Университет ИТМО

Факультет безопасности информационных технологий

Дисциплина:

«Современные инструменты анализа данных»

Отчет по лабораторной работе №1

«Кластеризация»

Выполнили:
Пастухова А.А. гр. N3253
Мариненков М. Д. гр. N3252

Проверила:
Гусарова Н. Ф.

Санкт-Петербург

2021г.

Задание:

1. Используйте метод K-средних и метод DBSCAN на самостоятельно сгенерированной выборке с количеством кластеров не менее 4. Для увеличения числа кластеров при генерации можно задать количество центров в функции make_blobs через параметр centers.
2. Используйте эти же два метода на датасете Mall_Customers.
3. Для каждого метода необходимо построить график.

Основные шаги по выполнению лабораторной работы

1. Импортируем необходимые библиотеки

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
%matplotlib inline
plt.style.use('ggplot')
plt.rcParams['figure.figsize']=(12,8)
```

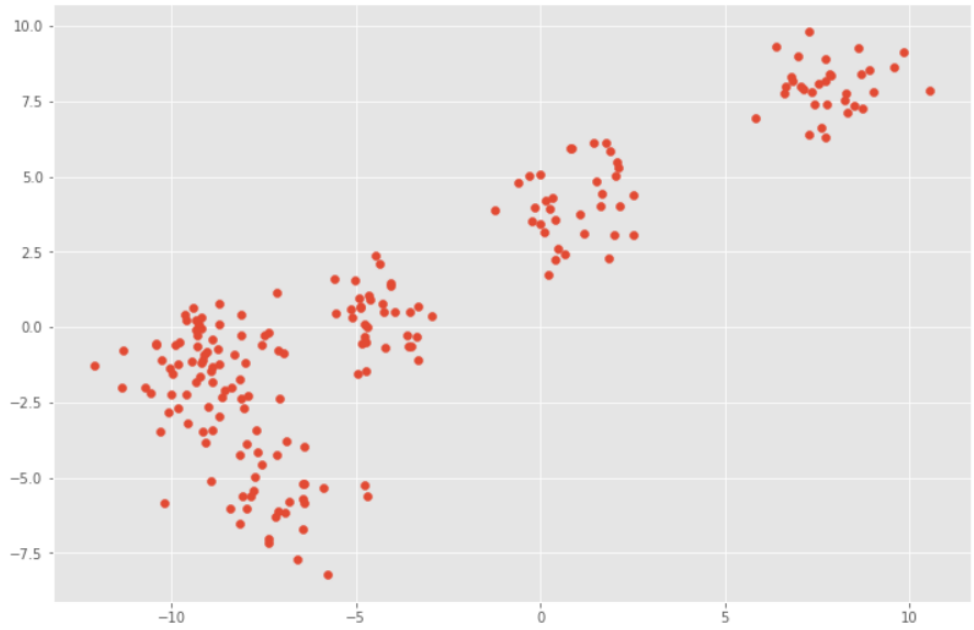
Воспользуемся библиотекой sklearn, чтобы сгенерировать "игрушечные" данные. Мы сгенерируем 200 объектов, имеющих 2 признака и разделенные на 5 кластеров. Кроме того, мы зададим значение random_state = 3 (цифра может быть любая), для повторяемости результатов

```
from sklearn.datasets import make_blobs
X,y = make_blobs(n_samples=200, random_state=3, centers=6)
```

Визуализируем данные, которые мы сгенерировали. Для визуализации в примере используется библиотека matplotlib. Можете использовать библиотеку plotly, которая строит интерактивные графики

```
plt.scatter(X[:,0], X[:,1])
```

```
<matplotlib.collections.PathCollection at 0x7ff80f984450>
```



Метод локтя

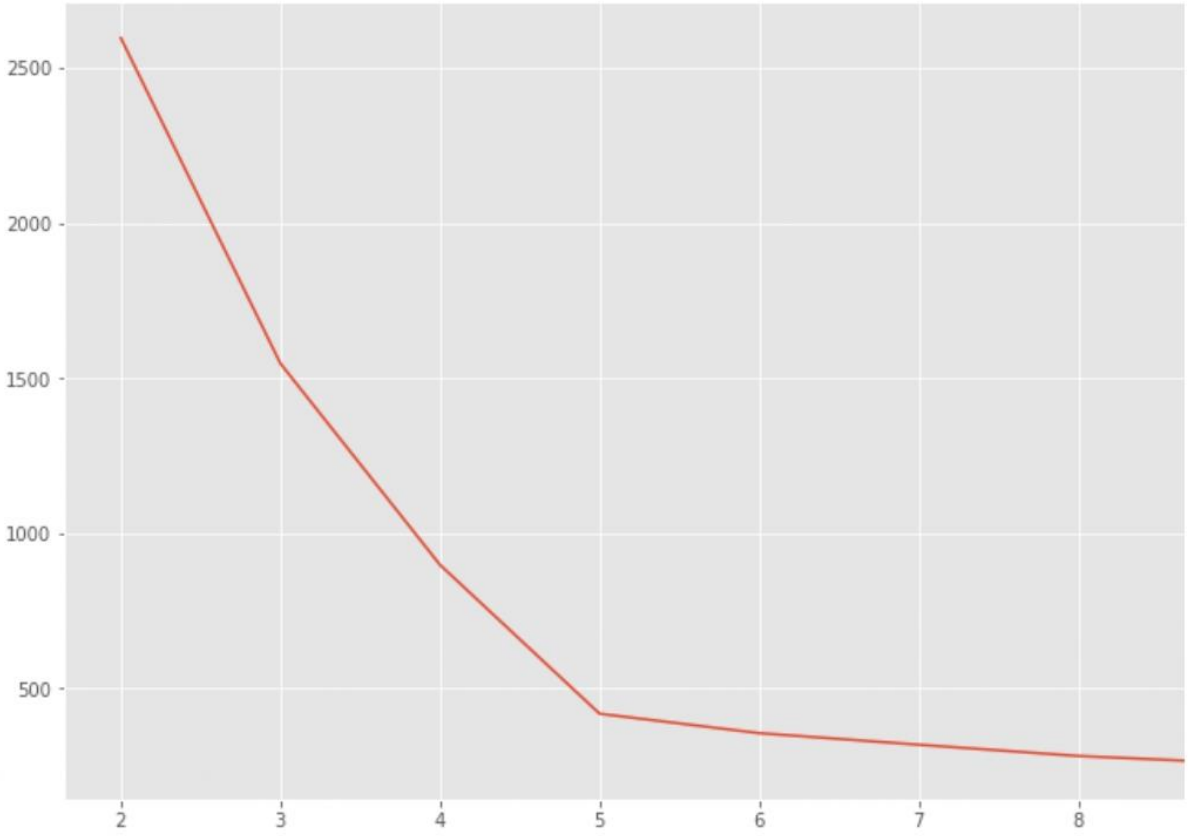
```
criteries = []
for k in range(2,10):
  kmeansModel=KMeans(n_clusters=k, random_state=3)
  kmeansModel.fit(X)
  criteries.append(kmeansModel.inertia_)
```

```
print(criteries)
```

```
[2597.6264043162364, 1548.2554957816237, 899.0997326465606, 419.1399294145061, 356.5
```

```
plt.plot(range(2,10), criteries)
```

```
[<matplotlib.lines.Line2D at 0x7ff806c7fe10>]
```




```
kmeansModel = KMeans(n_clusters=5,random_state=0)
```

Обучим модель

```
kmeansModel.fit(X)
```

```
KMeans(n_clusters=5, random_state=0)
```

После обучения мы можем получить метки кластеров, взяв атрибут класса KMeans под названием labels_

```
labels = kmeansModel.labels_
```

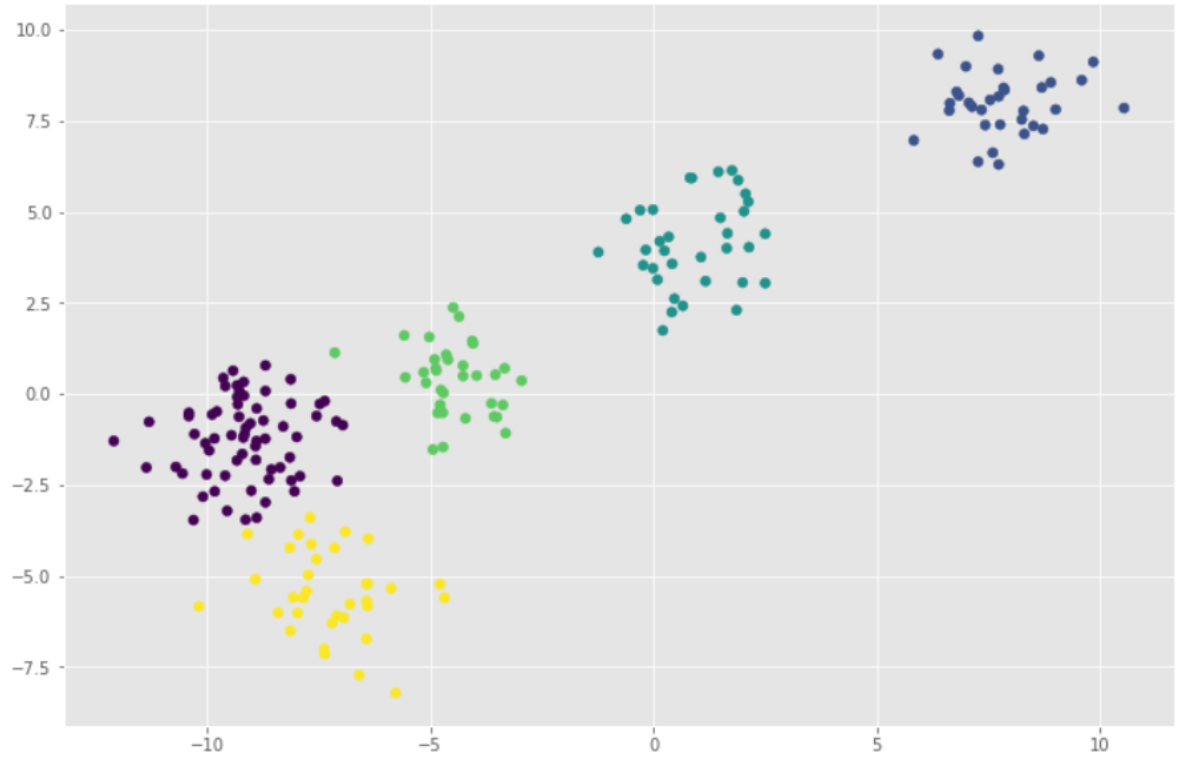
```
print(labels)
```

```
[3 2 4 1 4 1 0 0 3 0 2 2 0 3 0 0 0 0 1 2 3 0 1 1 3 4 0 0 0 0 3 4 2 0 2 1 2
 3 2 0 0 1 3 2 2 3 2 4 0 1 1 1 0 1 2 1 0 1 2 3 1 2 1 0 3 1 4 0 1 1 2 2 0 2
 1 0 2 3 3 0 3 2 0 3 3 0 3 0 4 4 2 1 0 4 0 1 4 0 4 4 4 3 0 0 0 1 3 1 0 1 4
 4 0 0 0 2 2 4 2 2 3 4 0 1 4 3 4 0 0 2 2 0 1 4 4 3 2 0 4 1 4 3 0 4 0 0 0 4
 0 3 0 2 3 0 3 0 0 2 0 0 3 1 1 3 2 4 0 2 4 0 3 4 0 0 1 4 0 0 0 3 0 4 3 2 0
 4 4 3 0 0 1 0 3 1 1 3 2 4 4 2]
```

Визуализируем полученные результаты, добавив в функцию scatter массив с метками классов

```
plt.scatter(X[:,0], X[:,1], c=labels)
```

```
<matplotlib.collections.PathCollection at 0x7ff80f984350>
```

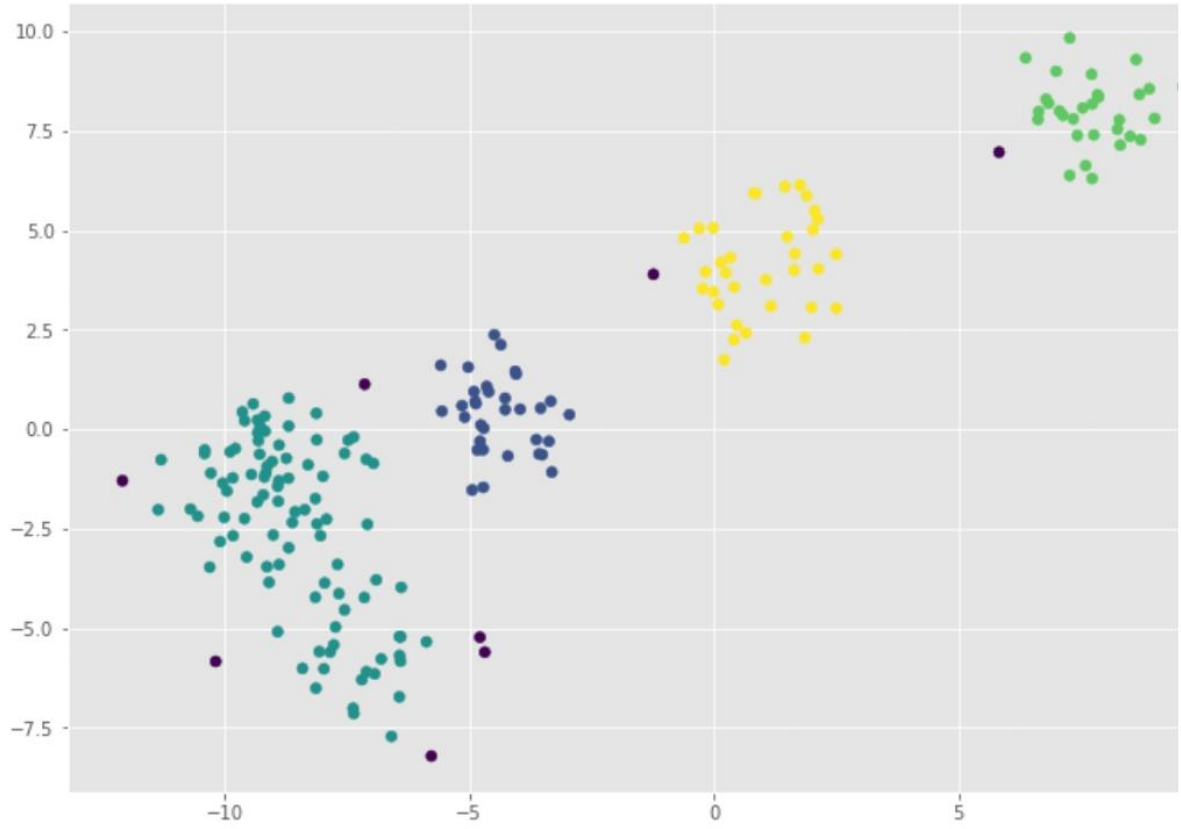


а теперь сделаем кластеризацию с помощью метода DBSCAN

```
from sklearn.cluster import DBSCAN

clustering = DBSCAN(eps=1, min_samples=5).fit_predict(X)
print(clustering)
plt.scatter(X[:,0], X[:,1], c=clustering);
```

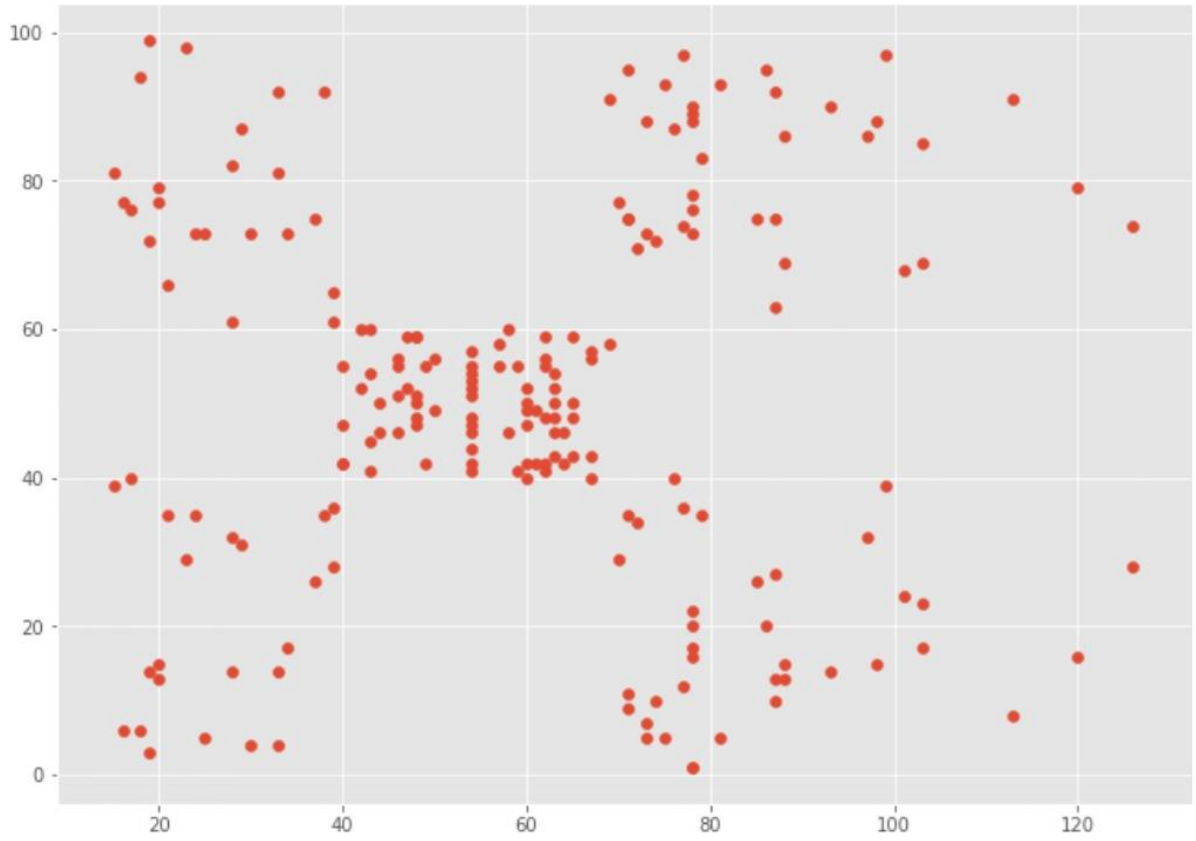
```
[ 0  3  1  2  1  2  1  1  0  1  3 -1  1  0  1  1  1  1  2  3  0  1  2  2
  0  1  1  1  1  1  0  1  3  1  3 -1  3  0  3  1  1  2  0  3  3  0  3  1
  1  2  2  2  1  2  3  2  1  2  3  0  2  3  2  1  0  2  1  1  2  2  3  3
  1  3  2  1  3  0  0  1  0  3  1  0  0  1  0  1  1  1  3  2  1 -1  1  2
  1  1  1  1  1  0  1  1  1  2  0  2  1  2  1  1  1  1  1  3  3  1  3  3
  0  1  1  2  1  0  1  1  1  3  3 -1  2  1  1  0  3  1  1  2 -1  0  1 -1
  1  1  1 -1  1  0  1  3  0  1  0  1  1  3  1  1  0  2  2  0  3  1  1  3
  1  1  0  1  1  1  2  1  1  1  1  0  1  1  0  3  1  1  1  0  1  1  2  1
  0  2 -1 -1  3  1  1  3]
```




```
data = pd.read_csv("Mall_Customers.csv")
```

Используем годовой доход и оценку трат

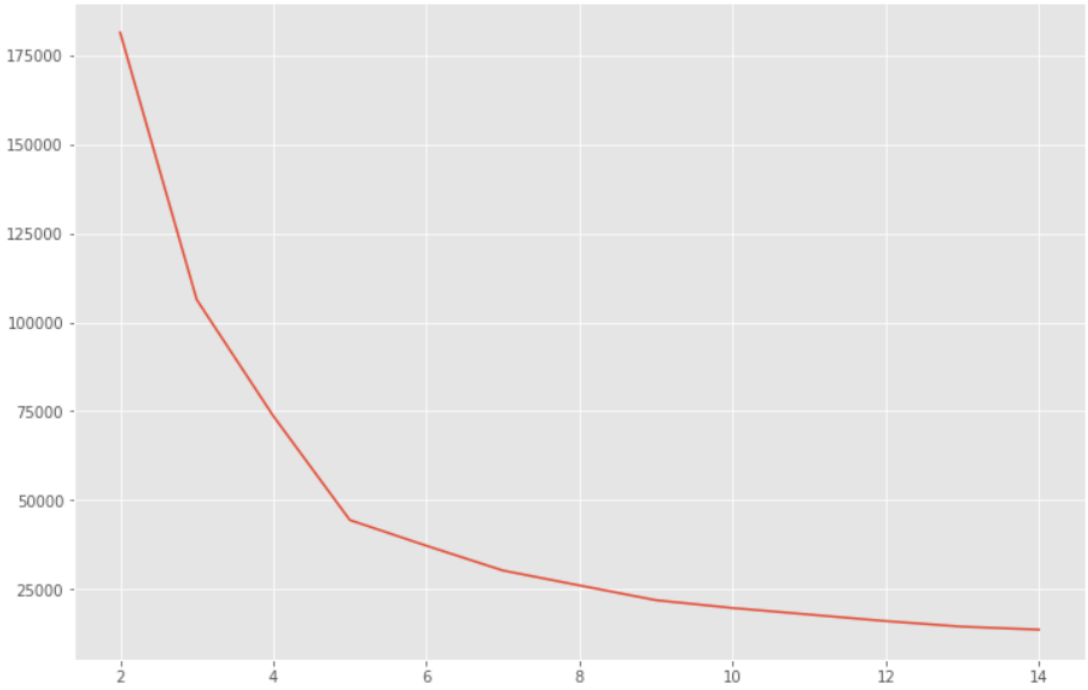
```
X = data[['Annual Income (k$)', 'Spending Score (1-100)']].iloc[: , :].values
plt.scatter(X[:,0], X[:,1]);
```



Используем метод локтя, чтобы найти количество кластеров

```
criteries = []
for k in range(2,15):
  kmeansModel=KMeans(n_clusters=k, random_state=3)
  kmeansModel.fit(X)
  criteries.append(kmeansModel.inertia_)
plt.plot(range(2,15), criteries)
```

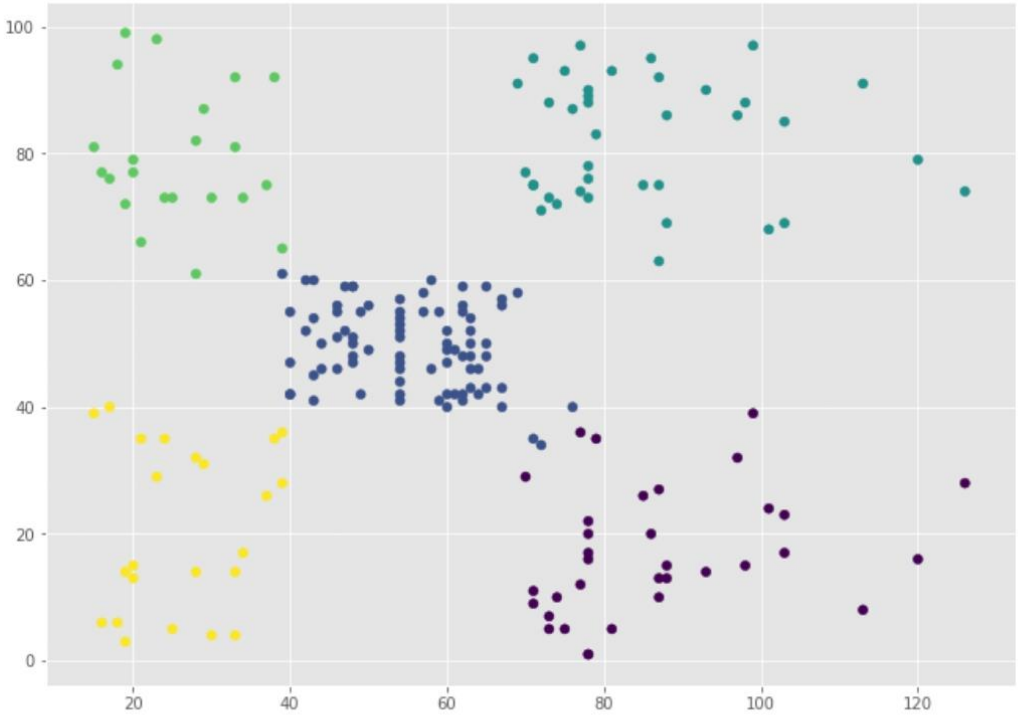
```
[<matplotlib.lines.Line2D at 0x7ff806a9d390>]
```



По графику k = 5, кластеризуем

```
kmeansModel=KMeans(n_clusters=5, random_state=0)
kmeansModel.fit(X)
labels = kmeansModel.labels_
plt.scatter(X[:,0], X[:,1], c=labels)
```

```
<matplotlib.collections.PathCollection at 0x7ff806b27490>
```



Кластеризуем при помощи DBSCAN

```
clustering = DBSCAN(eps=10, min_samples=7).fit_predict(X)
print(clustering)
plt.scatter(X[:,0], X[:,1], c=clustering);
```

```
[ 2  1  0  1  2  1  0 -1  0  1  0 -1  0  1  0  1  2  1  2 -1  2  1  0  1
  0  1  2 -1  2  1  0  1  0 -1  0  1  0  1  2  1  2 -1  2  2  2  1  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  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
  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  2  2  3  2  3  2  3  4  3  4  3  2  3  4  3  4  3  4  3  4  3  2  3
  4  3  2  3  4  3  4  3  4  3  4  3  4  3  4  3  2  3  4  3  4  3  4  3
  4 -1  4  3  4  3  4  3  4  3  4  3 -1  3  4  3 -1  3 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1]
```