УСРС-1 “Кластерный анализ при неизвестном числе кластеров”
Кобака Ф.А. 18ДКК-1 ФЦЭ

Работа была выполнена на языке программирования python3. Далее представлены важные отрывки кода. С результатами их выполнения:

Для того чтобы комбинировать объекты по подгруппам была написана функция:

```
def ful_combinator(objects, sizes, appendics = []):
    result = []
    f = False
    for i in combinations(objects, sizes[0]):
        i = list(i)
        for_next = list(set(objects) - set(i))
        if f == False:
            appendics.append(i)
            f = True
        else:
            appendics[len(appendics) - 1] = i

        if len(sizes) != 1:
            for j in ful_combinator(for_next, sizes[1:], copy.copy(appendics)):
                j.insert(0,i)
                result.append(j)
        else:
            result.append([objects])
    return result
```

Первым аргументом передается массив каких либо объектов описанных на ЯП python3, к примеру, чисел. Вторым массив размеров класстов на которые будут разбиваться приведенные ранее объекты. Третий аргумент технический потому его следует оставить пустым. Например разобьем 5 чисел от 1 до 5 на два класса, 2 объекта в перовом и 3 во втором. Из коммандной строки.

```
>>> for i in ful_combinator([1,2,3,4,5], [2,3]): print(i)
...
[[1, 2], [3, 4, 5]]
[[1, 3], [2, 4, 5]]
[[1, 4], [2, 3, 5]]
[[1, 5], [2, 3, 4]]
[[2, 3], [1, 4, 5]]
[[2, 4], [1, 3, 5]]
[[2, 5], [1, 3, 4]]
[[3, 4], [1, 2, 5]]
[[3, 5], [1, 2, 4]]
[[4, 5], [1, 2, 3]]
>>>
```

К сожалению сформировать алгоритм так, чтобы не было пересекающихся разбиений мне пока не удалось.

Теперь как это может быть использовано для решения поставленной задачи, на примере моего варианта 1 при разбиении на два кластера:

```
# изначально генерируем описанной выше функцией комбинации из чисел от 1 до 12
comb = ful_combinator(list(range(0,12)), [6,6])

# готовим pandas таблицу для того чтобы записать результаты
result_data = pd.DataFrame(columns = ["разбиение", "значение ФКР"])

print(len(comb))

# тут перебирам все выше подготовленные комбинации и вычисляем для них ФКР
for i in comb:
   variant = str(i[0]) + '-' + str(i[1])
   result_data = result_data.append({"разбиение" : variant, "значение ФКР" : functional5_8([
df.iloc[i[0]], df.iloc[i[1]]])}, ignore_index= True)
```

Вычисление ФКР проводиться заранее подготовленной функцией. Тут привдено для формулы 5.8, но у меня подготовлены и для других функционалов

```
def functional5_8(data, metrics_computer = eucl_metrics, info = False):
   #data = [[cluster1],[cluster2], ...] cluster[i] = pandas.dataframe

   # сумма квадратов попарных расстояний
   sq_dist_sum = 0

   # перебираем данные кластеры
   for i,cluster in enumerate(data):

      size = len(cluster.index)

      if info:
         print("cluster:" + str(i) + "==================================
==============")
         print(cluster)

      # то для скольки объектов попарные дистанции мы уже вычислили

      for h,j in enumerate(cluster.index):
         for k in cluster.index[h+1:]:

            if info:
               print("items " + str(j) + " and " + str(k) + "----------")
               print("item "+ str(j) + ":" + str(cluster.loc[j].values.tolist()))
               print("item "+ str(k) + ":" + str(cluster.loc[k].values.tolist()))
```

```
            print("distance: " + str(metrics_computer(cluster.loc[j], cluster.loc[k])))

            sq_dist_sum += metrics_computer(cluster.loc[j], cluster.loc[k])**2


    return sq_dist_sum
```

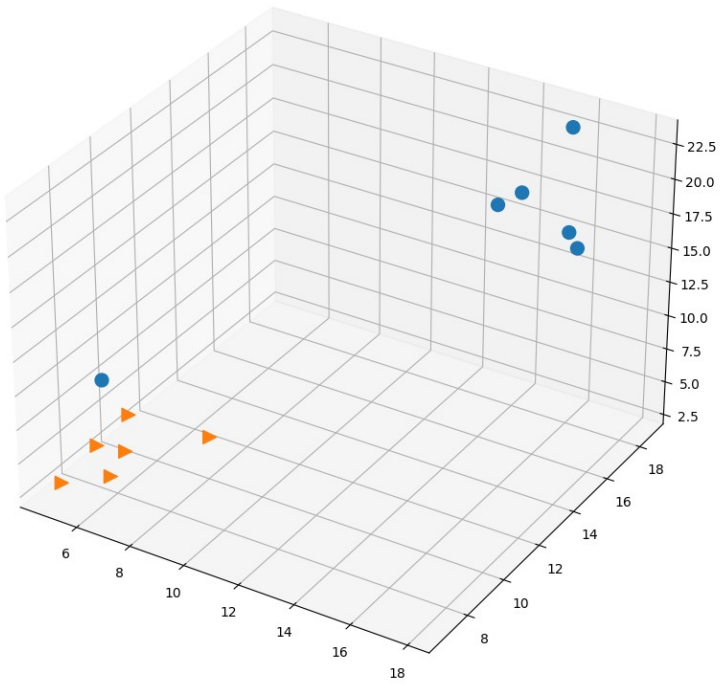
В результате будет получена pandas таблица с результатами. Все что остается это найти строчку с минимальным значением ФКР, это можно сделать так:

```
min_fkr_index = result_data[['значение ФКР']].idxmin()
print(comb[int(min_fkr_index)])  # выведет в консоль комбинацию с наименьшим ФКР
```

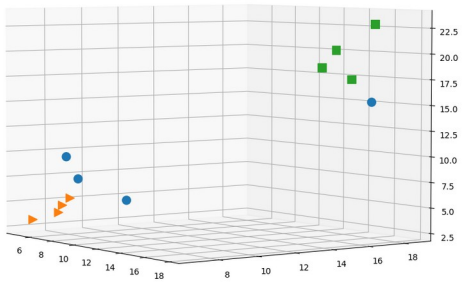
В pandas таблицы очень легко сохранить в excel, что я и сделал. В moodle я приложу excel файл с значениями ФКР для двух и трех кластеров на разных листах. Третий приложить не получается — размер файла превышает максимально допустимый к отправке в moodle, но по первому требованию могу приложить к письму.

В результате при разбиении на два класса наилучшим оказалось разбиение [4, 5, 6, 8, 10, 11]-[0, 1, 2, 3, 7, 9]

Его визуализация на рисунке



При разбиении на 3 класса [3, 6, 7, 8]-[0, 1, 2, 9]-[10, 11, 4, 5]



И при разбиении на 4. [0, 3, 8]-[4, 10, 11]-[5, 6, 7]-[1, 2, 9]