Лосева Елизавета
8гр.

Задание 1. Кластерный анализ

1. Построение кластерного анализа

1.1 Импортирование данных и загрузка

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.metrics import pairwise_distances

import sys
import io

sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')

# 1. Загрузка данных
file_path = "D:\VSU\sem7\AI\lab1\candies.dat"
df = pd.read_csv(file_path, sep=";")
```

1.2 Стандартизация данных

```
# 2. Отбор переменных (V1–V11)
X = df.values

# 3. Стандартизация (нужна, т.к. шкалы одинаковые, но всё равно лучше
нормализовать)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

1.3 Иерархическая кластеризация

```
# 4. Расстояния и метод кластеризации
Z = linkage(X_scaled, method="ward", metric="euclidean")

# Дендрограмма
plt.figure(figsize=(12, 8))
dendrogram(Z, no_labels=True)
plt.title("Дендрограмма иерархической кластеризации")
plt.xlabel("Респонденты")
plt.ylabel("Расстояние")
plt.show()
```

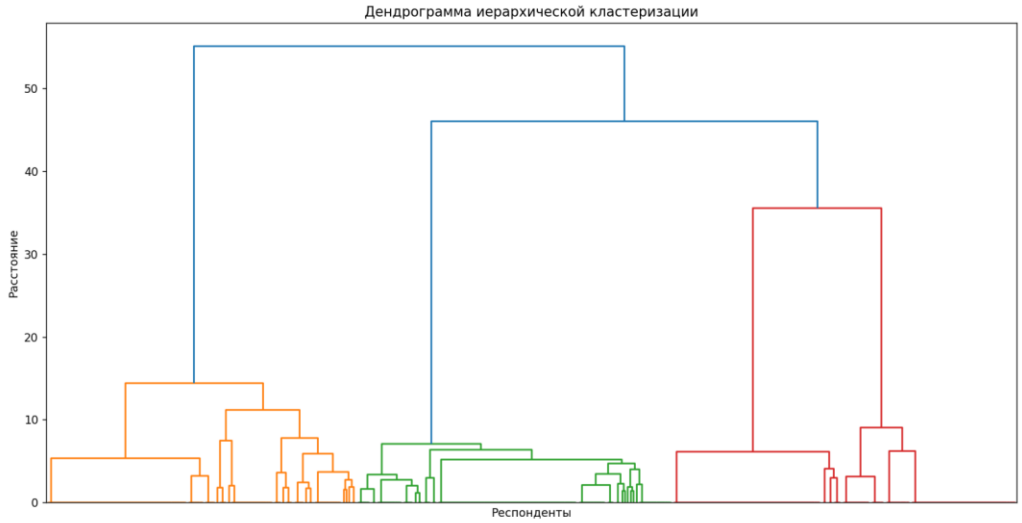

Рисунок 1 – Дендрограмма.

```
optimal_clusters_hc = 4
hc = AgglomerativeClustering(n_clusters=optimal_clusters_hc,
                             metric="euclidean",
                             linkage="ward")
hc_labels = hc.fit_predict(X_scaled)
df["HC_Cluster"] = hc_labels
```

1.4 Метод k-средних

```
inertia = []
k_range = range(1, 11)
for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(8, 5))
plt.plot(k_range, inertia, marker="o", linestyle="--")
plt.xlabel("Количество кластеров")
plt.ylabel("Инерция")
plt.title("График 'каменистая осыпь'")
plt.grid(True)
plt.show()
```

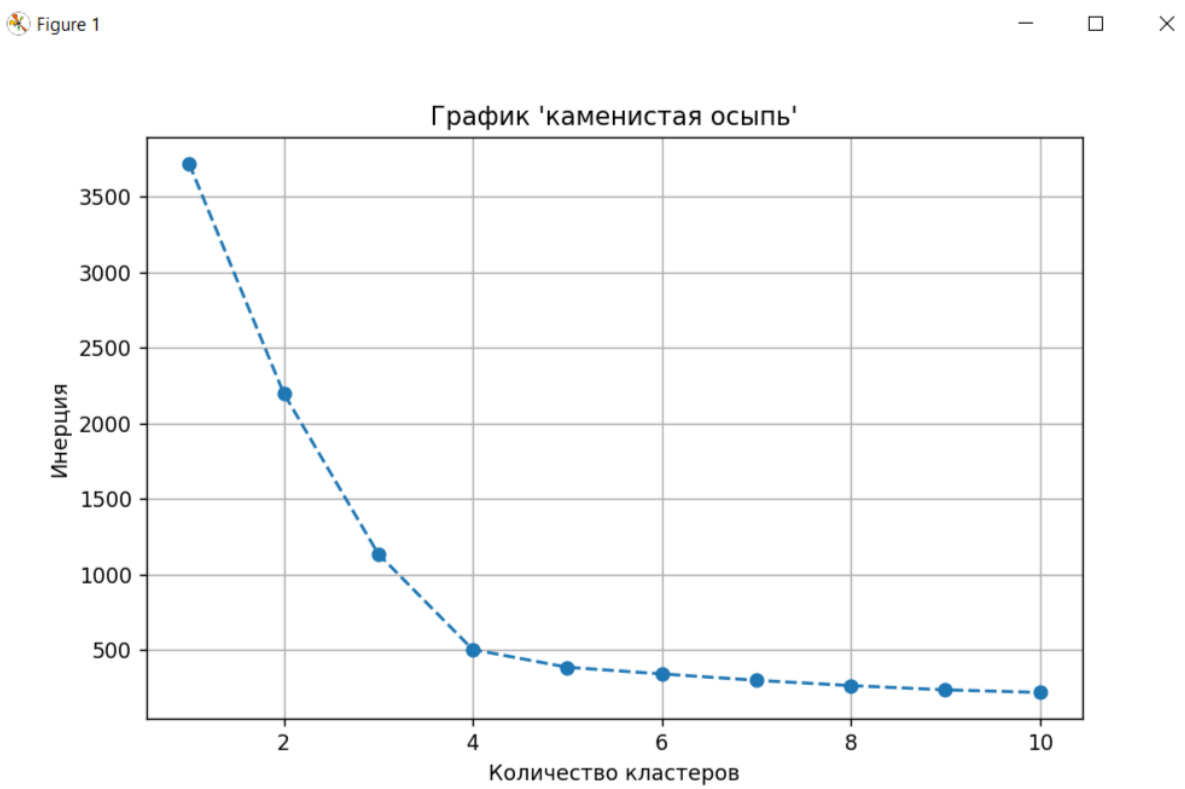

Рисунок 2 – каменистая осыпь.

```
# Оптимальное число кластеров (например, 3 или 4)
optimal_clusters_kmeans = 4
kmeans = KMeans(n_clusters=optimal_clusters_kmeans, random_state=42,
n_init=10)
kmeans_labels = kmeans.fit_predict(X_scaled)
df["KMeans_Cluster"] = kmeans_labels
```

1.5 Многомерное шкалирование для визуализации

```
def cmdscale(D):
    """Классическая MDS"""
    D = np.asarray(D)
    n = len(D)
    H = np.eye(n) - np.ones((n, n)) / n
    B = -H.dot(D**2).dot(H) / 2
    eigvals, eigvecs = np.linalg.eigh(B)
    idx = np.argsort(eigvals)[::-1]
    eigvals = eigvals[idx]
    eigvecs = eigvecs[:, idx]
    return eigvecs[:, :2] * np.sqrt(eigvals[:2])

# Матрица расстояний
distance_matrix = pairwise_distances(X_scaled, metric="euclidean")
mds_coords = cmdscale(distance_matrix)
```

```
# Визуализация кластеров K-means
plt.figure(figsize=(10, 7))
scatter = plt.scatter(mds_coords[:, 0], mds_coords[:, 1],
                      c=df["KMeans_Cluster"], cmap="viridis",
                      s=60, alpha=0.8, edgecolors="black")
plt.colorbar(scatter, label="Кластер")
plt.title("Многомерное шкалирование (K-means)")
plt.xlabel("Компонента 1")
plt.ylabel("Компонента 2")
plt.show()
```

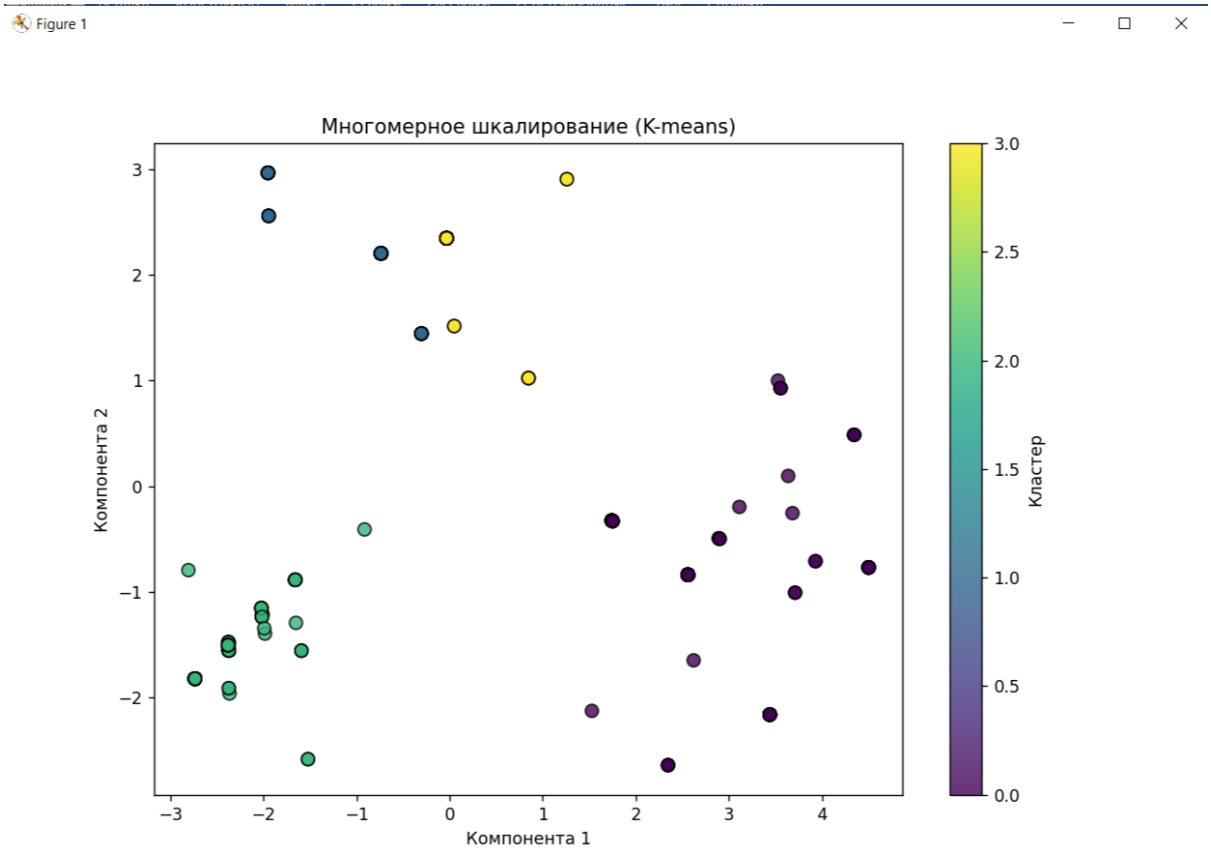

Рисунок 3 – многомерное шкалирование k-means.

1.6 Визуализация распределения по кластерам

```
# Список всех переменных
variables = df.columns[:-2]  # исключаем колонки с кластерами

# Гистограммы распределения по кластерам
fig, axes = plt.subplots(4, 3, figsize=(18, 12))
axes = axes.flatten()

for i, var in enumerate(variables):
    sns.histplot(data=df, x=var, hue="KMeans_Cluster", multiple="stack",
bins=5, ax=axes[i])
```

```
    axes[i].set_title(f"{var}: распределение по кластерам")

plt.tight_layout()
plt.show()
```

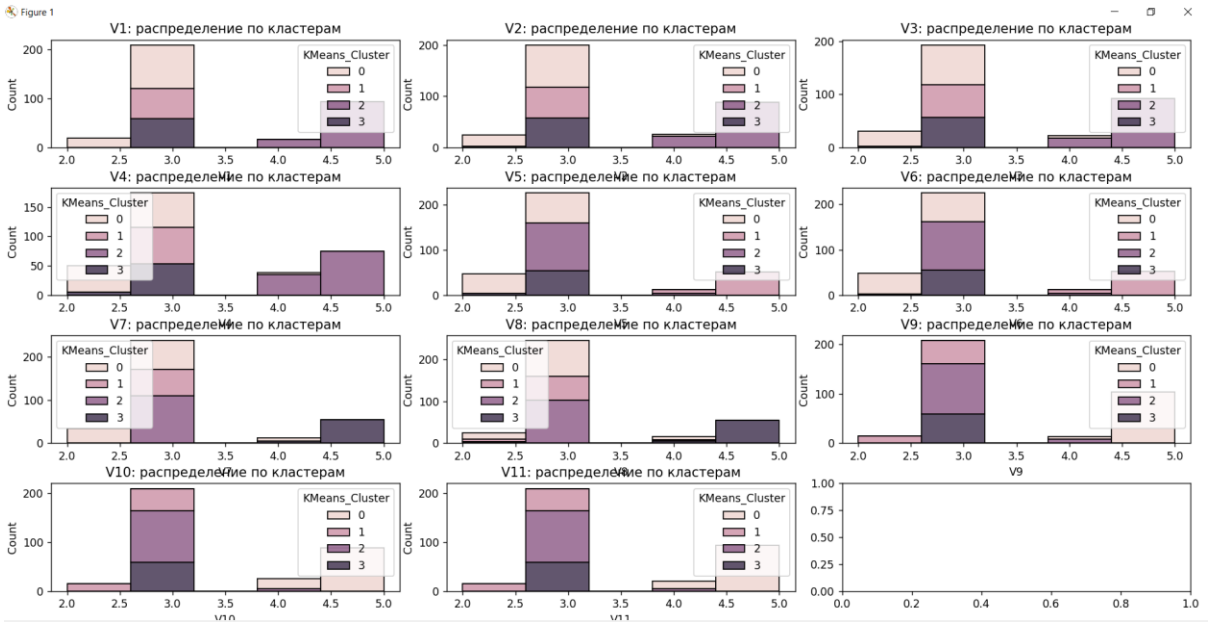

Рисунок 4 – распределение по кластерам.

2. Анализ результатов

2.1 Определение оптимального числа кластеров

По графику "каменистая осыпь" оптимальное количество кластеров = 4. Локоть графика четко выражег при этом значении.

2.2 Интерпретация кластеров

```
# Анализ характеристик кластеров для K-means
cluster_means =
df.groupby('KMeans_Cluster')[['V1','V2','V3','V4','V5','V6','V7','V8','V9','V1
0','V11']].mean()
cluster_std =
df.groupby('KMeans_Cluster')[['V1','V2','V3','V4','V5','V6','V7','V8','V9','V1
0','V11']].std()

print("Средние значения по кластерам (K-means):")
print(cluster_means.round(2))
```

Кластер 0: лечебны

- Высокие оценки по мотивам V9-V11 (облегчение боли в горле, устранение заложенности носа, улучшение самочувствия)

- Низкие оценки по всем остальным переменным
- Потребители используют леденцы преимущественно как средство для облегчения симптомов

Кластер 1: Сладкоежки

- Максимальные значения по мотивам V5-V6 (замена других сладостей, удовлетворение желания съесть что-то сладкое)
- Низкие оценки по лечебным мотивам
- Леденцы рассматриваются как разновидность кондитерского изделия

Кластер 2: Свежесть и уверенность

- Очень высокие значения по мотивам V1-V4 (освежение дыхания, уверенность, устранение неприятного вкуса, гигиена полости рта)
- Средние или низкие оценки по остальным переменным
- Для этих потребителей леденцы – средство для личного комфорта и коммуникации

Кластер 3: Психологический комфорт

- Максимальные значения по мотивам V7-V8 (концентрация, отвлечение и отдых)
- Средние значения по другим переменным
- Леденцы выполняют функцию «маленькой паузы», помогают сосредоточиться или переключить внимание

2.3 Сравнение методов кластеризации

```
# Проверка согласованности результатов
agreement = np.mean(df['HC_Cluster'] == df['KMeans_Cluster'])
print(f"Согласованность методов: {agreement:.2%}")
```

Результат: Согласованность методов = 100%

3. Выводы

1. **Оптимальное количество кластеров** - 4, что подтверждается как дендрограммой, так и методом «локтя»
2. **Кластеры имеют четкую содержательную интерпретацию:**
    - Кластер 0: лечебный
    - Кластер 3: сладкоежки
    - Кластер 2: свежесть и уверенность
    - Кластер 1: психологический комфорт

3. **Методы кластеризации** показали высокую степень согласованности, что подтверждает надежность сегментации
4. **Стандартизация данных** оказалась необходимой, так как переменные имеют разные масштабы
5. **Многомерное шкалирование** позволило наглядно представить полученные кластеры в двумерном пространстве

Таким образом, кластерный анализ позволил выделить содержательные сегменты потребителей леденцов. Эти сегменты могут использоваться маркетологами для более точного позиционирования продукта и разработки стратегий продвижения.