Кобак Ф.А. 18ДКК-1

Отчет по практическому заданию на 18.12.2020

Вариант 1

1)

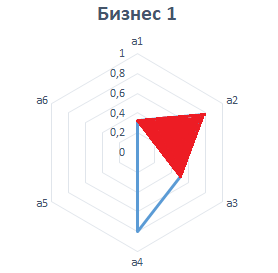
И так нормированные значения в таблице

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Бизнес-единица |  |  |  |  |  |  |
| 1 | 0.3226 | 0.7692 | 0.5 | 0.8039 | 0 | 0 |
| 2 | 0.0968 | 0.7692 | 0.7143 | 0.5098 | 0.1818 | 0.0577 |
| 3 | 0.1935 | 0.6923 | 0.5714 | 0.8824 | 0.0909 | 0.1731 |

2)

3) Принцип подсчёта площади многоугольников нарисованных синим проиллюстрирую на примере “Бизнес 1”.

Каждый из таких многоугольников можно представить как сложенные вместе 6 треугольников. Один из таких треугольников зарисован красным на следующем рисунке.



Мы знаем длинны двух его сторон – и , и угол между ними 60 градусов.

Далее пользуясь школьной формулой легко найдём его площадь:

, аналогично для других треугольников.

Сложив площади всех этих треугольников получим ответ.

Формула для площади шестиугольника выделенного синим примет вид.

Для моего варианта получаем

=0.433(0.3226\*0.7692+0.7692\*0.5+0.5\*0.8039+0+0+0) =0.4480

=0.433\*(0.0968\*0.7692+0.7692\*0.7143+0.7143\*0.50998+0.50998\*0.1818 + 0.1818\*0.0577+0.0577\*0.0968) =0.475

=0.433\*(0.1935\*0.6923+0.6923\*0.5714+0.5714\*0.8824+0.8824\*0.0909+0.0909\*0.1731+0.1731\*0.1935)=0.5037.

Для того чтобы подсчитать площадь правильного шестиугольника – основания диаграммы воспользуемся тем же принципом. Он состоит из 6 треугольников площадью 0.433, получаем его площадь

Таким образом получаем, показатели конкурентно способности

Первого бизнеса

4) Получаем, что в рейтинге из трех бизнес единиц наибольшую конкурентоспособность будет иметь третья. Второе место получает вторая единица и на последнем месте оказалась вторая.

5) Было разработано две функции для python

первая нормирует разнонаправленные переменные

def norm\_multidirectional(x):

    '''функция для снормирования  разнонаправленных переменных'''

    width = len(x[1]);

    height = len(x);

    a = [];

    for i in range(0, height):

        a.append([]);

        for j in range(0, width):

            a[i].append(0);

    for j in range(0, width):

        temp = [];

        for i in range(0, height):

            temp.append(x[i][j]);

        print(temp);

        max\_j = max(temp);

        min\_j = min(temp);

        for i in range(0, height):

            temp2 = (temp[i] - min\_j)/(max\_j - min\_j);

            a[i][j] = temp2;

    return a;

Вторая функция строит диаграммы по нормированным переменным

def build\_diagramm(a):

    labels=np.array(['a1', 'a2', 'a3', 'a4', 'a5', 'a6']);

    angles=np.linspace(0, 2\*np.pi, len(labels), endpoint=False);

    angles=np.concatenate((angles,[angles[0]]));

    stats=np.concatenate((a,[a[0]]));

    fig = plt.figure();

    ax = fig.add\_subplot(111, polar=True);

    ax.plot(angles, stats, 'o-', linewidth=2);

    ax.fill(angles, stats, alpha=0.25);

    angles=np.linspace(0, 2\*np.pi, len(labels), endpoint=False);

    ax.set\_thetagrids(angles \* 180/np.pi, labels)

    plt.show();

Для приведения в действие можно использовать команды python

from hello import norm\_multidirectional;

from hello import build\_diagramm;

x = [[3, 2.6, 2.4, 113, 47, 0.3],

    [ 2.3, 2.6,2.7,98,49,0.6],

    [2.6,2.5,2.5,117,48,1.2],

    [4.3,2.5,2.4,91,55,2.3],

    [2.9,2.8,2.1,99,49,2.6],

    [2.4,3.1,3.1,89,52,5.5],

    [5.1,1.6,2.1,79,58,2.4],

    [3.4,2,1.7,72,57,1.6],

    [2,2.9,2.7,123,50,3.2],

    [4.5,2.9,2.8,80,53,4.2]];

a = norm\_multidirectional(x);

for i in range(0, 10):

    print(a[i]);

build\_diagramm(a[0]);

build\_diagramm(a[1]);

build\_diagramm(a[2]);

В результате получим

Это матрица нормированных переменных

[0.32258064516129037, 0.6666666666666666, 0.49999999999999994, 0.803921568627451, 0.0, 0.0]

[0.09677419354838705, 0.6666666666666666, 0.7142857142857144, 0.5098039215686274, 0.18181818181818182, 0.05769230769230769]

[0.19354838709677424, 0.6, 0.5714285714285714, 0.8823529411764706, 0.09090909090909091, 0.17307692307692304]

[0.7419354838709677, 0.6, 0.49999999999999994, 0.37254901960784315, 0.7272727272727273, 0.3846153846153846]

[0.2903225806451613, 0.7999999999999998, 0.2857142857142858, 0.5294117647058824, 0.18181818181818182, 0.44230769230769235]

[0.12903225806451613, 1.0, 1.0, 0.3333333333333333, 0.45454545454545453, 1.0]

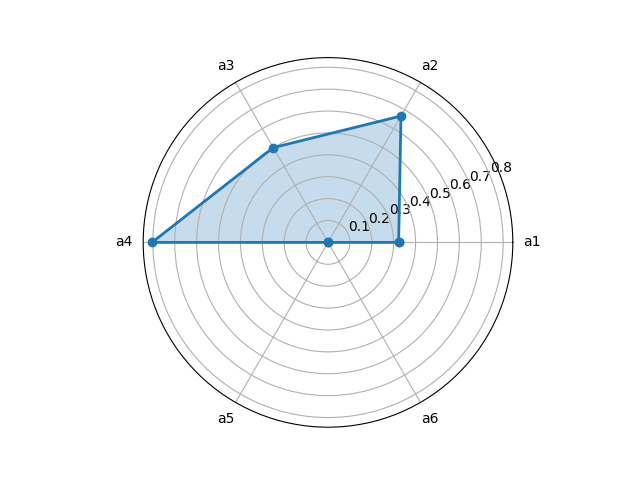
[1.0, 0.0, 0.2857142857142858, 0.13725490196078433, 1.0, 0.40384615384615385]

[0.4516129032258065, 0.2666666666666666, 0.0, 0.0, 0.9090909090909091, 0.25]

[0.0, 0.8666666666666666, 0.7142857142857144, 1.0, 0.2727272727272727, 0.5576923076923077]

*[0.8064516129032259, 0.8666666666666666, 0.7857142857142856, 0.1568627450980392, 0.5454545454545454, 0.75]*

Это построенные по матрице диаграммы

**

