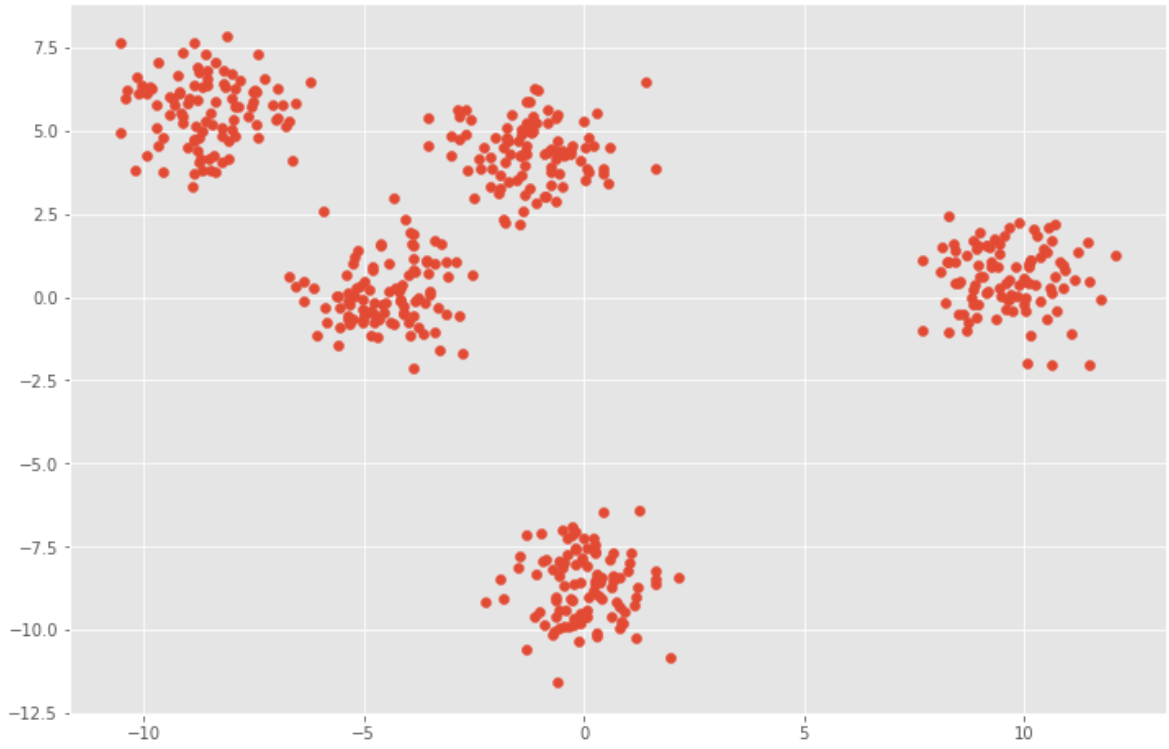
# Лабораторная работа №4

Для первого пункта нам нужно создать свой датасет.

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

from sklearn.datasets import make_blobs
X,y = make_blobs(n_samples=500, random_state=7, centers = 5)

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




Используя размер кластеров 2:

```
from sklearn.cluster import KMeans
kmeansModel = KMeans(n_clusters=2) # мы пока не знаем количество кластеров
plt.scatter(X[:,0], X[:,1], c=kmeansModel.labels_) # визуализируем данные
```

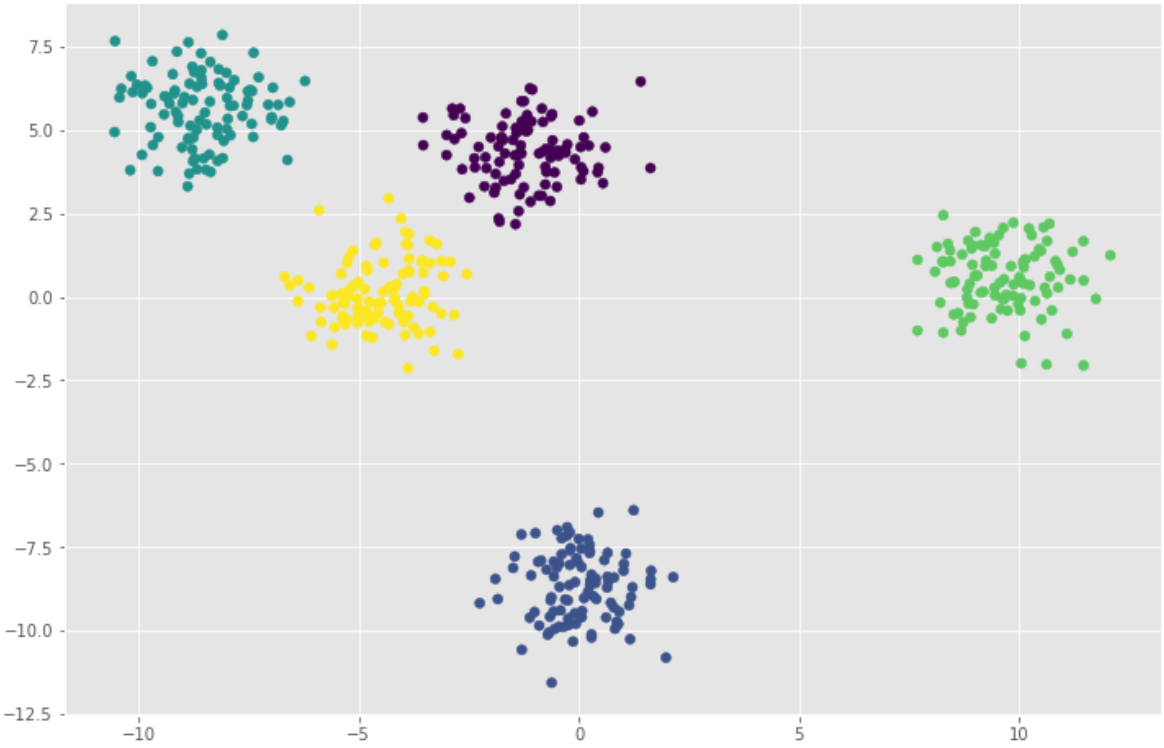
Мы получим не совсем корректную кластеризацию, в том смысле, что, на самом деле, у нас больше кластеров, чем 2:

Мы разумеется, могли бы на глаз определить количество кластеров из графика, но лучше использовать более объективный метод, как метод Локтя:

```
criteries = []
for k in range(2,10):
  kmeansModel=KMeans(n_clusters=k, random_state=7)
  kmeansModel.fit(X)
  criteries.append(kmeansModel.inertia_)
plt.plot(range(2,10), criteries)
```

Как мы видим производная ближе к нулю сильнее с точки 5. Таким образом, оптимальное значение кластеров - это 5

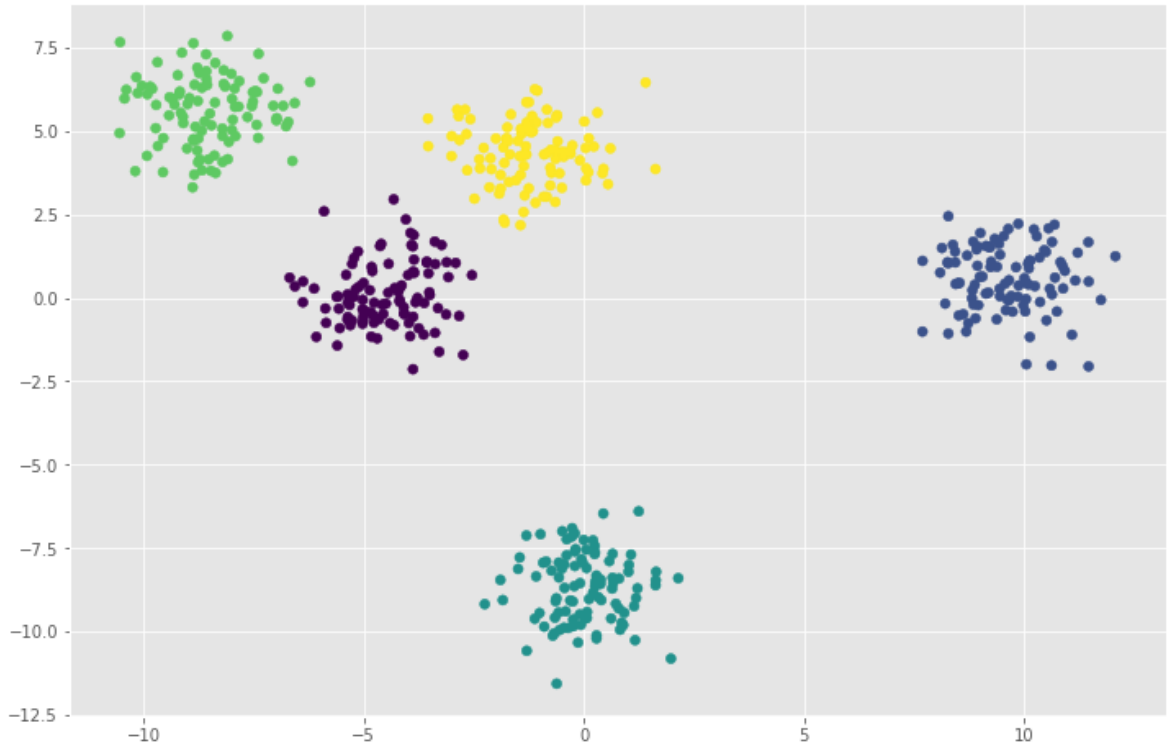
Изменив параметры кластеризации, получим:




Делаем тоже самое с **DBSCAN**:

```
from sklearn.cluster import DBSCAN

clustering = DBSCAN(eps=1.55, min_samples=5).fit_predict(X)
print(clustering)
plt.scatter(X[:,0], X[:,1], c=clustering);
```



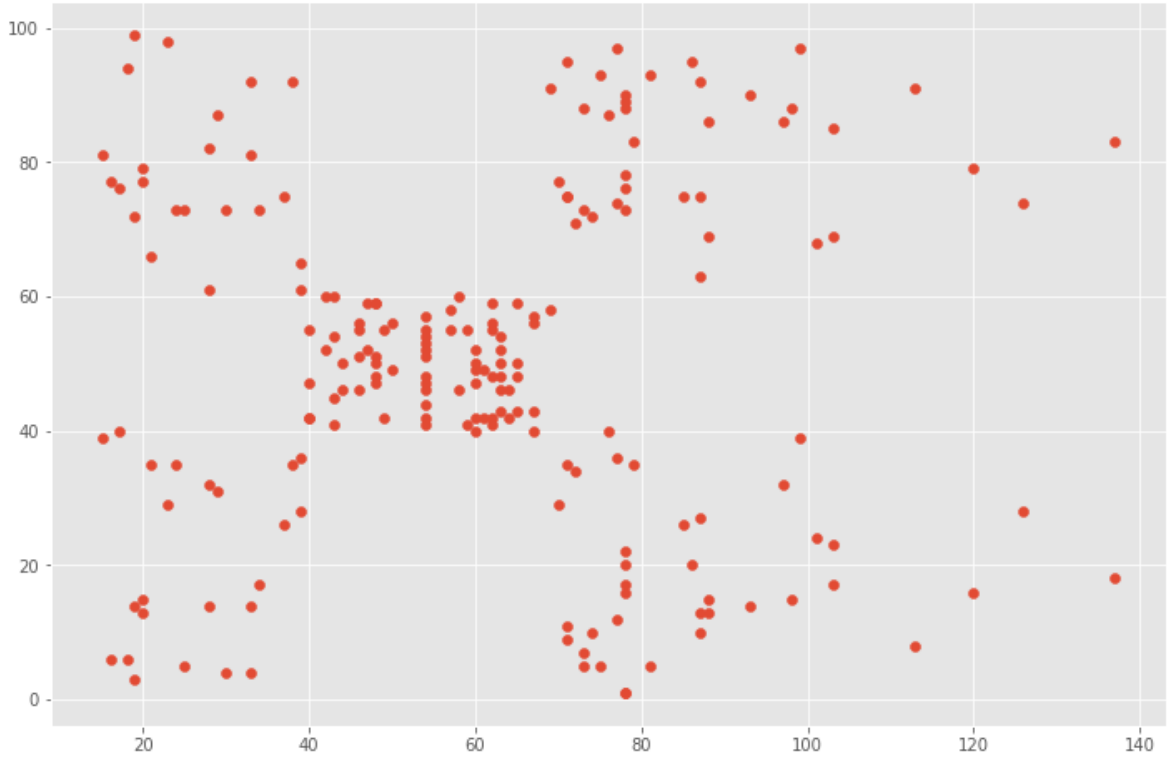

## Работа с реальными данными

После тренировки на "игрушечных данных", нам предлагают пройти задания с реальными данными

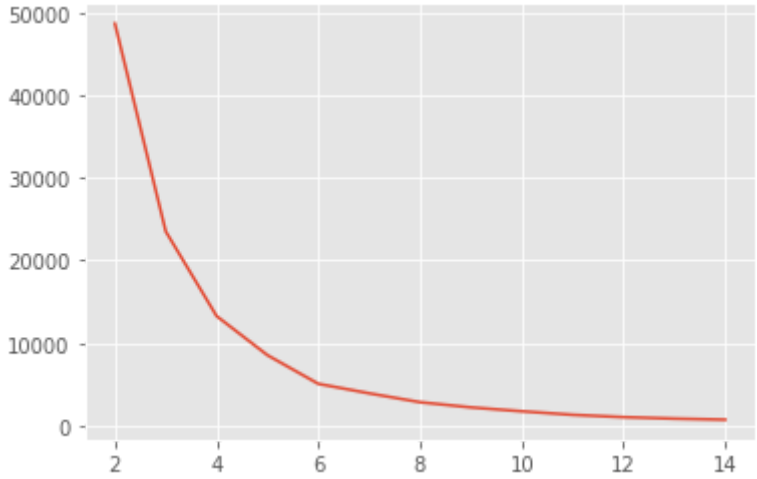
Так как данные расположены во внешнем файле, то нам нужно его считать и сделать срезы

```
import pandas as pd
from sklearn import preprocessing
X = pd.read_csv('data/Mall_Customers.csv').values  # считываем данные
```

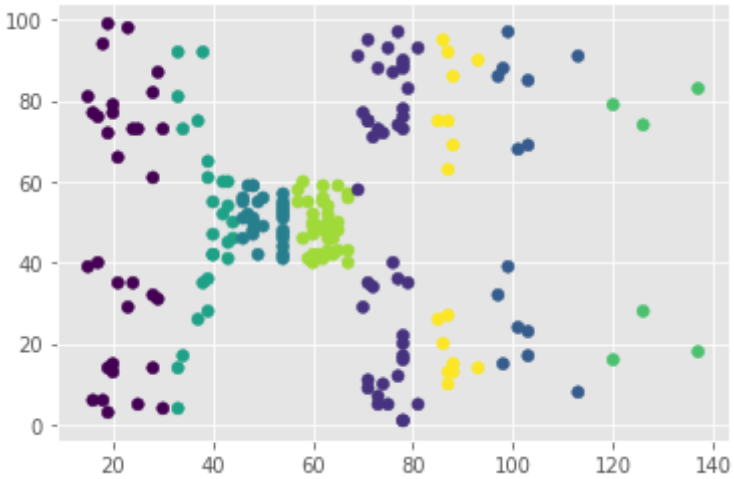
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')
plt.rcParams['figure.figsize']=(12,8)
data_1 = X[:,3]
data_2 = X[:,4]
plt.scatter(X[:,3], X[:,4])
```




Метод Локтя:

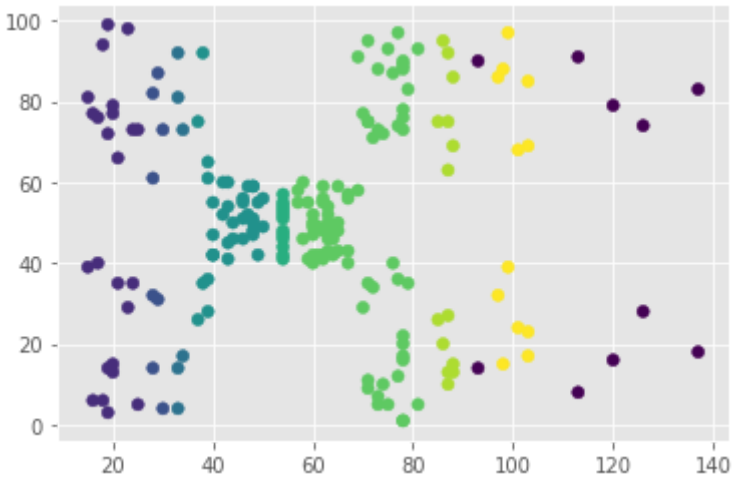



```
kmeansModel = KMeans(n_clusters=8) # используем k вычисленный через метод Локтя
kmeansModel.fit(X[:,3:4]) # обучаем модель
plt.scatter(X[:,3], X[:,4], c=kmeansModel.labels_) # визуализируем данные
```



```
from sklearn.cluster import DBSCAN

clustering = DBSCAN(eps=2.8, min_samples=5).fit_predict(X[:,3:4])
plt.scatter(X[:,3], X[:,4], c=clustering);
```




## Вывод

Кластеризация оказалась действительно интересной задачей, думаю самая интересная из всех семи предложенных. Здесь мы использовали помимо самой кластеризации метод Локтя, чтобы оптимально выбрать входные параметры кластеризации.

## Authors


|  |  |
|---|---|
| Arthur Kupriyanov | Artyom Kolokolov |

Группа: P3212