МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования

**"Южно-Уральский государственный университет**

**(национальный исследовательский университет)"**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**ОТЧЕТ**о выполнении лабораторной работы № 5

по дисциплине  
«Технологии аналитической обработки информации»

|  |  |
| --- | --- |
|  | Выполнил: студент группы КЭ-403  Разуев Г.А.  Проверил:  Преподаватель кафедры СП  Гоглачев А.И. |

Челябинск-2025

ОГЛАВЛЕНИЕ

[1. ЗАДАНИЕ 2](#_Toc191729743)

[2. РЕАЛИЗАЦИЯ ПРОГРАММЫ 3](#_Toc191729744)

[3. ЭКСПЕРИМЕНТЫ 5](#_Toc191729745)

# ЗАДАНИЕ

1. Разработайте программу, которая выполняет кластеризацию заданного набора данных с помощью алгоритмов k‑Means и k-Medoids. Параметрами программы являются набор данных и число кластеров. Программа должна выдавать координаты точек и назначенные им кластера, а также значение ошибки кластеризации.
2. Проведите эксперименты на наборе данных customers (сведения о клиентах банка: скачать zip-архив с данными в формате CSV и описанием).
3. Выполните визуализацию полученных результатов в следующем виде:
   1. точечный график, на котором цвет точки отражает принадлежность кластеру;
   2. зависимость ошибки кластеризации от параметра k.
4. Доработайте программу, добавив в список ее параметров долю зашумленных объектов набора. Дополнительно к ранее реализованным функциям программа должна вносить шум в набор данных: случайным образом изменить заданную долю объектов набора.
5. Проведите эксперименты на ранее выбранных наборах данных, варьируя долю зашумленных объектов (1%, 3%, 5%, 10%) и используя различные значения параметра k (из интервала 3..9).
6. Выполните визуализацию полученных результатов указанным выше способом.
7. Подготовьте отчет о выполнении задания и загрузите отчет в формате PDF в систему. Отчет должен представлять собой связный и структурированный документ со следующими разделами:
   1. формулировка задания;
   2. гиперссылка на каталог репозитория с исходными текстами, наборами данных и др. сопутствующими материалами;
   3. рисунки с результатами визуализации;
   4. пояснения, раскрывающие смысл полученных результатов.

# РЕАЛИЗАЦИЯ ПРОГРАММЫ

Реализация программы представлена в классе «ClusteringEstimator», который инкапсулирует логику по обучению и использованию алгоритмов кластеризации, а также по выбору оптимального количества кластеров для алгоритма и зашумлении данных при обучении. Один из ключевых атрибутов класса – алгоритм кластеризации (KMeans или KMedoids). Данный класс реализует стандартный интерфейс Scikit-Learn (имеет методы «fit» и «predict»), однако помимо данных метод «fit» имеет параметр доли зашумленных данных «noise\_fraction», что позволяет внедрять шум в данные при необходимости. Для выбора оптимального колчичества кластеров в классе реализован статический метод «choose\_optimal\_n\_clusters», который обучает несколько вариаций указанного алгоритма с переданными данными со всеми возможными значениями параметра k из переданного диапазона, для каждой вариации считает метрику качества для выбранного метода анализа («inertia» для метода локтя и показатель силуэта для метода силуэта). Далее метод визуализирует зависимость метрики качества от значения параметра k. Также данный класс реализует метод «plot\_clusters\_with\_tsne», который позволяет снизить размерность исходных данных до двух осей и визуализировать их, присвоив цвета в соответствии с метками кластеров.

Код реализованной программы и всех проведенных экспериментов находится в репозитории по ссылке https://github.com/ke103rga/analytics.

Программа была протестирована полном наборе данных «customers» Данные предварительно были отмасштабированы, для повышения качества кластеризации. Код подбора оптимального количества кластеров представлен в листинге 1. Результат представлен на рисунке 1.

Листинг 1 – Код подбора оптимального количества кластеров

ClusteringEstimator.choose\_optimal\_n\_clusters(

    X=scaled\_X, n\_clusters\_range=[3, 15],

    method='elbow', algorithm='k-means'

)

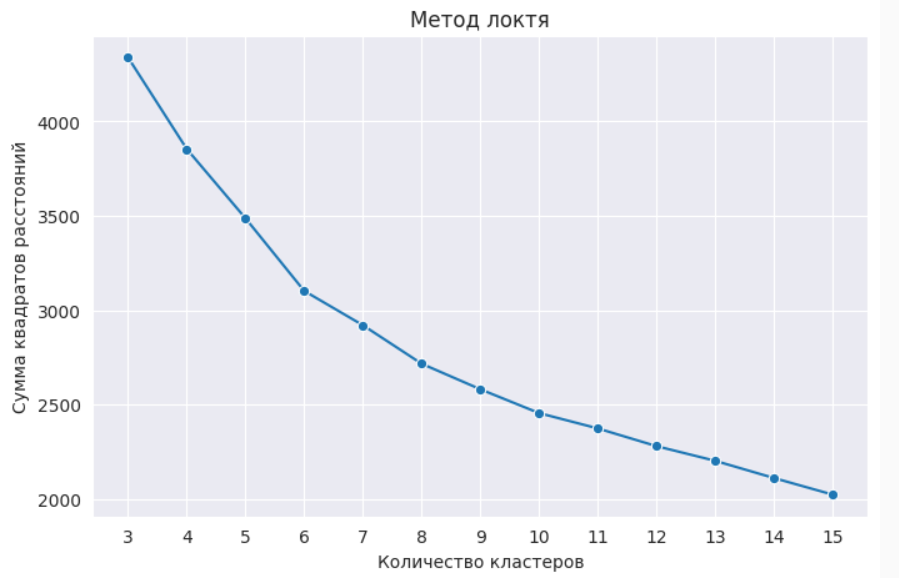


Рисунок 1 – Подбор оптимального значения k

В данном случае метод локтя не позволял сделать однозначный вывод о значении параметра k, поэтому дополнительно был применен метод силуэта. В итоге было выбрано значение k = 8. Код обучения алгоритма и визуализации меток кластеров представлен в листинге 2. Результат представлен на рисунке 2.

Листинг 2 – Код обучения алгоритма и визуализации меток кластеров

kmeans = ClusteringEstimator(algorithm='k-means', n\_clusters=8)

kmeans.fit(scaled\_X)

kmeans.plot\_clusters\_with\_tsne()

)

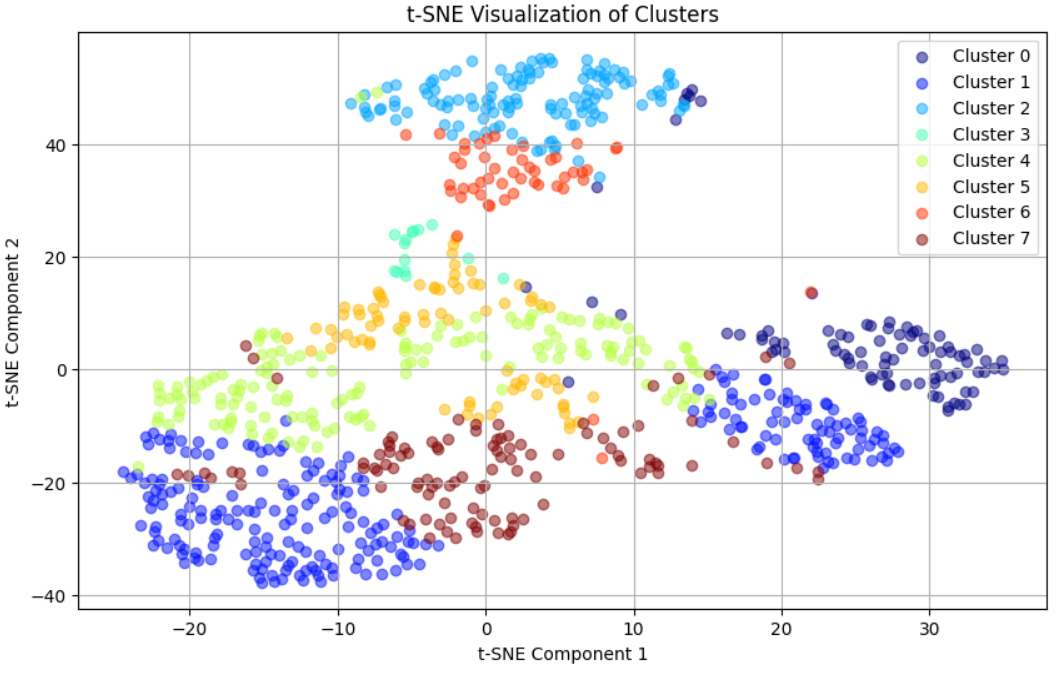


Рисунок 2 – Визуализации меток кластеров

# ЭКСПЕРИМЕНТЫ

Были проведены эксперименты на ранее выбранных наборах данных, варьируя долю зашумленных объектов (1%, 3%, 5%, 10%) и используя различные значения параметра k (из интервала 3..9). Код проведения эксперимента представлен в листинге 3.

Листинг 3 – Код проведения экспериментов

for noise\_fraction in noise\_fractions:

    for n\_clusters in n\_clusters\_range:

*# K-Means*

        kmeans\_estimator = ClusteringEstimator(n\_clusters, 'k-means')

        kmeans\_estimator.fit(pd.DataFrame(scaled\_X),noise\_fraction)

        results.append({

            'method': 'kmeans',

            'n\_clusters': n\_clusters,

            'noise\_fraction': noise\_fraction,

            'inertia': kmeans\_estimator.inertia

        })

*# K-Medoids*

        kmedoids\_estimator = ClusteringEstimator(n\_clusters, 'k-medoids')

        kmedoids\_estimator.fit(pd.DataFrame(scaled\_X), noise\_fraction)

        results.append({

            'method': 'kmedoids',

            'n\_clusters': n\_clusters,

            'noise\_fraction': noise\_fraction,

            'inertia': kmedoids\_estimator.inertia

        })

На основании проведенных экспериментов была выполнена визуализация результатов с использованием линейной диаграммы, на которой ось X – значение параметра k, ось Y – метрика качества кластеризации. Для каждого алгоритма была построена своя сравнительные диаграммы, где каждая линия – значения метрики качества при конкретной доли зашумленности данных, которые представлены на рисунках 3, 4.

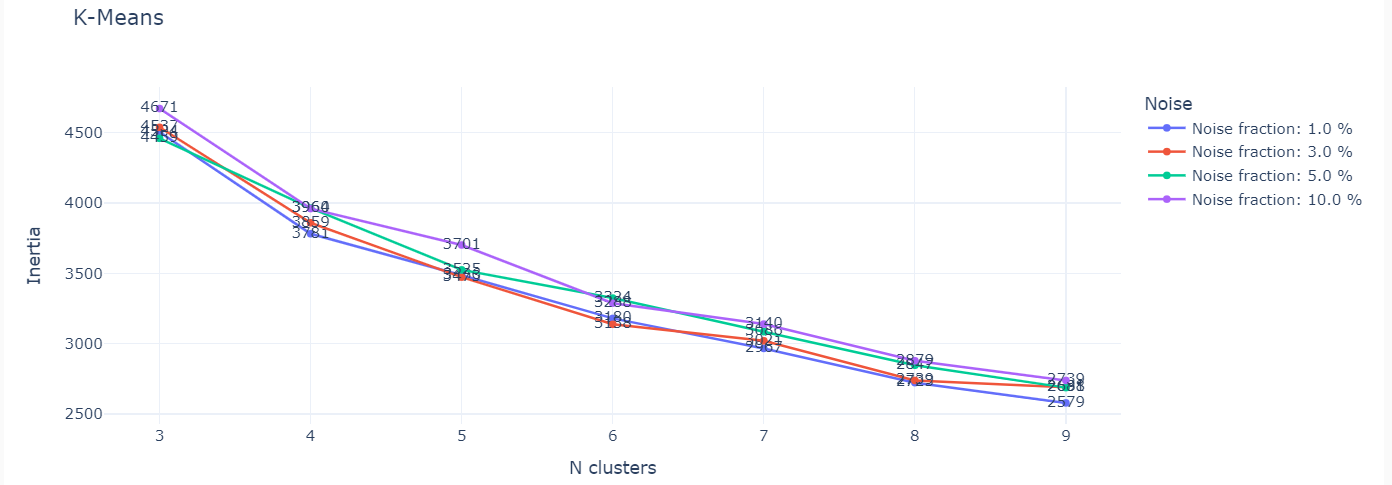


Рисунок 3 – Сравнение качества кластеризации для «K-Means»

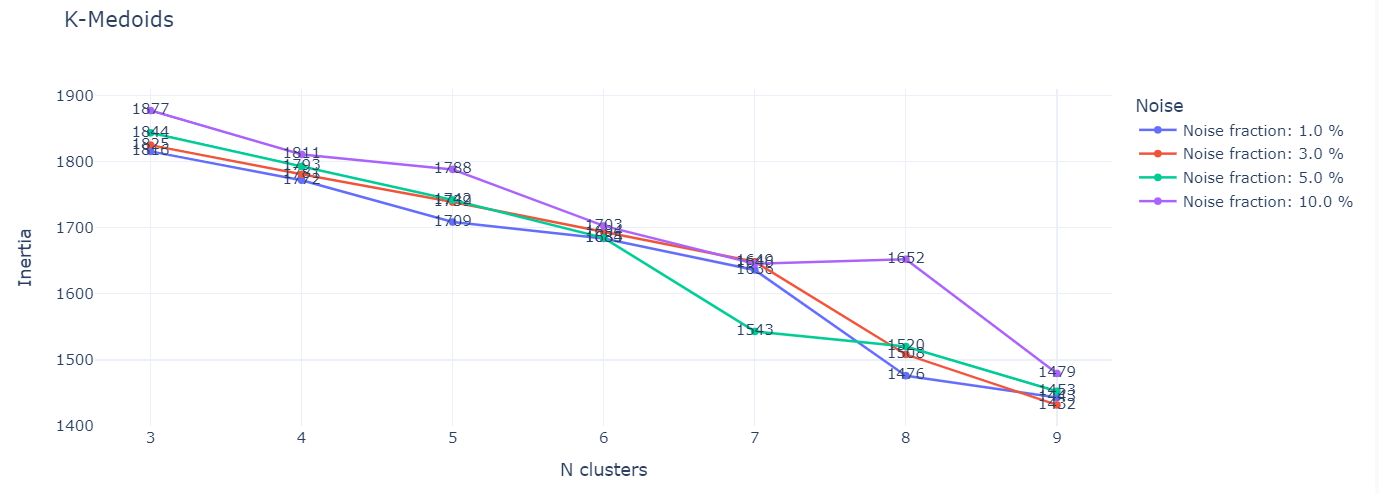


Рисунок 4 – Сравнение качества кластеризации для «K-Medoids»

Из анализа данных графиков можно сделать ряд выводов:

1. Почти для любого значения параметра k актуально что чем выше доля зашумленности данных чем хуже показатели качества (исключение составляет всего одна комбинация параметров), что говорит о том, что шум в данных снижает качество кластеризации;
2. Однако отличие в значении метрики качества чаще всего очень небольшое (особенно если считать в долях стандартного отклонения ошибки), что может свидетельствовать либо о том, что зашумленность слабо влияет на качество кластеризации, либо что данные изначально были зашумлены. Однозначно сформулировать этот вывод можно проведя тесты на других наборах данных и замеряя уровень статистической значимости между метриками качества;
3. Показатели качества у алгоритма «K-Medoids» ощутимо лучше при каждом значении параметра k. Чтобы нагляднее это увидеть на рисунке 5 отдельно представлено сравнения метрик качества алгоритмов при доле зашумленности в 1%.

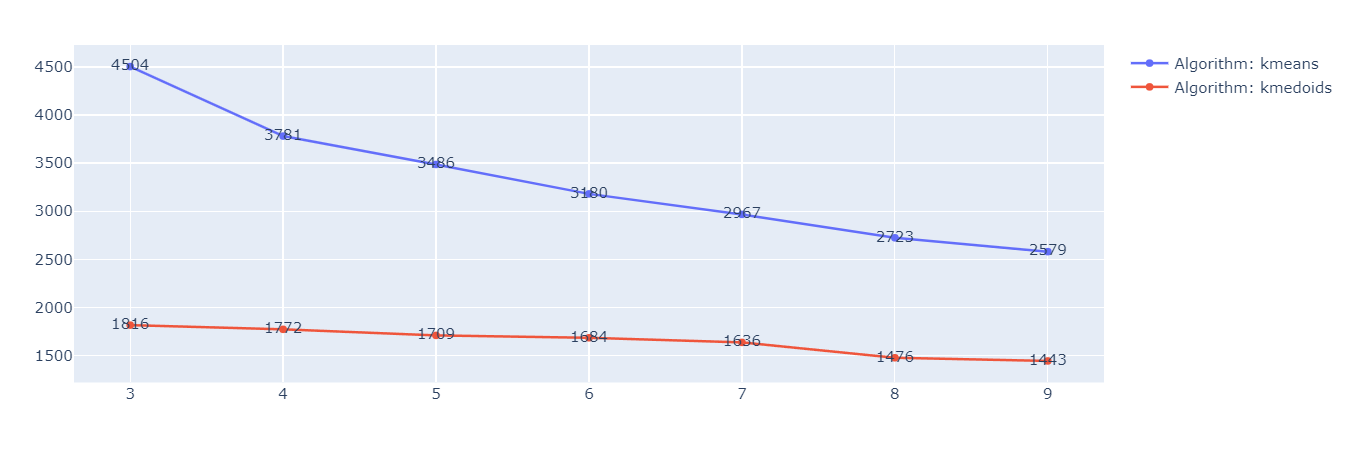


Рисунок 5 – сравнения метрик качества алгоритмов при доле зашумленности в 1%

На этом графике отчетливо видно колоссальное преимущество «K-Medoids» по метрике «inertia», что почти гарантировано является сигналом что для конкретно этих данных оптимальнее всего использовать именно его, но для принятия окончательного решения нужно также провести анализ силуэтов внутри каждого кластера.