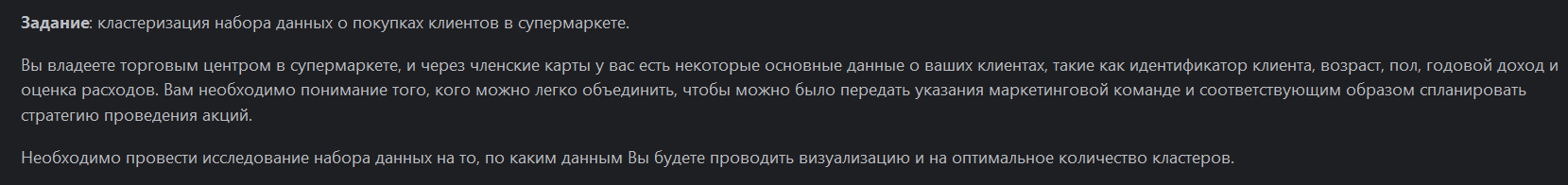
# Группа 5 Панфилов Валерий Александрович Лабораторная работа №4

# «Проведение кластеризации над множеством»

Задание: 

Код:

from sklearn import datasets  
from sklearn.cluster import KMeans  
from sklearn.metrics import silhouette\_score, davies\_bouldin\_score,v\_measure\_score  
import matplotlib.pyplot as plt  
import numpy as np  
  
import pandas as pd  
  
  
df = pd.read\_csv('Mall\_Customers.csv')  
del df['CustomerID']  
del df['Gender']  
  
  
#Определение алгоритма  
model = KMeans(n\_clusters=6)  
values = df.values  
model.fit(values)  
data = model.predict(values)  
  
  
#Кластеры по "годовому доходу от покупателя" и "способности потратьть деньги"  
for i in range(0, len(values)):  
 if data[i] == 0:  
 c1 = plt.scatter(values[i, 1], values[i, 2], c='r', marker='\*')  
 elif data[i] == 1:  
 c2 = plt.scatter(values[i, 1], values[i, 2], c='g', marker='.')  
 elif data[i] == 2:  
 c3 = plt.scatter(values[i, 1], values[i, 2], c='b', marker='+')  
 elif data[i] == 3:  
 c4 = plt.scatter(values[i, 1], values[i, 2], c='m', marker='o')  
 elif data[i] == 4:  
 c5 = plt.scatter(values[i, 1], values[i, 2], c='c', marker='x')  
 elif data[i] == 5:  
 c6 = plt.scatter(values[i, 1], values[i, 2], c='y', marker='x')  
plt.legend([c1, c2, c3, c4, c5, c6], ['Кластер 1', 'Кластер 2', 'Кластер 3', 'Кластер 4'  
, 'Кластер 5', 'Кластер 6'])  
plt.title('k-means. Разделение на n кластеров')  
plt.show()  
  
#Кластеры по "возрасту" и "способности потратьть деньги"  
for i in range(0, len(values)):  
 if data[i] == 0:  
 c1 = plt.scatter(values[i, 0], values[i, 2], c='r', marker='\*')  
 elif data[i] == 1:  
 c2 = plt.scatter(values[i,0], values[i, 2], c='g', marker='.')  
 elif data[i] == 2:  
 c3 = plt.scatter(values[i, 0], values[i, 2], c='b', marker='+')  
 elif data[i] == 3:  
 c4 = plt.scatter(values[i, 0], values[i, 2], c='m', marker='o')  
 elif data[i] == 4:  
 c5 = plt.scatter(values[i, 0], values[i, 2], c='c', marker='x')  
 elif data[i] == 5:  
 c6 = plt.scatter(values[i, 0], values[i, 2], c='y', marker='x')  
plt.legend([c1, c2, c3, c4, c5, c6], ['Кластер 1', 'Кластер 2', 'Кластер 3', 'Кластер 4'  
, 'Кластер 5', 'Кластер 6'])  
plt.title('k-means. Разделение на n кластеров')  
plt.show()  
  
#Кластеры по "возрасту" и "годовому доходу от покупателя  
for i in range(0, len(values)):  
 if data[i] == 0:  
 c1 = plt.scatter(values[i, 0], values[i, 1], c='r', marker='\*')  
 elif data[i] == 1:  
 c2 = plt.scatter(values[i,0], values[i, 1], c='g', marker='.')  
 elif data[i] == 2:  
 c3 = plt.scatter(values[i, 0], values[i, 1], c='b', marker='+')  
 elif data[i] == 3:  
 c4 = plt.scatter(values[i, 0], values[i, 1], c='m', marker='o')  
 elif data[i] == 4:  
 c5 = plt.scatter(values[i, 0], values[i, 1], c='c', marker='x')  
 elif data[i] == 5:  
 c6 = plt.scatter(values[i, 0], values[i, 1], c='y', marker='x')  
plt.legend([c1, c2, c3, c4, c5, c6], ['Кластер 1', 'Кластер 2', 'Кластер 3', 'Кластер 4'  
, 'Кластер 5', 'Кластер 6'])  
plt.title('k-means. Разделение на n кластеров')  
plt.show()  
  
#Форматирование данных  
scores= []  
res\_silhouette = []  
vmeasure\_score =[]  
db\_score = []  
for i in range(2,12):  
 model = KMeans(n\_clusters=i, random\_state=0).fit(values)  
 preds = model.predict(values)  
 print("Score for number of cluster(s) {}: {}".format(i,model.score(values)))  
 scores.append(-model.score(values))  
 silhouette = silhouette\_score(values,preds)  
 res\_silhouette.append(silhouette)  
  
print("Silhouette score for number of cluster(s) {}: {}".format(i,silhouette))  
  
#Метод Локтя  
plt.figure(figsize=(7,4))  
plt.title("Метод локтя",fontsize=16)  
plt.scatter(x=[i for i in range(2,12)],y=scores,s=150,edgecolor='k')  
plt.grid(True)  
plt.xlabel("Номер кластера",fontsize=14)  
plt.ylabel("Оценка силуэта",fontsize=15)  
plt.xticks([i for i in range(2,12)],fontsize=14)  
plt.yticks(fontsize=15)  
plt.show()  
  
#Метод Силуэта  
plt.figure(figsize=(7,4))  
plt.title("Метод Силуэта",fontsize=16)  
plt.scatter(x=[i for i in range(2,12)],y=res\_silhouette,s=150,edgecolor='k')  
plt.grid(True)  
plt.xlabel("Номер кластера",fontsize=14)  
plt.ylabel("Оценка силуэта",fontsize=15)  
plt.xticks([i for i in range(2,12)],fontsize=14)  
plt.yticks(fontsize=15)  
plt.show()

Результат выполнения:

