**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине

«Прикладные интеллектуальные системы и экспертные системы»

Студент Первушин О. С.

Группа М-ИАП-23-1

Руководитель Кургасов В. В.

доцент, канд. пед. наук

Липецк 2023 г.

Цель работы

Получить практические навыки решения задачи кластеризации фактографических данных в среде Jupiter Notebook. Научиться проводить настраивать параметры методов и оценивать точность полученного разбиения.

Задание кафедры

Загрузить выборки согласно варианту задания

Отобразить данные на графике в пространстве признаков. Поскольку решается задача кластеризации, то подразумевается, что априорная информация о принадлежности каждого объекта истинному классу неизвестна, соответственно, на данном этапе все объекты на графике должны отображаться одним цветом, без привязки к классу.

Провести иерархическую кластеризацию выборки, используя разные способы вычисления расстояния между кластерами: расстояние ближайшего соседа (single), дальнего соседа (complete), Уорда (Ward). Построить дендрограммы для каждого способа. Размер графика должен быть подобран таким образом, чтобы дендрограмма хорошо читалась.

Исходя из дендрограмм выбрать лучший способ вычисления расстояния между кластерами.

Для выбранного способа, исходя из дендрограммы, определить количество кластеров в имеющейся выборке. Отобразить разбиение на кластеры и центроиды на графике в пространстве признаков (объекты одного кластера должны отображаться одним и тем же цветом, центроиды всех кластеров – также одним цветом, отличным от цвета кластеров)

Рассчитать среднюю сумму квадратов расстояний до центроида, среднюю сумму средних внутрикластерных расстояний и среднюю сумму межкластерных расстояний для данного разбиения. Сделать вывод о качестве разбиения.

Провести кластеризацию выборки методом k-средних. для k [1, 10].

Сформировать три графика: зависимость средней суммы квадратов расстояний до центроида, средней суммы средних внутрикластерных расстояний и средней суммы межкластерных расстояний от количества кластеров. Исходя из результатов, выбрать оптимальное количество кластеров 2

Составить сравнительную таблицу результатов разбиения иерархическим методом и методом k-средних.

Ход работы

Для большего удобства все изображения, приведенные в данной работе, находятся в одном каталоге с файлом отчета, поэтому при невозможности на рисунке что-либо различить, просьба открыть оригинальный файл.

Для реализации приведенных заданий нам потребуются следующие библиотеки языка Python, приведенные в листинге 1.

Листинг 1 — Библиотеки и модули

from scipy.spatial import distance  
from sklearn.metrics.pairwise import uclidean\_distances, pairwise\_distances\_argmin\_min, pairwise\_distances  
from scipy.cluster.hierarchy import linkage, dendrogram, fcluster  
from sklearn.cluster import Kmeans  
from sklearn.datasets import make\_blobs  
import matplotlib.pyplot as plt  
import scipy.cluster.hierarchy as shc  
import numpy as np  
import warnings

Загрузим выборку, состоящую из 150 точек в соответствие с вариантом, и отобразим данные на графике одним цветом. Код приведен в листинге 2, визуализация — на рисунке 1.

Листинг 2 — Загрузка и визуализация выборки

# Генерация синтетических данных «blobs»  
x, y = make\_blobs(n\_samples=n\_samples, random\_state=random\_state, cluster\_std=cluster\_std, centers=centers)  
  
# Визуализация сгенерированных данных без привязки к классам  
plt.figure('Визуализация сгенерированных данных без привязки к классам')  
plt.scatter(x[:, 0], x[:, 1], c='b')  
plt.show()

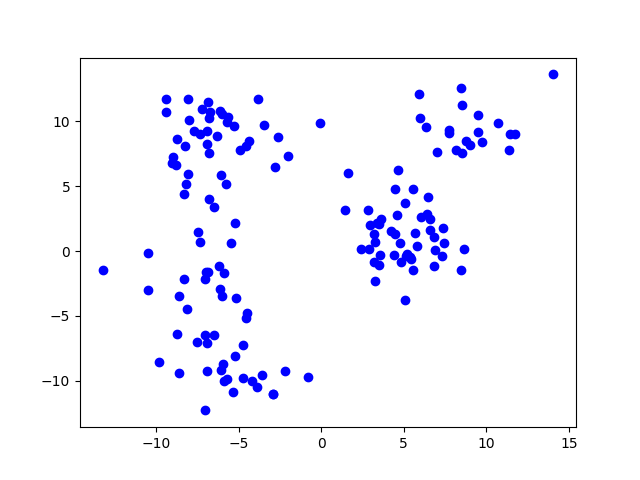


Рисунок 1 — Визуализация сгенерированных данных без привязки к классам

Проведем иерархическую кластеризацию выборки, используя разные способы вычисления расстояния между кластерами: расстояние ближайшего соседа (single), дальнего соседа (complete), Уорда (Ward). Построим дендрограммы для каждого способа. Код, реализующий иерархическую кластеризацию приведенными методами и визуализацию представлен на листинге 3. Дендрограммы приведены на рисунках 2 — 4.

Листинг 3 — Иерархическая кластеризация различными методами

# Создание подзаголовков для дендрограмм  
methods = ['single', 'complete', 'ward']  
  
# Создание и отрисовка дендрограмм для каждого способа  
for i, method in enumerate(methods):  
 plt.figure(f»Dendrogram ({method} linkage)», figsize=(10, 5))  
 plt.title(f»Dendrogram ({method} linkage)»)  
 dend = shc.dendrogram(shc.linkage(x, method=method))  
 plt.show()

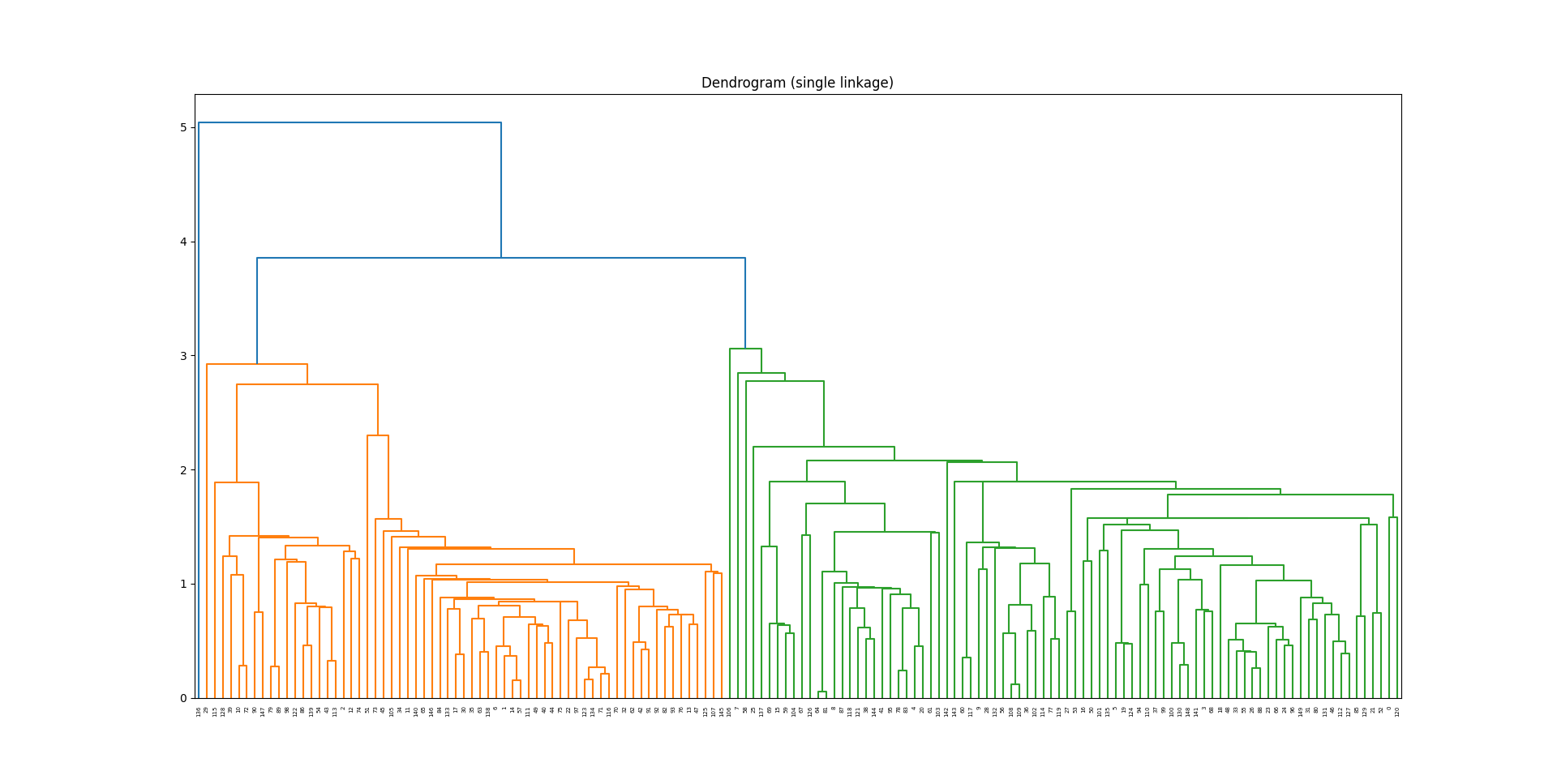


Рисунок 2 — Дендрограмма (single linkage)

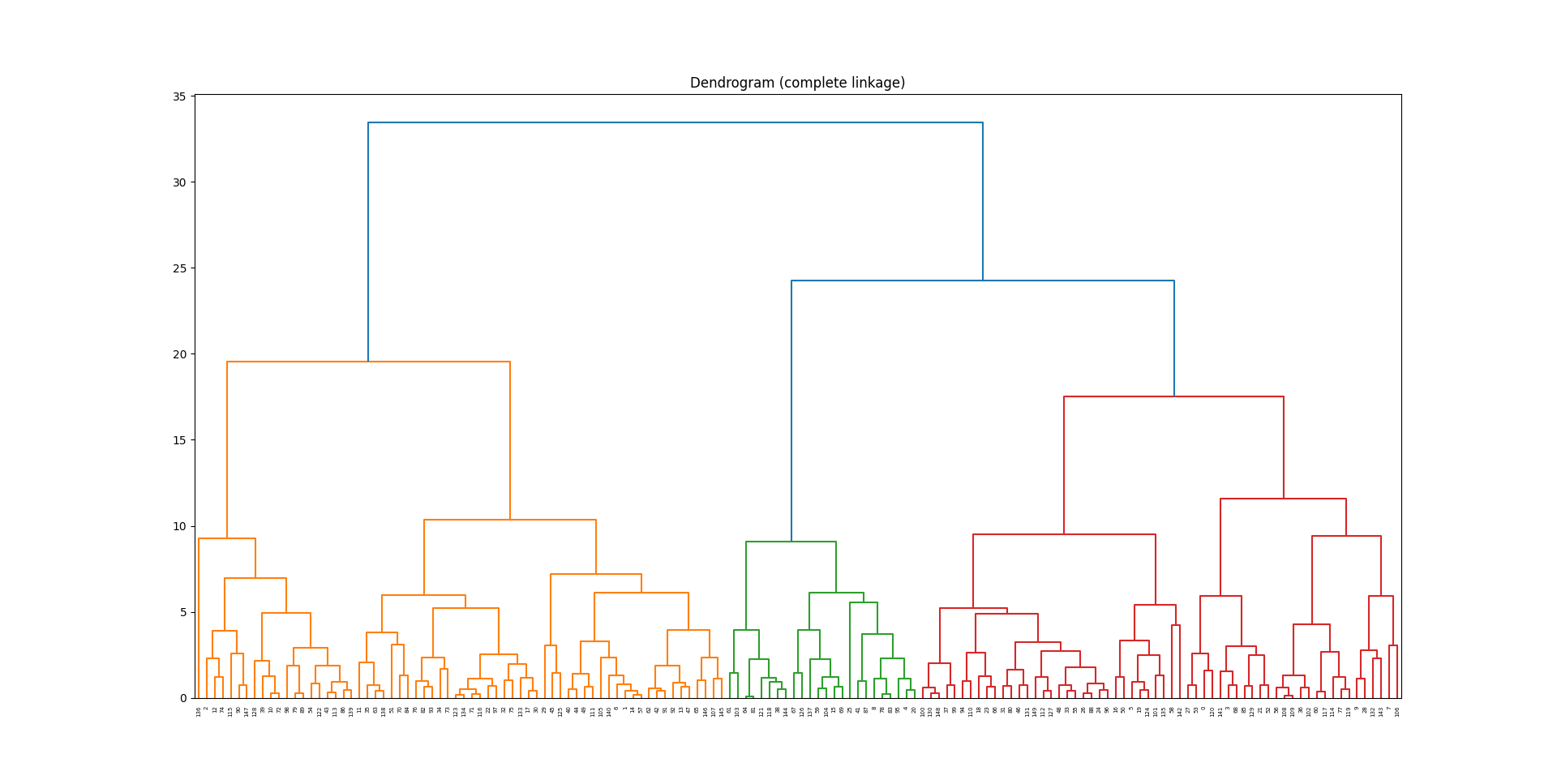


Рисунок 3 — Дендрограмма (complete linkage)

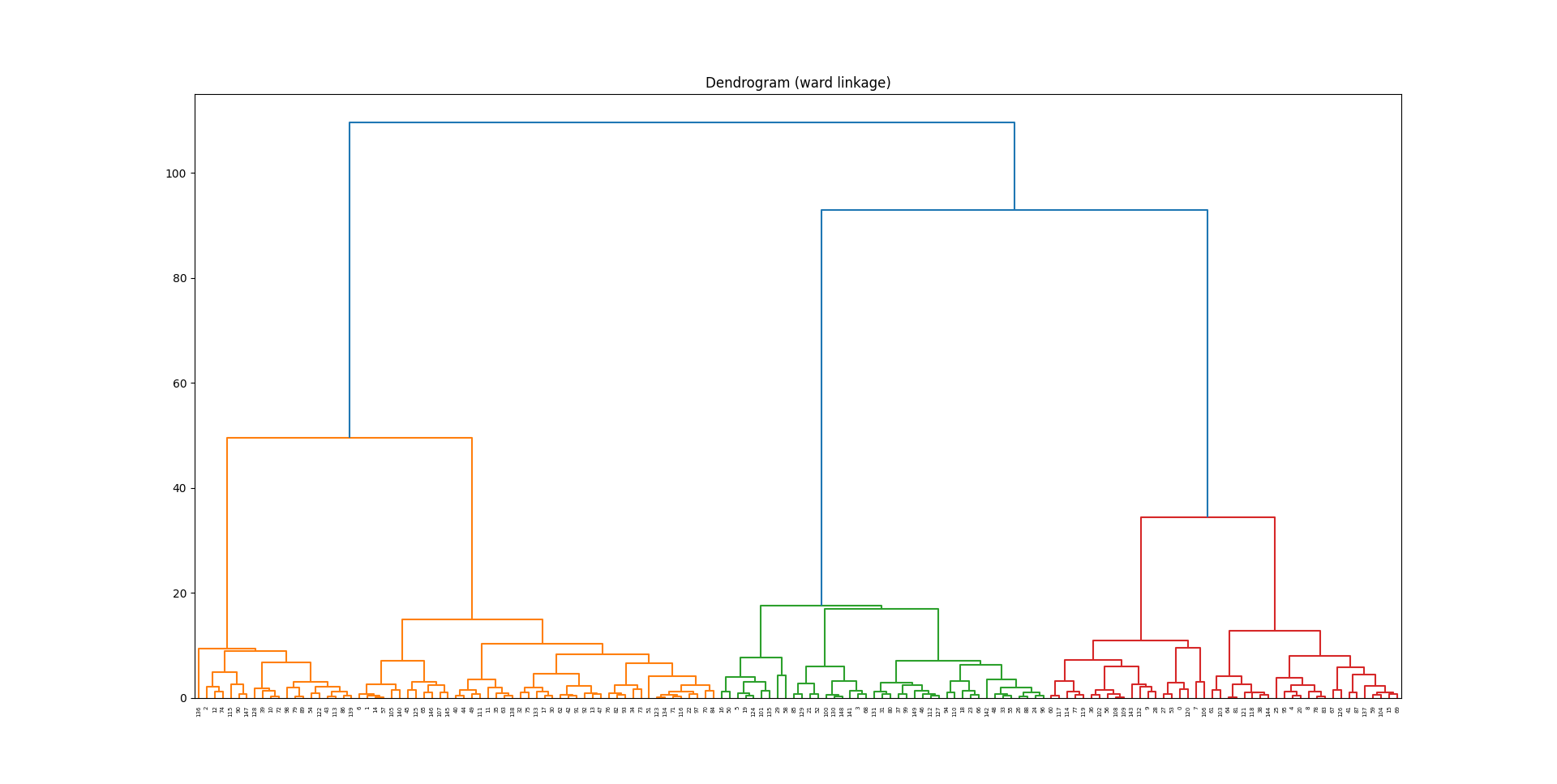


Рисунок 4 — Дендрограмма (ward linkage)

Исходя из дендрограмм лучшими способами вычисления расстояния являются метод дальнего соседа и Уорда. В дальнейшем будем использовать метод дальнего соседа.

Отобразим разбиение на кластеры и центроиды на графике в пространстве признаков. Код, реализующий визуализацию представлен на листинге 4. График распределения точек по кластерам с изображением центроидов приведен на рисунке 5.

Листинг 4 — Визуализация разбиения

Z = linkage(x, method='complete')  
k = 3  
labels = fcluster(Z, k, criterion='maxclust')  
centroids = np.zeros((k, x.shape[1]))  
for i in range(1, k + 1):  
 centroids[i - 1, :] = np.mean(x[labels == i, :], axis=0)  
  
# Предположим, что x - это матрица объектов (признаки), labels - вектор с метками кластеров, centroids - матрица  
# центроидов  
  
# создаем график  
plt.figure(figsize=(8, 6))  
  
# отображаем объекты выборки  
for i in np.unique(labels):  
 plt.scatter(x[labels == i, 0], x[labels == i, 1], label='Cluster ' + str(i))  
  
# отображаем центроиды  
plt.scatter(centroids[:, 0], centroids[:, 1], s=300, c='black', marker='x', label='Centroids')  
  
# добавляем легенду  
plt.legend()  
plt.show()

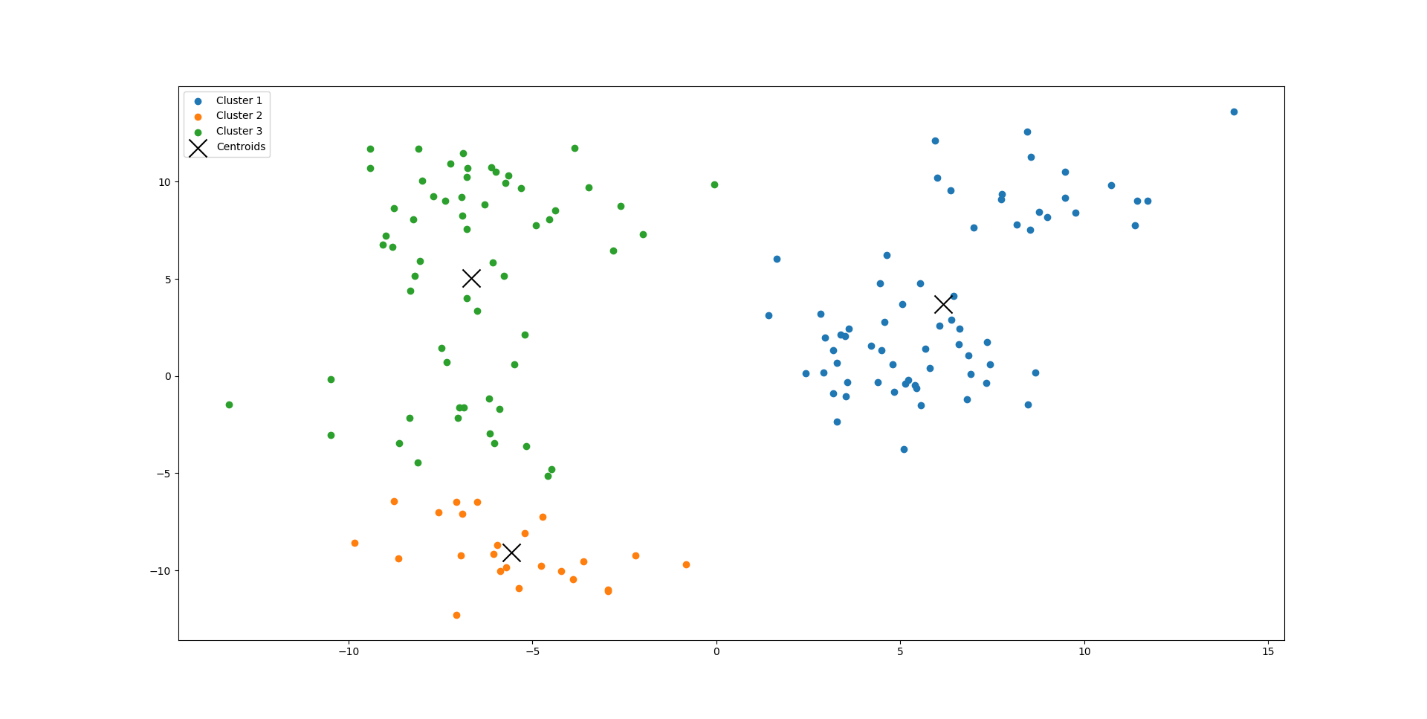


Рисунок 5 — Визуализация разбиения на кластеры (Figure 1)

Рассчитаем среднюю сумму квадратов расстояний до центроида, среднюю сумму средних внутрикластерных расстояний и среднюю сумму межкластерных расстояний для данного разбиения (лист. 5). Результаты работы данной части кода приведены на рисунке 6. По данным параметрам можно сказать, что кластеризация разнородных данных вполне удовлетворительна, хоть и не является идеальной.

Листинг 5 — Расчёт заданных параметров

# Рассчитаем среднюю сумму квадратов расстояний до центроида  
intra\_cluster\_distances = np.zeros(k)  
for I in range(1, k + 1):  
 centroid = centroids[I – 1]  
 cluster\_points = x[labels == I, :]  
 distances = distance.cdist(cluster\_points, [centroid], ‘euclidean’)  
 intra\_cluster\_distances[I – 1] = np.sum(distances \*\* 2) / distances.shape[0]  
  
average\_intra\_cluster\_distance = np.mean(intra\_cluster\_distances)  
  
# Рассчитаем среднюю сумму внутрикластерных расстояний  
inter\_cluster\_distances = np.zeros((k, k))  
for I in range(1, k + 1):  
 for j in range(1, k + 1):  
 distance\_matrix = distance.cdist(x[labels == I, :], x[labels == j, :], ‘euclidean’)  
 inter\_cluster\_distances[I – 1, j – 1] = np.mean(distance\_matrix)  
  
average\_inter\_cluster\_distance = np.mean(inter\_cluster\_distances)  
  
# Рассчитаем среднюю сумму межкластерных расстояний  
within\_cluster\_distances = np.zeros(k)  
for I in range(1, k + 1):  
 for j in range(1, k + 1):  
 if I != j:  
 distance\_matrix = distance.cdist(x[labels == I, :], x[labels == j, :], ‘euclidean’)  
 within\_cluster\_distances[I – 1] += np.sum(distance\_matrix) / (  
 distance\_matrix.shape[0] \* distance\_matrix.shape[1])  
  
average\_within\_cluster\_distance = np.mean(within\_cluster\_distances)  
  
# Выведем результаты  
print(“Средняя сумма квадратов расстояний до центроида:”, average\_intra\_cluster\_distance)  
print(“Средняя сумма внутрикластерных расстояний:”, average\_within\_cluster\_distance)  
print(“Средняя сумма межкластерных расстояний:”, average\_inter\_cluster\_distance)

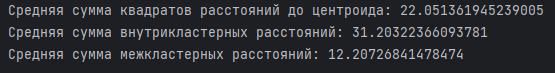


Рисунок 6 — Рассчитанные параметры

Проведем кластеризацию выборки методом k-средних. для k [1, 10]. Сформируем три графика: зависимость средней суммы квадратов расстояний до центроида, средней суммы средних внутрикластерных расстояний и средней суммы межкластерных расстояний от количества кластеров (листинг 6). Графики представлены на рисунке 7.

Листинг 6 — Кластеризация и формирование графиков

# Подготовка списков для сохранения метрик  
avg\_cluster\_distances = []  
avg\_intra\_cluster\_distances = []  
avg\_inter\_cluster\_distances = []  
  
# Проведение кластеризации для различных k от 1 до 10  
for k in range(1, 11):  
 kmeans = KMeans(n\_clusters=k)  
 kmeans.fit(x)  
  
 # Расчет средней суммы квадратов расстояний до центроида  
 avg\_cluster\_distances.append(kmeans.inertia\_ / len(x))  
  
 # Расчет средней суммы средних внутрикластерных расстояний  
 closest, \_ = pairwise\_distances\_argmin\_min(x, kmeans.cluster\_centers\_)  
 avg\_intra\_cluster\_distances.append(  
 np.mean(np.min(pairwise\_distances(x, kmeans.cluster\_centers\_, metric='euclidean')[closest], axis=1)))  
  
 # Расчет средней суммы межкластерных расстояний  
 avg\_inter\_cluster\_distances.append(kmeans.score(x) / len(x))  
  
# Построение графиков  
plt.figure('Addiction', figsize=(15, 5))  
  
# Зависимость средней суммы квадратов расстояний до центроида  
plt.subplot(1, 3, 1)  
plt.plot(range(1, 11), avg\_cluster\_distances, marker='o')  
plt.xlabel('Number of clusters')  
plt.ylabel('Average cluster distances')  
plt.title('Average cluster distances vs. Number of clusters')  
  
# Зависимость средней суммы средних внутрикластерных расстояний  
plt.subplot(1, 3, 2)  
plt.plot(range(1, 11), avg\_intra\_cluster\_distances, marker='o')  
plt.xlabel('Number of clusters')  
plt.ylabel('Average intra-cluster distances')  
plt.title('Average intra-cluster distances vs. Number of clusters')  
  
# Зависимость средней суммы межкластерных расстояний  
plt.subplot(1, 3, 3)  
plt.plot(range(1, 11), avg\_inter\_cluster\_distances, marker='o')  
plt.xlabel('Number of clusters')  
plt.ylabel('Average inter-cluster distances')  
plt.title('Average inter-cluster distances vs. Number of clusters')  
  
plt.show()

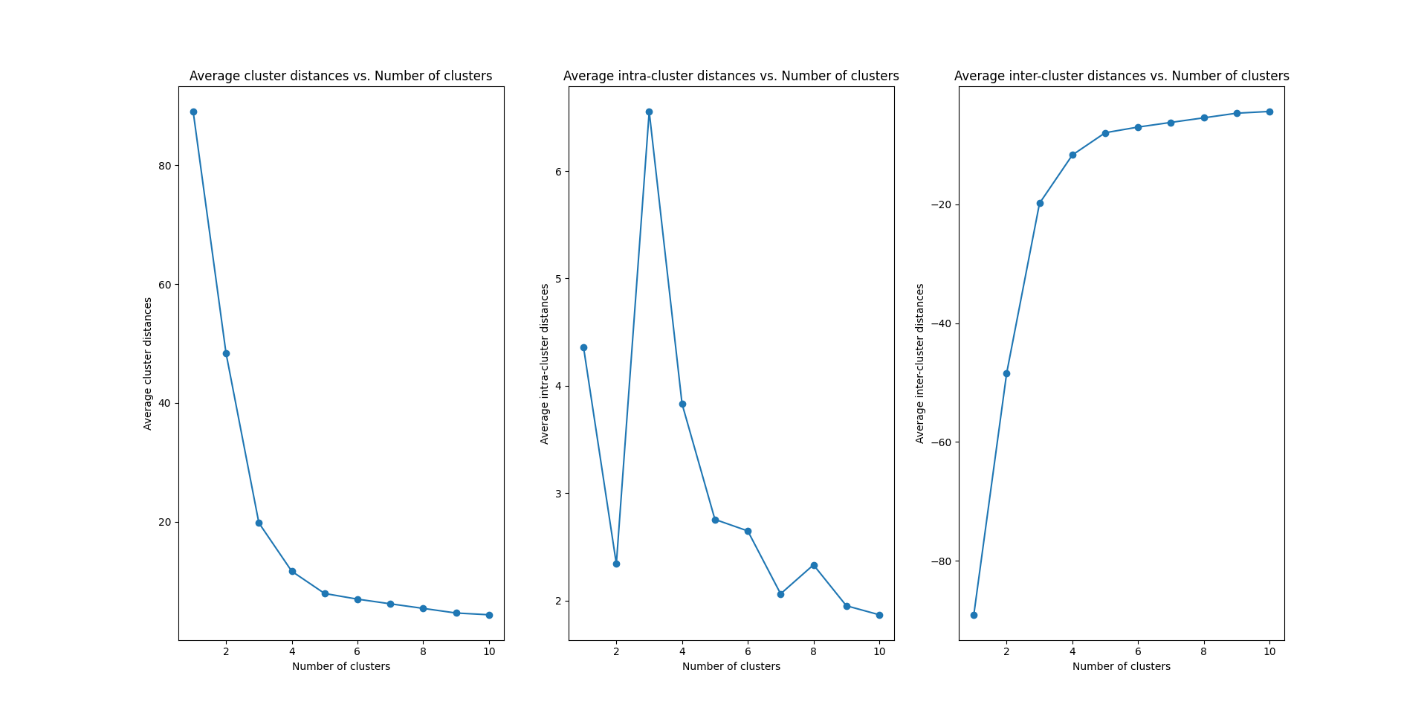


Рисунок 6 — Графики зависимостей (Addiction)

Вывод

Мы получили практические навыки решения задачи кластеризации фактографических данных, научились настраивать параметры методов и оценивать точность полученного разбиения.