Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Лабораторная работа №5**

Дисциплина: «Системы искусственного интеллекта»

«Алгоритмы кластеризации данных»

Вариант 12

Выполнил студент

группы ИВТАПбд-41

Меховников Е.А.

Проверил:

ассистент кафедры «ВТ»

Хайруллин И. Д.

Ульяновск, 2024

**Цель работы**

1. Произвести масштабирование признаков (scaling).
2. С использованием библиотеки sklearn написать программу с использованием алгоритмов кластеризации данных, позволяющую разделить исходную выборку на классы, соответствующие предложенной вариантом задаче.
3. Провести эксперименты и определить наилучший алгоритм кластеризации, параметры алгоритма. Необходимо использовать не менее 3-х алгоритмов. Данные экспериментов необходимо представить в отчете (графики, ход проведения эксперимента, выводы).

# Ход работы

Варианту 12 соответствует датасет «NYSK». Вначале была подключена библиотека scikit-learn, затем были загружены данные из XML-файла. Текстовые данные были преобразованы в числовые с помощью метода TF-IDF. После этого данные были масштабированы с использованием StandardScaler, а затем была уменьшена размерность с помощью метода главных компонент (PCA), чтобы получить двухмерное представление данных для визуализации.

*Листинг 1. Подготовка данных*

|  |
| --- |
| import xml.etree.ElementTree as ET  import pandas as pd  from sklearn.feature\_extraction.text import TfidfVectorizer  from sklearn.preprocessing import StandardScaler  from sklearn.decomposition import PCA  # Чтение данных из XML файла  tree = ET.parse('nysk.xml')  root = tree.getroot()  documents = []  for doc in root.findall('document'):  text = doc.find('text').text  documents.append(text)  # Создание DataFrame  df = pd.DataFrame(documents, columns=['text'])  # Векторизация текстов с использованием TF-IDF  vectorizer = TfidfVectorizer(stop\_words='english', max\_features=1000)  X = vectorizer.fit\_transform(df['text'])  # Масштабирование данных  scaler = StandardScaler(with\_mean=False)  X\_scaled = scaler.fit\_transform(X)  # Преобразование данных для визуализации с использованием PCA  X\_dense = X\_scaled.toarray()  pca = PCA(n\_components=2)  X\_pca = pca.fit\_transform(X\_dense) |

Для группировки данных были применены три алгоритма кластеризации: KMeans, AgglomerativeClustering и DBSCAN. Качество полученных кластеров оценивалось с помощью метрики Silhouette Score.

KMeans - это популярный алгоритм кластеризации, который разбивает набор данных на k групп (кластеров), минимизируя расстояние между объектами внутри каждого кластера. Алгоритм начинается с случайного выбора k центров кластеров, а затем каждый объект данных назначается к ближайшему центру. (Рис. 1).

Agglomerative Clustering - это иерархический метод кластеризации, который строит дерево кластеров (дендрограмму) путем последовательного объединения объектов. Алгоритм начинается с того, что каждый объект данных считается отдельным кластером. Затем, на основе близости, кластеры последовательно объединяются в более крупные кластеры, пока количество объектов не достигнет заданного числа (Рис. 2).

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) - это алгоритм кластеризации, который использует плотность данных для определения кластеров. Он может находить кластеры произвольной формы и справляться с шумом в данных. DBSCAN работает с двумя ключевыми параметрами: eps (максимальное расстояние между точками, которые должны быть объединены в один кластер) и min\_samples (минимальное количество точек, необходимое для формирования региона). (Рис. 3).

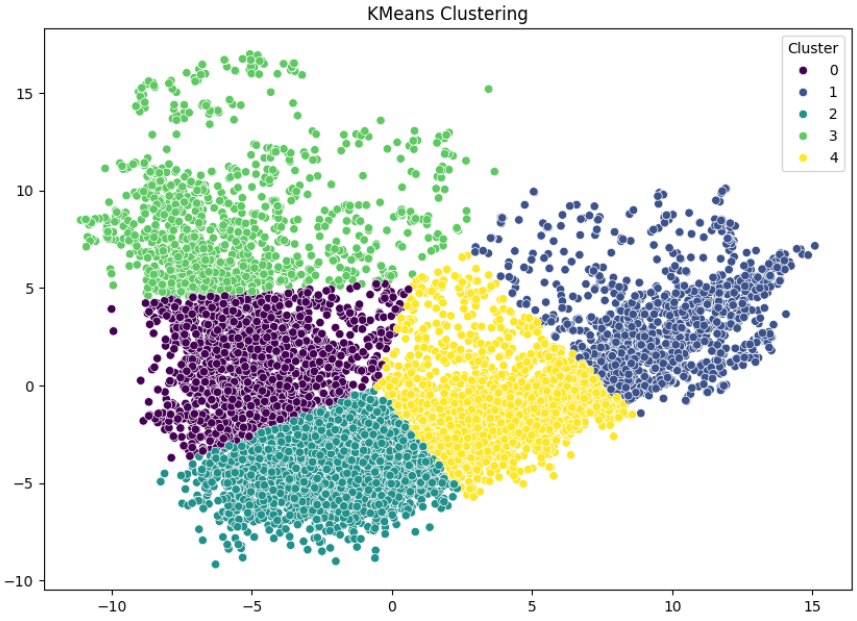
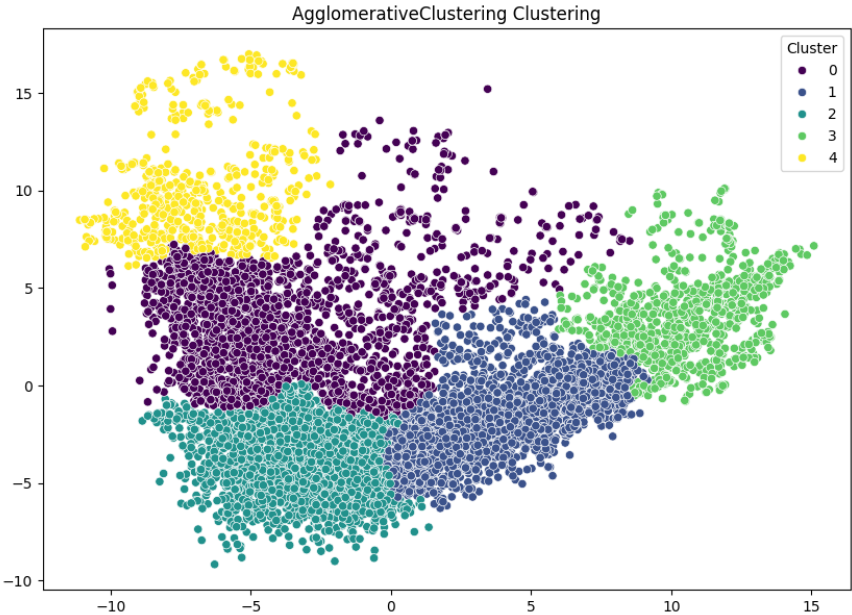
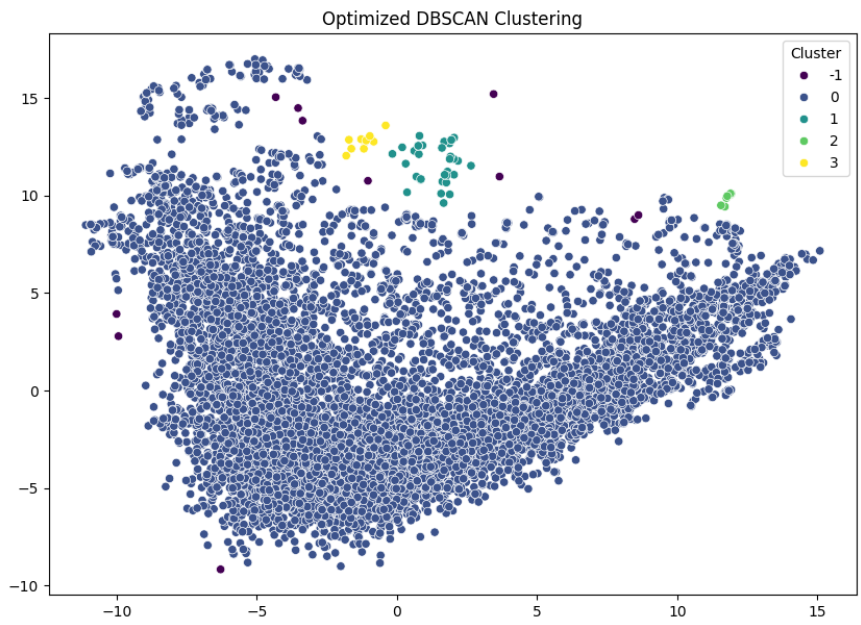
Рис. 1 – KMeans

Рис. 2 – Agglomerative Clustering

Рис. 3 – DBSCAN

**Вывод**

В этой работе проводилась кластеризация текстовых данных с использованием различных алгоритмов. Результаты экспериментов показали, что алгоритм KMeans является наиболее эффективным для данного набора данных. Он достиг наивысшего значения Silhouette Score, что свидетельствует о том, что он лучше всего сгруппировал данные в кластеры.