|  |  |  |
| --- | --- | --- |
| Лабораторная работа № 7. Кластеризация | Студент | Бокова О. Д. |
| Группа | ИВТ-363 |
| Преподаватель | Фокин Р.О. |
| Оценка |  |
| Дата | 17.04.2023 |

Задание (одно на выбор)

1. Выполните кластеризацию данных с использованием k-means для

своего или произвольного набора данных из репозитория UC Irvine

Machine Learning Repository. Оцените качество кластеризации в зависимости от числа итераций и значения k. Дайте рекомендации по выбору числа

кластеров.

2. Выполните кластеризацию данных с использованием MeanShift

для своего или произвольного набора данных из репозитория UC Irvine

Machine Learning Repository. Проанализируйте как изменения параметра h

при реализации MeanShift влияют на результат?

Код:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import silhouette\_score

# Загрузка данных

df = pd.read\_csv("biofam.csv", delimiter = ';')

# Удаление столбца id

df.drop(columns=['idhous'], inplace=True)

df.drop(columns=['nat\_1\_02'], inplace=True)

df.drop(columns=['plingu02'], inplace=True)

df.drop(columns=['p02r01'], inplace=True)

df.drop(columns=['p02r04'], inplace=True)

df.drop(columns=['cspfaj'], inplace=True)

df.drop(columns=['cspmoj'], inplace=True)

# Замена значений столбца sex

df['sex'].replace({0: 'woman', 1: 'man'}, inplace=True)

# Обработка пропущенных значений

df.replace(' ', np.nan, inplace=True)

df.dropna(inplace=True)

# Нормализация данных

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(df.drop(columns=['sex']))

# Кластеризация методом k-means

def kmeans\_f(X, n\_clusters):

    kmeans = KMeans(n\_clusters=n\_clusters, random\_state=42)

    kmeans.fit(X)

    return kmeans

# Вычисление индекса силуэта для каждого числа кластеров

def silhouette(X):

    silhouette\_scores = []

    for n\_clusters in range(2, 11):

        kmeans = kmeans\_f(X, n\_clusters)

        silhouette\_scores.append(silhouette\_score(X, kmeans.labels\_))

    return silhouette\_scores

# Визуализация индекса силуэта в зависимости от числа кластеров

silhouette\_scores = silhouette(X\_scaled)

plt.plot(range(2, 11), silhouette\_scores)

plt.xlabel('Number of clusters')

plt.ylabel('Silhouette score')

plt.show()

# Кластеризация методом k-means

n\_clusters = 4

kmeans = kmeans\_f(X\_scaled, n\_clusters)

# Добавление меток кластеров в датафрейм

df['cluster'] = kmeans.labels\_

# Анализ кластеров

means = df.groupby(['cluster']).mean()

print(means)

# График анализа кластеров

fig, ax = plt.subplots(1, len(df.columns)-1, figsize=(20,5))

for i, col in enumerate(df.columns[:-1]):

    for cluster in range(n\_clusters):

        ax[i].hist(df[df['cluster'] == cluster][col], alpha=0.5, label=f'Cluster {cluster}')

        ax[i].set\_title(col)

plt.legend()

plt.show()

Результат:

