

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики-процессов управления

Программа бакалавриата

“Большие данные и распределенная цифровая платформа”

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Алгоритмы и структуры данных»

на тему «Обезличивание датасета»

Вариант – 2

**Студент гр. 23Б15-пу
Сериков К.Г.**

**Преподаватель
Дик А.Г.**

Санкт-Петербург

2024 г.

Оглавление

1. Цель работы.....	3
2. Описание задачи (формализация задачи).....	3
3. Теоретическая часть.....	4
4. Основные шаги программы.....	5
5. Блок схема программы.....	6
6. Описание программы.....	7
7. Рекомендации пользователя.....	8
8. Рекомендации программиста.....	8
9. Исходный код программы.....	8
10. Контрольный пример.....	9
11. Вывод.....	11
12. Источники.....	11

Цель работы

Целью лабораторной работы является разработка системы обезличивания датасета для списка визитов к врачу. При работе с такими данными необходимо учитывать требования по защите персональной информации и возможности восстановления исходных данных.

Теоретическая часть

Основные методы обезличивания данных

Для обезличивания медицинских данных в данной работе используются следующие методы обезличивания:

1. Маскирование данных

Маскирование данных — это метод, при котором часть информации скрывается, оставляя только обобщенные или частичные значения. Этот метод подходит для полей, таких как паспорт и СНИЛС, где важно сохранить структуру данных, но не позволить восстановить точные значения.

2. Локальное обобщение

Локальное обобщение заключается в замене детализированных значений данных на более общие категории или диапазоны. Например, вместо точного стоимости анализов можно указать диапазон (например, 0-3500 руб.).

3. Локальное подавление

Локальное подавление применяется к отдельным строкам данных, где существует риск утечки информации. В этом случае, данные в этих строках могут быть полностью скрыты или удалены, если они представляют угрозу для конфиденциальности.

Оценка уровня анонимности данных

К-анонимность — это метрика, используемая для оценки уровня защиты данных. Её суть заключается в том, что каждый набор данных должен содержать по крайней мере K записей, которые невозможно различить по заданным квази-идентификаторам. Чем больше значение K , тем выше уровень анонимности.

Квази-идентификаторы представляют собой поля, которые по отдельности не уникальны, но в комбинации могут позволить идентифицировать конкретного человека.

Расчёт К-анонимности осуществляется путём группировки данных по квази-идентификаторам и подсчёта количества записей в каждой группе. Если для каких-либо комбинаций квази-идентификаторов количество записей меньше установленного порога K , такие записи считаются недостаточно анонимными.

Основные шаги программы

1. **Запуск программы:** Запуск основного файла (main.py).
2. **Ввод пользователя:** Пользователь поочерёдно выбирает, какие данные необходимо обезличить, отвечая на вопросы (y/n).
3. **Обезличивание ФИО:** Поле ФИО заменяется на пол пациента.
4. **Обезличивание паспортных данных:** Поле СНИЛС маскируется под формат **** *.
5. **Обезличивание СНИЛС:** Поле СНИЛС маскируется под формат ***-**-****.
6. **Обезличивание симптомов:** Симптомы классифицируются как внутренние, внешние или смешанные (при наличии и тех, и других).
7. **Обезличивание врачей:** Врачи распределяются по медицинским отделениям.
8. **Обезличивание анализов:** Остаётся только первый анализ из списка, который классифицируется по категориям.
9. **Обезличивание стоимости:** Стоимость преобразуется в диапазон вместо точного значения.
10. **Обезличивание банковских карт:** Сохраняется только название банка.
11. **Расчёт К-анонимности:** Выполняется группировка данных и расчёт К-анонимности для оценки уровня обезличенности.
12. **Локальное подавление:** Удаляются строки, где К-анонимность ниже 5, если таких строк не более 5% от общего числа записей.
13. **Вывод результата:** Отображаются 5 минимальных значений К-анонимности и процентное соотношение этих записей от общего количества.
14. **Сохранение данных:** Обезличенный набор данных сохраняется в файл anon_dataset.csv.

Блок схема программы

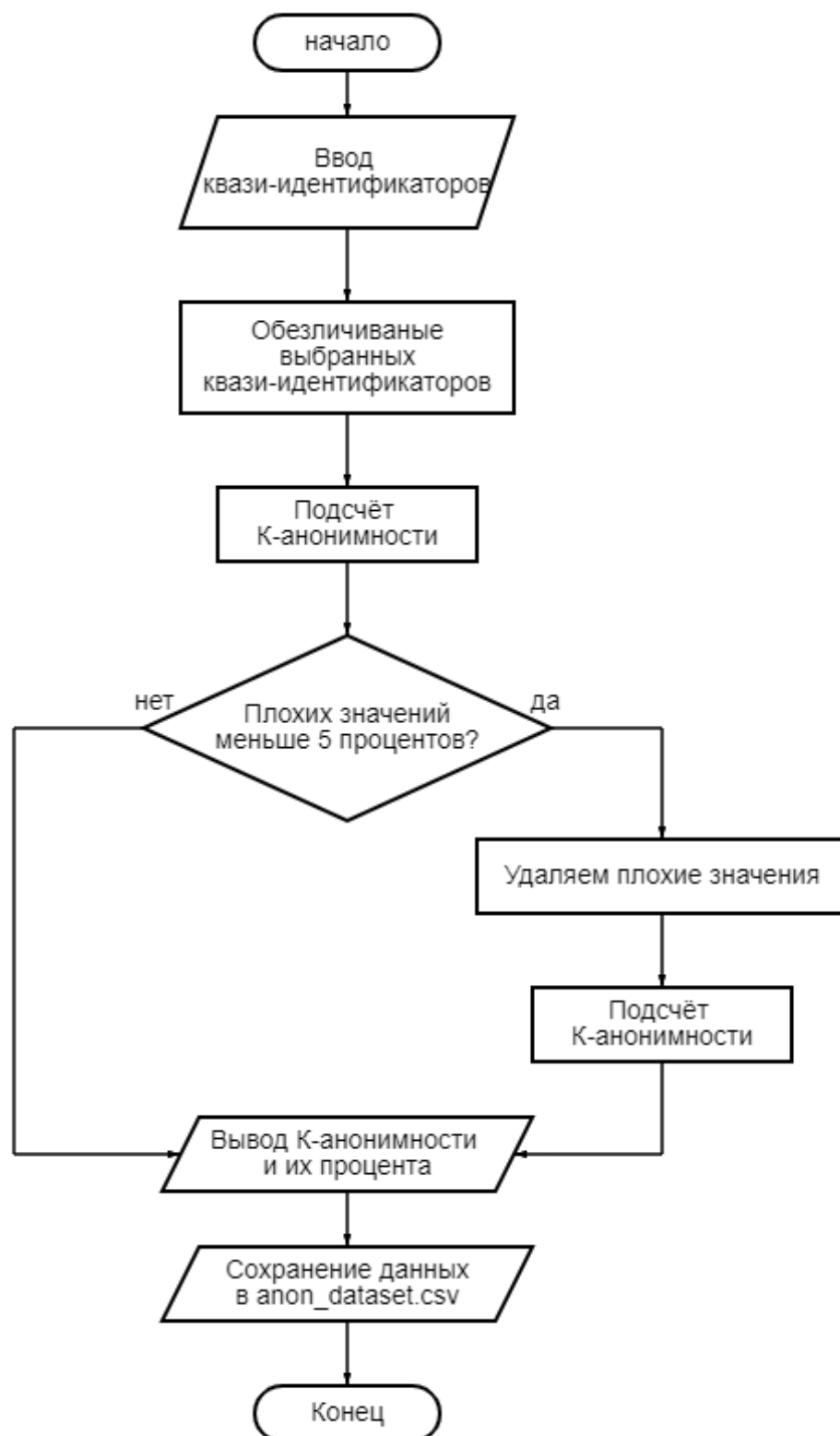


Рис 1. Блок-схема основной программы

Описание программы

Программная реализация написана на языке Python 3.13.0 с использованием библиотеки pandas [1]. Программа нацелена на обезличивания данных для списка визитов к врачу. В процессе разработки программы использовались 2 файла и 13 функций, каждая из которых имеет чётко определённое назначение:

Таблица 1. functions.py

Функция	Описание	Возвращаемое значение
anonymize_fullname	Замена ФИО на пол.	str
anonymize_passport	Маскировка паспорта. Остаётся .	str
anonymize_snils	Обезличивание СНИЛС. Остаётся -- .	str
anonymize_symptoms	Обезличивание симптомов	str
anonymize_doctor	Обезличивание врача. Распределение по отделениям.	str
anonymize_analyses	Обезличивание анализов. Распределение первого по категориям.	str
anonymize_cost	Обезличивание стоимости. Распределение на диапазоны	str
anonymize_card	Обезличивание банковской карты. Остаётся только банк.	str

Таблица 2. main.py

Функция	Описание	Возвращаемое значение
input_quasi_identifiers	Ввод квази-идентификаторов.	str
anonymize_data	Вызов всех функция для обезличивания.	str
calculate_k_anonymity	Подсчёт K-anonymity	str
remove_bad_k_anonymity_records	Локальное подавление	str
print_k_values	Вывод K-anonymity и их процента	str

Рекомендации пользователя

Для запуска программы убедитесь, что у вас установлен Python. Код можно запустить в среде разработки или через командную строку, используя консоль для выбора квази-идентификаторов. Также убедитесь, что все файлы программы находятся в одной директории для корректного выполнения. Запуск программы производится через файл `main.py`. Важно периодически проверять корректность данных перед генерацией походов. Перед запуском убедитесь, что ваш файл `dataset.xml` правильно отформатирован и содержит минимум 50000 строк.

Рекомендации программиста

Поддерживайте актуальную версию Python для обеспечения работоспособности программы на современных системах. Уделяйте внимание четкому именованию переменных и функций. Регулярно проводите тестирование программы на различных входных данных, чтобы убедиться в её надежности и корректности.

Исходный код программы

<https://github.com/romplle/spbu-algorithms-and-data-structures/>

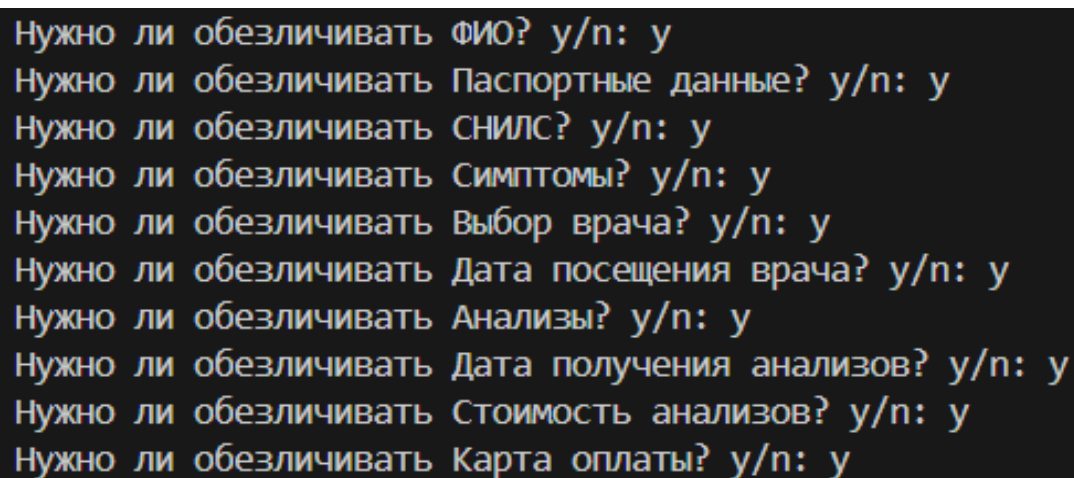
Контрольный пример

1. Запуск программы.

Для запуска программы используйте файл main.py.

2. Выбора квази-идентификаторов.

После запуска программы пользователю предложено выбрать какие квази-идентификаторы нужно обезличить ("ФИО", "Паспортные данные", "СНИЛС", "Симптомы", "Выбор врача", "Дата посещения врача", "Анализы", "Дата получения анализов", "Стоимость анализов", "Карта оплаты") (Рис. 2).

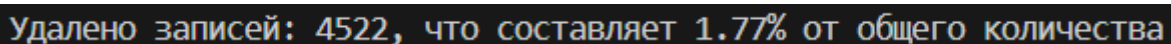


```
Нужно ли обезличивать ФИО? y/n: y
Нужно ли обезличивать Паспортные данные? y/n: y
Нужно ли обезличивать СНИЛС? y/n: y
Нужно ли обезличивать Симптомы? y/n: y
Нужно ли обезличивать Выбор врача? y/n: y
Нужно ли обезличивать Дата посещения врача? y/n: y
Нужно ли обезличивать Анализы? y/n: y
Нужно ли обезличивать Дата получения анализов? y/n: y
Нужно ли обезличивать Стоимость анализов? y/n: y
Нужно ли обезличивать Карта оплаты? y/n: y
```

Рис 2. пример выбора квази-идентификаторов

3. Локальное подавление

Если в таблице остались уникальные значения и их меньше 5 процентов, то запустится функция локального подавления (Рис. 3)



```
Удалено записей: 4522, что составляет 1.77% от общего количества
```

Рис 3. пример ввода количества билетов

4. Вывод результатов

Программа рассчитывает значения К-анонимности и выводит их на экран, чтобы пользователь мог оценить уровень анонимизации данных (Рис. 4).

```
K-anonymity: 5 (0.0863)
K-anonymity: 6 (0.0899)
K-anonymity: 7 (0.0736)
K-anonymity: 8 (0.0692)
K-anonymity: 9 (0.0700)
```

Рис 4. пример вывода К-анонимности и их процента

5. Запись данных

В самом конце программа сохраняет обезличенный датасет в anon_dataset.csv (Рис. 5 и Рис. 6).

```
Данные успешно сохранены в файл 'anon_dataset.csv'.
```

Рис 5. пример сохранения данных

	A	B	C	D	E	F	G	H	I	J
1	ФИО	Паспортные данные	СНИЛС	Симптомы	Выбор врача	Дата посещения врача	Анализы	Дата получения анализов	Стоимость анализов	Карта оплаты
2	M	**** *	***_***_*** **	Смешанные	Терапевтическое		2023 Клинические анализы		2023 0-3500 руб.	Альфа-Банк
3	M	**** *	***_***_*** **	Внутренние	Инфекционное		2024 Другие анализы		2024 0-3500 руб.	Сбербанк
4	M	**** *	***_***_*** **	Смешанные	Терапевтическое		2024 Инструментальные исследования		2024 3501-7000 руб.	Сбербанк
5	M	**** *	***_***_*** **	Внутренние	Инфекционное		2024 Клинические тесты на жидкости		2024 3501-7000 руб.	ВТБ
6	M	**** *	***_***_*** **	Смешанные	Хирургическое		2023 Клинические анализы		2023 3501-7000 руб.	Альфа-Банк
7	Ж	**** *	***_***_*** **	Внутренние	Приемное		2024 Функциональные исследования		2024 7001+ руб.	Т-банк
8	Ж	**** *	***_***_*** **	Неизвестные	Травматологическое		2024 Инструментальные исследования		2024 3501-7000 руб.	Сбербанк
9	Ж	**** *	***_***_*** **	Внутренние	Гериатрическое		2024 Инструментальные исследования		2024 0-3500 руб.	Т-банк
10	M	**** *	***_***_*** **	Неизвестные	Терапевтическое		2023 Клинические анализы		2023 0-3500 руб.	Т-банк

Рис 6. пример обезличенных данных

Вывод

В рамках данной работы были исследованы принципы генерации синтетических данных, применительно к моделированию посещений людей к врачу. Разработан алгоритм, который учитывает особенности врачей, симптомов и анализов. Было реализовано программное обеспечение для автоматической генерации датасета, включающего такие данные, как личные данные пассажиров, информация о симптомах, анализах и платежных системах. Программа позволяет настраивать параметры генерации банковских карт оплаты, обеспечивая соответствие заданным требованиям и реалистичность получаемого датасета.

Источники

1. Pandas documentation // Pandas URL: <https://pandas.pydata.org/docs/> (дата обращения: 10.10.2024).