Кластеризация

Найти в данных группирование, при котором элементы в одном кластере (группе) будут похожи друг на друга больше, чем на элементы из других кластеров

Группирование объектов по подобию признаков. Метод k-средних (k-means)

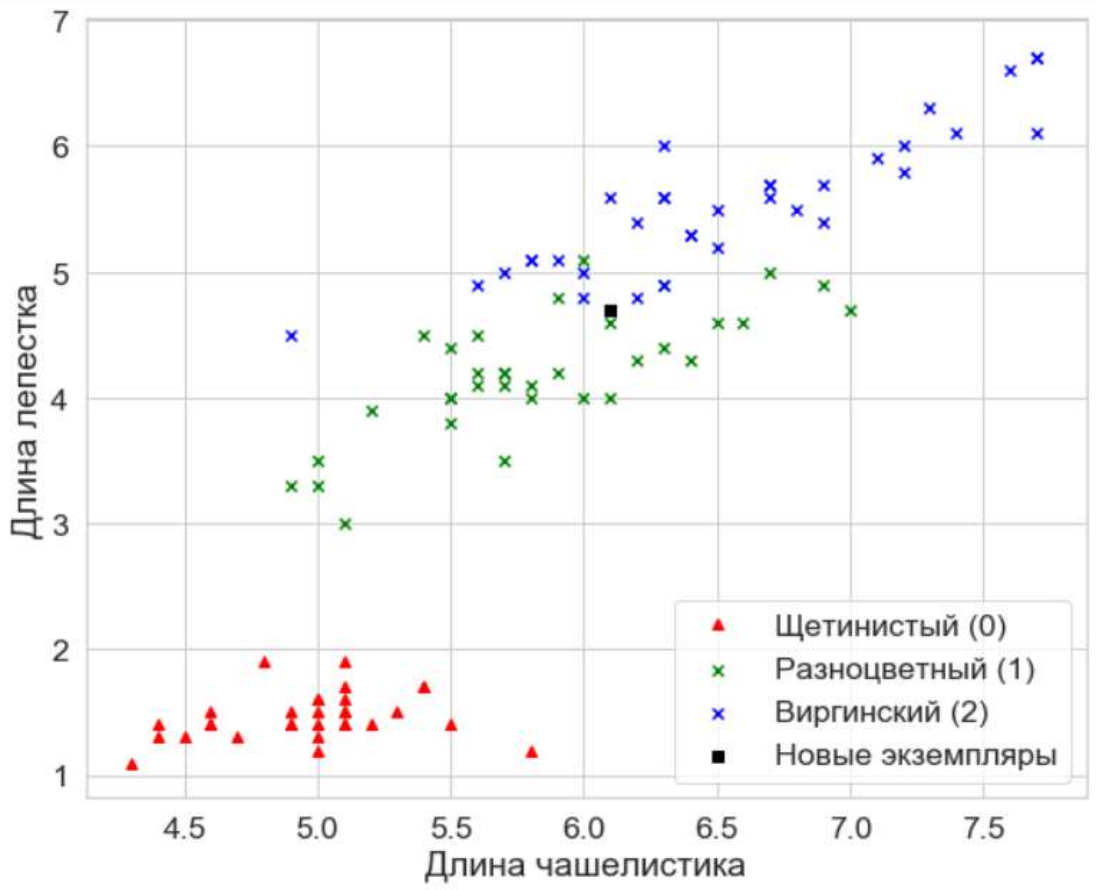
- Хотим минимизировать суммарное квадратичное отклонение точек кластера от их центра

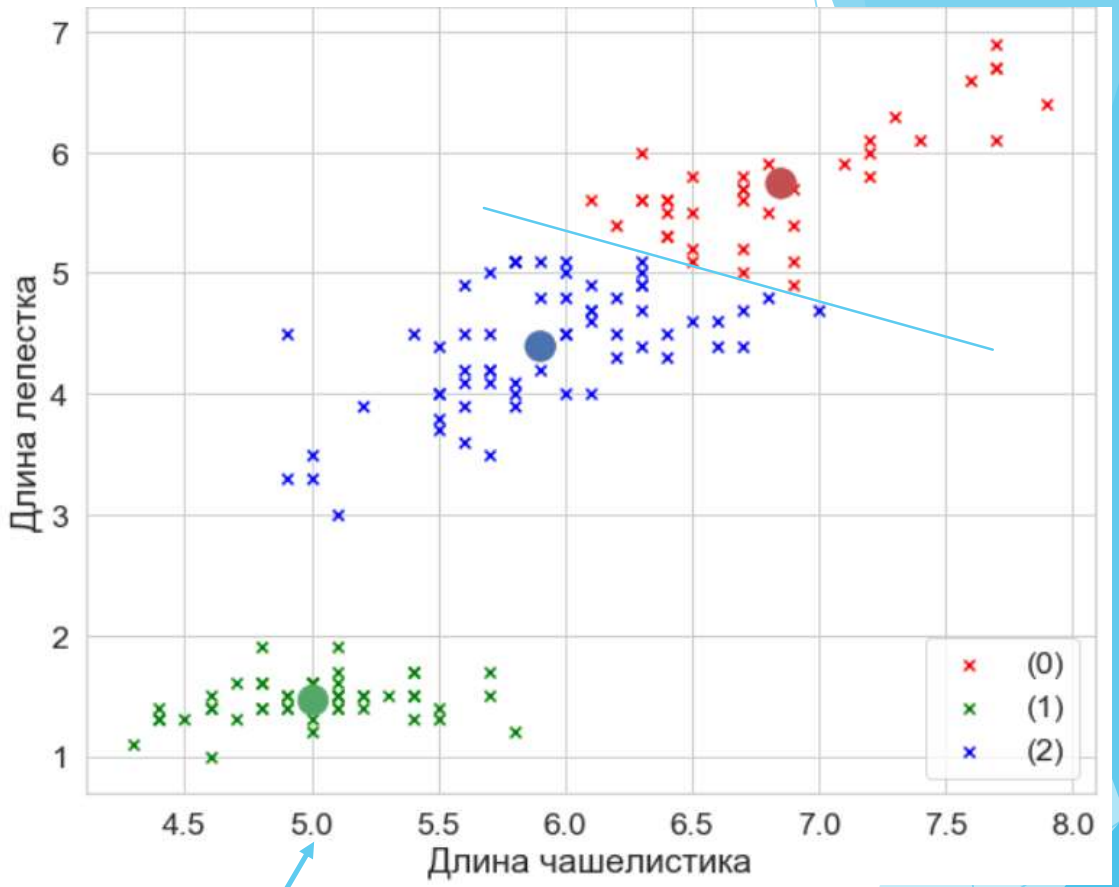
$$\sum_{i=1}^{k} \sum_{x \in S_i} (x - \mu_i)^2 \to min$$

- k – количество кластеров, S_i – кластер, μ_i - центр кластера/центроид (среднее координат всех точек кластера)

1. Случайным образом выбрать k центров кластера (из образцов) в качестве начальных центров
2. Перебрать все образцы и назначить каждый из них ближайшему центроиду μ_i
3. На основе образцов каждого кластера получить реальный центр масс (центроид)
4. Повторять шаги 2 и 3 пока изменяется состав кластеров или пока не будет достигнуто максимальное число итераций

Пример на плоскости






```
kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
kmeans.labels_
```

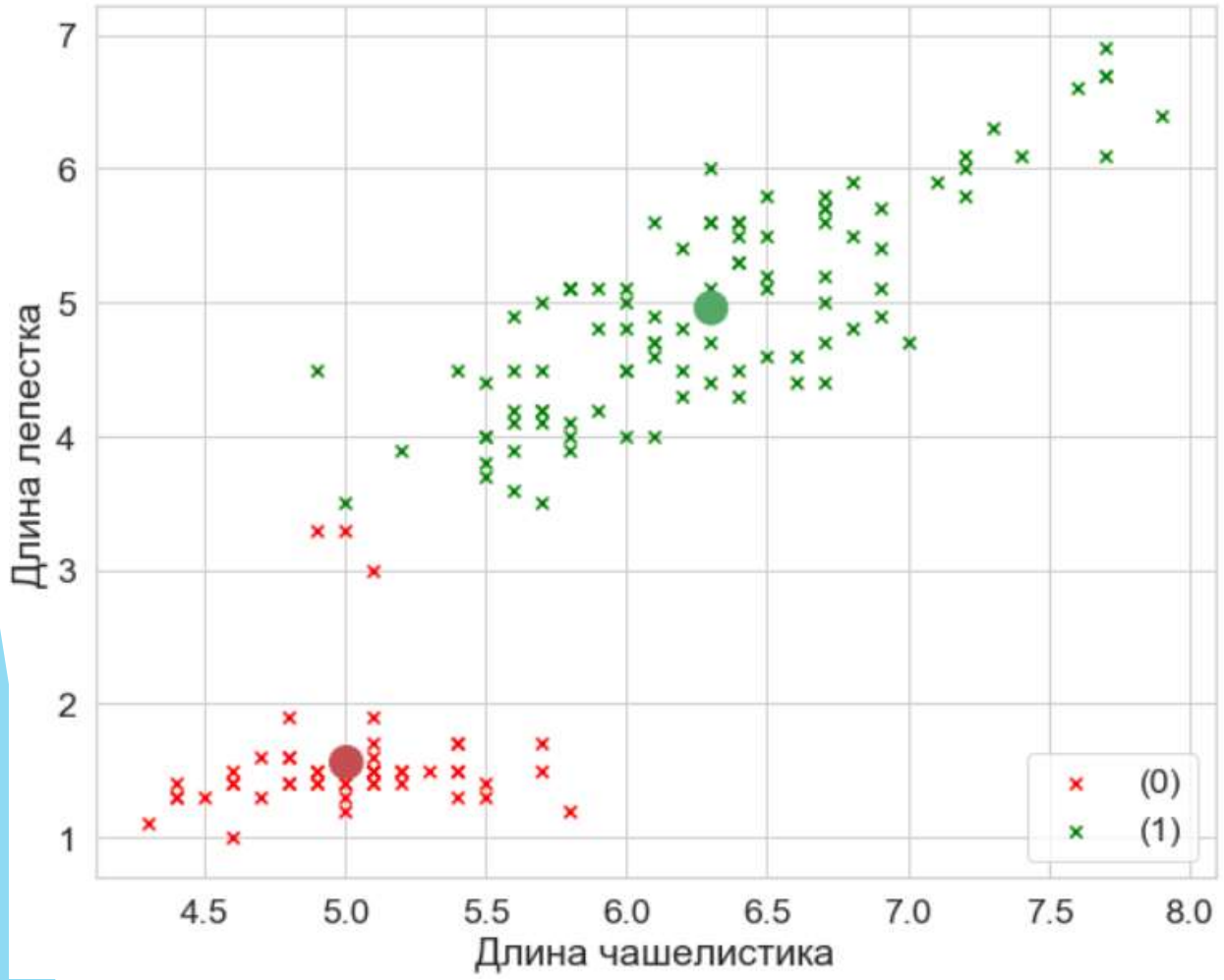
```
1 kmeans.cluster_centers_
```

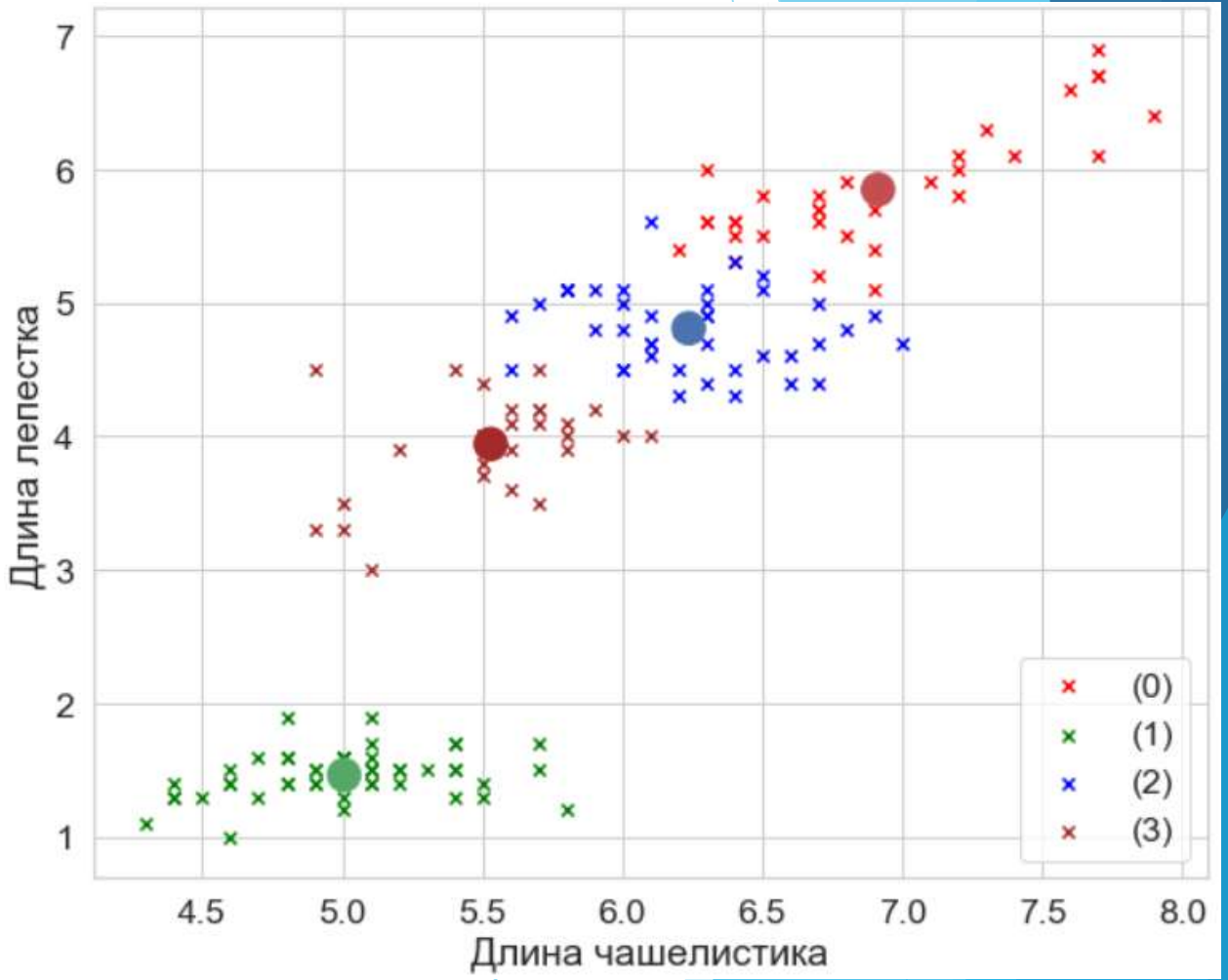
```
array([[6.85      , 3.07368421, 5.74210526, 2.07105263],
       [5.006     , 3.428     , 1.462     , 0.246     ],
       [5.9016129 , 2.7483871 , 4.39354839, 1.43387097]])
```

```
1 X[kmeans.labels_ == 1][:,0].mean()
```

```
5.006
```

Пример на плоскости (k=2, k=4)





K-Means

- Легко реализовать
- Эффективен с вычислительной точки зрения

далее

- Результат зависит от выбора исходных центров кластеров
- Необходимо заранее указывать количество кластеров k.
- Идентификация кластеров со сферической формой

- Важно! Так как это метрический алгоритм, необходимо чтобы признаки были одинакового масштаба

Выбор исходных центров кластеров K-Means++

- Идея - выбор начальных центроидов далеко друг от друга

- Выбрать первый центроид μ_1 случайно
- Для каждого образца x_i, не являющимся центроидом, найти квадратичное расстояние до всех центроидов $D(x, \mu_i)^2$
- Выбрать x_c в качестве нового центроида используя вероятность

$$\frac{D(x_c, \mu)^2}{\sum_i D(x_i, \mu)^2}$$

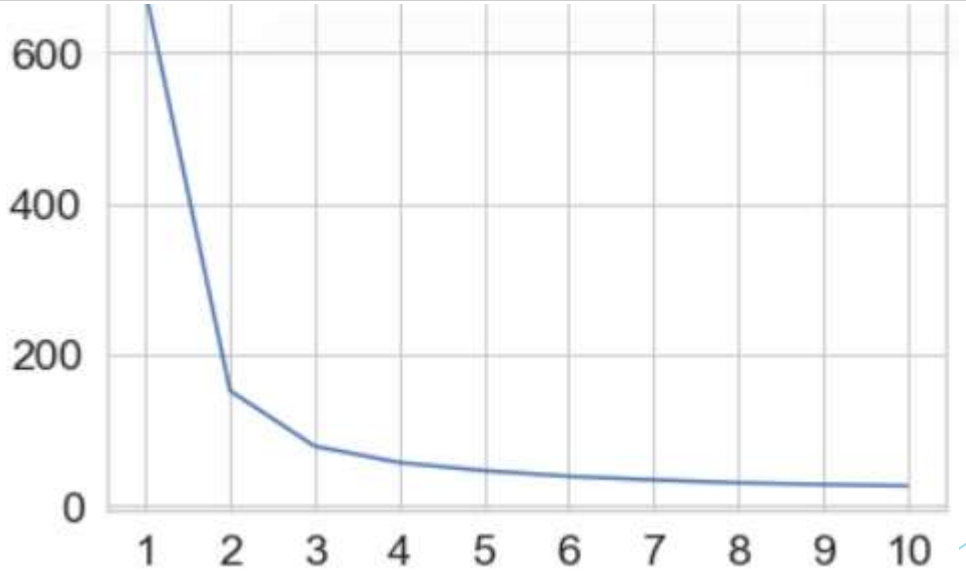
- Повторять шаги пока не выберется k центроидов
- Продолжить работу по классическому k-means

```
kmeans = KMeans(n_clusters=3, init='k-means++')
```

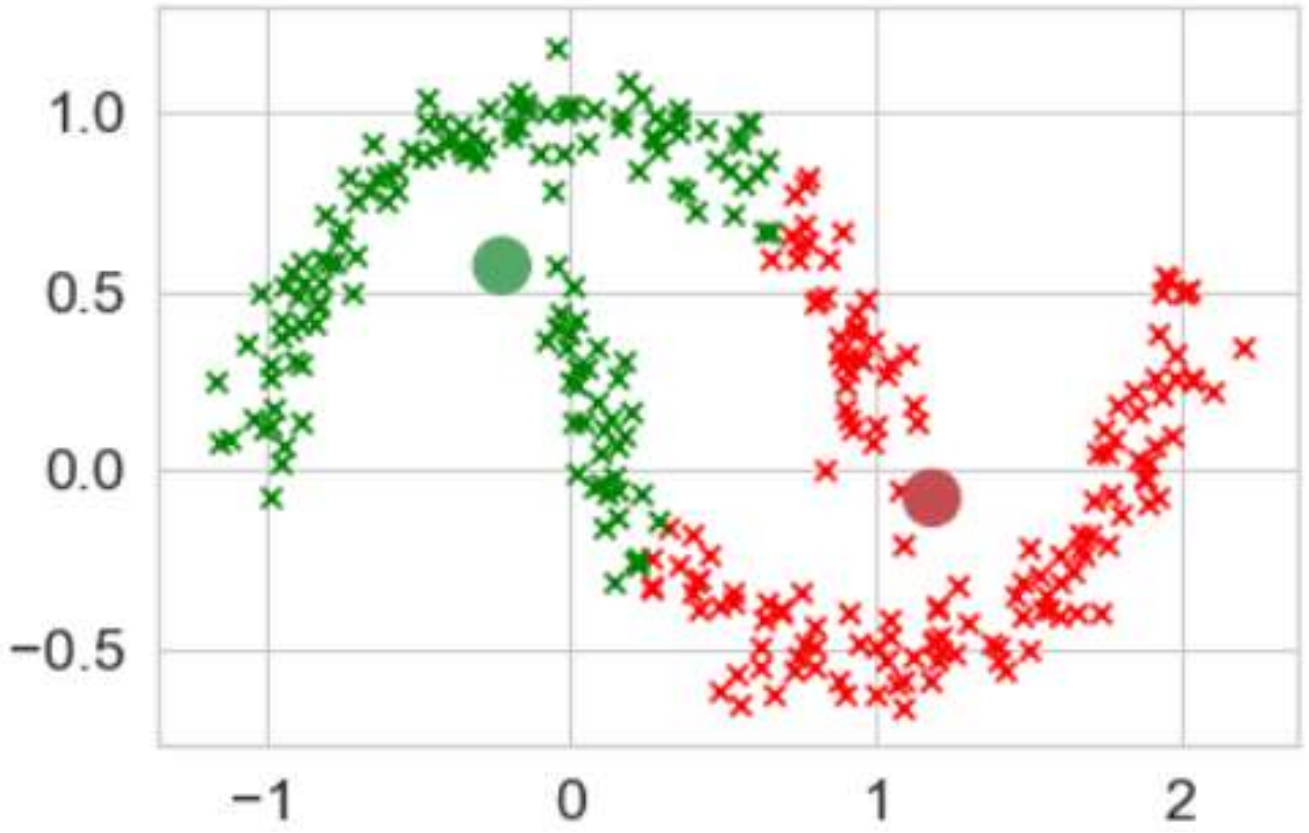
Выбор количества кластеров
Метод локтя

- В алгоритме k-means есть параметр инерция – сумма квадратов расстояний образцов до ближайшего центроида
- По мере увеличения k эта сумма убывает
- Вариант - в качестве k взять ту величину, после которой инерция не уменьшается существенно

```
inertia = [KMeans(n_clusters=i).fit(X).inertia_ for i in range(1,11)]
```



Идентификация кластеров сферической формы



Алгоритм кластеризации с использованием нахождения областей высокой плотности. DBSCAN – Density-Based Spatial Clustering of Applications with Nose

Каждому образцу назначается метка с применением перечисленных ниже критериев:

- Точка считается ядерной (core point) если по крайней мере указанное количество (MinPts) соседних точек попадают внутрь заданного радиуса ε.
- Точка считается граничной (border point) если число соседей меньше MinPts в пределах ε, но лежащая внутри радиуса ε ядерной точки.
- Все остальные точки считаются шумовыми (noise points).

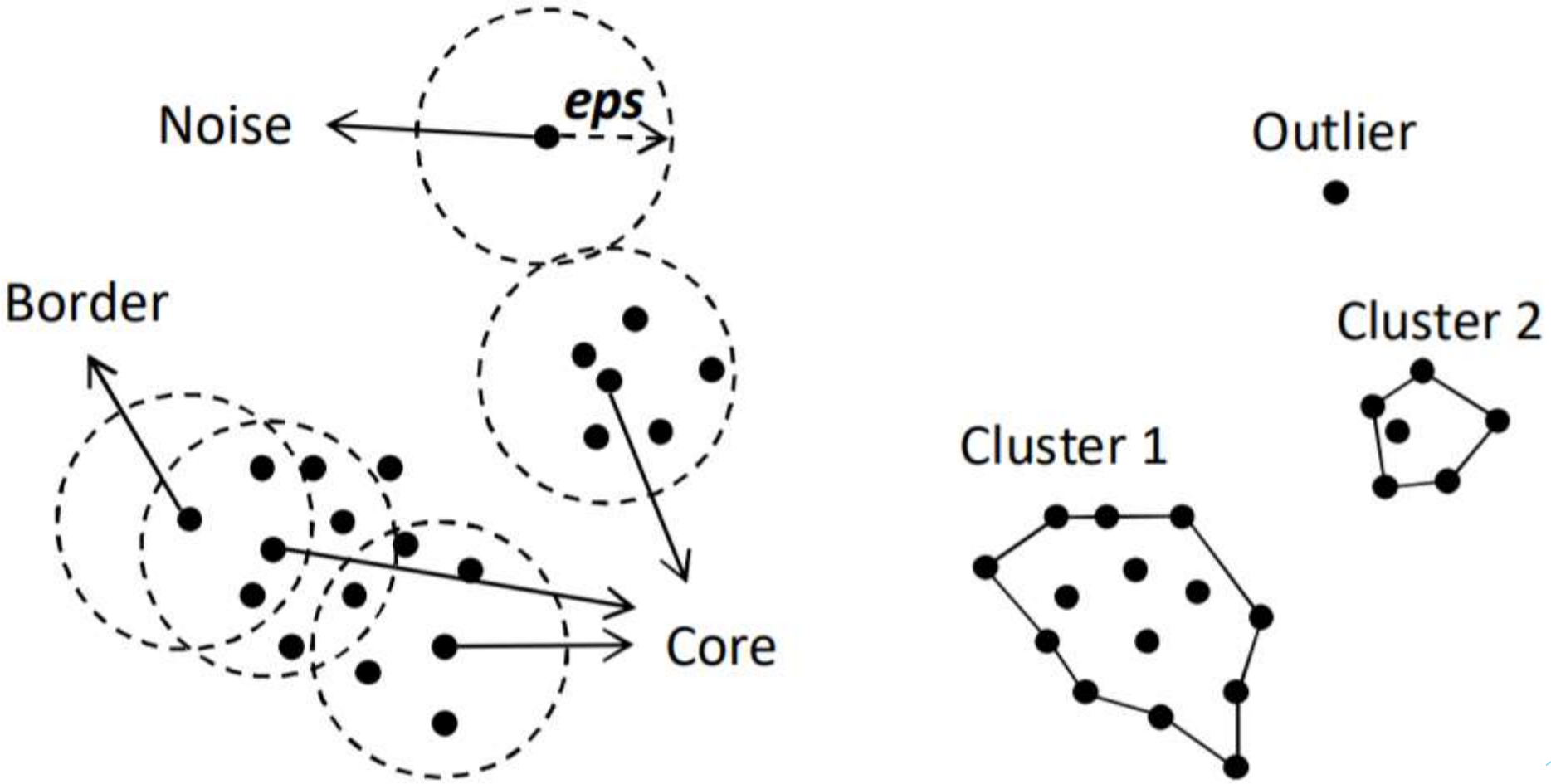
Алгоритм, особенности

- Сформировать отдельный кластер для каждой ядерной точки или связной группы ядерных точек (Ядерные точки считаются связными, если расположены не дальше чем ε).
- Назначить каждую граничную точку кластеру, к которому принадлежит соответствующая ей ядерная точка.

- Не выдвигает допущения относительно сферической формы кластеров.
- Необязательно назначает каждую точку кластеру (могут остаться точки не входящие ни в один кластер)
- Нет такого понятия как центроид – можем сами посчитать среднее

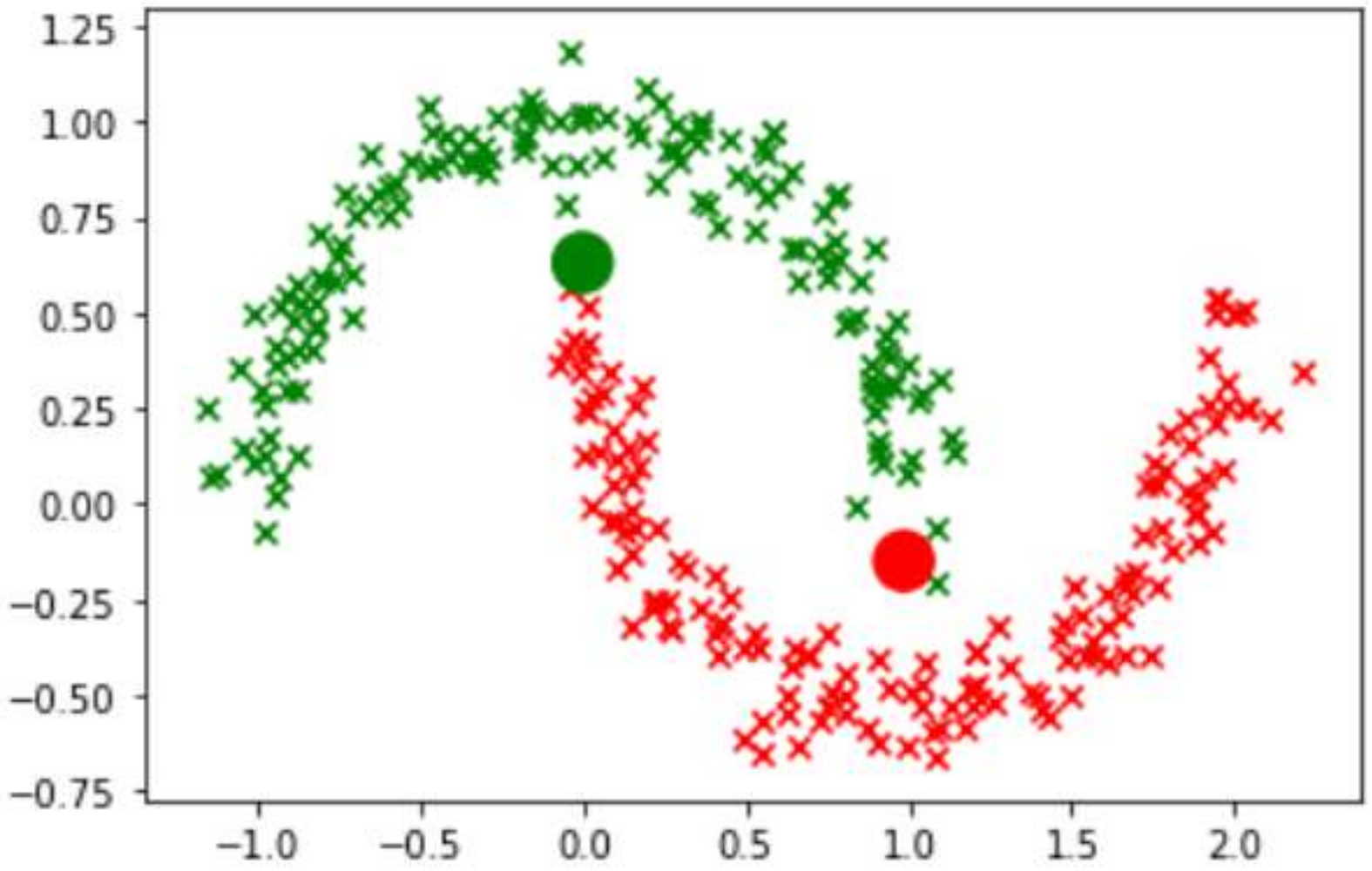
Пример работы

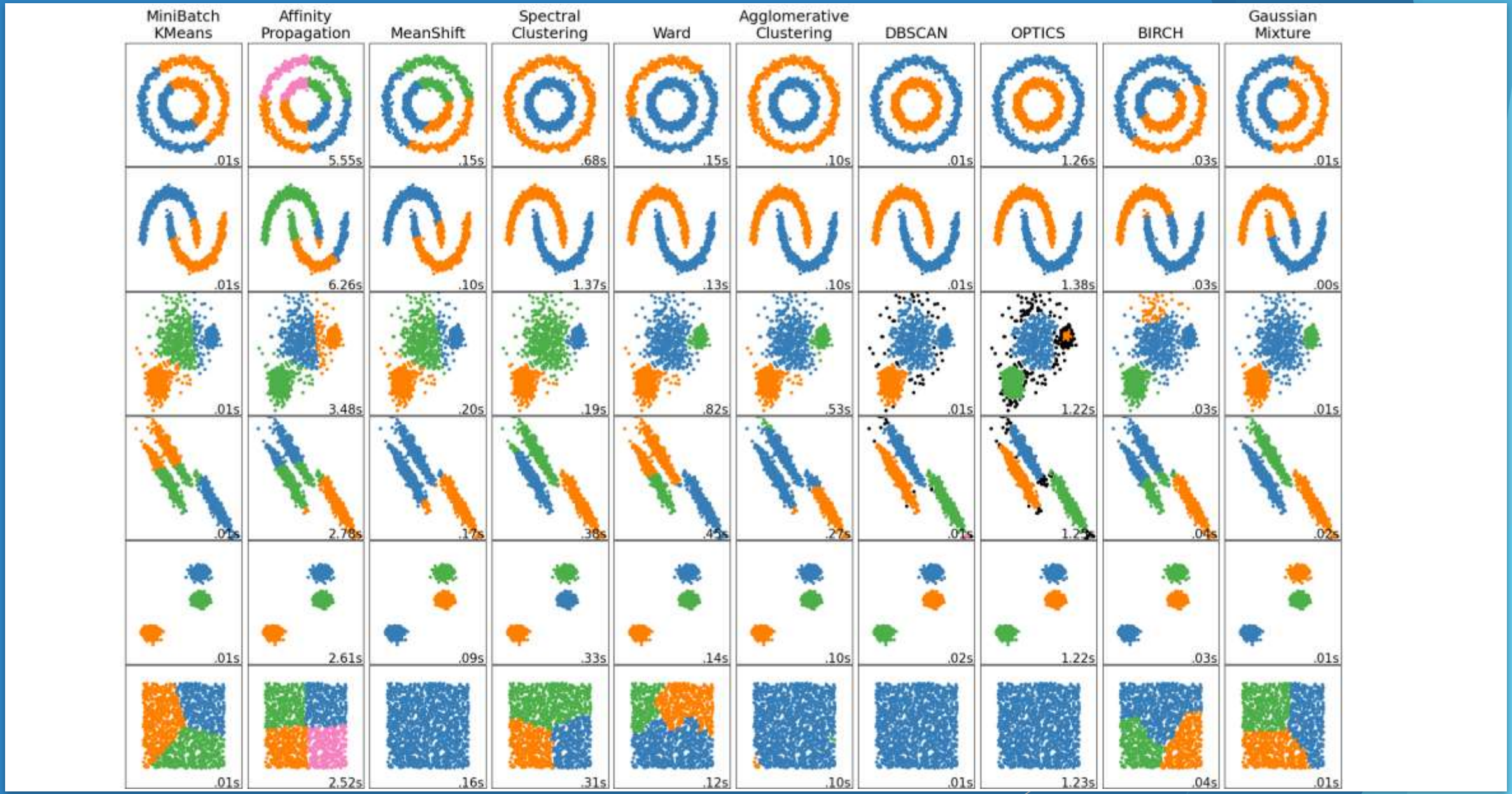
MinPts=5



https://www.researchgate.net/publication/318099337_Clustering_validation

Идентификация кластеров DBSCAN





https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

Даны флаги стран (всех) 

1. Нарисовать их на плоскости, предварительно понизив размерность до 2х.
2. Сгруппировать с помощью методов кластеризации в группы

Визуально оценить какие группы получились, сделать выводы

Домашнее задание

Размеры картинок:

82*100

143*100

180*90

x3 RGB

Различных цветов: 42661

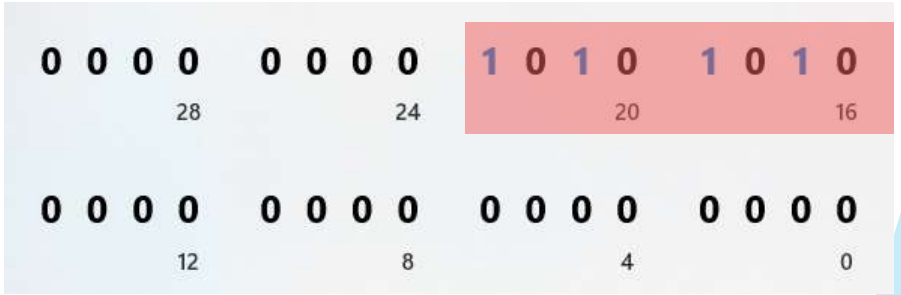
Каким образом сделать описания так, чтобы выровнять количество признаков?

Данные

Преобразование цвета

- Всего комбинаций 256*256*256=**16 777 216**
- Вариант: каждый компонент делить на какую-то величину, например на 85 (очень грубое округление, считаем эти цвета за один). Всего цветов 4*4*4=**64**
- Переводим в одно число. **RGBint = (red<<16) + (green<<8) + blue**

[254, 38, 0]
[254, 84, 0]
[254, 84, 84]
[170, 38, 0]
[170, 84, 0]
[170, 84, 84]



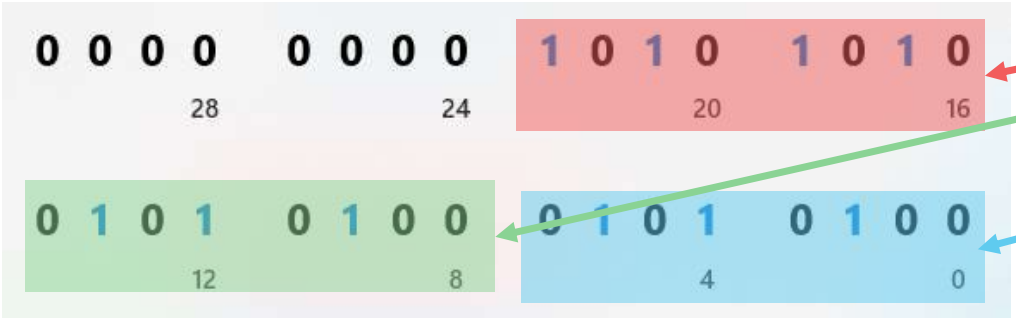
170

170 << 16 = 11 141 120

84 << 8 = 21504

84

11 162 708



Описание цветов флага

- Отбираем только те цвета, которые используются более чем 2% от общей площади флага
- Преобразовываем цвет в словарь:

ключ - цвет (как на предыдущем слайде);

значение – процент заполнения (от общей площади)

{170: 0.6609550561797752, синий
11184810: 0.027303370786516856, остальное
16777215: 0.2825842696629213} белый

```
1 fileDics
```

```
{'100px-Flag_of_Switzerland.svg.png': {16777215: 0.1908, 16711680: 0.784},
 '100px-Flag_of_the_Vatican_City.svg.png': {16755200: 0.5007,
  16777215: 0.4061,
  11184810: 0.0386},
 '117px-Flag_of_Niger.svg.png': {16777215: 0.26153846153846155,
  11141120: 0.3795726495726496,
  43520: 0.33},
 '125px-Flag_of_Monaco.svg.png': {16777215: 0.5, 11141120: 0.5},
 '133px-Flag_of_Gabon.svg.png': {11184640: 0.32, 21930: 0.33, 21845: 0.33},
```

Переход от цветов к признакам

- Получили 27 уникальных цвета
- По аналогии с кодированием OneHot для категориальных признаков необходимо выполнить кодирование цветов флагов. В соответствующее поле цвета ставить процентное использования цвета в конкретном флаге (в формате сотых долей – как в словарях)

- Таблица признаков 206(кол-во флагов) * 27(цвета)

Домашнее задание

- Получить таблицу описания флагов
- Применить алгоритм понижения размерности с параметром n_components=2 (PCA или KernelPCA). При необходимости произвести подбор гиперпараметров
- После преобразования получим две координаты – возьмём их за (x,y) и отобразим на плоскости
- Описать полученные закономерности