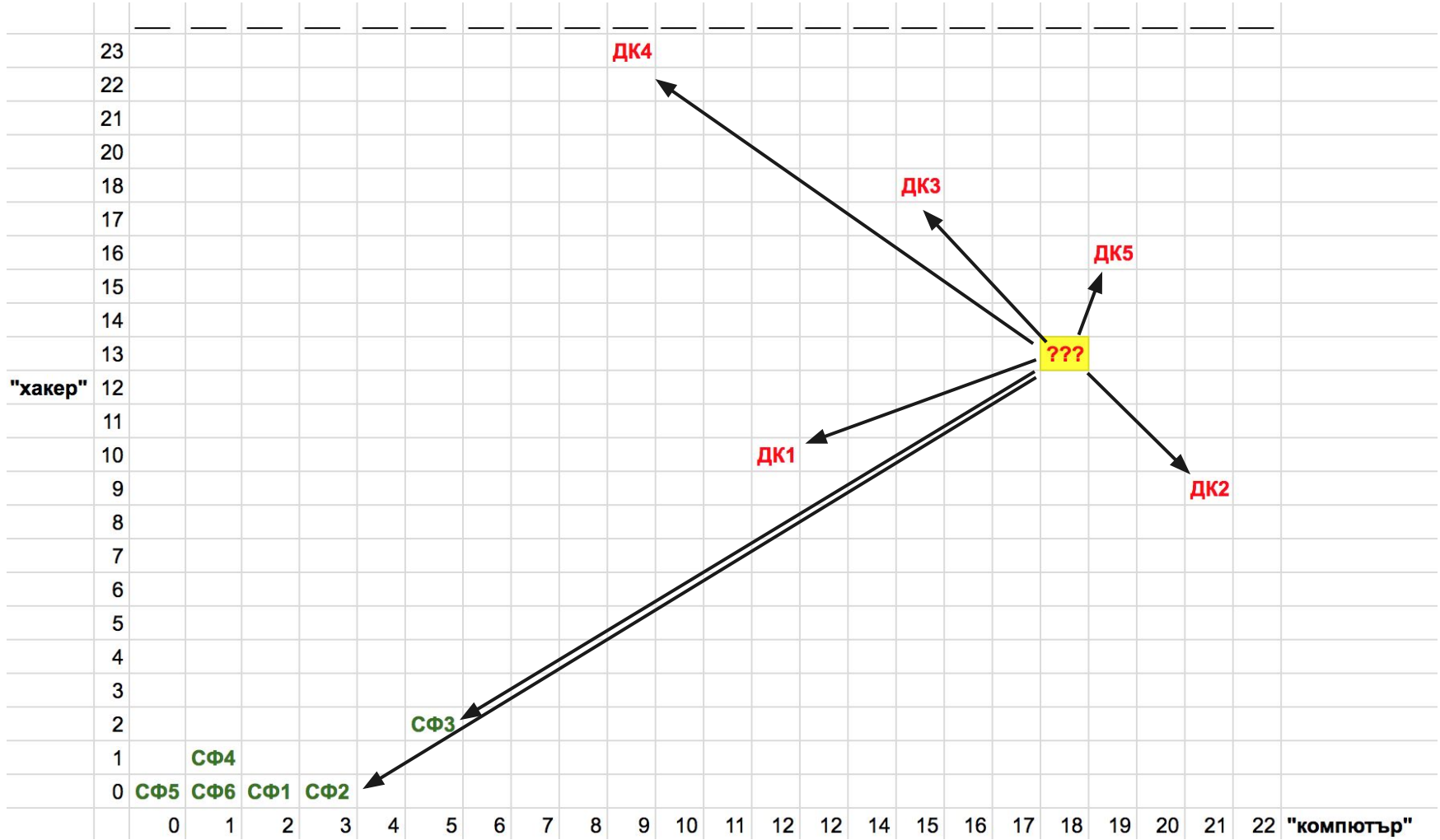

ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"

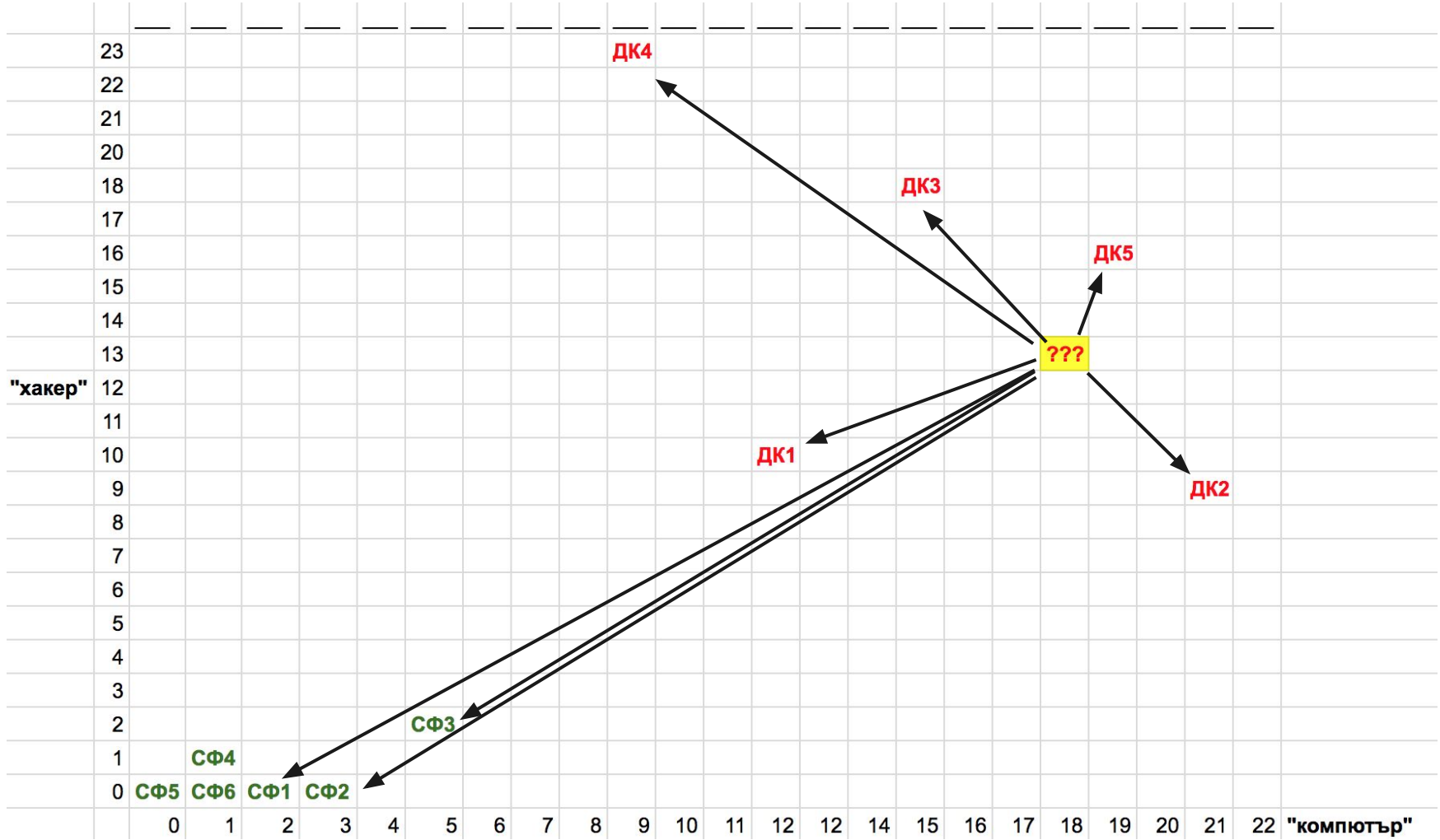

ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"

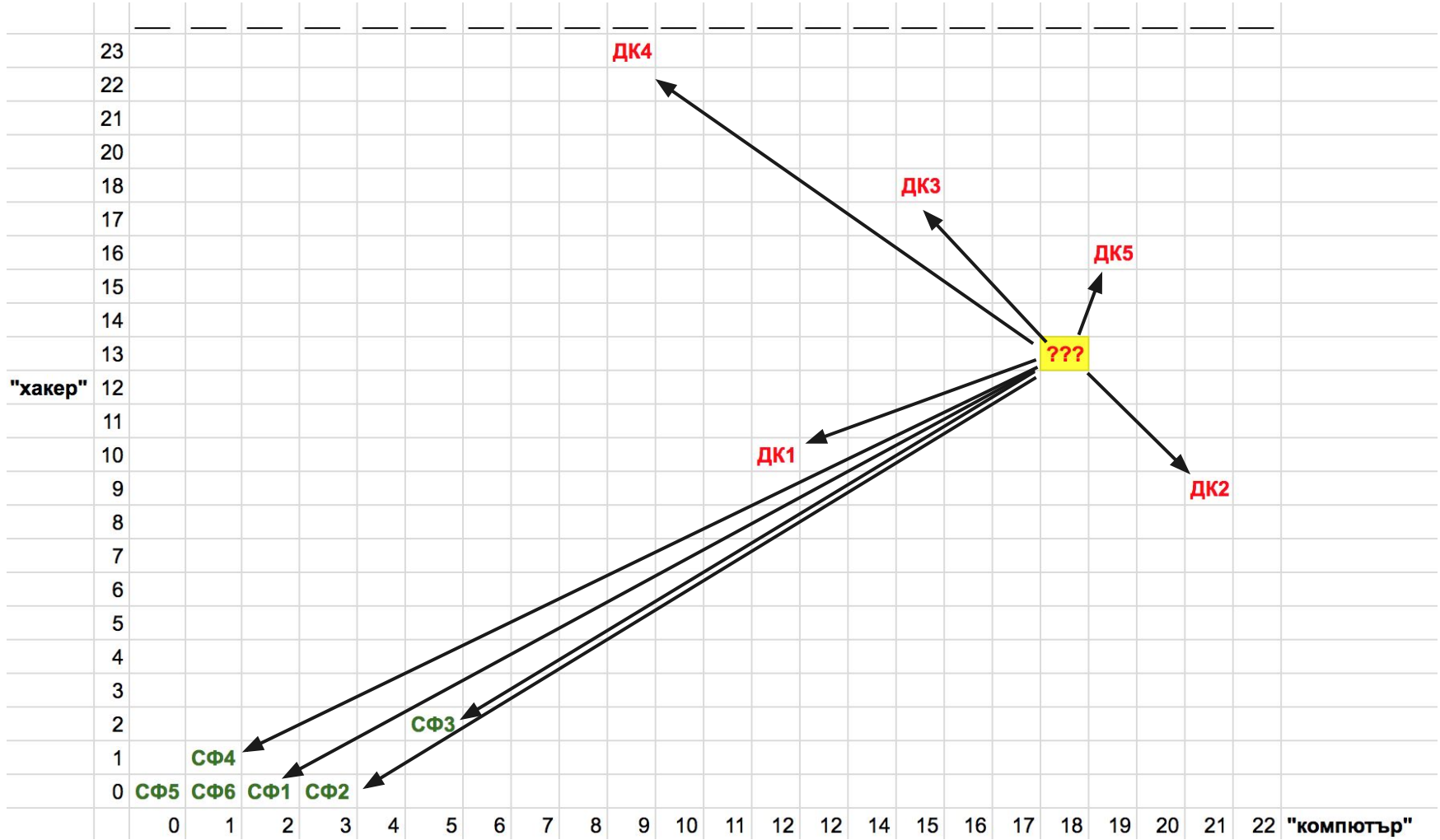

ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"

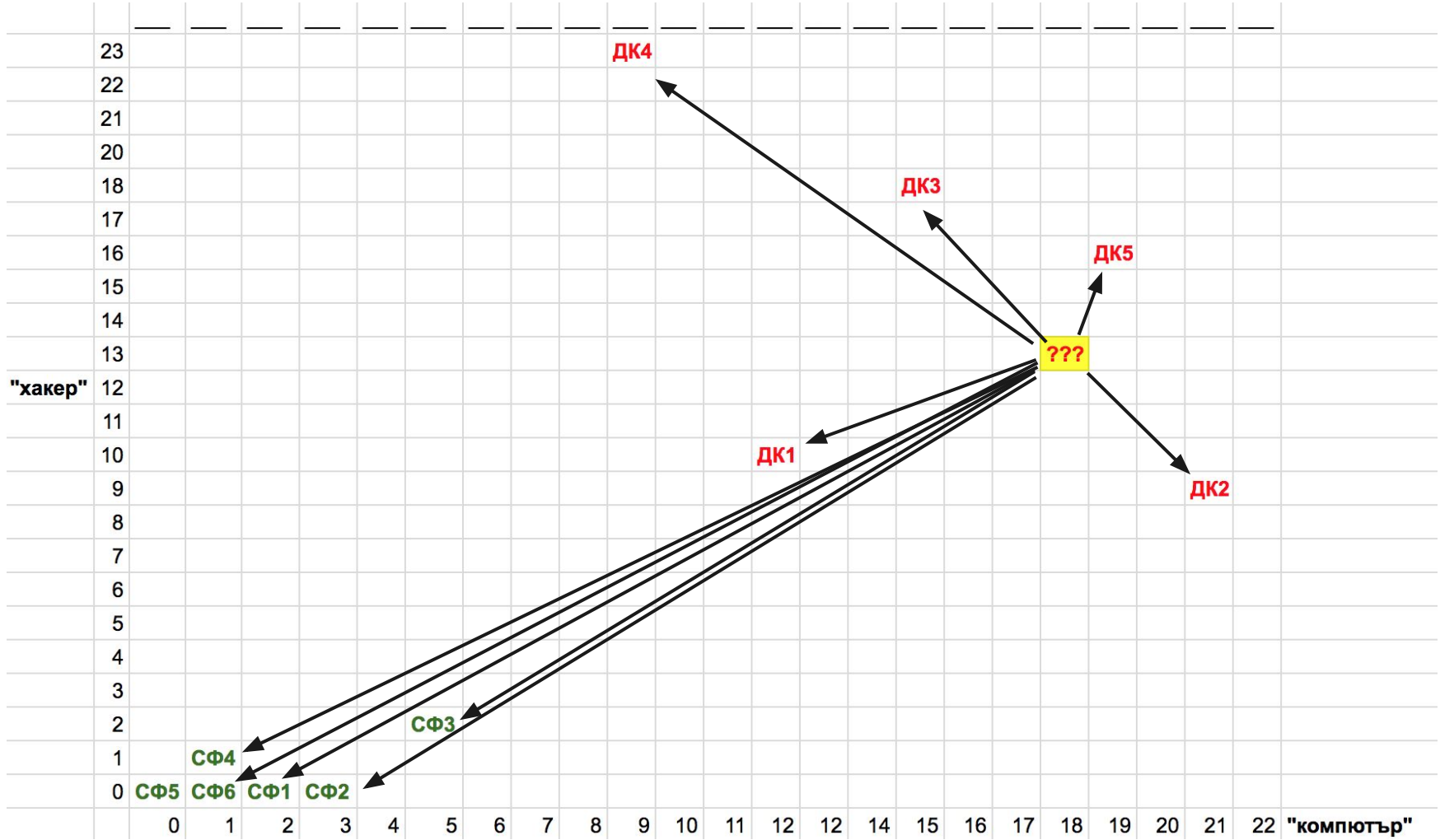

ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"

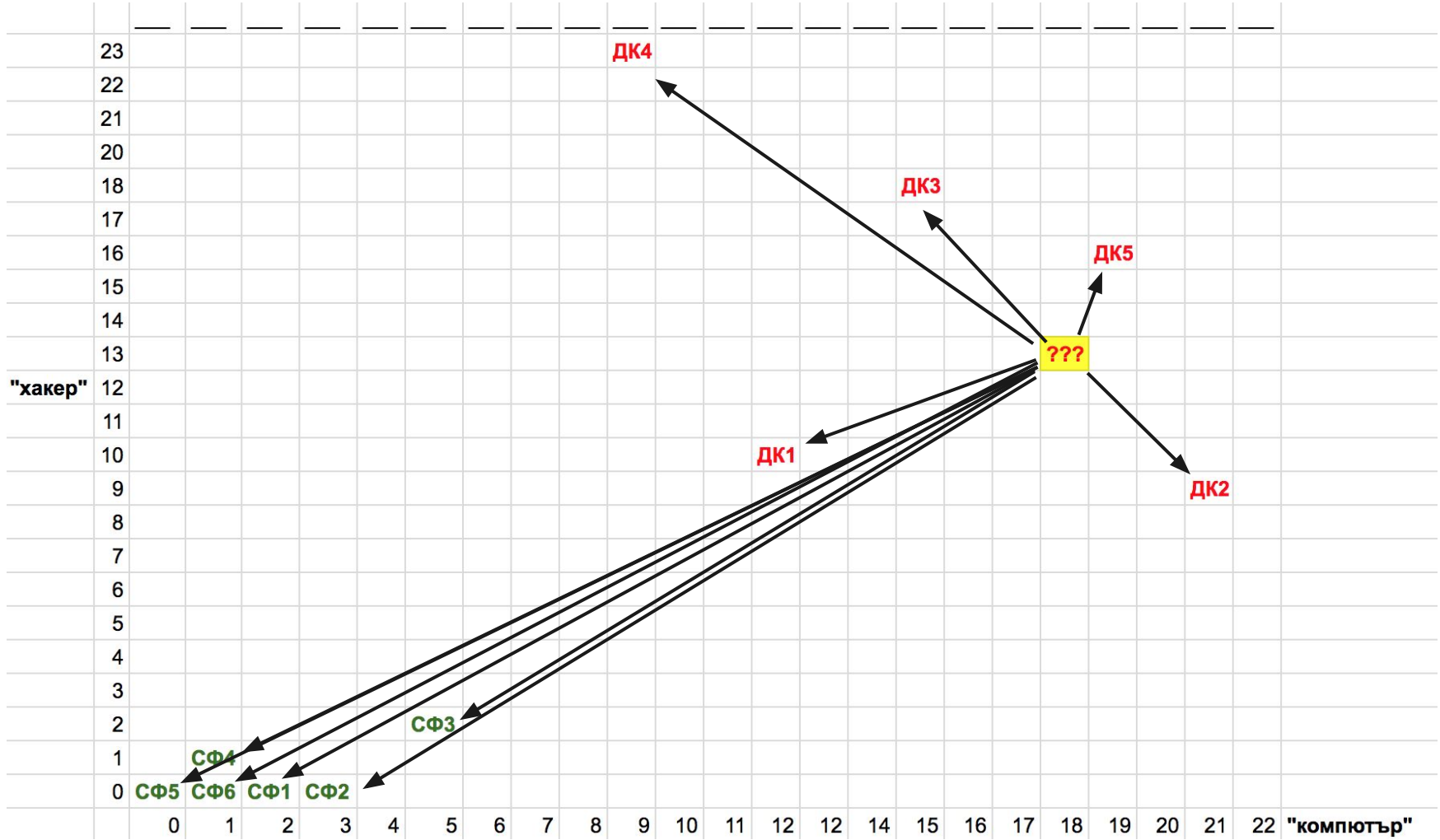

ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"


ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"


ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"


ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"


ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"


ДК4
ДК3
ДК5
???
ДК1
ДК2
СФ3
СФ4
СФ5
СФ6
СФ1
СФ2
"хакер"
"компютър"

Наблюдения

- “Най-близките” статии са статии от “Дунавски Компютър”
- “Разстояние” или “дистанция” е от координатите на повторенията на “компютър” и “хакер” за всяка статия.

Сортиране на разстоянита

класация разстояние **К**	неизвестна статия до
1	**ДК5**
2	**ДК2**
3	**ДК3**
4	**ДК1**
5	**ДК4**
6	**СФ3**
7	**СФ2**
8	**СФ1**
9	**СФ4**
10	**СФ6**
11	**СФ5**

- Кои са най-близките съседи?

К= 1 : ДК5 - статията от Дунавски Компютър

К= 2 : ДК5, ДК2 - статията от Дунавски Компютър

К= 3 : ДК5, ДК2, ДК3 - статията от Дунавски Компютър

Сортиране на разстоянита

брой съседи **К**	съседи	Гласуване	заключение за неизвестната статия
1	**ДК5**	**1**:0	**Дунавски Компютър**
2	**ДК5, ДК2**	**2**:0	**Дунавски Компютър**
3	**ДК5, ДК2, ДК3**	**3**:0	**Дунавски Компютър**
4	**ДК5, ДК2, ДК3, ДК1**	**4**:0	**Дунавски Компютър**
5	**ДК5, ДК2, ДК3, ДК1, ДК4**	**5**:0	**Дунавски Компютър**
6	**ДК5, ДК2, ДК3, ДК1, ДК4, СФ3**	**5**:1	**Дунавски Компютър**
7	**ДК5, ДК2, ДК3, ДК1, ДК4, СФ3, СФ2**	**5**:3	**Дунавски Компютър**
8	**ДК5, ДК2, ДК3, ДК1, ДК4, СФ3, СФ2, СФ1**	**5**:3	**Дунавски Компютър**
9	**ДК5, ДК2, ДК3, ДК1, ДК4, СФ3, СФ2, СФ1, СФ4**	**5**:4	**Дунавски Компютър**
10	**ДК5, ДК2, ДК3, ДК1, ДК4, СФ3, СФ2, СФ1, СФ4, СФ6**	**5**:5	???????????
11	**ДК5, ДК2, ДК3, ДК1, ДК4, СФ3, СФ2, СФ1, СФ4, СФ6, СФ5**	**5**:6	**Северозападен Фермер**

- Очевидно най-подходящата стойност за К е между 3 и 6 - първите 10-30% от най близките съседи.

Наблюдения

1. Близки съседи K= 4
2. Измерения на пространството на състоянията (feature space): m=2
   a. (“компютър”,”хакер”)
3. Класове (labels): l=2
   a. *labes* =(“Дунавски компютър”,”Северозападен Фермер”)
   b. Стойности на класове: $Y_i = label_1 . \text{или}. label_2$
4. Записи: n=11:

$$labels = (label_1, label_2)$$

$$\begin{aligned} X_1 &= (x_{1,1}, x_{1,2}, \ldots, x_{1,m}) & Y_1 \\ X_2 &= (x_{2,1}, x_{2,2}, \ldots, x_{2,m}) & Y_2 \\ \ldots & & \ldots \\ X_n &= (x_{n,1}, x_{n,2}, \ldots, x_{n.,m}) & Y_n \end{aligned}$$

Наблюдения

$X_1 = (12, 10)$ $Y_1 =$” ДК ”
$X_2 = (21, 9)$ $Y_2 =$” ДК ”
$X_3 = (15, 18)$ $Y_3 =$” ДК ”
$X_4 = (9, 23)$ $Y_4 =$” ДК ”
$X_5 = (19, 16)$ $Y_5 =$” ДК ”
$X_6 = (2, 0)$ $Y_6 =$” СФ ”
$X_7 = (3, 0)$ $Y_7 =$” СФ ”
$X_8 = (5, 2)$ $Y_8 =$” СФ ”
$X_9 = (1, 0)$ $Y_9 =$” СФ ”
$X_{10} = (0, 0)$ $Y_{10} =$” СФ ”
$X_{11} = (1, 0)$ $Y_{11} =$” СФ ”

к=4

feature space: m = 2

Labels = 2

Data entries n = 11

X_i and Y_i (*i=1..11*)

Наблюдения

Неизвестен текст с брой на думите “компютър”,”хакер”:$U = (u_1, u_2, .. u_m)$

$$U = (18, 13)$$

Разстояние (дистанция) $d_i = \sqrt{\sum_{i=1}^{n} (u_j - x_{i,j})^2}$

$$d_4 = \sqrt{\sum_{i=1}^{2} (u_j - x_{4,j})^2} = (u_1 - x_{4,1})^2 + (u_2 - x_{4,2})^2 =$$

$$(18 - 9)^2 + (13 - 23)^2 = (-9)^2 + (-10)^2 = 181$$

sqrt(181) = **13.45**

Алгоритъм

1. Избери параметър за класация на най-близките съседи K = 4.
2. Определи параметрите m и n
3. Зареди в паметта векторите данни $X_i(i = 1..n)$ и стойности на класовете $Y_i(i = 1..n)$, които могат да бъдат $label_1, lablel_2, .. label_L$
4. Въведи нова стойност да бъде класифицирана $U = (u_1, u_2, .. u_m)$
5. За всеки вектор данни изчислете всички дистанции до елементите $X_i(i = 1..n)$, използвайки формулата $d_i = \sqrt{\sum_{i=1}^{n} (u_j - x_{i,j})^2}$
6. От всички дистанции изберете най-късите k дистанции $d_1, d_2, \dots d_k$
7. Новата стойност $U = (u_1, u_2, .. u_m)$ принадлежи към класа $label$, който има най-голяма честота в $d_1, d_2, \dots d_k$

А сега малко Python

Програма на Python за изчисляване дистанциите между две точки от миналата лекция:

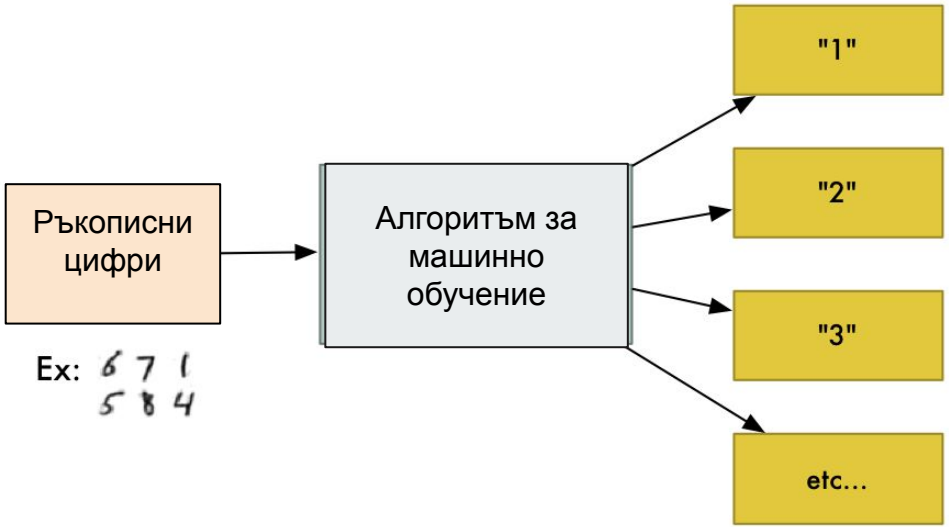
```
from math import *

def euclidian_distance(x,y):
  return sqrt(sum(pow(a-b,2) for a,b in zip(x,y)))

print euclidian_distance([0,3,4,5],[7,6,3,-1])
```

Изчислява дистанцията между точки U=(0,3,4,5) и X=(7,6,3,-1) и резултатът е 9.746794344

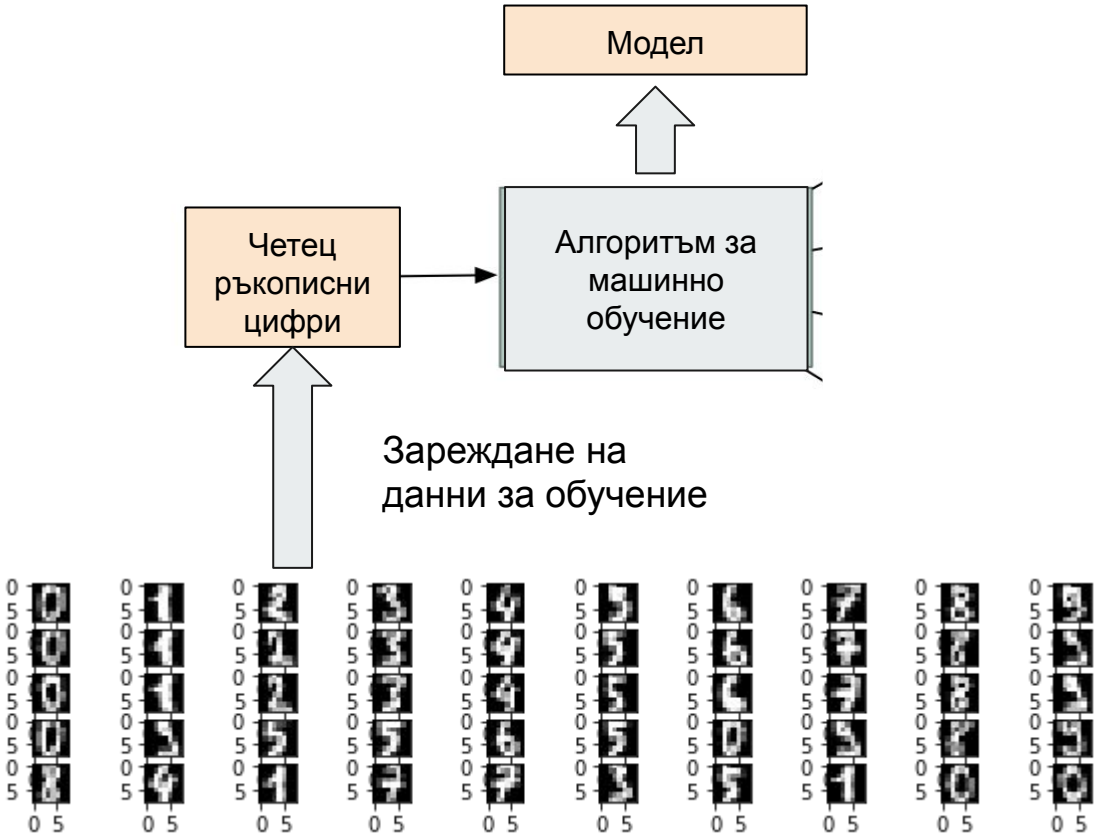
Да дефинираме проблема за разпознаване на ръкописни цифри



Имаме две фази:

1. Обучение и създаване на модел
2. Разпознаване на ръкописни цифри

Фаза 1 - обучение

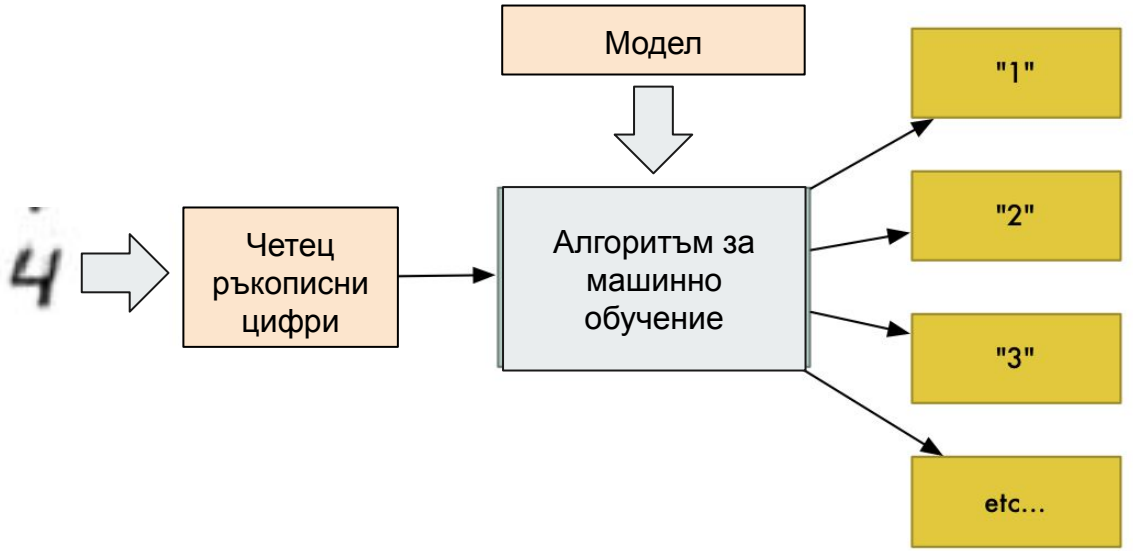


- Данните за обучение се зареждат в компютърната система.

- Алгоритъмът за машинно обучение създава **модел**.

- Моделът се използва за разпознаване на образи.

Фаза 2 - разпознаване на цифри

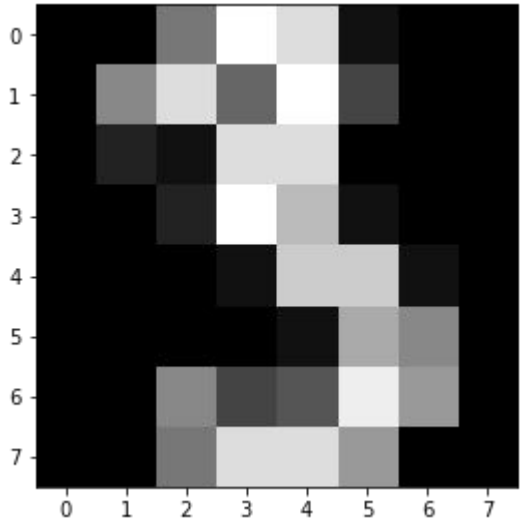


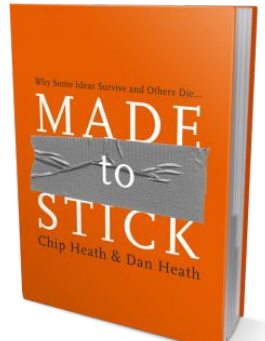
- Въвеждаме непозната ръкописна цифра.

- Използвайки модела, алгоритъмът за машинно обучение разпознава цифрите.

Метод с К най-близки съседи

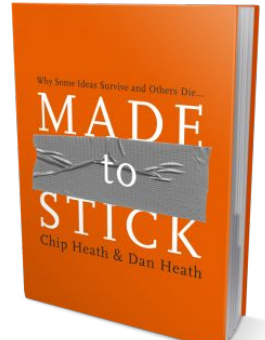
- Имаме 64 (8 x 8) измерения - колкото е растера на цифрата.
- Трябва да кажем към кой клас (1,2,3,4,5,6,7,8 или 9) принадлежи цифрата.
- **Да намерим най-близките съседи.**





Демонстрация на разпознаване на ръкописни числа с езика Python

- **А сега ще ви представя кратка демонстрация на разпознаване на ръкописни числа, използвайки езика Python**
- **Използват се библиотеки специално проектирани за машинно обучение**
- **Можете да експериментирате в онлайн интерпретатора https://repl.it/ Трябва да изберете Python**



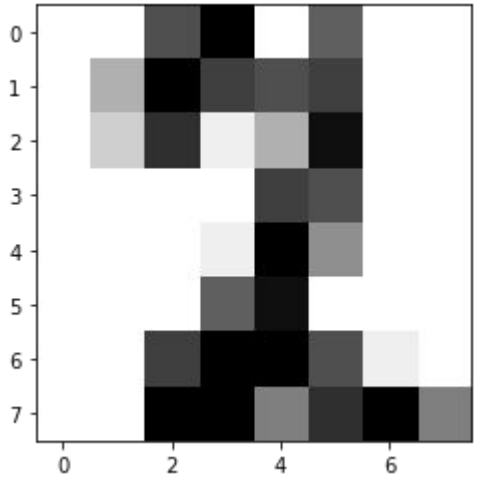
```
#  Зареждане на библиотеки за машинно обучение
from sklearn.datasets import load_digits
import numpy as np
from sklearn.neighbors import KNeighborsClassifier

# Зареждане на данните за машинно обучение
digits = load_digits()
x_train = digits.data
y_train = digits.target


# обучение: x_train е масивът с 1347 записа за цифри,
# y_train - цифрите
neigh = KNeighborsClassifier(n_neighbors=5, metric='euclidean')
neigh.fit(x_train, y_train)
```

 # Тест с данни на потребителя - растер с данни за числото 2.

```
two = [  0.,  0., 11., 16.,  0., 10.,  0.,  0.,
         0.,  5., 16., 12., 11., 12.,  0.,  0.,
         0.,  3., 13.,  1.,  5., 15.,  0.,  0.,
         0.,  0.,  0.,  0., 12., 11.,  0.,  0.,
         0.,  0.,  0.,  1., 16.,  7.,  0.,  0.,
         0.,  0.,  0., 10., 15.,  0.,  0.,  0.,
         0.,  0., 12., 16., 16., 11.,  1.,  0.,
         0.,  0., 16., 16.,  8., 13., 16.,  8.]
```




```
np_two = np.array(two)

prediction = neigh.predict(np_two.reshape(1,-1))
```

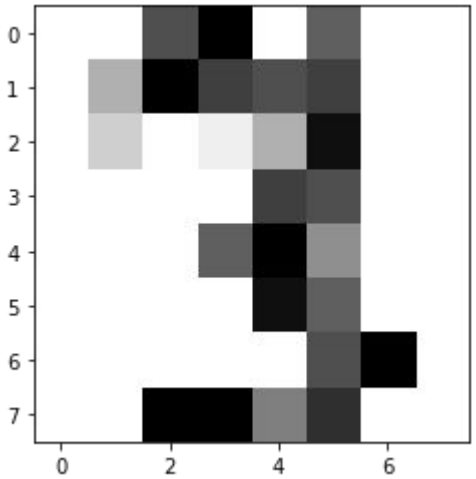
 # Разпознато число

```
print("predicted value ", list(prediction))
```

[2]

 # Тест с данни на потребителя - растер с данни за числото 3.

```
three = [   0.,  0.,  11.,  16.,  0.,  10.,  0.,  0.,
            0.,  5.,  16.,  12.,  11.,  12.,  0.,  0.,
            0.,  3.,  0.,   1.,   5.,  15.,  0.,  0.,
            0.,  0.,  0.,   0.,  12.,  11.,  0.,  0.,
            0.,  0.,  0.,   1.,  16.,  7.,  0.,  0.,
            0.,  0.,  0.,  0.,  15.,  10.,  0.,  0.,
            0.,  0.,  0.,  0.,  0.,  11.,  16.,  0.,
            0.,  0.,  16.,  16.,  8.,  13.,  0.,  0.]
```




```
np_three = np.array(three)

prediction = neigh.predict(np_three.reshape(1,-1))
```

 # Разпознато число

```
print("predicted value ", list(prediction))
```

[3]

БЛАГОДАРЯ И ДО НОВИ СРЕЩИ ?

Thank you! Danke ! Merci !

Литература

K Nearst Neighbours :
https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/

Recognizing Hand Written Digits
http://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html#sphx-glr-auto-examples-classification-plot-digits-classification-py