ИТМО

Современные инструменты анализа данных
Лабораторная работа №1

Кластеризация

ФИО студента: Готовко Алексей Владимирович
Направление подготовки: 09.03.04 (СППО)
Учебная группа: P32101
Практик: Змиевский Данил Александрович

Санкт-Петербург
2022г.

1 Настройка среды

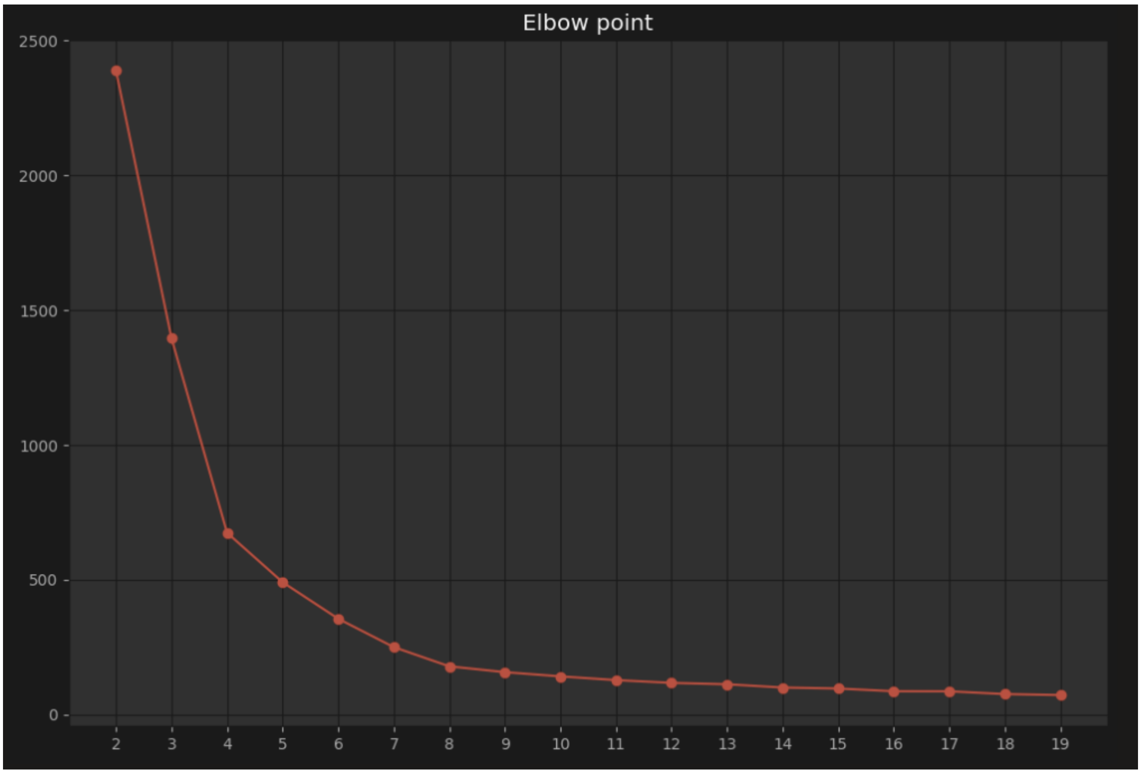
```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans, DBSCAN

plt.style.use("ggplot")
plt.rcParams["figure.figsize"] = (12, 8)

def show_result(x, labels, plot_name):
    plt.scatter(x[:, 0], x[:, 1], c=labels)
    plt.title(plot_name)
    plt.show()
```

2 Генерация данных

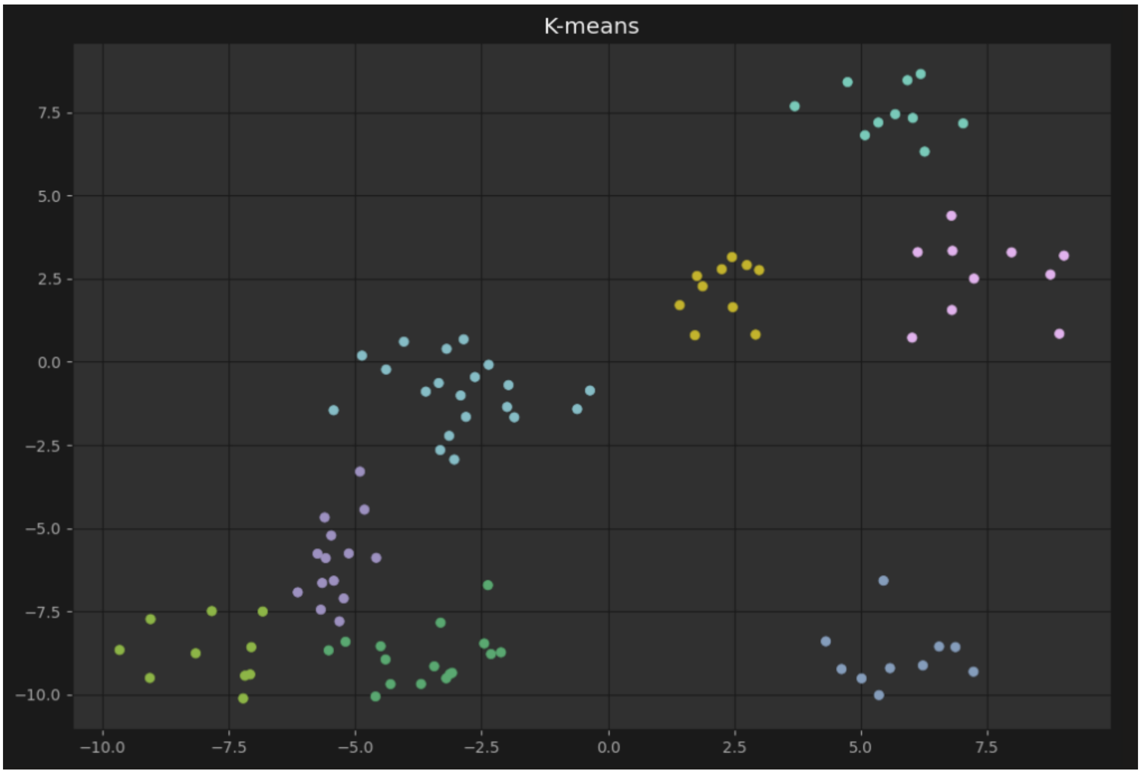
```
# n_samples    -- total number of points equally divided among clusters
# centers      -- the number of centers to generate
# random_state -- determines random number generation for dataset creation
# iterations   -- how many iterations we are going to make while determining optimal
                                              clusters number (only for K-Means)
n_samples = 100
centers = 10
random_state = 420
iterations = centers * 2

x, y = make_blobs(n_samples=n_samples, centers=centers, random_state=random_state)
```

3 K-means

```
criteria_arr = []
for k in range(2, iterations):
    kmeans_model = KMeans(n_clusters=k, random_state=random_state)
    kmeans_model.fit(x)

    criteria = kmeans_model.inertia_
    criteria_arr.append(criteria)
```

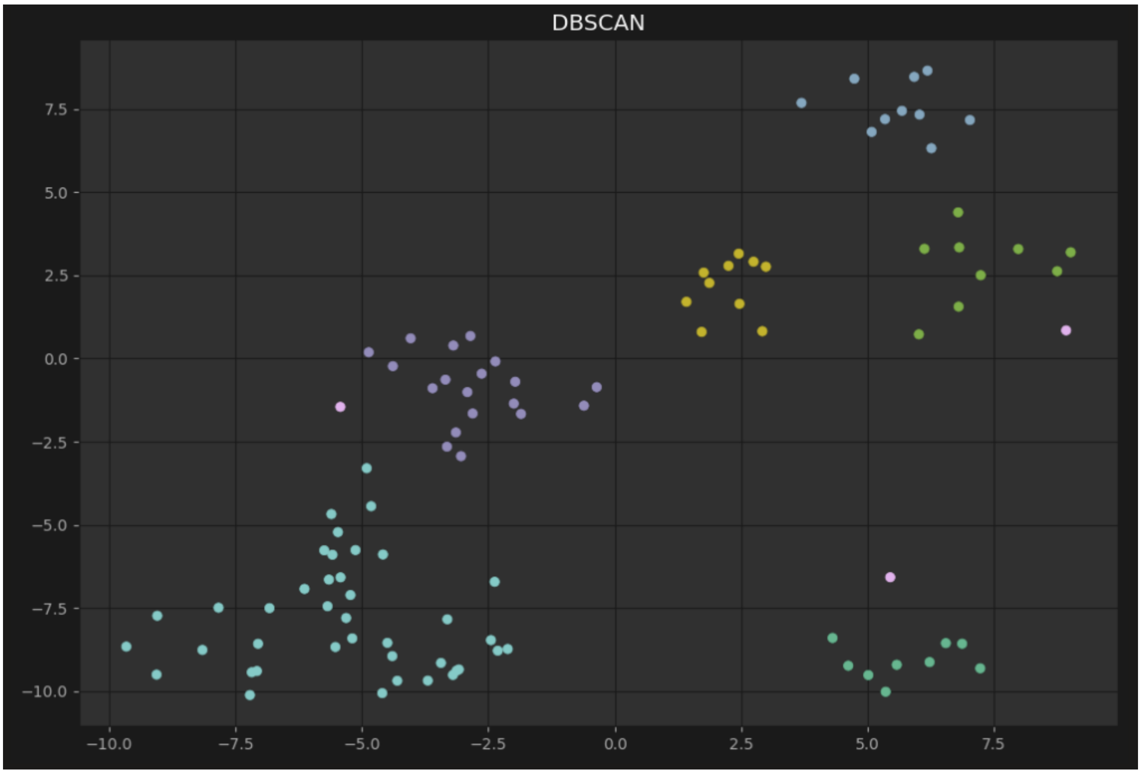


```
clusters = 8
kmeans_model = KMeans(n_clusters=clusters)
kmeans_model.fit(x)
labels = kmeans_model.labels_
show_result(x, labels, "K-means")
```



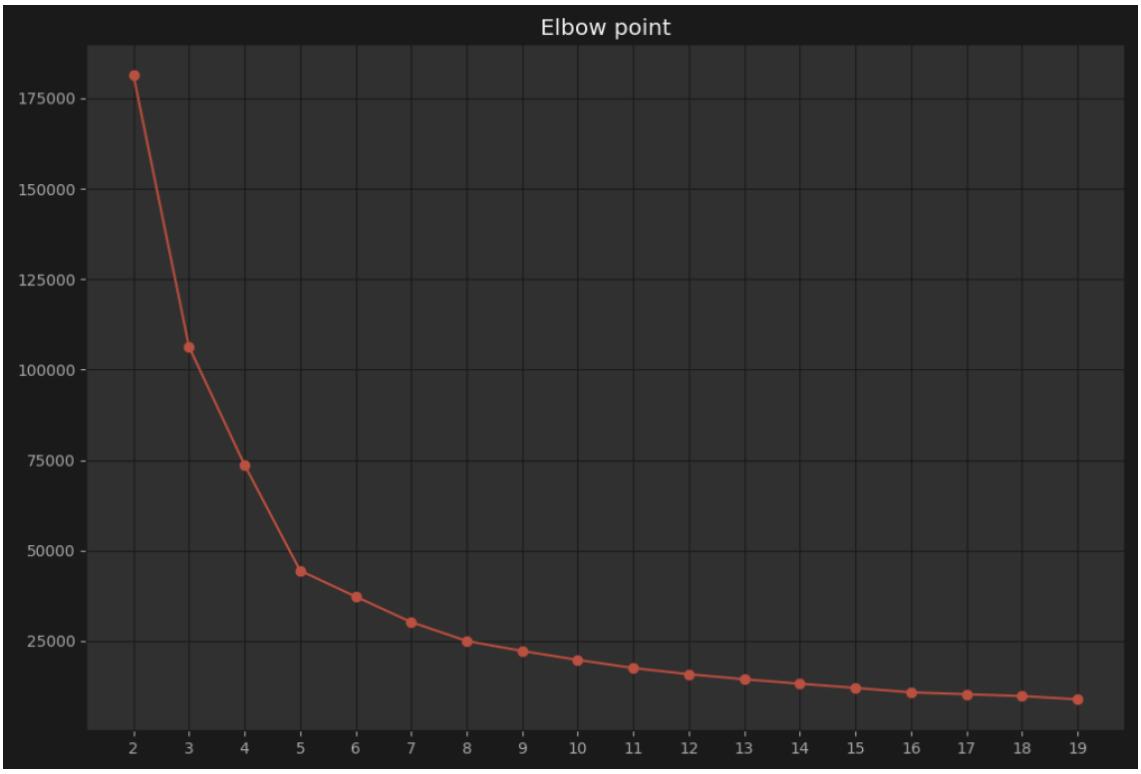
4 DBSCAN

```
clustering = DBSCAN(eps=1.5, min_samples=2).fit_predict(x)
print(clustering)
# [ 0  0  1  1  2  0  1  1  2  2  0  2  0  3  0  0  2  2  0  1  0  1  2  4
#  2  2  3  5  2  1  0  0  4  2  2  2  5  1 -1  2  5  2  2  5  2  5  2  0
#  3  2  2  2  0  3  2  1  5  5  4  0  3  2  2  3  3  2 -1  2  3  4  5  0
#  2  2  4  0  4  1  2  3  2  2  2  2  2  5  4  2  0  2  2  2  0  0  4 -1
#  5  2  4  2]

show_result(x, clustering, "DBSCAN")
```

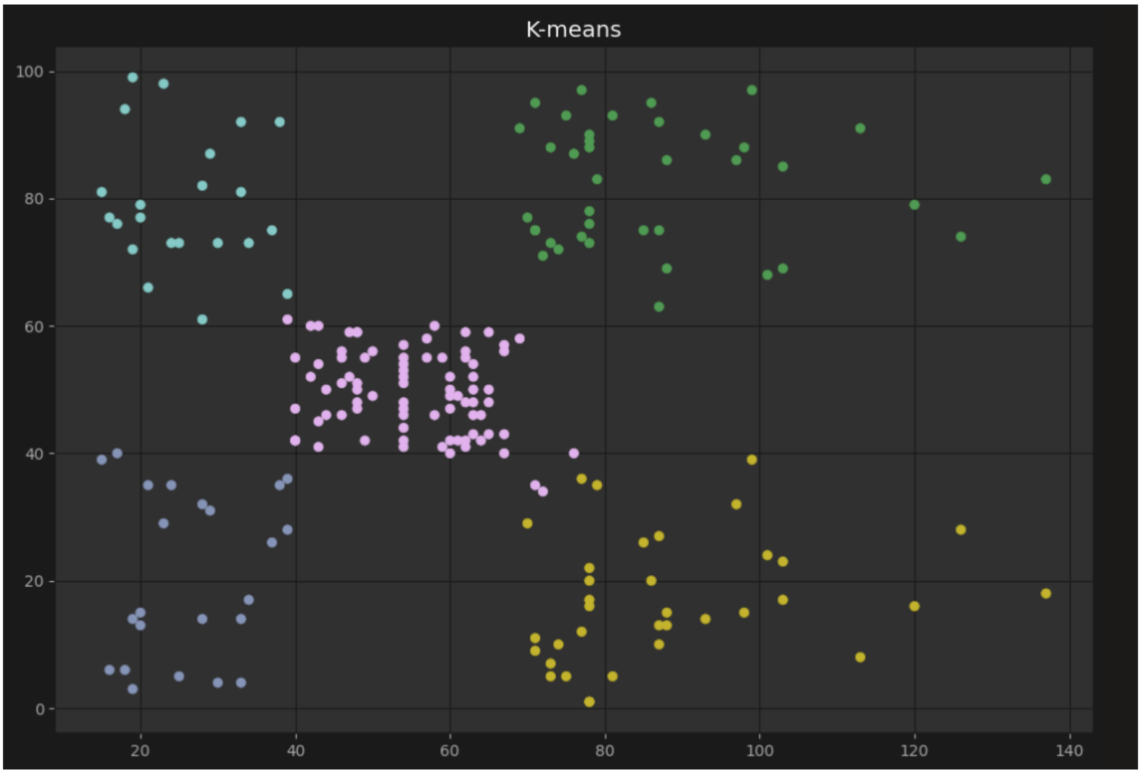


5 Датасет покупателей магазина

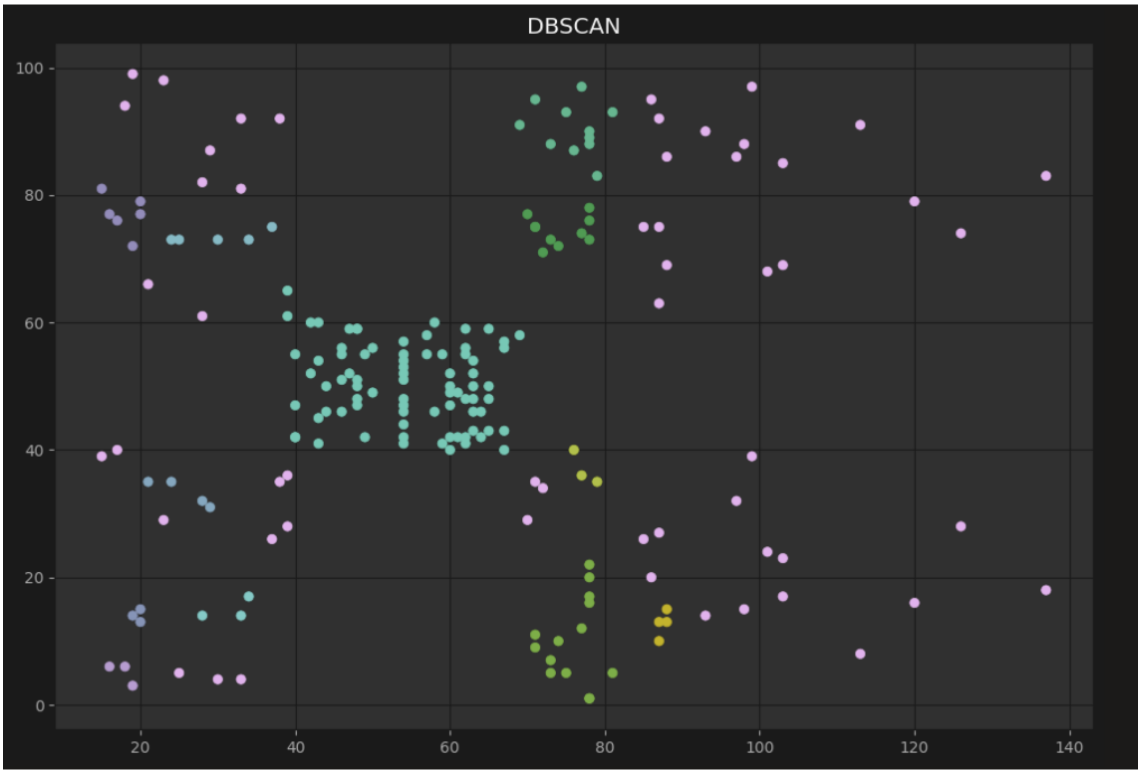
```
df = pd.read_csv("Mall_Customers.xls")
print(df)
#      CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
# 0             1    Male   19                  15                      39
# 1             2    Male   21                  15                      81
# 2             3  Female   20                  16                       6
# 3             4  Female   23                  16                      77
# 4             5  Female   31                  17                      40
# ..          ...     ...  ...                 ...                     ...
# 195         196  Female   35                 120                      79
# 196         197  Female   45                 126                      28
# 197         198    Male   32                 126                      74
# 198         199    Male   32                 137                      18
# 199         200    Male   30                 137                      83
#
# [200 rows x 5 columns]

x = df[["Annual Income (k$)", "Spending Score (1-100)"]].iloc[: , :].values

criteria_arr = []
for k in range(2, iterations):
    kmeans_model = KMeans(n_clusters=k)
    kmeans_model.fit(x)

    criteria = kmeans_model.inertia_
    criteria_arr.append(criteria)

plt.plot(range(2, iterations), criteria_arr, marker="o")
plt.xticks([i for i in range(2, iterations)])
plt.title("Elbow point")
```



```
clusters = 5
kmeans_model = KMeans(n_clusters=clusters)
kmeans_model.fit(x)
labels = kmeans_model.labels_
show_result(x, labels, "K-means")
```



```
clustering = DBSCAN(eps=5, min_samples=3).fit_predict(x)
print(clustering)
# [-1  1  0  1 -1  1  0 -1  0  1  2 -1  2  1  2  1  3 -1 -1 -1  3  4 -1  4
#   5 -1  3 -1  3 -1 -1  4 -1 -1  5 -1  5  4 -1  4 -1 -1 -1  6 -1  6  6  6
#   6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
#   6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
#   6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
#   6  6  6  7 -1  8 -1  7  9  8  9  8 -1  8  9  7  9  8  9  8  9  7 10  7
#   9  7 10  8  9  7  9  7  9  8  9  7  9  8  9  8 10  7  9  7 -1 -1 -1 -1
#  -1 -1 11 -1 11 -1 11 -1 11 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
#  -1 -1 -1 -1 -1 -1 -1 -1]

show_result(x, clustering, "DBSCAN")
```


DBSCAN