

# Лабораторная работа №4

Для первого пункта нам нужно создать свой датасет.

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 %matplotlib inline
5 plt.style.use('ggplot')
6 plt.rcParams['figure.figsize']=(12,8)
7
8 from sklearn.datasets import make_blobs
9 X,y = make_blobs(n_samples=500, random_state=7, centers = 5)
10
11 plt.scatter(X[:,0], X[:,1])
```



Используя размер кластеров 2:

```
1 from sklearn.cluster import KMeans
2 kmeansModel = KMeans(n_clusters=2) # мы пока не знаем количество кластеров
3 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:

```

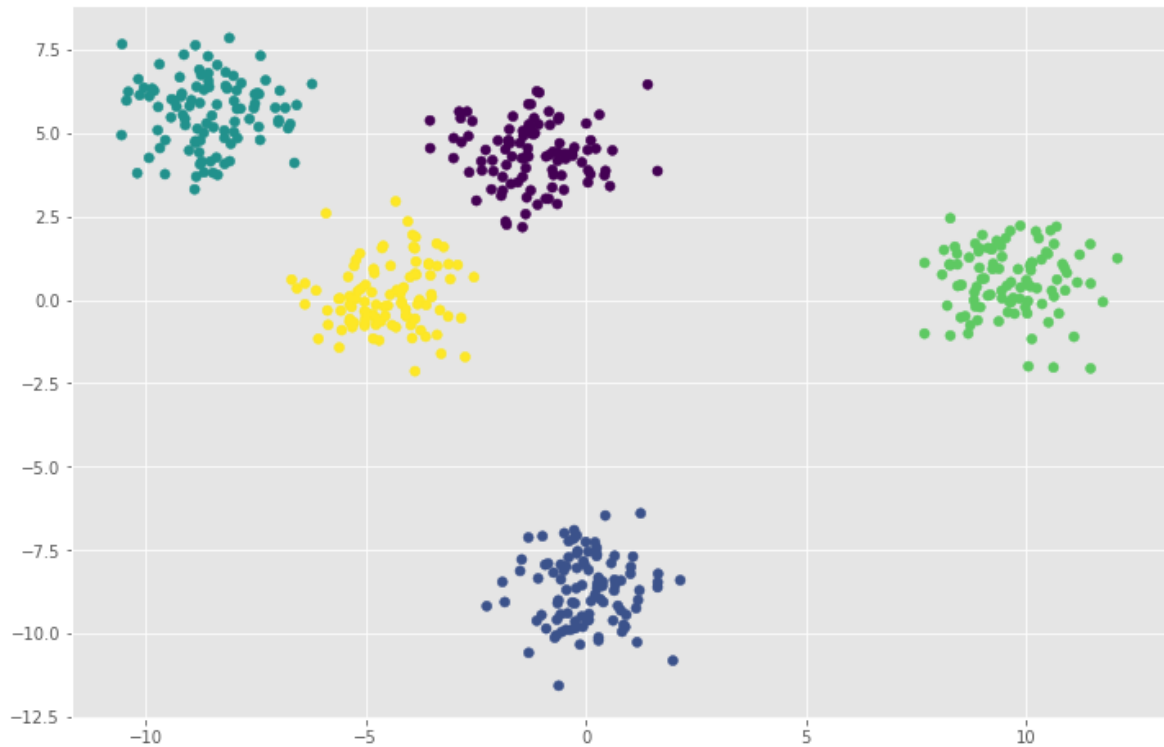
1 | criteries = []
2 | for k in range(2,10):
3 |     kmeansModel=KMeans(n_clusters=k, random_state=7)
4 |     kmeansModel.fit(X)
5 |     criteries.append(kmeansModel.inertia_)
6 | plt.plot(range(2,10), criteries)

```



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

Изменив параметры кластеризации, получим:

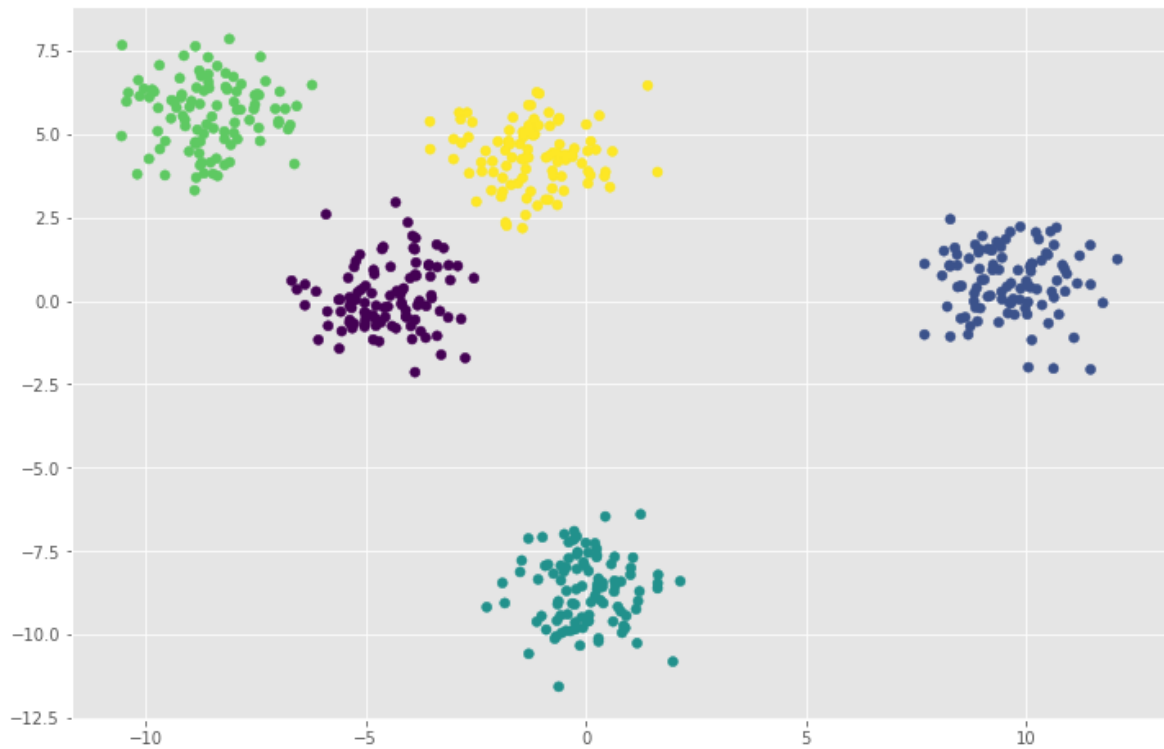


Делаем тоже самое с **DBSCAN**:

```

1 | from sklearn.cluster import DBSCAN
2 |
3 | clustering = DBSCAN(eps=1.55, min_samples=5).fit_predict(X)
4 | print(clustering)
5 | plt.scatter(X[:,0], X[:,1], c=clustering);

```



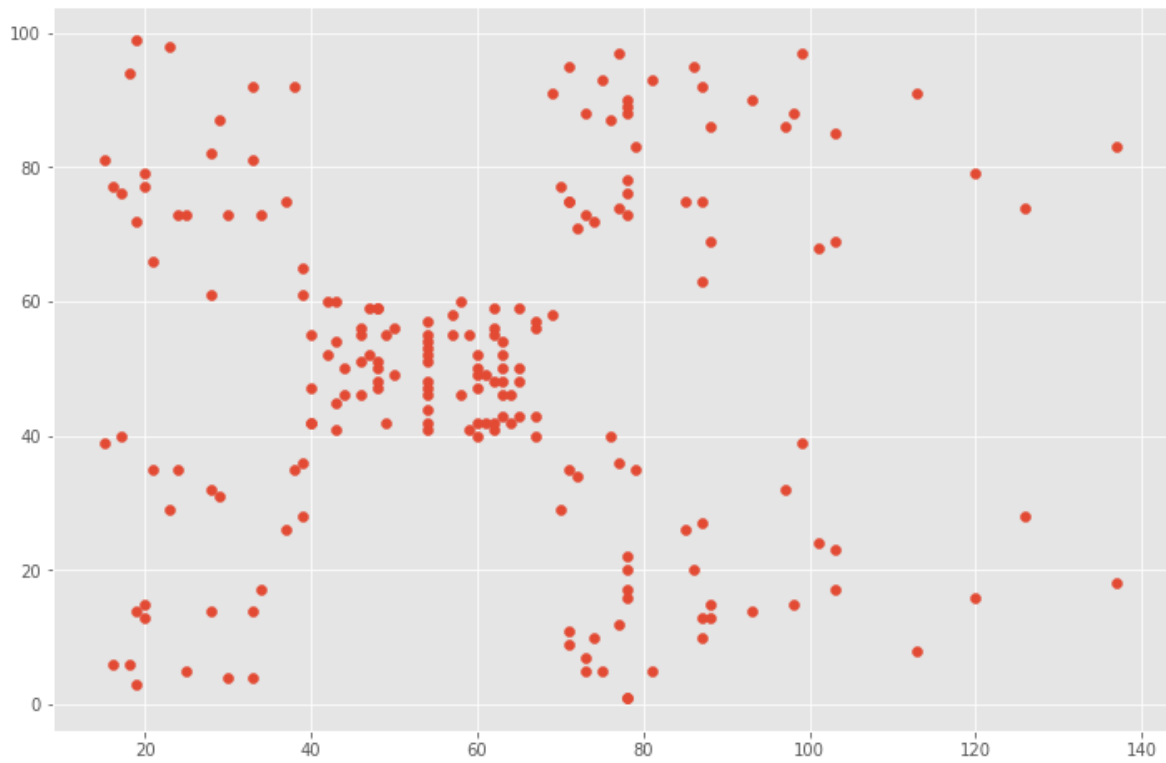
## Работа с реальными данными

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

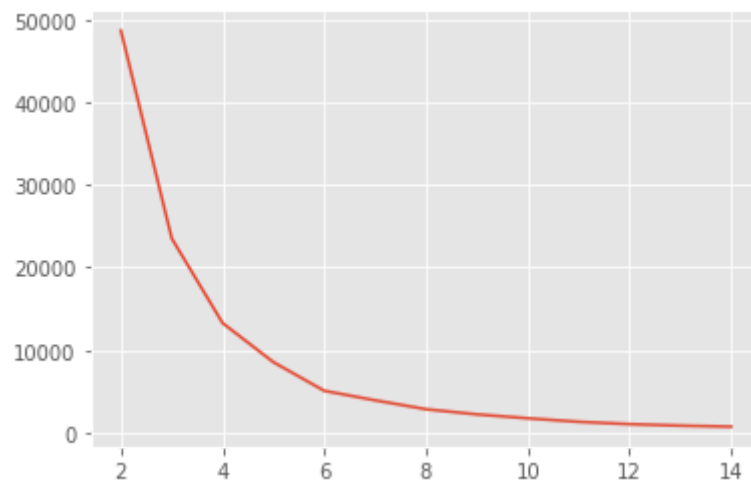
Так как данные расположены во внешнем файле, то нам нужно его считать и сделать срезы

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

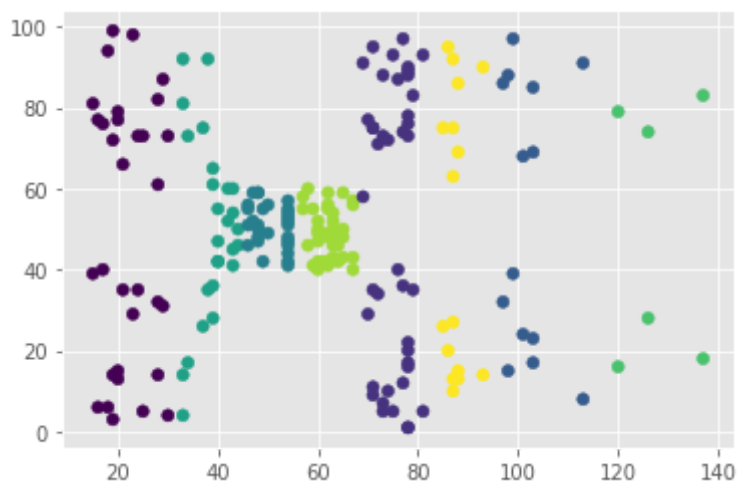
```
1 import matplotlib.pyplot as plt
2 %matplotlib inline
3 plt.style.use('ggplot')
4 plt.rcParams['figure.figsize']=(12,8)
5 data_1 = x[:,3]
6 data_2 = x[:,4]
7 plt.scatter(x[:,3], x[:,4])
```



Метод Локтя:



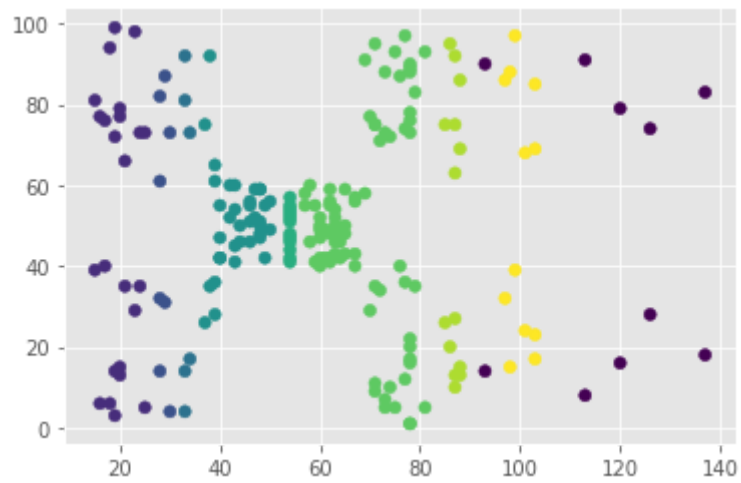
```
1 kmeansModel = kmeans(n_clusters=8) # используем k вычисленный через метод
  Локтя
2 kmeansModel.fit(X[:,3:4]) # обучаем модель
3 plt.scatter(X[:,3], X[:,4], c=kmeansModel.labels_) # визуализируем данные
```



```

1 from sklearn.cluster import DBSCAN
2
3 clustering = DBSCAN(eps=2.8, min_samples=5).fit_predict(X[:,3:4])
4 plt.scatter(X[:,3], X[:,4], c=clustering);

```



## Authors



[Arthur Kupriyanov](#)



[Artyom Kolokolov](#)

Группа: P3212