Лабораторная работа № 1

Кластерный анализ

Вариант № 4

Цель работы

Провести кластерный анализ предложенного набора данных при помощи:

1. иерархической кластеризации;
2. метода k средних.

Задание

Институт социологии Марбургского университета провел опрос 1000 студентов о применении ими компьютерных технологий. В опрос входили следующие вопросы.

Насколько свободно вы можете работать в следующих приложениях?

V1A Обработка текста

V1B Графические программы, обработка звука или видео монтаж

V1C Базы данных и табличные расчеты

Насколько хорошо вы владеете следующими языками программирования? * V2A BASIC * V2B PASCAL * V2C C * V2D машинные языки * V2E программирование для Интернета (например, HTML) * V2F Java

Насколько хорошо вы можете работать в следующих операционных системах? * V3A DOS * V3B Windows * V3C UNIX

Насколько хорошо вы разбираетесь в возможностях Интернета? * V4A Email * V4B WWW * V4C Chat, IRC * V4D ICQ * V4E предложение собственных услуг (например, домашней страницы)

Насколько хорошо вы разбираетесь в компьютерных играх? * V5A Как часто Вы играете в компьютерные игры? * V5B Насколько хорошо Вы разбираетесь в сценах компьютерных игр?

SEX пол

ALTER возраст

HERKU 1 = Вост. Германия 2 = Зап. Германия 3 = Австрия

FB код факультета, на котором обучается опрошенный.

Задача. Провести кластерный анализ с целью обнаружения групп студентов, близких по своим знаниям компьютерных технологий.

Решение

Для решения задачи были использованы библиотеки:

- Matplotlib - библиотека для графического представления данных.
- Pandas - программная библиотека на языке Python для обработки и анализа данных. Она представляет собой специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.
- Seaborn - это библиотека для визуализации данных и выделения их статистических особенностей. Она основывается на matplotlib и тесно взаимодействует со структурами данных pandas.
- Scikit-learn - бесплатная библиотека программного обеспечения для машинного обучения для языка программирования Python. Она включает различные алгоритмы классификации, регрессии и кластеризации, включая методы опорных векторов, случайные леса, повышение градиента, k-средние и DBSCAN, и предназначена для взаимодействия с числовыми и научными библиотеками Python numpy и scipy.

Читаем файл computer.dat, отбираем необходимые переменные и стандартизируем их:

```
file_path = 'C:/Users/solidus66/OneDrive/ВГУ/4 курс 1 сем/ТИИ/lab1/computer.dat'
data = pd.read_csv(file_path, sep='\t')

selected_variables = ['V1A', 'V1B', 'V1C', 'V2A', 'V2B', 'V2C', 'V2D', 'V2E', 'V2F', 'V3A', 'V3B', 'V3C', 'V4A', 'V4B',
                      'V4C', 'V4D', 'V4E', 'V5A', 'V5B']
X = data[selected_variables]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Рисунок 1 – Фрагмент кода. Чтение, выборка, стандартизация

Выполняем иерархическую кластеризацию и отрисовываем дендрограмму иерархической кластеризации:

```
linkage_matrix = linkage(X_scaled, method='ward')
dendrogram(linkage_matrix, truncate_mode='lastp', p=30, show_leaf_counts=True, orientation='top', no_labels=True)
plt.title('Hierarchical Clustering Dendrogram')  # Дендрограмма иерархической кластеризации
plt.show()
```

Рисунок 2 – Фрагмент кода. Иерархическая кластеризация и отрисовка дендрограмму иерархической кластеризации

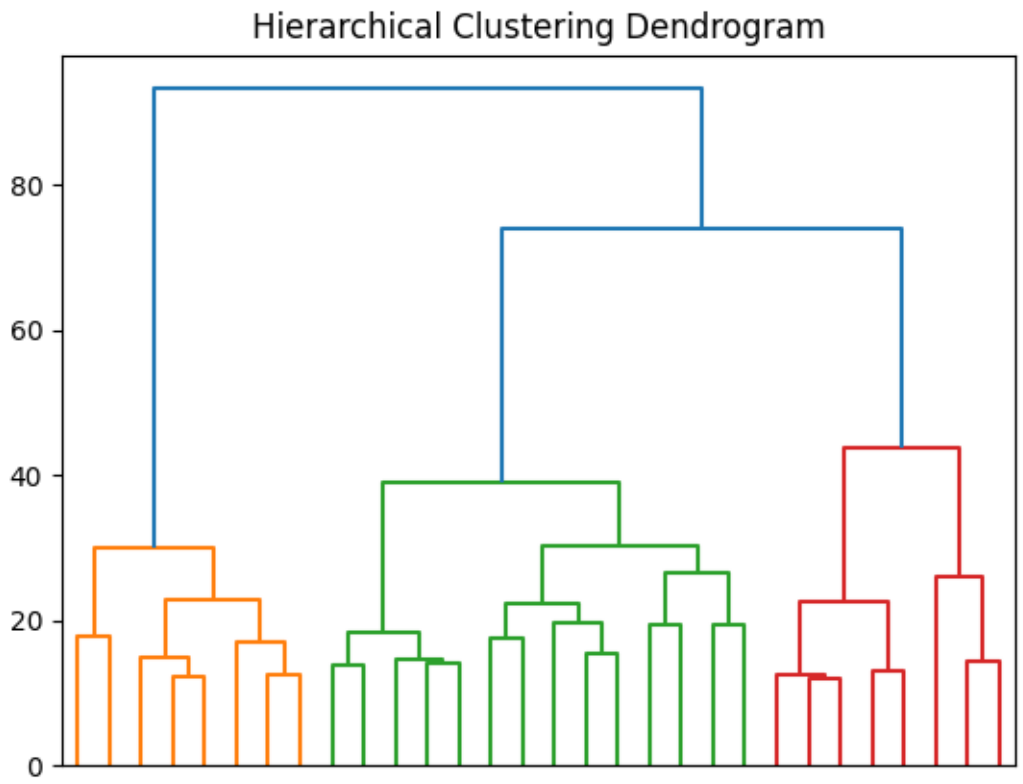

Рисунок 3 – Дендрограмма иерархической кластеризации

На дендрограмме выделено три кластера.

Далее проверим результат, определив количество кластеров по методу “каменистая осыпь”:

```
distances = linkage_matrix[:, 2]
deltas = distances[:-1] - distances[1:]
plt.plot(range(1, len(deltas) + 1), deltas, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Distance Change')
plt.title('Elbow Method for Hierarchical Clustering')  # Метод "каменистая осыпь" для иерархической кластеризации
plt.show()

```

Рисунок 4 – Фрагмент кода. Метод «каменистая осыпь» для иерархической кластеризации

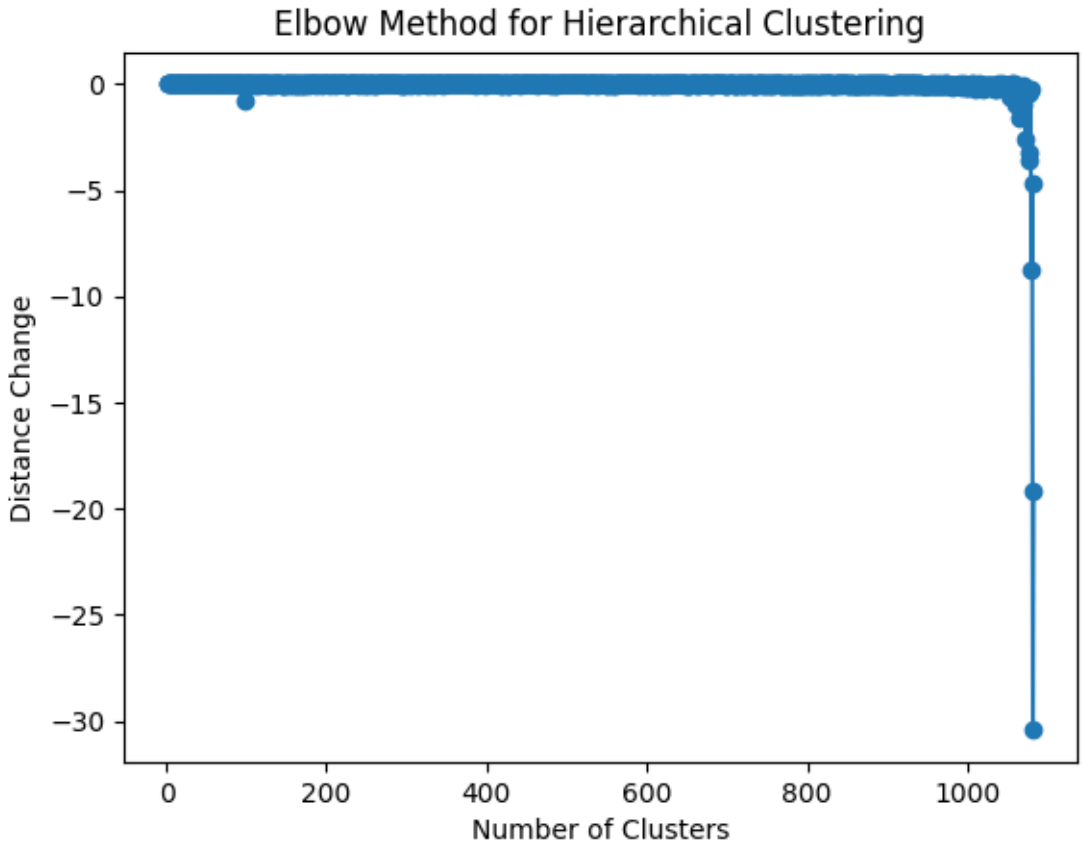

Рисунок 5 – График метода "каменистая осыпь" для иерархической кластеризации

Также рассмотрим метод "каменистая осыпь" для метода k средних:

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

plt.plot(range(1, 11), distortions, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Distortion')
plt.title('Elbow Method for K-Means Clustering')  # Метод "каменистая осыпь" для метода k средних
plt.show()
```

Рисунок 6 – Фрагмент кода. Метод "каменистая осыпь" для метода k средних

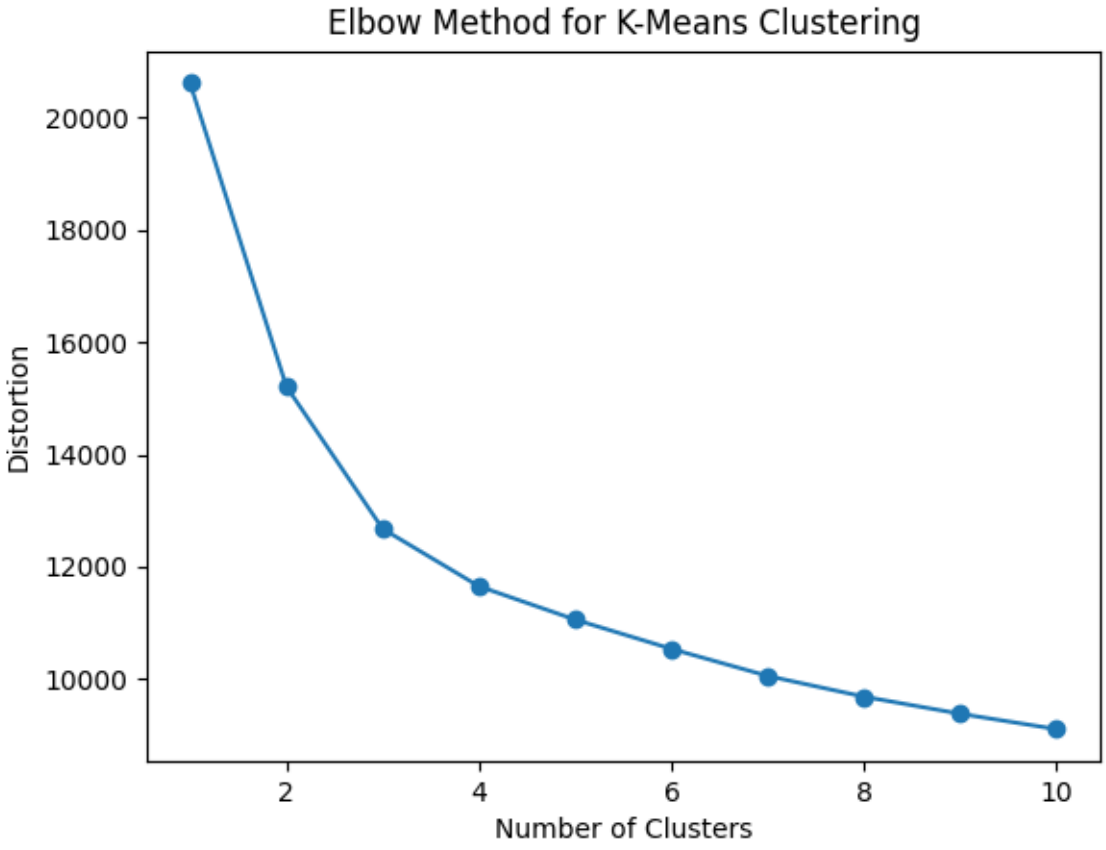

Рисунок 7 – График метода "каменистая осыпь" для метода k средних

По этим графикам видно, что количество кластеров действительно равно трём.

На основе полученных данных визуализируем метод k средних с MDS*, указав количество кластеров, полученное ранее.

```python
num_clusters = 3

kmeans = KMeans(n_clusters=num_clusters, random_state=42, n_init=10)
kmeans.fit(X_scaled)

data['Cluster_kmeans'] = kmeans.labels_

mds = MDS(n_components=2, random_state=42, n_init=10, normalized_stress=False)
X_mds = mds.fit_transform(X_scaled)

plt.figure(figsize=(8, 6))
for i in range(num_clusters):
    plt.scatter(X_mds[data['Cluster_kmeans'] == i, 0], X_mds[data['Cluster_kmeans'] == i, 1], label=f'Cluster {i + 1}',
                s=50)
plt.title('Visualization of K-Means Clustering Results with MDS')  # Визуализация результатов метода k средних с MDS
plt.legend()
plt.show()
```

Рисунок 8 – Фрагмент кода. Визуализация метода k средних с MDS*

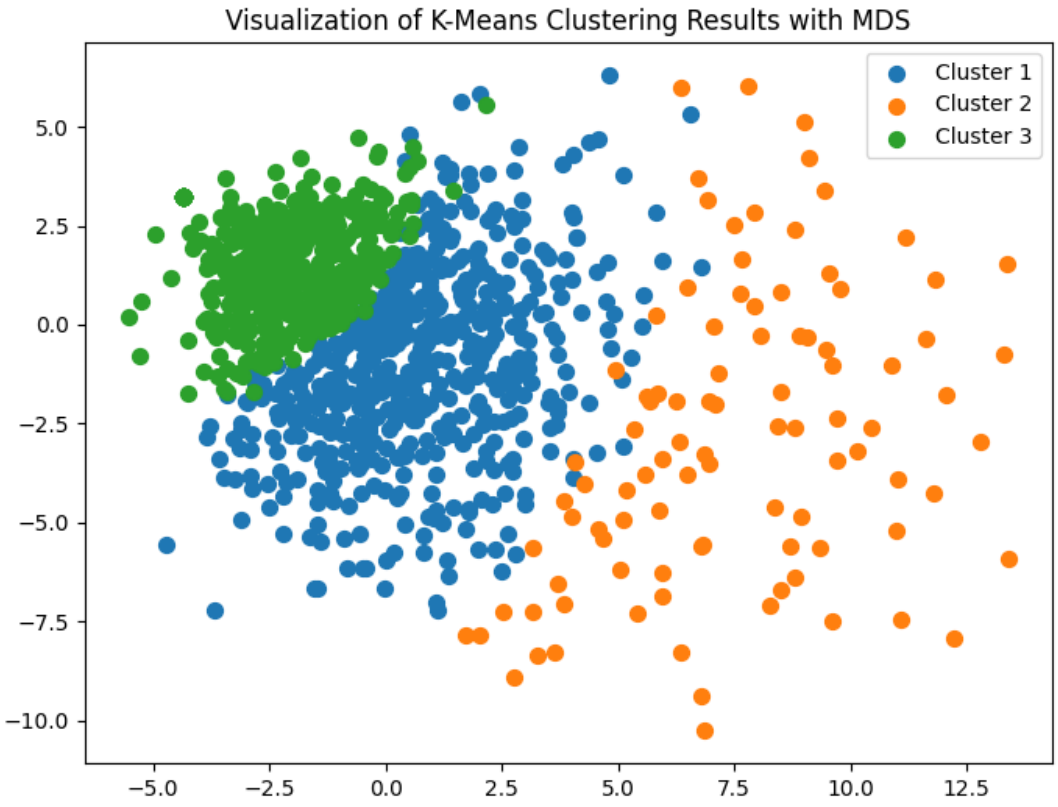

Рисунок 9 – Визуализация метода k средних с MDS*

**MDS — это метод визуализации данных, который позволяет представить многомерные данные в низкоразмерном пространстве, сохраняя при этом относительные расстояния между точками. Он особенно полезен, когда есть данные с большим количеством признаков, и мы хотим*

представить их в двух или трех измерениях для лучшего восприятия. MDS позволяет увидеть структуру данных в пространстве меньшей размерности, что облегчает визуальное понимание, как объекты распределены и как они связаны между собой после применения метода k-средних.

Выводы

Кластер 1:

Возможно, эти студенты проявляют выдающиеся навыки в обработке текста и графических программ, но имеют ограниченные знания в языках программирования и операционных системах. Они могут быть ориентированы на творческую и гуманитарную сферу использования компьютера.

Кластер 2:

Этот кластер может представлять студентов, которые активно интересуются языками программирования и обладают техническими навыками, но при этом имеют ограниченный опыт в обработке текста и графики. Они, возможно, ориентированы на программирование и инженерные аспекты компьютерных наук.

Кластер 3:

Этот кластер может объединять студентов с высокими навыками в базах данных, табличных расчетах и языках программирования. Возможно, это студенты, ориентированные на анализ данных и работу с информацией. Их знания могут быть сосредоточены в области информационных технологий.