Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

ЛАБОРАТОРНАЯ РАБОТА

по теме:

**Кластеризация данных с использованием алгоритма K-средних**

по курсу Методы принятия решений и машинное обучение

Студент:Котюшева Н.Э., 211-361

Преподаватель:Гаврилов А.И., к.т.н.

Москва 2024

**Лабораторная работа 3**

**Кластеризация данных с использованием алгоритма K-средних**

**Цель работы:**

1. Изучение и применение алгоритма k-средних для кластеризации данных.

**Задачи:**

1. Изучение теории кластеризации и алгоритма K-средних.
2. Реализация алгоритма k-средних на языке Python.
3. Применение алгоритма k-средних на реальных данных для выявления групп схожих объектов.
4. Оценка и интерпретация результатов кластеризации.

**Отчет о выполнении**

**1. Теоретическая часть работы**

Кластеризация – это задача разделения набора данных на группы, называемые кластерами, таким образом, чтобы объекты внутри одного кластера были максимально схожи между собой, а объекты из разных кластеров отличались друг от друга. Алгоритм k-средних является одним из наиболее распространенных и простых методов кластеризации. Он основан на итеративном разделении объектов на k кластеров, где k - заданное число кластеров. Процесс кластеризации осуществляется путем минимизации суммарного квадратичного отклонения (SSE) между объектами и центроидами кластеров. Подробный обзор шагов алгоритма k-средних:

Шаг 1: Инициализация центроидов

Выбираются случайные k объектов из набора данных в качестве начальных центроидов.

Центроиды представляют центральные точки каждого кластера.

Шаг 2: Назначение объектов к ближайшим центроидам

Каждый объект назначается к ближайшему центроиду на основе некоторой метрики расстояния (обычно евклидово расстояние).

Объекты, находящиеся ближе к одному центроиду, считаются более похожими друг на друга, чем на объекты, находящиеся ближе к другим центроидам.

Шаг 3: Перерасчет центроидов

Вычисляются новые центроиды путем вычисления среднего значения всех объектов, назначенных к каждому кластеру.

Это обновление центроидов на основе текущего разделения объектов на кластеры.

Шаг 4: Повторение шагов 2 и 3

Процесс назначения объектов к ближайшим центроидам и перерасчета центроидов повторяется до сходимости.

Сходимость достигается, когда изменение центроидов становится незначительным или достигается максимальное число итераций.

Шаг 5: Завершение алгоритма

В результате работы алгоритма каждый объект будет назначен к одному из k кластеров.

Кластеры могут быть интерпретированы как группы объектов, близких друг к другу, а объекты внутри каждого кластера будут более похожи друг на друга, чем на объекты из других кластеров.

Применение алгоритма k-средних может быть полезно в различных областях, например:

Маркетинговые исследования: Кластеризация клиентов позволяет выявить сегменты схожих покупателей для более эффективной маркетинговой стратегии.

Анализ социальных сетей: Кластеризация пользователей позволяет выявить группы схожих интересов или поведения в социальных сетях.

Медицинская диагностика: Кластеризация пациентов может помочь выявить группы схожих заболеваний для более точной диагностики и лечения.

**2. Практическая часть**

В ходе лабораторной работы был загружен датасет с набором данных об ирисах: длина чашелистика, ширина чашелистика, длина лепестка, ширина лепестка и разновидность.

На начальном этапе производилась предобработка данных.

Затем была реализована функция, которая реализует алгоритм k-средних. В ней реализован итеративный процесс пересчета центроидов и перераспределение объектов по кластерам. Для измерения расстояния между объектами используется Евклидово расстояние.

Далее реализованный алгоритм k-средних применяется на подготовленных данных. Определяется оптимальное число кластеров k=3.

Результат кластеризации представляется на графике. Также выводится коэффициент силуэта.

**3. Результаты**

Результат кластеризации представлен на рисунке 1.

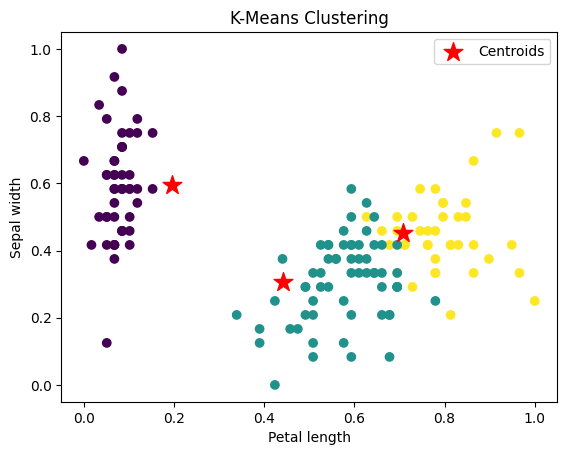


Рисунок 1 – Результат кластеризации

Анализ полученных результатов показывает, что коэффициент силуэта равен 0.76 (насколько данные в одном кластере отличаются от данных в другом). Данные были поделены на три кластера. Каждая точка соответствует одному цветку, а её цвет обозначает принадлежность к определённому кластеру. Красные звезды отмечают положение центроидов этих кластеров.

**Вывод:** При выполнении лабораторной работы был исследован и использован алгоритм K-средних для кластеризации данных. Предварительно данные были обработаны путем масштабирования. Результаты кластеризации были визуализированы, а качество оценено с помощью коэффициента силуэта.

Полученные результаты демонстрируют, что алгоритм K-средних успешно выявляет группы похожих объектов, однако требуется тщательный выбор количества кластеров и предварительная обработка данных.

Коэффициент силуэта показывает, что кластеры сформировались неплохо, хотя и не идеально, что может объясняться выбранным числом кластеров или спецификой самих данных.

В общем, алгоритм K-средних оказался простым и эффективным методом для кластеризации данных.

**Приложение**

**4. Код**

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

import numpy as np

from sklearn.metrics import silhouette\_score

# Загрузка данных из CSV

data = pd.read\_csv('iris.csv')

# Масштабирование данных

# Масштабирует данные так, чтобы они лежали в диапазоне [0, 1].

# Это важно для алгоритма K-средних, так как он основан на расстоянии.

scaler = MinMaxScaler()

scaled\_data = scaler.fit\_transform(data.drop(['variety'], axis=1))

scaled\_data = pd.DataFrame(scaled\_data, columns=['sepal.length', 'sepal.width', 'petal.length', 'petal.width'])

def kmeans(data, k, max\_iterations=100):

    n\_samples, n\_features = data.shape

    centroids = data[np.random.choice(n\_samples, k, replace=False)]

    for \_ in range(max\_iterations):

        distances = np.sqrt(((data - centroids[:, np.newaxis])\*\*2).sum(axis=2))

        labels = np.argmin(distances, axis=0)

        new\_centroids = np.array([data[labels == i].mean(axis=0) for i in range(k)])

        if np.allclose(centroids, new\_centroids):

            break

        centroids = new\_centroids

    return labels, centroids

labels, centroids = kmeans(scaled\_data.values, 3, 100)

# Добавление меток кластеров к исходным данным

# Добавляет новый столбец 'cluster' в DataFrame data, содержащий метки кластеров.

scaled\_data['cluster'] = labels

plt.scatter(scaled\_data['petal.length'], scaled\_data['sepal.width'], c=scaled\_data['cluster'], cmap='viridis')

plt.scatter(centroids[:, 0], centroids[:, 1], marker='\*', s=200, c='red', label='Centroids')

plt.xlabel('Petal length')

plt.ylabel('Sepal width')

plt.title('K-Means Clustering')

plt.legend()

plt.show()

silhouette\_avg = silhouette\_score(scaled\_data, labels) # насколько данные в одном кластере отличаются от данных в другом

print(f"Коэффициент силуэта: {silhouette\_avg}")

**5. Контрольные вопросы**

1. Что такое кластеризация и для чего она используется в машинном обучении?

Кластеризация – это задача разделения набора данных на группы, называемые кластерами, таким образом, чтобы объекты внутри одного кластера были максимально схожи между собой, а объекты из разных кластеров отличались друг от друга. Алгоритм k-средних является одним из наиболее распространенных и простых методов кластеризации. Он основан на итеративном разделении объектов на k кластеров, где k - заданное число кластеров. Процесс кластеризации осуществляется путем минимизации суммарного квадратичного отклонения (SSE) между объектами и центроидами кластеров.

2. Как работает алгоритм k-средних?

Подробный обзор шагов алгоритма k-средних:

Шаг 1: Инициализация центроидов

Выбираются случайные k объектов из набора данных в качестве начальных центроидов.

Центроиды представляют центральные точки каждого кластера.

Шаг 2: Назначение объектов к ближайшим центроидам

Каждый объект назначается к ближайшему центроиду на основе некоторой метрики расстояния (обычно евклидово расстояние).

Объекты, находящиеся ближе к одному центроиду, считаются более похожими друг на друга, чем на объекты, находящиеся ближе к другим центроидам.

Шаг 3: Перерасчет центроидов

Вычисляются новые центроиды путем вычисления среднего значения всех объектов, назначенных к каждому кластеру.

Это обновление центроидов на основе текущего разделения объектов на кластеры.

Шаг 4: Повторение шагов 2 и 3

Процесс назначения объектов к ближайшим центроидам и перерасчета центроидов повторяется до сходимости.

Сходимость достигается, когда изменение центроидов становится незначительным или достигается максимальное число итераций.

Шаг 5: Завершение алгоритма

В результате работы алгоритма каждый объект будет назначен к одному из k кластеров.

Кластеры могут быть интерпретированы как группы объектов, близких друг к другу, а объекты внутри каждого кластера будут более похожи друг на друга, чем на объекты из других кластеров.

3. Как выбрать оптимальное число кластеров k в алгоритме k-средних?

Для выбора оптимального значения k можно использовать следующие методы:

- Метод локтя: строится график, показывающий зависимость суммы квадратов расстояний от центроидов (inertia) от числа кластеров K. Оптимальным считается значение K, соответствующее точке "локтя" на графике, где снижение inertia начинает замедляться.

- Коэффициент силуэта: рассчитывается коэффициент силуэта, который оценивает, насколько правильно объекты распределены по своим кластерам. Оптимальное значение K определяется там, где коэффициент силуэта достигает своего максимального значения.

4. Какие метрики используются для оценки качества кластеризации?

Сумма квадратов расстояний до центроидов (SSE): меньшее значение указывает на лучшую кластеризацию.

Коэффициент силуэта: оценивает степень принадлежности каждого объекта к своему кластеру. Значения варьируются от -1 до 1, причем близость к 1 означает лучшее распределение.