

Санкт-Петербургский Политехнический университет имени Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

ЛАБОРАТОРНАЯ РАБОТА №3
«КЛАСТЕРИЗАЦИЯ»

по дисциплине «Статистическое моделирование случайных процессов и систем»

Выполнил студент гр. 3530904/70103

Русаков Е.С.

Преподаватель

Селин И.

Санкт-Петербург
2020

Оглавление

Задание.....	3
1. Метод k средних.....	4
2. Сравнение методов кластеризации.....	5
1) Датасет clustering_1.csv.....	5
2) Датасет clustering_2.csv.....	6
3) Датасет clustering_3.csv.....	7
Вывод.....	8
3. Сжатие изображения.....	8
4. Анализ распределения голосов.....	10
Приложение.....	12
1. Метод k средних.....	12
2. Сравнение методов кластеризации.....	12
3. Сжатие изображений.....	13
4. Анализ распределения голосов.....	14

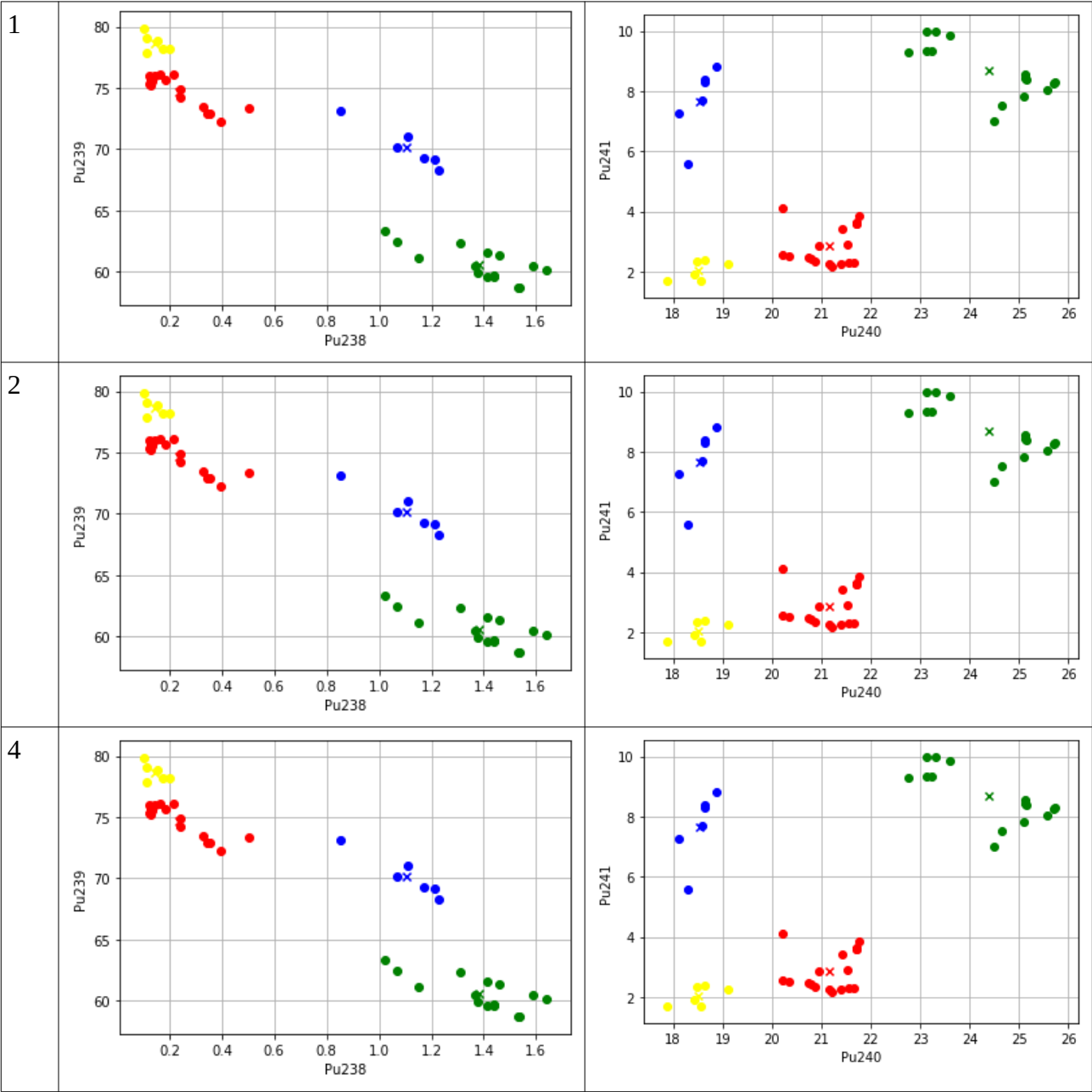
Задание

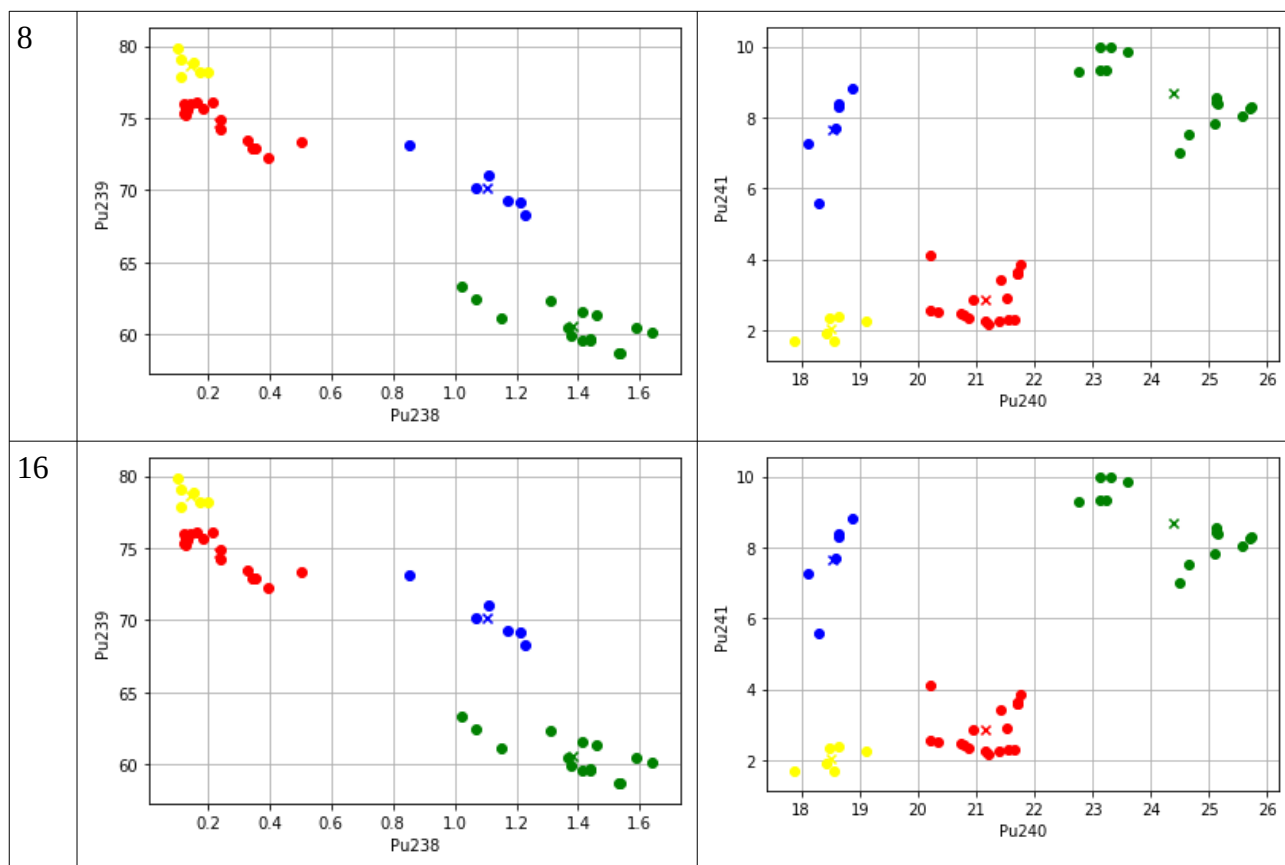
1. Разбейте множество объектов из набора данных `pluton.csv` на 3 кластера с помощью `k-means`. Сравните качество разбиения в зависимости от максимального числа итераций алгоритма и использования стандартизации.
2. Разбейте на кластеры множество объектов из наборов данных `clustering_1.csv`, `clustering_2.csv` и `clustering_3.csv` с помощью `k-means`, `DBSCAN` и иерархической кластеризации. Определите оптимальное количество кластеров (где это применимо). Какой из методов сработал лучше и почему?
3. Осуществите сжатие цветовой палитры изображения (любого, на ваш выбор). Для этого выделите n кластеров из цветов всех пикселей изображения и зафиксируйте центра этих кластеров. Создайте изображение с цветами из сокращенной палитры (цвета пикселей только из центров выделенных кластеров). Покажите исходное и сжатое изображения.
4. Постройте дендрограмму для набора данных `votes.csv` (число голосов, поданных за республиканцев на выборах с 1856 по 1976 год). Строки представляют 50 штатов, а столбцы - годы выборов (31). Проинтерпретируйте полученный результат.

Ход работы

1. Метод k средних

Будем искать кластеры в выборке pluton.csv. Кластеризацию будем проводить методами класса `k_means`, поставляемого библиотекой `Sklearn.cluster`. Результаты работы алгоритма в зависимости от числа итераций представлена в таблице. На каждое испытание приводится по два графика (две проекции), поскольку данные расположены в четырёхмерном пространстве. Экземпляры помечены разными цветами соответственно найденным кластерам. Крестом отмечены центры обнаруженных кластеров.





Как видно, алгоритм со стандартными параметрами хорошо определяет центры кластеров уже с первой итерации. На последующих итерациях они не сдвигаются и никаких изменений в распределении элементов между кластерами не происходит.

Для объективной оценки качества кластеризации используем метрику `silhouette_score` из пакета `Sklearn`. На всех значениях числа итераций показатель остаётся всё тем же: 0.6732197311637, что говорит о том, что с первой же итерации кластеры найдены наилучшим образом.

2. Сравнение методов кластеризации

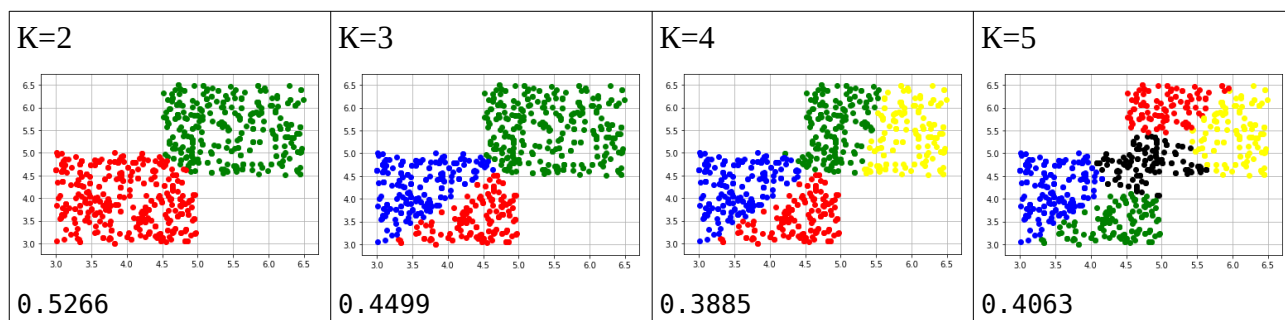
Будут сравниваться три метода кластеризации: *k* средних, плотностно-пространственный метод и иерархический метод на нескольких датасетах.

1) Датасет `clustering_1.csv`

Поскольку данные представлены в двумерном пространстве, результаты кластеризации можно легко визуализировать.

Метод k средних

В таблице представлена визуализация разбиения данных на *K* классов и значение метрики `silhouette_score`.



Из таблицы можно заключить, что самым успешным есть разбиение на два кластера. Для больших количеств искомых кластеров значение метрики значительно меньше.

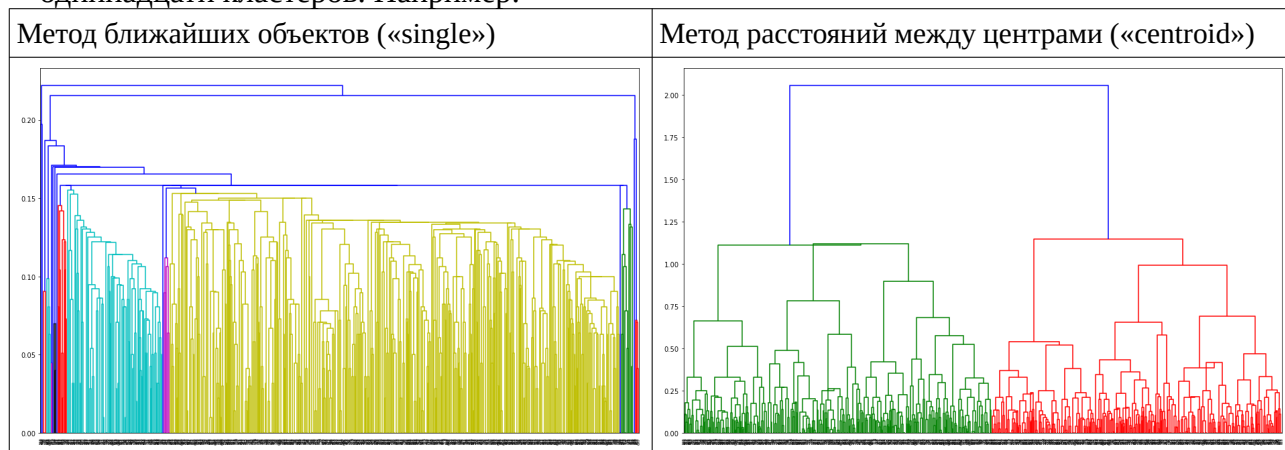
Плотностно-пространственный метод

Процедура DBSCAN из библиотеки Sklearn, реализующая этот метод, не смогла найти каких-либо кластеров в предложенной выборке, объединив все данные в единственный кластер.

Иерархическая кластеризация

Для реализации и построения дендрограмм использовались использовались методы linkage dendrogram из библиотеки scipy.cluster.hierarchy.

В зависимости от метода определения близости объектов алгоритм выделяет от двух до одиннадцати кластеров. Например:



Заключение

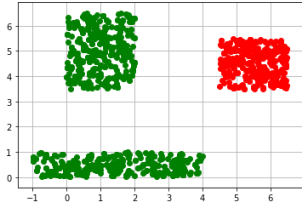
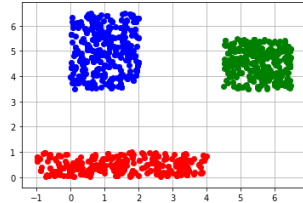
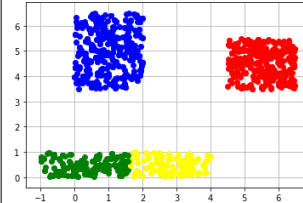
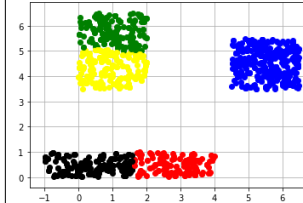
В данном случае нельзя всецело опереться на один из методов. Благо, данные позволяют себя визуализировать и подключить человеческое мышление, говорящее, что хоть в данных и отсутствуют явные кластеры (что и заключил метод DBSCAN), но из геометрических соображений их можно разделить на две группы, как это сделал метод k средних.

2) Датасет clustering_2.csv

Данные этого датасета также расположены в двухмерном пространстве, что позволяет пользоваться визуализацией для правильного заключения.

Метод k средних

Проведём аналогичные предыдущему разу исследования и поместим результаты в таблицу:

K=2	K=3	K=4	K=5
			
0.5305	0.7066	0.6868	0.5885

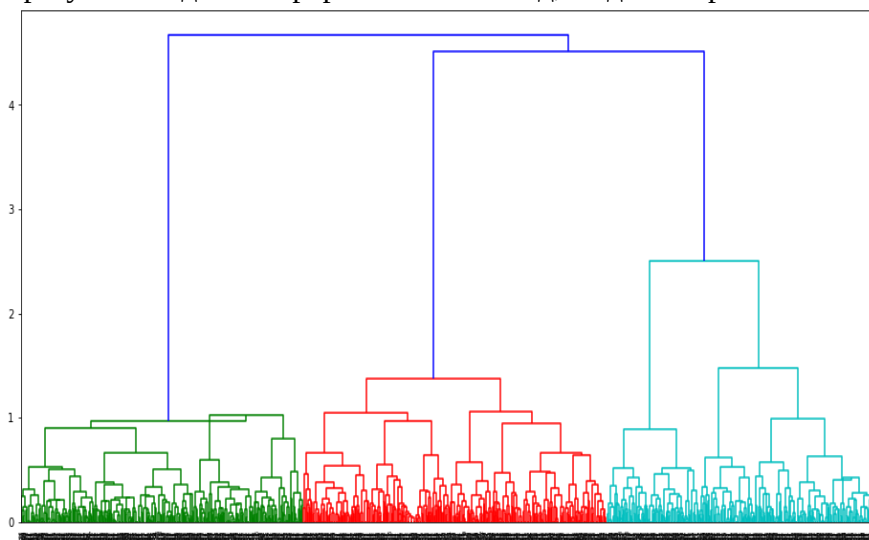
Наибольшие значения метрики наблюдаются при K=3, следовательно, по методу K средних оптимальным будет деление предложенных данных на три кластера.

Плотностно-пространственный метод

Процедура плотностно-пространственного анализа обнаружила наличие трёх кластеров аналогично методу K средних при K=3.

Иерархический метод

Аналогичные результаты дал и иерархический метод, выделив три явных кластера:



Заключение

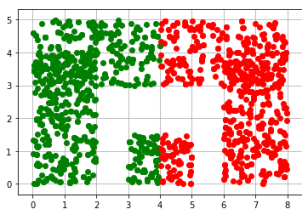
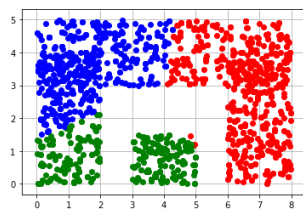
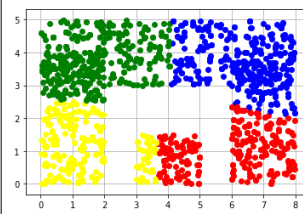
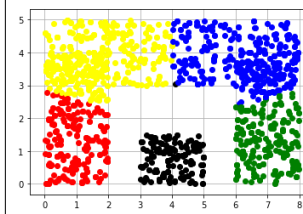
По результатам всех трёх методов можно заключить, что в датасете clustering_2.csv можно выделить три явных кластера.

3) Датасет clustering_3.csv

И этот датасет представляет собой двумерные данные.

Метод K средних

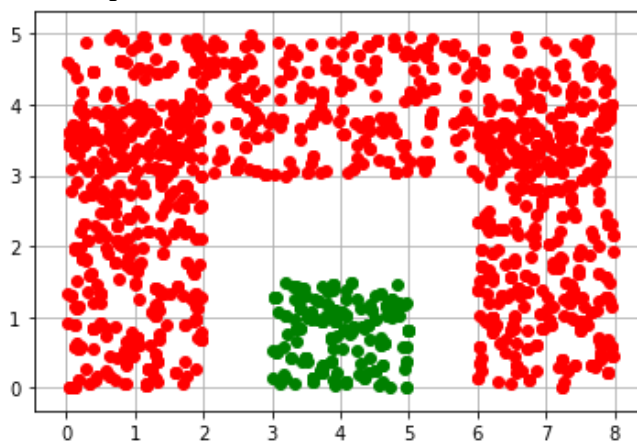
Результаты работы метода представлены в таблице

K=2	K=3	K=4	K=5
			
0.5118	0.4789	0.4324	0.4778

Как можно видеть, метод не справился со сложной геометрической формой объектов.

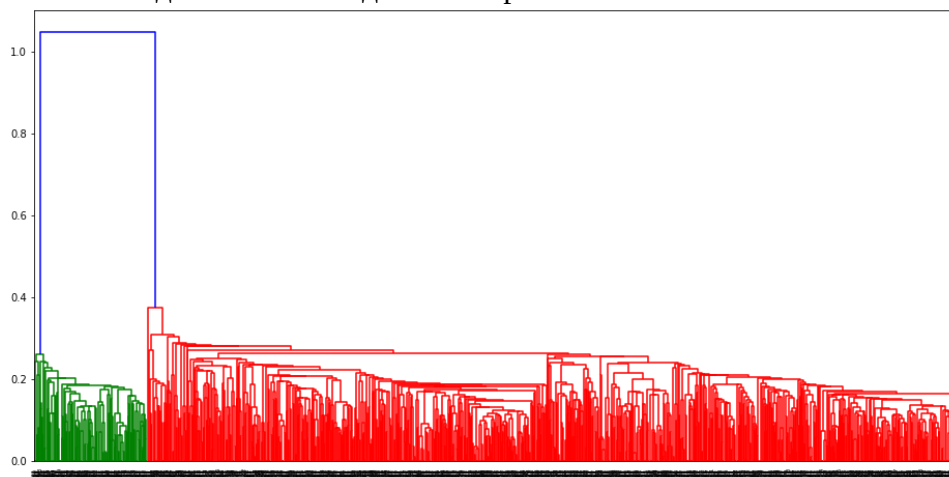
Метод плотностно-пространственной кластеризации

Метод обнаружил два кластера:



Иерархический метод

Иерархический метод также нашёл два кластера:



Заключение

Данный датасет насчитывает в себе два кластера.

Вывод

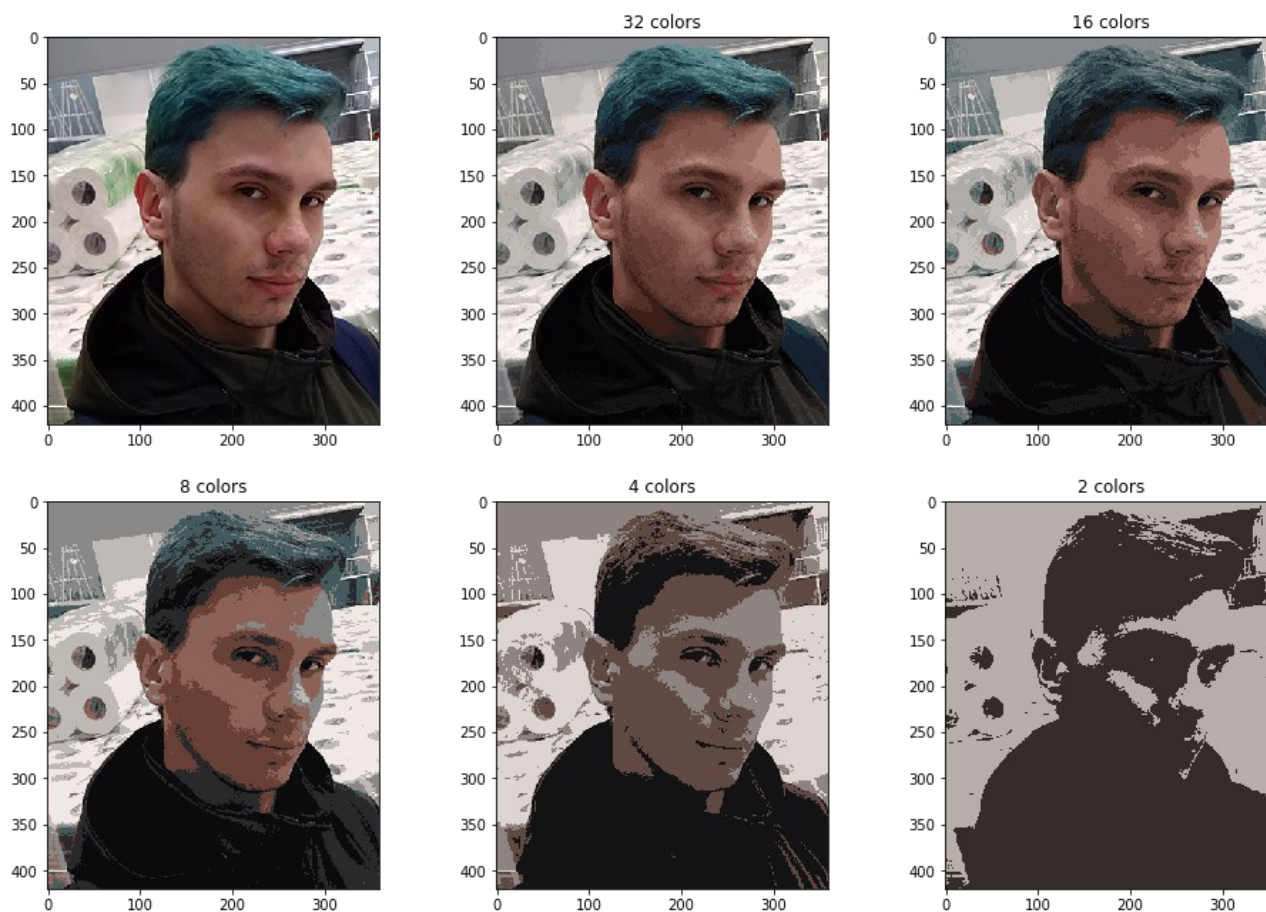
В случае, когда имеется хотя бы два кластера, все методы успешно его находят. Трудности возникают у метода К средних, когда кластер всего один, поскольку суть своей метод не предназначен для работы в таких условиях. Некластеризуемость можно выявить по отсутствию явного максимального значения метрики `silhouette_score` при каком-либо предполагаемом числе кластеров. Также с некластеризуемостью хорошо справляется плотностно-пространственный анализ.

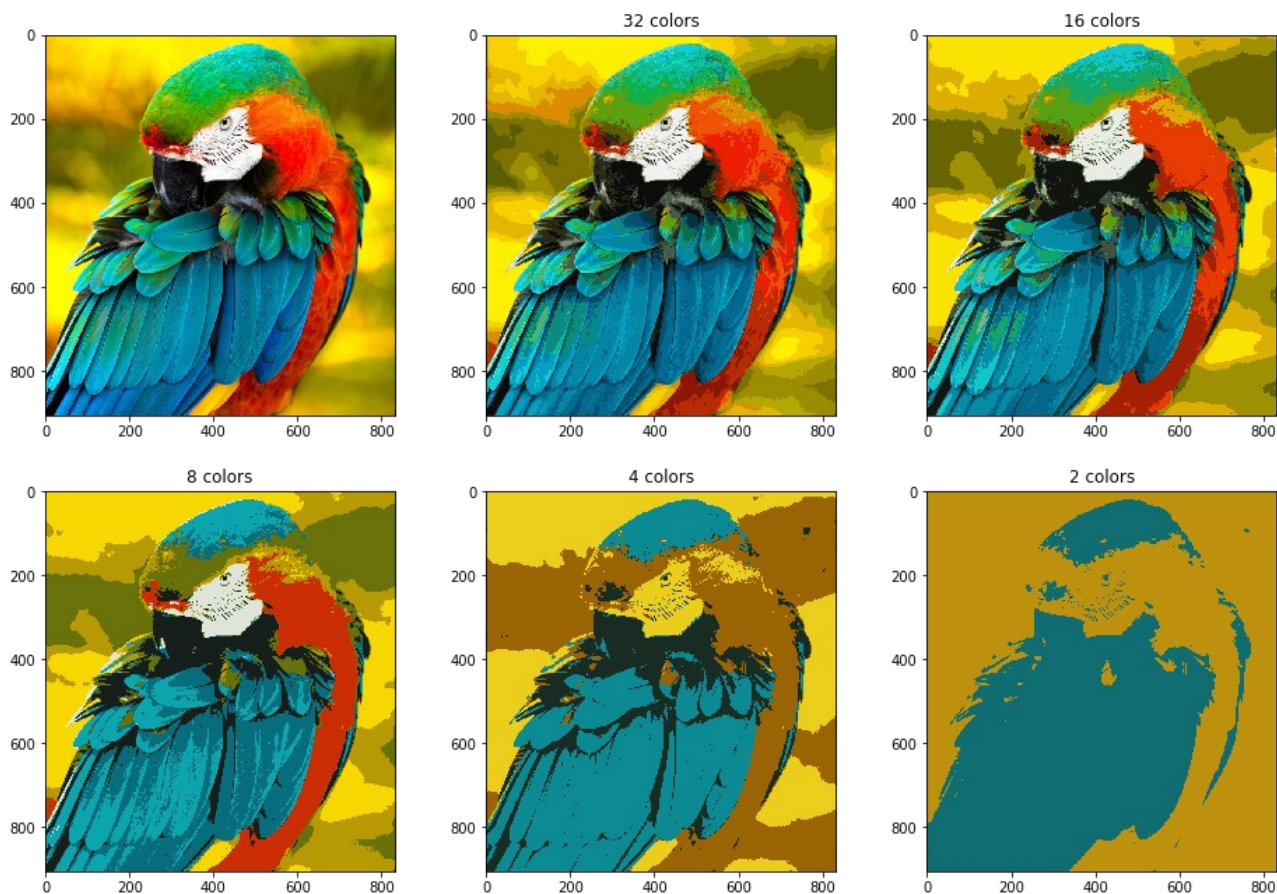
3. Сжатие изображения

Для эксперимента по сжатию картинки возьмём изображения случайного синеволосого парня и цветного попугая из интернета. Картинка в системе RGB представляет собой двумерный массив пикселей с тремя цветовыми компонентами, стало быть поиск кластеров пройдёт в трёхмерном пространстве, где каждая точка – пиксель исходной картинки.

С помощью метода К средних выделим N кластеров в трёхмерном пространстве цветов, а затем заменим исходные значения цветов координатами центров кластеров.

Результаты уменьшения палитры представлены ниже.

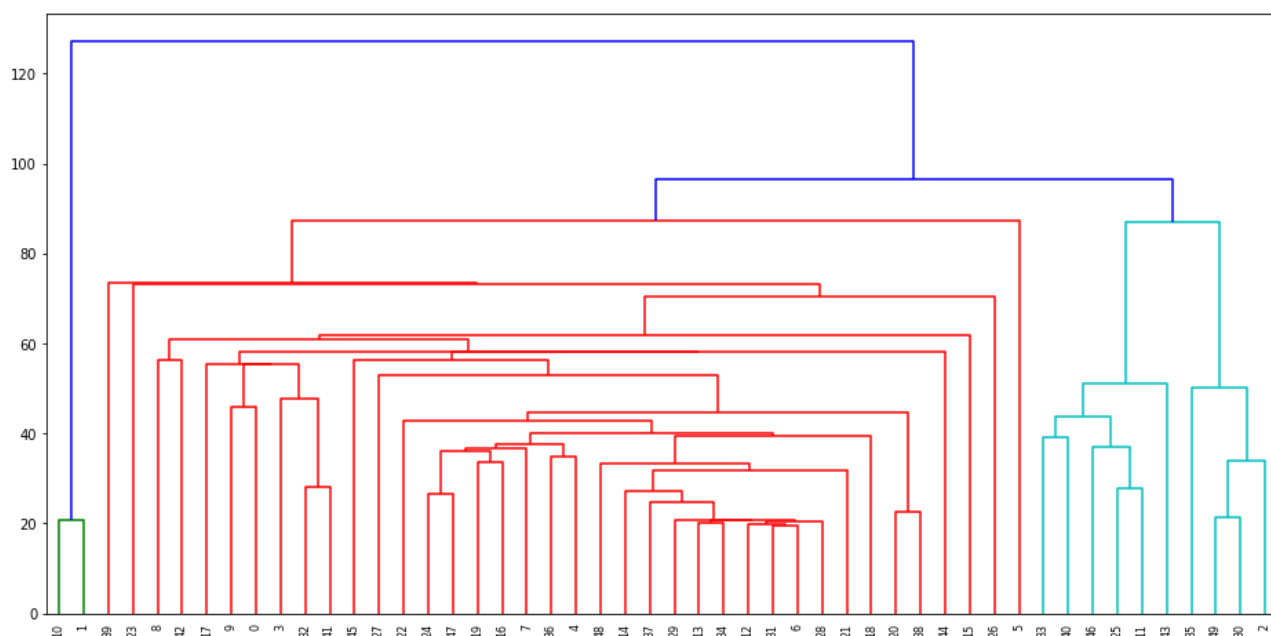




Как результат можно отметить, что при палитре в 32 цвета среднестатистическая картинка слабо отличима от оригинала. Разница становится явной на оригиналах с очень разнообразной цветовой палитрой. Картинка сохраняет узнаваемость и при восьми цветах, при четырёх становится похожа на произведение художника-абстракциониста, а при двух – на граффити.

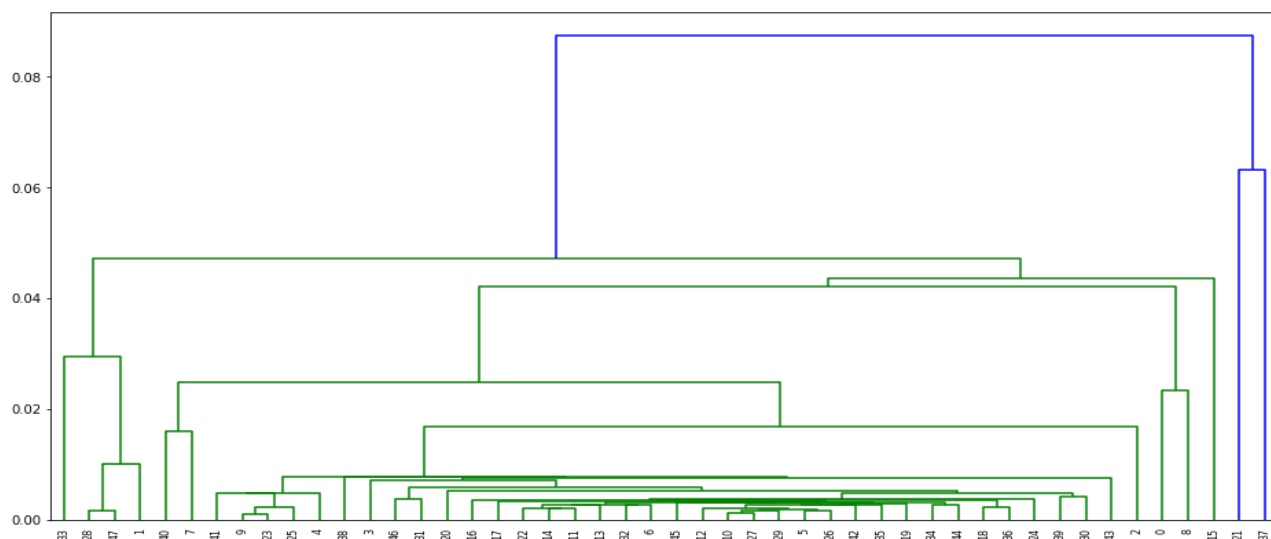
4. Анализ распределения голосов

Файл `votes.csv` содержит данные о проценте голосов отданных на Республиканскую партию США по штатам на протяжении более века. Некоторые данные отсутствуют и были заменены на нули. По всем данным построена дендрограмма, она приведена ниже:



Дендрограмма должна отражать схожесть в динамике предпочтений электората за период. Однако из-за большого количества изначально отсутствующих данных рисунок не отражает действительного положения вещей. Например, два левых штата схожи между собой только потому, что в них много нулей. Отсюда, для получения более точной картины потребуются поработать с данными, а именно: убрать сильно искажённые штаты и уменьшить период анализа, потому что в первые полвека подотчётного периода выпало много значений.

Теперь рассматриваются только выборы с 1900 года, кроме штатов №№1 и 10.



Хоть алгоритм выделил только два кластера, стоит проанализировать его вручную. Сразу заметны большое скопление похожих штатов в центре диаграммы: там динамика голосов избирателей очень схожа. Остальные штаты, судя по длине соединяющих их рёбер, так складно в группы не объединяются.

Приложение

1. Метод k средних

```
data = pd.read_csv('pluton.csv')

X = data.to_numpy()

# тест кластеризации
iterations = [1, 2, 4, 8, 16, 32]
colors = ['red', 'green', 'blue', 'yellow']
gridsize = (1, 2)

for itr in iterations:
    clusterizator = KMeans(n_clusters=4, max_iter=itr, n_jobs=4)
    clusterizator.fit(X)
    pred = clusterizator.predict(X)

    print(f'Silhouette metrics: {silhouette_score(X, pred)}')
    print(f'Silhouette samples: {silhouette_samples(X, pred)}')

    for i in range(len(X)):
        plt.scatter(x=X[i, 0], y=X[i, 1], color=colors[pred[i]])
    for idx ,center in enumerate(clusterizator.cluster_centers_):
        plt.scatter(center[0], center[1], color=colors[idx],
marker='x')
    plt.grid(True)
    #plt.title(f'K = {itr}')
    plt.xlabel('Pu238')
    plt.ylabel('Pu239')
    plt.show()

    for i in range(len(X)):
        plt.scatter(x=X[i, 2], y=X[i, 3], color=colors[pred[i]])
    for idx ,center in enumerate(clusterizator.cluster_centers_):
        plt.scatter(center[2], center[3], color=colors[idx],
marker='x')

    plt.grid(True)
    plt.xlabel('Pu240')
    plt.ylabel('Pu241')
    plt.show()
```

2. Сравнение методов кластеризации

```
def visualize_2dim_classes(X, clusters):
    colors = ['red', 'green', 'blue', 'yellow', 'black', '0.35', 'cyan',
'magenta', '0.65', '#AAAAFF']
    for idx in range(len(X)):
        plt.scatter(x=X[idx, 0], y=X[idx, 1], color=colors[clusters[idx]])
    plt.grid(True)
    plt.show()

def dbscan_analysis(X):
    clusterizer = DBSCAN(n_jobs=4)
```

```

    pred = clusterizer.fit_predict(X)
    if (len(np.unique(pred)) > 1):
        print(f'Обнаружено кластеров: {len(np.unique(pred))}\nSilhouette
metrics: {silhouette_score(X, pred)}')
        visualize_2dim_classes(X, pred)
    else:
        print('Обнаружен только один кластер')

# метод K-средних
def k_means_analysis(X):
    supposed_classes = [i for i in range(2, 11)]

    for cls in supposed_classes:
        clusterizer = KMeans(n_clusters=cls, n_jobs=4)
        pred = clusterizer.fit_predict(X)
        if (len(np.unique(pred)) > 1):
            print(f'K: {cls}, Silhouette metrics: {silhouette_score(X, pred)}')
            visualize_2dim_classes(X, pred)
        else:
            print('Обнаружен только один кластер')

data1 = pd.read_csv('clustering_1.csv', sep='\t', header=None)
data1
X = data1.to_numpy()
plt.scatter(x=X[:, 0], y=X[:, 1])
plt.grid(True)
plt.title('clustering_1.csv')
plt.show()

# метод K-средних
k_means_analysis(X)

# метод DBSCAN
dbscan_analysis(X)

# метод деревьев
mergins = linkage(X, method='centroid')
plt.figure(figsize=(16,10))
dendrogram(mergins)
plt.show()

# для остальных датасетов аналогично

```

3. Сжатие изображений

```

def img_cluster_compressor(img_path, num_clusters):
    old_img = plt.imread(img_path) / 255.
    old_shape = old_img.shape
    fig, ((ax1, ax2, ax3), (ax4, ax5, ax6)) = plt.subplots(2, 3, figsize=(16,
11))
    ax1.imshow(old_img)
    old_img = old_img.reshape(old_img.shape[0] * old_img.shape[1], 3)

    for ax in [ax2, ax3, ax4, ax5, ax6]:
        ax.set_title(f'{num_clusters} colors')
        new_img = np.zeros(old_img.shape)
        if (num_clusters < 2):

```

```

        colors = [np.mean(old_img[:, 0]), np.mean(old_img[:, 1]),
np.mean(old_img[:, 2])]
        preds = [0] * len(old_img)
    else:

        clusterizer = KMeans(n_jobs=4, n_clusters=num_clusters)
        preds = clusterizer.fit_predict(old_img)
        colors = clusterizer.cluster_centers_

    for idx in range(len(old_img)):
        new_img[idx] = colors[preds[idx]]
    new_img = new_img.reshape(old_shape)
    plt.imsave(img_path + f'_{num_clusters}_c.jpg',
np.uint8(np.around(new_img*255)))
    ax.imshow(new_img)
    num_clusters = int(np.floor(num_clusters / 2))

img_cluster_compressor('example_x60.jpg', 32)
img_cluster_compressor('popug.jpg', 32)

```

4. Анализ распределения голосов

```

data3 = pd.read_csv('votes.csv')
data3 = data3.fillna(0.)

X = data3.to_numpy()

from scipy.cluster.hierarchy import linkage, dendrogram

mergins = linkage(X, optimal_ordering=True, method='single')

plt.figure(figsize=(16, 8))
dendrogram(mergins)

plt.show()

data3.drop(labels=['X1856', 'X1860', 'X1864', 'X1868', 'X1872', 'X1876',
'X1880', 'X1884', 'X1888', 'X1892'],
        inplace=True, axis=1)

data3.drop(labels=[1, 10], inplace=True, axis=0)
# избавляемся от лишних данных и строим новую дендрограмму
X2 = data3.to_numpy()
mergins = linkage(X2, optimal_ordering=True, method='single', metric='cosine')

plt.figure(figsize=(16, 8))
dendrogram(mergins)
plt.show()

```