МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Машинное обучение»

Тема: Кластеризация (k-средних, иерархическая)

Студент гр. 6307 _____ Новиков Б.М.

Преподаватель _____ Жангиров Т.Р.

2020

ЗАГРУЗКА ДАННЫХ

1. Загрузка данных в датафрейм

Выведем 5 первых строк:

	0	1	2	3	4
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

K-MEANS

1. Проведем кластеризацию методом k-средних

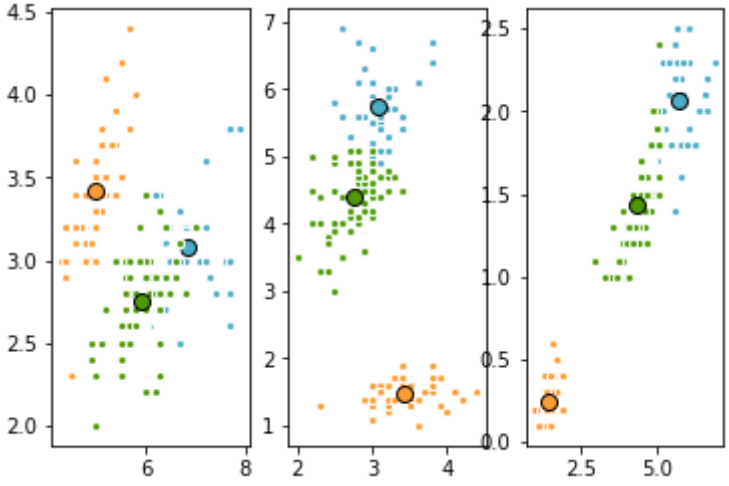
k_means = KMeans(init='k-means++', n_clusters=3, n_init=15)
k_means.fit(no_labeled_data)

2. Получим центры кластеров и определим наблюдения в какой кластер попали

k_means_cluster_centers = k_means.cluster_centers_
k_means_labels = pairwise_distances_argmin(no_labeled_data, k_means_cluster_centers)

```
[[6.85       3.07368421 5.74210526 2.07105263]
 [5.006      3.418      1.464      0.244     ]
 [5.9016129  2.7483871  4.39354839 1.43387097]]

[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 0 0 0 0 2 0 0 0 0
 0 0 2 2 0 0 0 0 2 0 2 0 2 0 0 2 2 0 0 0 0 2 0 0 0 2 0 0 0 2 0 0 0 2 0 0
 0 2]
```

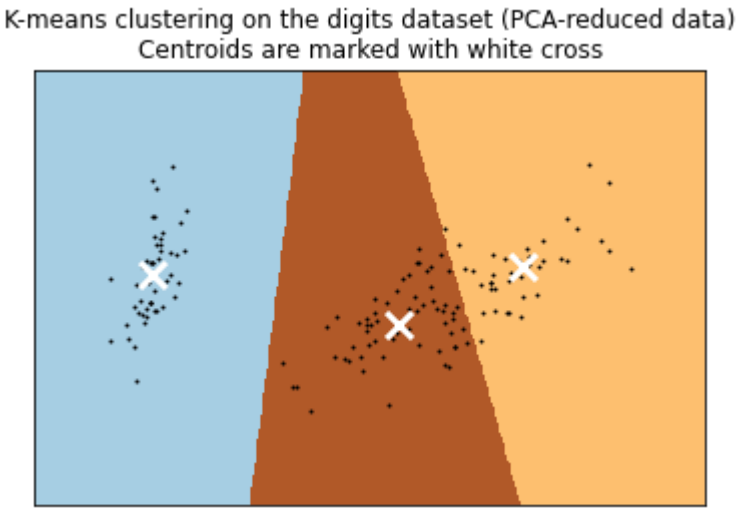
3. Построим результаты классификации для признаков попарно



Как влияет значение параметра n_init?
Количество запусков алгоритма с разными сидами для центроида.

4. Уменьшить размерность данных до 2, используя метод главных компонент и нарисуйте карту для всей области значений, на которой каждый кластер занимает определенную область со своим цветом.

```
reduced_data = PCA(n_components=2).fit_transform(no_labeled_data)

kmeans = KMeans(init='k-means++', n_clusters=3, n_init=15)
kmeans.fit(reduced_data)
```



5. Исследуем работу алгоритма k-средних при различных параметрах init.

```
k_means = KMeans(init='k-means++', n_clusters=10)
k_means.fit(no_labeled_data)
```

При параметре init k-means++ k_means.inertia_:

```
26.171482323232333
```

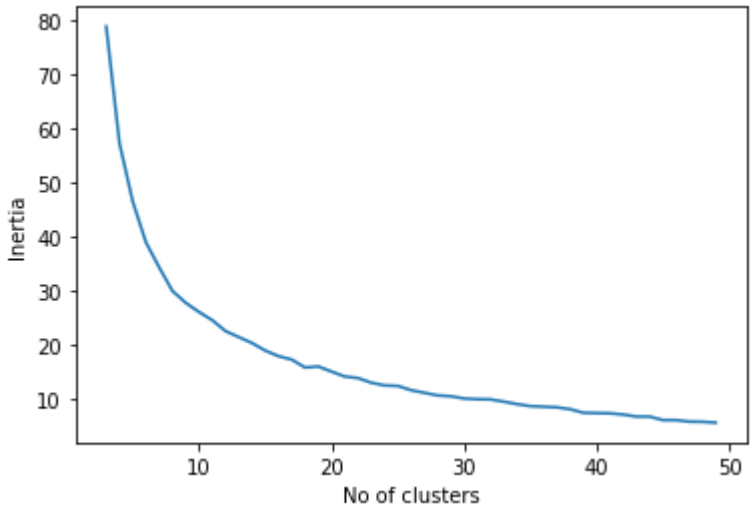
```
for i in range(3):
    k_means = KMeans(init='random', n_clusters=10)
    k_means.fit(no_labeled_data)
```

При параметре init random k_means.inertia_ (3 запуска):

```
26.382150394101185
26.93803227732022
26.37322510822512
```

6. Определить наилучшее количество методом локтя.

```
inertia_values = []
range_of_n = range(3, 50)

for n in range_of_n:
    k_means = KMeans(init='k-means++', n_clusters=n)
    k_means.fit(no_labeled_data)
    inertia_values.append(k_means.inertia_)

plt.xlabel("No of clusters")
plt.ylabel("Inertia")
plt.plot(range_of_n, inertia_values)

plt.show()
```

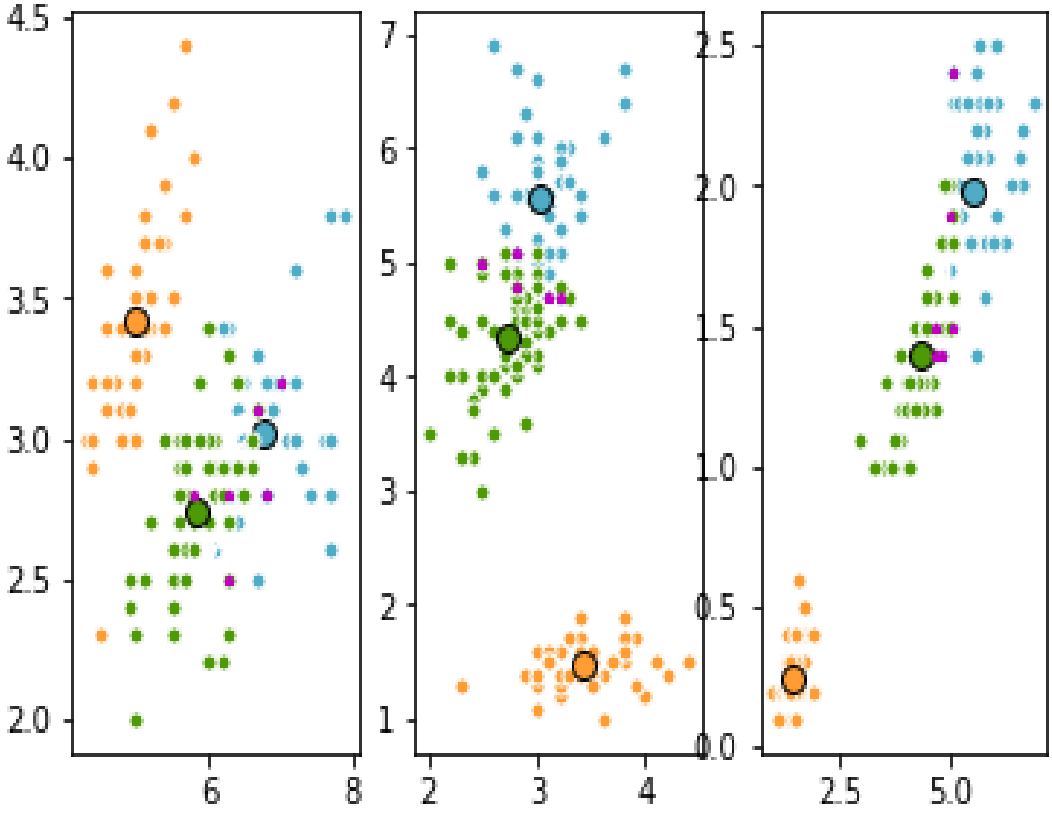


Как видно по графику, 12 кластеров — наилучшее количество

7. Проведем кластеризацию, используя пакетную кластеризацию k-средних. В чем отличие от обычного метода k-средних. Постройте диаграмму рассеяния, на которой будут выделены точки, которые для разных методов попали в разные кластеры.

```
mb_k_means = MiniBatchKMeans(init='k-means++', n_clusters=3, n_init=15)
mb_k_means.fit(no_labeled_data)

mb_k_means_cluster_centers = mb_k_means.cluster_centers_
mb_k_means_labels = pairwise_distances_argmin(no_labeled_data, mb_k_means_cluster_centers)
```

```
[[5.86983409 2.72767722 4.40082956 1.44426848]
 [5.0146789  3.42220183 1.47422018 0.24963303]
 [6.85803357 3.08884892 5.71822542 2.0618705 ]]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2
 2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 2 0 2 2 2 0 2 2 2 0 2
 2 0]
```



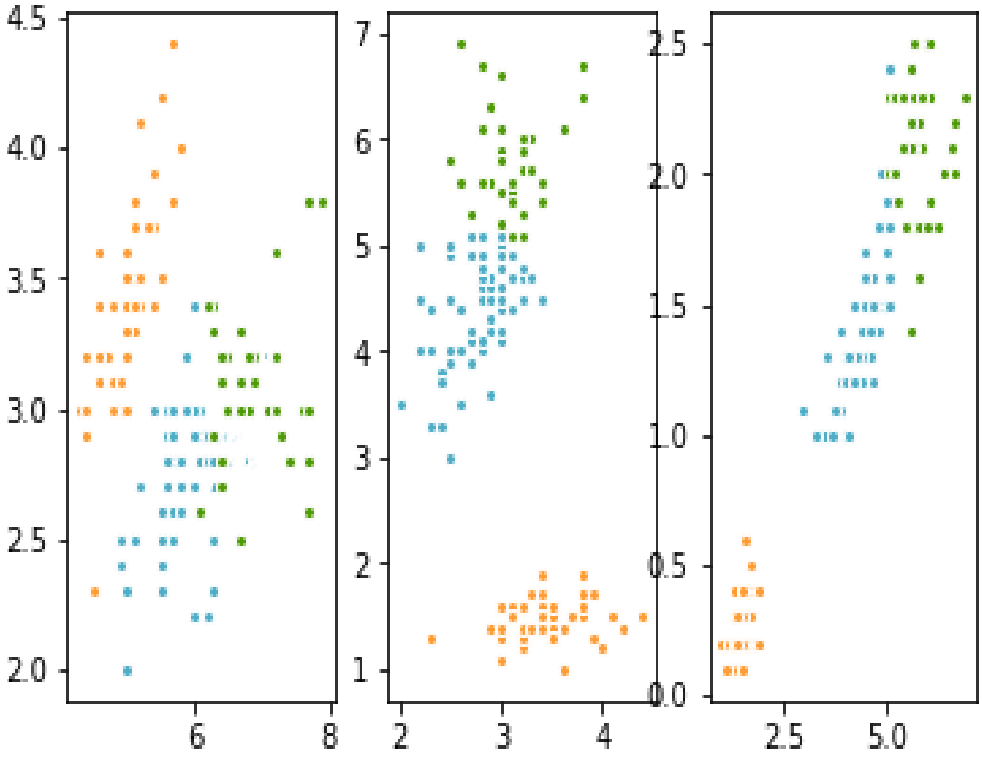
На данном графике точки, которые попали в другой кластер, выделены цветом magenta.

ИЕРАРХИЧЕСКАЯ КЛАСТЕРИЗАЦИЯ

1. Проведем иерархическую кластеризацию на тех же данных

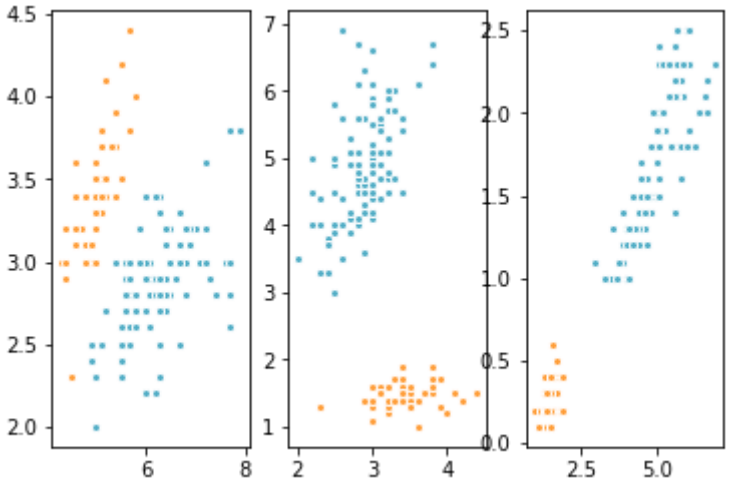
```
hier = AgglomerativeClustering(n_clusters=3, linkage='average')
hier = hier.fit(no_labeled_data)
hier_labels = hier.labels_
```

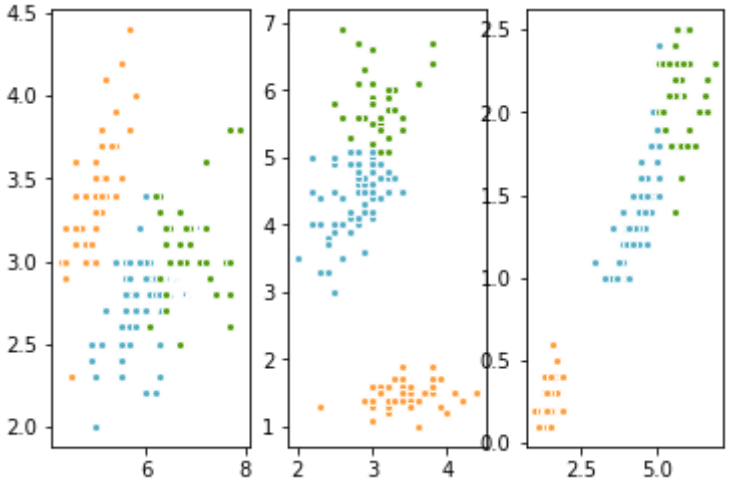
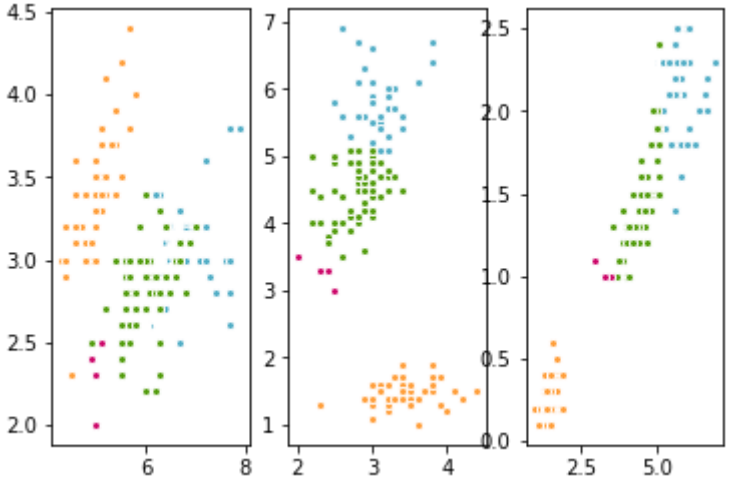
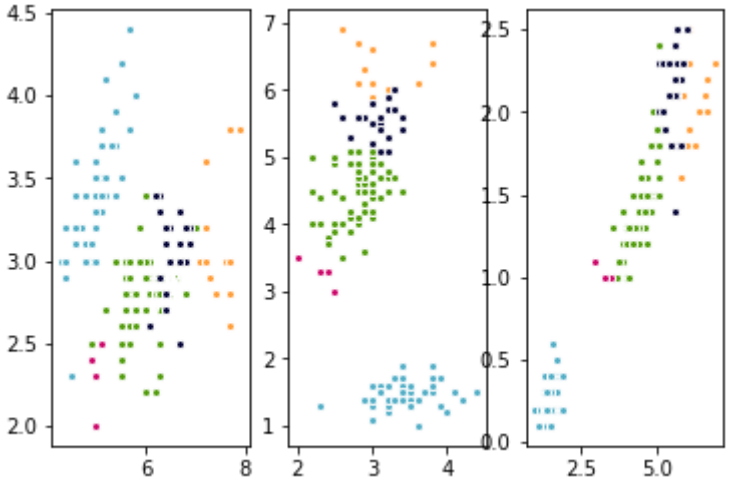
2. Отобразим результаты кластеризации



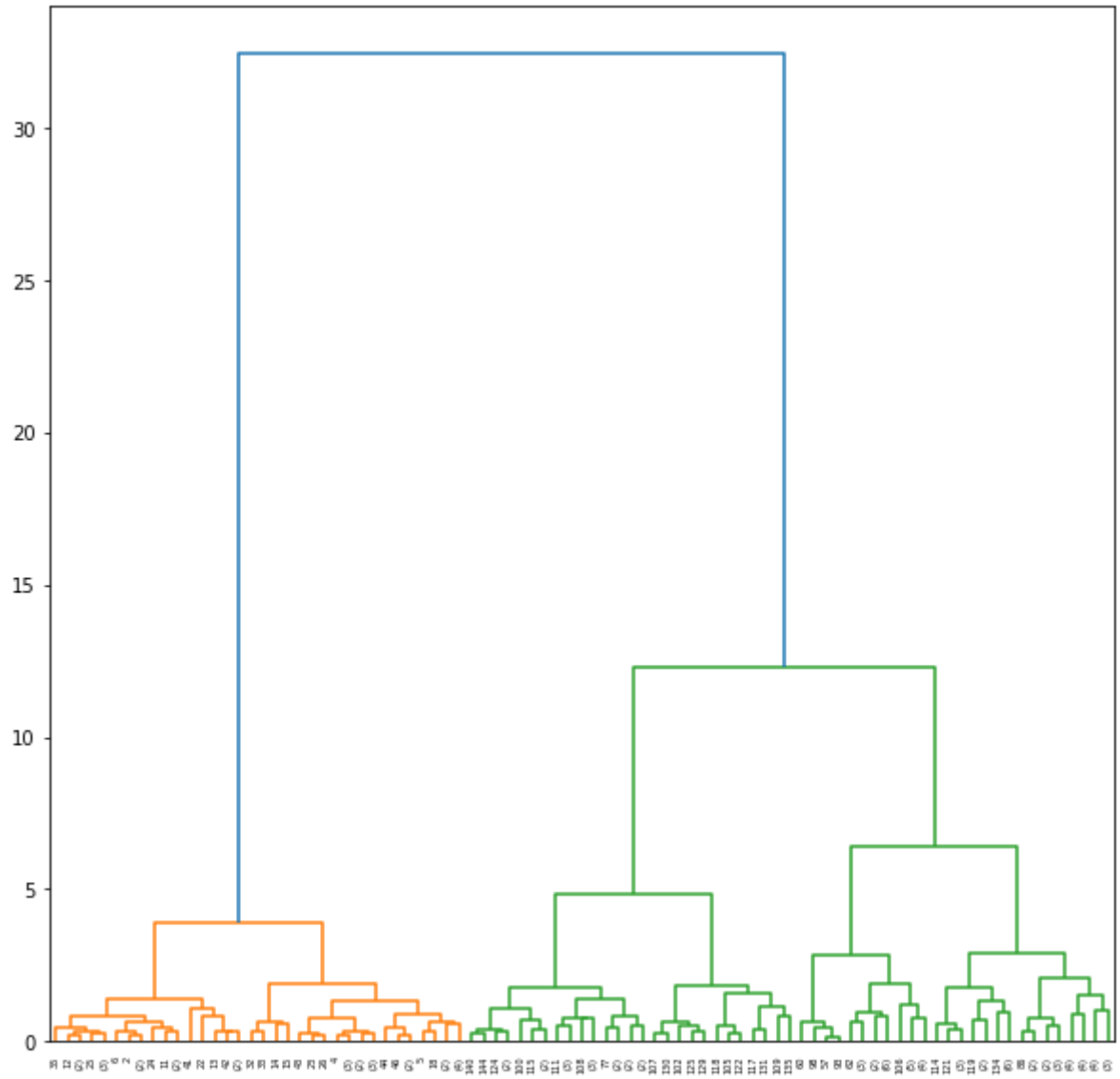
В чем отличие от метода k-средних?

3. Проведем исследование для различного размера кластеров (2-5)



4. Нарисуем дендограмму до уровня 6



5. Сгенерируем случайные данные в виде двух колец

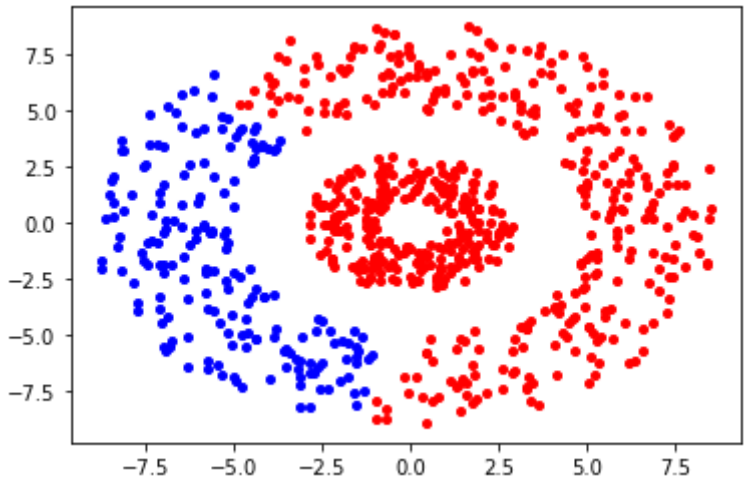
```
data1 = np.zeros([250,2])

for i in range(250):
    r = random.uniform(1, 3)
    a = random.uniform(0, 2*math.pi)
    data1[i,0] = r*math.sin(a)
    data1[i,1] = r*math.cos(a)
    data2 = np.zeros([500,2])

    for i in range(500):
        r = random.uniform(5, 9)
        a = random.uniform(0, 2*math.pi)
        data2[i,0] = r*math.sin(a)
        data2[i,1] = r*math.cos(a)
```

```
data = np.vstack((data1, data2))
```

6. Проведем иерархическую кластеризацию

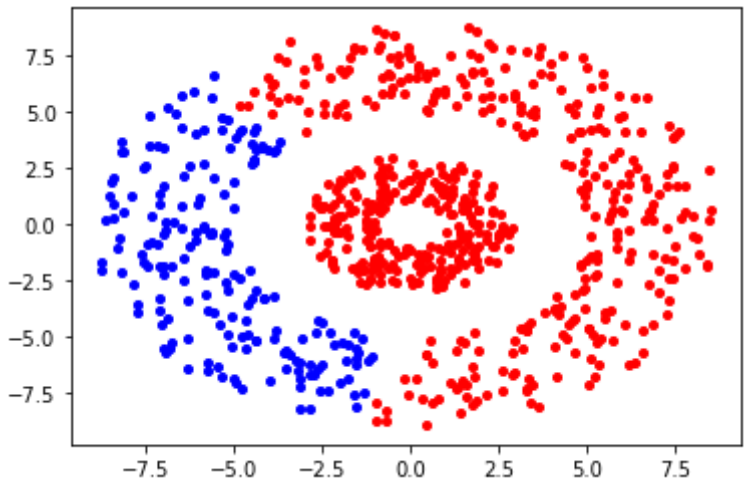
```
hier = AgglomerativeClustering(n_clusters=2, linkage='ward')
hier = hier.fit(data)
hier_labels = hier.labels_
```

7. Выведем полученные результаты

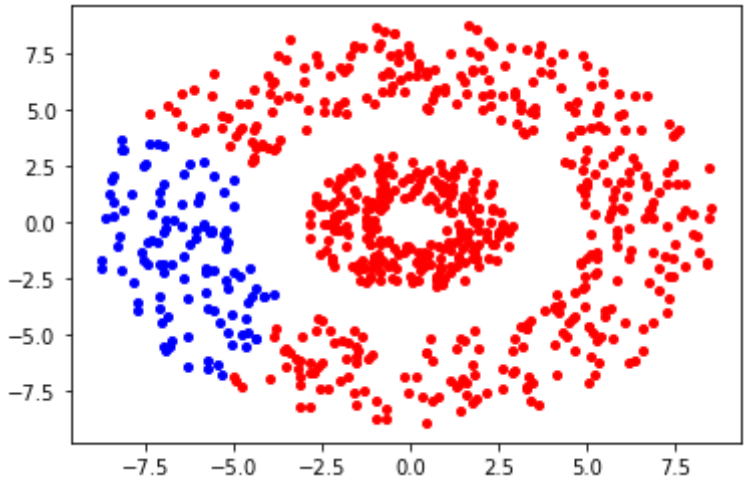


8. Исследуем кластеризацию при всех параметрах linkage. Отобразим и обоснуем полученные рез-ты. Для каких случаев какой тип связи будет работать лучше всего.

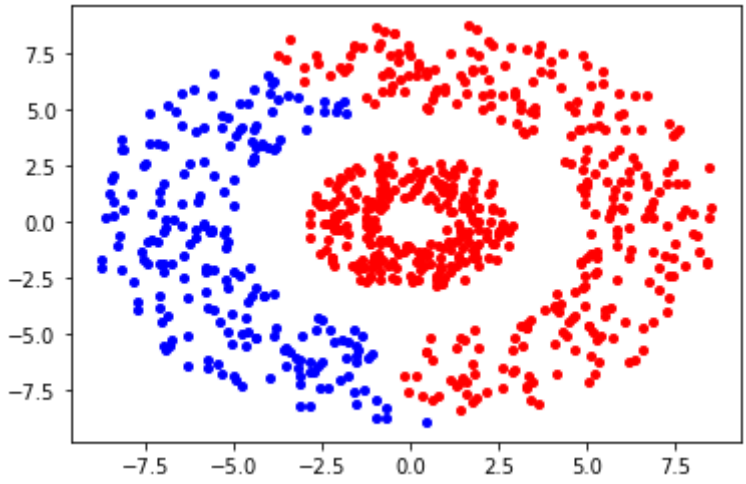
Параметры: 'ward', 'complete', 'average', 'single'



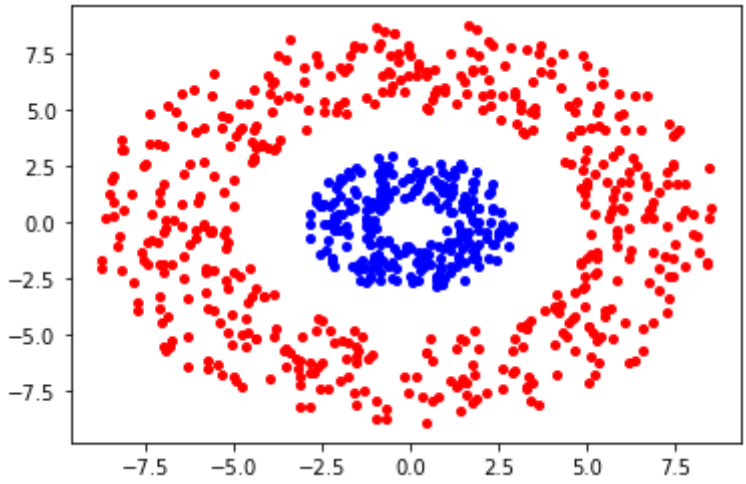
минимизирует дисперсию объединенных кластеров



средняя дистанция каждого наблюдения множества



максимальная дистанция между всеми наблюдениями двух множества



минимальная дистанция между всеми наблюдениями двух множеств