|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | | | |
|  | Институт информационных технологий (ИТ) |
|  | Кафедра прикладной математики |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 6** | |
| **по дисциплине «Технологии и инструментарий анализа больших данных»** | |
|  | |
| Выполнил студент группы ИКБО-20-21 | Сидоров С.Д. |
| Проверил ассистент кафедры ПМ ИИТ | Тетерин Н.Н. |

Москва 2024

**Практическая работа**

1. Найти данные для кластеризации. Данные в группе не должны повторяться. Если признаки в данных имеют очень сильно разные масштабы, то необходимо данные предварительно нормализовать.

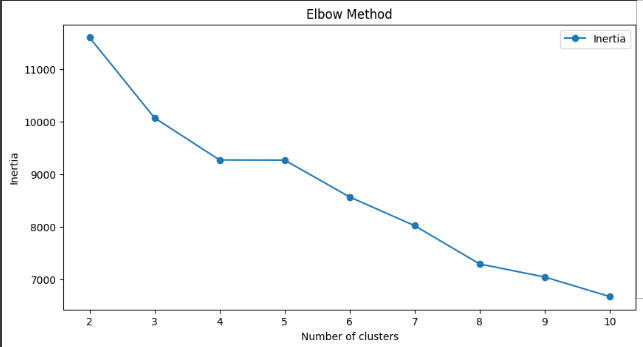
Листинг 1:

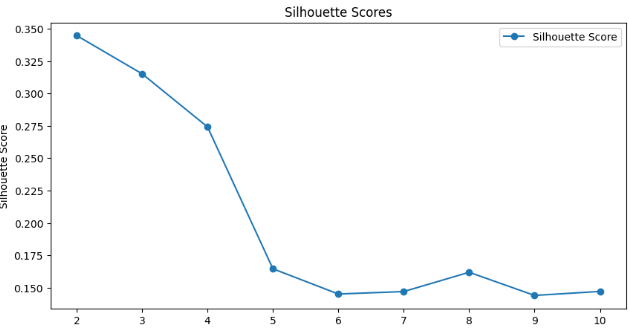
|  |
| --- |
| cancer = datasets.load\_breast\_cancer()  X = cancer.data # признаки  y = cancer.target # метки классов (не используем для кластеризации)  # Нормализация данных  scaler = StandardScaler()  X\_scaled = scaler.fit\_transform(X)  # Преобразуем в DataFrame для наглядности  df = pd.DataFrame(X\_scaled, columns=cancer.feature\_names)  df.head() |

1. Провести кластеризацию данных с помощью алгоритма k-means. Использовать «правило локтя» и коэффициент силуэта для поиска оптимального количества кластеров.

Листинг 2:

|  |
| --- |
| # Поиск оптимального числа кластеров по "правилу локтя"  inertia = []  silhouette\_scores = []  cluster\_range = range(2, 11)  for k in cluster\_range:  kmeans = KMeans(n\_clusters=k, random\_state=42)  kmeans.fit(X\_scaled)  inertia.append(kmeans.inertia\_)  silhouette\_scores.append(silhouette\_score(X\_scaled, kmeans.labels\_))  # Визуализация "правила локтя"  plt.figure(figsize=(10, 5))  plt.plot(cluster\_range, inertia, marker='o', label='Inertia')  plt.title('Elbow Method')  plt.xlabel('Number of clusters')  plt.ylabel('Inertia')  plt.legend()  plt.show()  # Визуализация коэффициента силуэта  plt.figure(figsize=(10, 5))  plt.plot(cluster\_range, silhouette\_scores, marker='o', label='Silhouette Score')  plt.title('Silhouette Scores')  plt.xlabel('Number of clusters')  plt.ylabel('Silhouette Score')  plt.legend()  plt.show()  # Кластеризация с оптимальным числом кластеров (на основе графиков)  optimal\_k = 2 # например, если оптимально 2 кластера  kmeans = KMeans(n\_clusters=optimal\_k, random\_state=42)  kmeans\_labels = kmeans.fit\_predict(X\_scaled) |

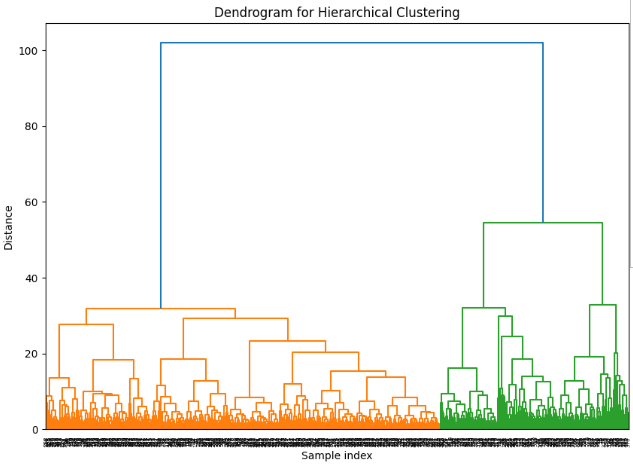




1. Провести кластеризацию данных с помощью алгоритма иерархической кластеризации

Листинг 3:

|  |
| --- |
| # Иерархическая кластеризация  linkage\_matrix = linkage(X\_scaled, method='ward')  plt.figure(figsize=(10, 7))  dendrogram(linkage\_matrix)  plt.title('Dendrogram for Hierarchical Clustering')  plt.xlabel('Sample index')  plt.ylabel('Distance')  plt.show() |



1. Провести кластеризацию данных с помощью алгоритма DBSCAN

Листинг 4:

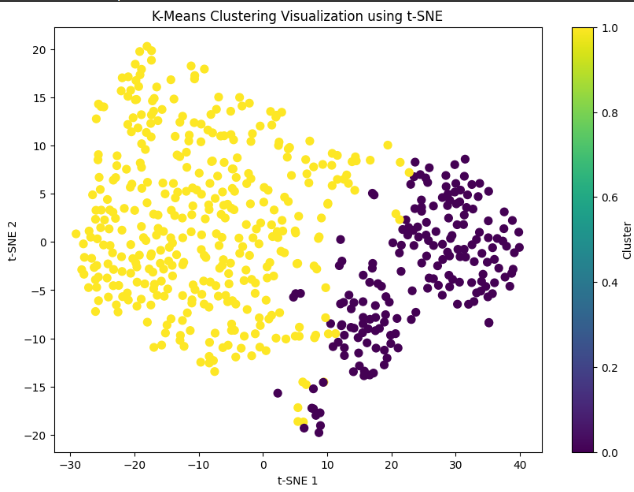
|  |
| --- |
| # Настройка DBSCAN (eps и min\_samples нужно подобрать эмпирически)  dbscan = DBSCAN(eps=2.0, min\_samples=4)  dbscan\_labels = dbscan.fit\_predict(X\_scaled)  # Количество уникальных кластеров  n\_clusters\_dbscan = len(set(dbscan\_labels)) - (1 if -1 in dbscan\_labels else 0)  print(f'Количество кластеров, найденных DBSCAN: {n\_clusters\_dbscan}') |

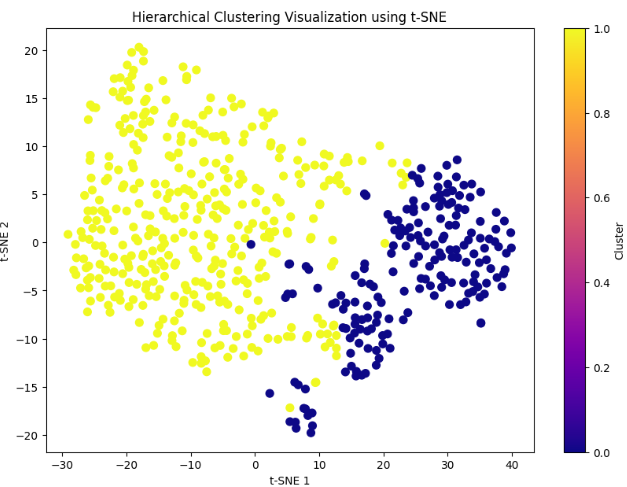


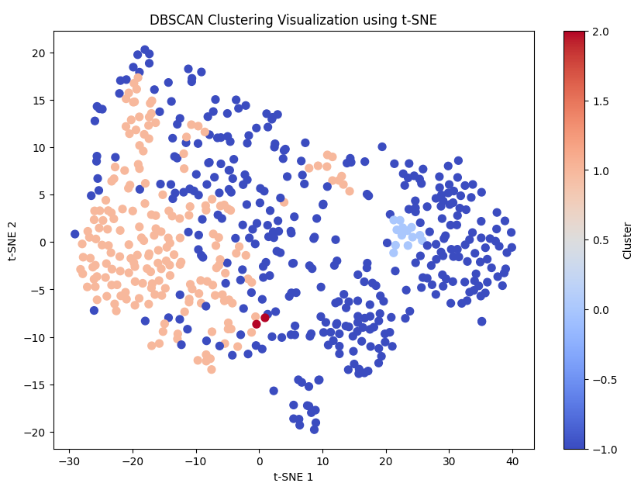
1. Визуализировать кластеризованные данные с помощью t-SNE или UMAP, если необходимо. Если данные трехмерные, то можно использовать трехмерный точечный график

Листинг 5:

|  |
| --- |
| # Преобразование данных с помощью t-SNE до 2-х измерений  tsne = TSNE(n\_components=2, random\_state=42)  X\_tsne = tsne.fit\_transform(X\_scaled)  # Визуализация кластеров для k-means  plt.figure(figsize=(10, 7))  plt.scatter(X\_tsne[:, 0], X\_tsne[:, 1], c=kmeans\_labels, cmap='viridis', s=50)  plt.title('K-Means Clustering Visualization using t-SNE')  plt.xlabel('t-SNE 1')  plt.ylabel('t-SNE 2')  plt.colorbar(label='Cluster')  plt.show()  # Визуализация кластеров для иерархической кластеризации  plt.figure(figsize=(10, 7))  plt.scatter(X\_tsne[:, 0], X\_tsne[:, 1], c=hierarchical\_labels, cmap='plasma', s=50)  plt.title('Hierarchical Clustering Visualization using t-SNE')  plt.xlabel('t-SNE 1')  plt.ylabel('t-SNE 2')  plt.colorbar(label='Cluster')  plt.show()  # Визуализация кластеров для DBSCAN  plt.figure(figsize=(10, 7))  plt.scatter(X\_tsne[:, 0], X\_tsne[:, 1], c=dbscan\_labels, cmap='coolwarm', s=50)  plt.title('DBSCAN Clustering Visualization using t-SNE')  plt.xlabel('t-SNE 1')  plt.ylabel('t-SNE 2')  plt.colorbar(label='Cluster')  plt.show() |







**Выводы:**

### 1. K-Means кластеризация + "Правило локтя" и коэффициент силуэта

"Правило локтя" (Elbow Method):

* На графике отображаются значения инерции (сумма квадратов расстояний между точками и их центроидами) в зависимости от количества кластеров.
* График выглядит как локоть (или колено), и оптимальное число кластеров находится в точке "перелома", где инерция начинает уменьшаться медленнее. Это число указывает на оптимальное количество кластеров.
* На этом графике мы можем выбрать оптимальное число кластеров — например, 3 кластера, если видим резкий перелом в этой точке.

Коэффициент силуэта (Silhouette Score):

* Этот график показывает, насколько хорошо объекты внутри одного кластера похожи друг на друга и насколько они отличаются от объектов из других кластеров.
* Чем выше коэффициент силуэта, тем лучше выполнена кластеризация. Оптимальное число кластеров — это то, при котором значение силуэта максимально.

### 2. Иерархическая кластеризация и дендрограмма

Дендрограмма:

* Этот график показывает дерево объединений данных на основе их схожести. Объекты данных начинают на "листьях" дерева, а по мере того, как схожие объекты объединяются, они поднимаются вверх по дереву.
* По дендрограмме можно определить оптимальное количество кластеров, "обрезав" дерево на определенном уровне по вертикали. Например, если мы хотим 3 кластера, мы "разрезаем" дерево на высоте, где остается ровно три ветви.

### 3. DBSCAN кластеризация

DBSCAN:

* DBSCAN группирует объекты на основе плотности: он объединяет точки, которые находятся близко друг к другу (по определенному радиусу eps).
* На графике с результатами DBSCAN мы можем увидеть кластеры разных цветов, а также шумовые точки (которые не принадлежат никакому кластеру) как точки с отдельным цветом (обычно -1).
* Этот метод полезен для обнаружения кластеров сложной формы и выявления аномалий в данных.

### 4. Визуализация кластеров с помощью t-SNE

* t-SNE — это метод уменьшения размерности, который помогает визуализировать высокоразмерные данные (такие как данные iris) в 2D-пространстве.
* Визуализация с t-SNE позволяет увидеть, как данные распределены по кластерам. На каждом графике для методов кластеризации (K-Means, иерархическая кластеризация, DBSCAN) мы видим, как объекты сгруппированы в 2D-пространстве.
* Различные кластеры окрашены в разные цвета, и можно визуально оценить, насколько данные разделены. Если точки внутри одного кластера сгруппированы плотнее, а между кластерами есть явные границы, это означает, что кластеризация выполнена хорошо.

### Что можно понять по графикам?

**K-Means**: На графике t-SNE для K-Means мы увидим 3 четких кластера, которые будут представлять три различных вида ирисов.

**Иерархическая кластеризация**: На графике t-SNE для иерархической кластеризации мы можем увидеть похожее распределение, однако форма и границы кластеров могут отличаться из-за того, что метод работает иначе.

**DBSCAN**: На графике t-SNE для DBSCAN можно увидеть кластеры, которые DBSCAN распознал, а также шумовые точки, которые не попали ни в один из кластеров. (Синие = (-1) это шумовые точки, как было найдено в 4 пункте у нас два кластера: белые и красные точки)