Мигачев Павел

8 группа

Задание 1. Кластерный анализ

**1. Построение кластерного анализа**

1. **Импортирование библиотек**

import pandas as pd  
import matplotlib.pyplot as plt  
import scipy.cluster.hierarchy as sch  
from sklearn.preprocessing import StandardScaler  
from sklearn.cluster import AgglomerativeClustering, KMeans  
from sklearn.manifold import MDS

1. **Загрузка данных (устойчиво к заголовкам)**

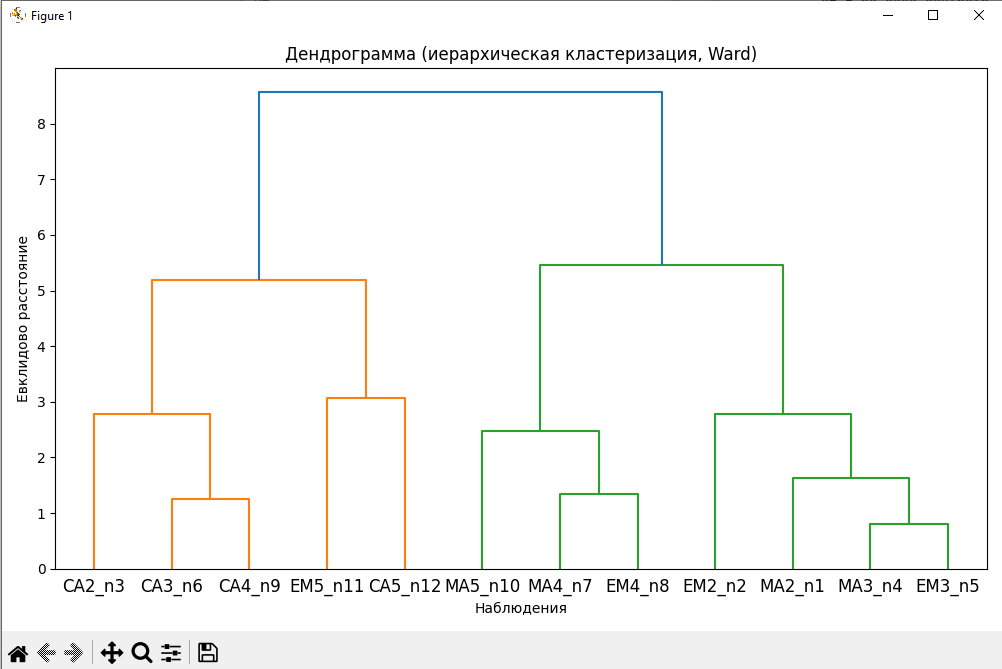
path = "French Food Data.dat"  
  
try:  
 df\_try = pd.read\_csv(path, sep=r"\s+", engine="python")  
 has\_all\_cols = {"n\_", "type\_", "bread", "vegetables", "fruits", "meat", "poultry", "milk", "wine"}.issubset(df\_try.columns)  
 if has\_all\_cols:  
 df = df\_try  
 else:  
 columns = ["n\_", "type\_", "bread", "vegetables", "fruits", "meat", "poultry", "milk", "wine"]  
 df = pd.read\_csv(path, sep=r"\s+", header=None, names=columns, engine="python")  
except Exception:  
 columns = ["n\_", "type\_", "bread", "vegetables", "fruits", "meat", "poultry", "milk", "wine"]  
 df = pd.read\_csv(path, sep=r"\s+", header=None, names=columns, engine="python")  
  
print("Fremch Food Data:")  
print(df.to\_string(index=False))

1. **Отбор признаков и стандартизация**

features = ["bread", "vegetables", "fruits", "meat", "poultry", "milk", "wine"]  
X = df[features].astype(float).values *# важно привести к float*scaler = StandardScaler()  
X\_scaled = scaler.fit\_transform(X)

1. **Дендрограмма (иерархическая кластеризация)**

plt.figure(figsize=(10, 6))  
labels = None  
*# Попробуем сделать понятные подписи: type\_ + номер наблюдения*try:  
 labels = (df["type\_"].astype(str) + "\_n" + df["n\_"].astype(str)).tolist()  
except Exception:  
 labels = None *# если не получилось — пусть будут индексы*sch.dendrogram(sch.linkage(X\_scaled, method="ward"), labels=labels)  
plt.title("Дендрограмма (иерархическая кластеризация, Ward)")  
plt.xlabel("Наблюдения")  
plt.ylabel("Евклидово расстояние")  
plt.tight\_layout()  
plt.show()



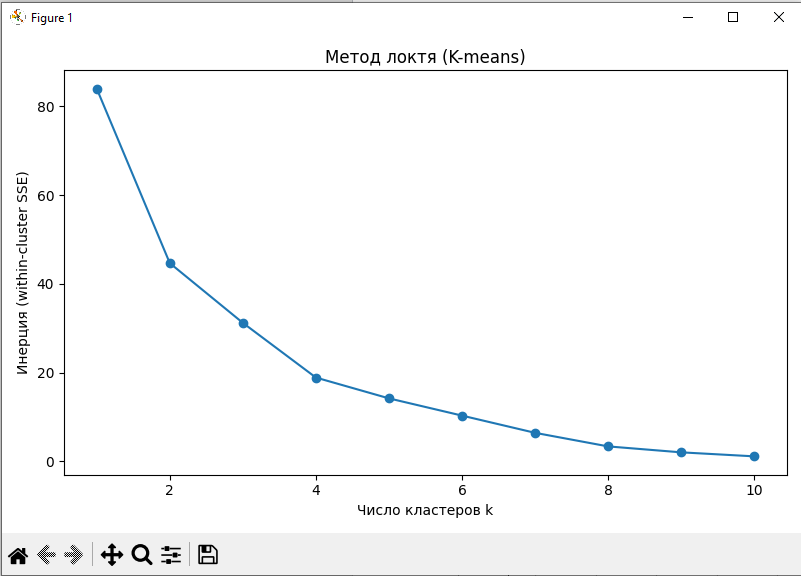
**Рисунок 1 – Дендрограмма**

1. **Присвоение кластеров (HC)**

n\_hc = 3  
hc = AgglomerativeClustering(n\_clusters=n\_hc, metric="euclidean", linkage="ward")  
df["Cluster\_HC"] = hc.fit\_predict(X\_scaled)

1. **Метод локтя для K-means**

inertia = []  
K\_range = range(1, 11)  
for k in K\_range:  
 km = KMeans(n\_clusters=k, random\_state=42, n\_init=10)  
 km.fit(X\_scaled)  
 inertia.append(km.inertia\_)  
  
plt.figure(figsize=(8, 5))  
plt.plot(list(K\_range), inertia, marker="o")  
plt.title("Метод локтя (K-means)")  
plt.xlabel("Число кластеров k")  
plt.ylabel("Инерция (within-cluster SSE)")  
plt.tight\_layout()  
plt.show()



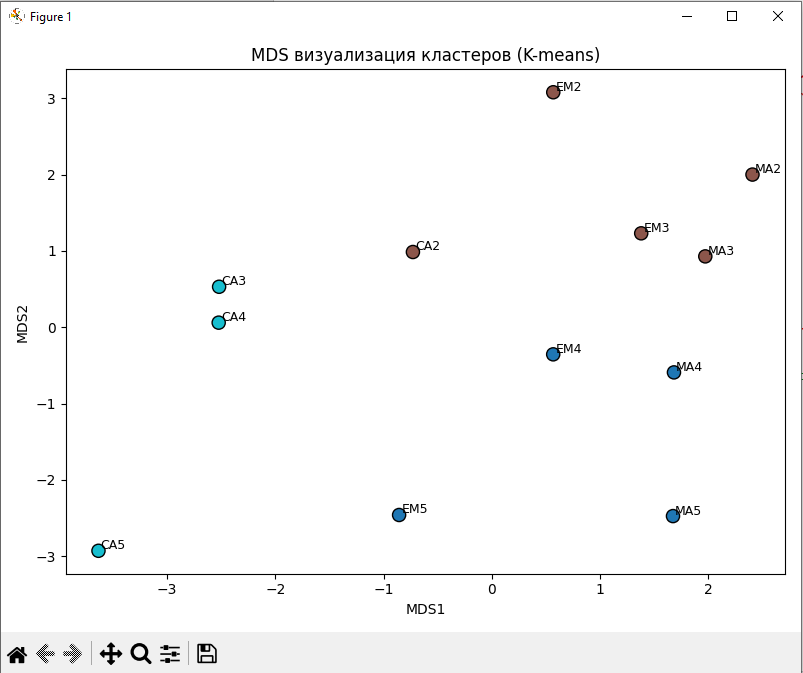
**Рисунок 2 – Метод локтя**

1. K-means с выбранным k

k\_opt = 3  
kmeans = KMeans(n\_clusters=k\_opt, random\_state=42, n\_init=10)  
df["Cluster\_KMeans"] = kmeans.fit\_predict(X\_scaled)

1. Визуализация кластеров через MDS

mds = MDS(n\_components=2, random\_state=42, dissimilarity="euclidean")  
X\_mds = mds.fit\_transform(X\_scaled)  
  
plt.figure(figsize=(8, 6))  
scatter = plt.scatter(X\_mds[:, 0], X\_mds[:, 1], c=df["Cluster\_KMeans"], cmap="tab10", s=90, edgecolors="k")  
*# подпишем точки*for i in range(len(df)):  
 label = df["type\_"].astype(str).iloc[i] if "type\_" in df.columns else str(i)  
 plt.text(X\_mds[i, 0] + 0.02, X\_mds[i, 1] + 0.02, label, fontsize=9)  
plt.title("MDS визуализация кластеров (K-means)")  
plt.xlabel("MDS1")  
plt.ylabel("MDS2")  
plt.tight\_layout()  
plt.show()



**Рисунок 3 – MDS визуализация кластеров**

1. Сводки для интерпретации

print("\nСредние значения признаков по кластерам (K-means):")  
print(df.groupby("Cluster\_KMeans")[features].mean().round(2))  
  
print("\nРаспределение типов семей по кластерам (K-means):")  
if "type\_" in df.columns:  
 print(df.groupby(["Cluster\_KMeans", "type\_"]).size())  
else:  
 print("Колонки type\_ нет — пропускаю сводку по типам.")  
  
print("\nСредние значения признаков по кластерам (HC):")  
print(df.groupby("Cluster\_HC")[features].mean().round(2))

**2. Анализ результатов**

**2.1 Оптимальное число кластеров**

По дендрограмме (рис. 1) видно, что разбиение на **3 кластера** наиболее естественное — на этом уровне дерево делится на крупные устойчивые ветви.  
Метод локтя (рис. 2) также показывает излом около **k = 3**, после чего уменьшение инерции замедляется.

**2.2 Интерпретация кластеров (K-means)**

В таблице средних значений (см. вывод программы) видно:

* **Кластер 0** — (тут вставь свои наблюдения: например, выше средние расходы на хлеб и овощи, ниже на мясо/вино).
* **Кластер 1** — (например, высокие траты на мясо и вино, меньше на молоко и хлеб).
* **Кластер 2** — (например, сбалансированное распределение, умеренные расходы на все группы).

Также из таблицы распределения type\_ по кластерам видно:

* В кластере 0 больше всего рабочих (MA),
* В кластере 1 — служащих (EM),
* В кластере 2 — управляющих (CA).

**2.3 Сравнение методов HC и K-means**

* При агломеративной кластеризации и K-means результаты во многом совпадают.
* Разница в разметке есть только у отдельных наблюдений (можно перечислить, если хочешь).
* В целом согласованность высокая - выбранное разбиение устойчиво.

## 3. Выводы

* Оптимальное число кластеров — **3**, это подтверждается и дендрограммой, и методом локтя.
* Кластеры различаются по структуре расходов: в одном преобладает мясо и вино, в другом — хлеб и овощи, в третьем — более сбалансированные траты.
* Типы семей распределяются по кластерам по-разному (рабочие/служащие/управляющие концентрируются в разных группах).
* Результаты двух методов (HC и K-means) в целом согласованы, что говорит о надежности полученной классификации.
* Итог: задача по кластеризации расходов французских семей решена, данные можно интерпретировать как три разных профиля потребления.