**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный университет**

**Факультет прикладной математики-процессов управления**

**Кафедра “фундаментальная информатика и информационные технологии”**

**отчет**

**по лабораторной работе №2**

**по дисциплине «Алгоритмы и структуры данных»**

**на тему «обезличивание датасета»**

**Вариант – 1**

|  |  |  |
| --- | --- | --- |
| Студент гр. 23Б16-пу |  | Сулимов А.С. |
| Преподаватель |  | Дик А.Г. |

**Санкт-Петербург**

**2023 г**

**Оглавление**

[Цель работы 3](#_Toc178951981)

[Задача 3](#_Toc178951982)

[Теоретическая часть 3](#_Toc178951983)

[Описание работы программы 5](#_Toc178951984)

[Блок-схема программы 1](#_Toc178951985)

[Описание программы 0](#_Toc178951986)

[Рекомендации пользователя 3](#_Toc178951987)

[Рекомендации для программиста 4](#_Toc178951988)

[Контрольный пример 5](#_Toc178951989)

[Вывод 7](#_Toc178951990)

[Список использованной литературы 7](#_Toc178951991)

# **Цель работы**

Обезличить датасет, в котором хранится информация о покупках в магазинах на территории Санкт-Петербурга для анализа поведения покупателей и их предпочтений в зависимости от различных факторов.

# **Задача**

# Изучить принципы обезличивания данных о покупках. Разработать алгоритм обезличивания датасета, содержащего информацию о покупках в различных магазинах с необходимой степенью защиты от деобезличивания, учитывая объем данных. Разработать алгоритм, который рассчитывает значение k-anonymity. Написать программу, которая по выбранным квази-идентификаторам обезличивает датасет и рассчитывает значение k-anonymity.

# **Теоретическая часть**

Существует несколько методов обезличивания данных: возмущение, микро-агрегация, перемешивание, локальное обобщение, агрегация, удаление атрибутов, локальное подавление, маскеризация, создание псевдонимов, семплинг, декомпозиция, генерация синтетических данных на основе имеющегося набора данных [1].

Чтобы определить, какие методы наиболее эффективны, необходимо проанализировать, какие данные нужно обезличить максимально, а какие целесообразно сохранить в наименее измененном виде. Также нужно ввести метрику для оценки уровня защищенности данных от деобезличивания. В данной работе используется метрика *k-anonymity*. Для расчета этого показателя данные в наборе делятся на группы по определенным атрибутам. Значение *k-anonymity* для группы равно количеству строк в этой группе. Наименьшее из всех значений *k-anonymity* указывает на степень защищенности датасета [2].

Теперь следует определить наиболее чувствительные данные в этом датасете. Это те данные, которые повторяются минимальное количество раз. Для таких данных необходимо уменьшить количество уникальных значений [3]. Например, номер банковской карты является уникальным для каждого покупателя, поэтому такие данные необходимо маскировать для достижения высокой степени защиты. Максимально обезличить данные можно с помощью таких методов, как удаление атрибута (полное удаление данных) и маскеризация (замена значений специальными символами или шаблонами). Для номеров карт в данном случае применяется маскеризация.

Далее, данные, такие как название магазина и категория товаров, могут быть менее чувствительными. Однако для анализа покупательского поведения важна информация о типе магазина и категории товара. Чтобы обезличить эти данные, можно применить метод локального обобщения — замена уникальных значений более обобщенными, но сохраняющими ключевые характеристики. Например, конкретные названия магазинов заменяются на типы магазинов: «электроника», «одежда», «продукты», что сохраняет полезность данных.

Дата и время покупки также полезны для анализа, но являются чувствительными с точки зрения идентификации. Метод возмущения (добавление небольших изменений в данные) позволяет снизить чувствительность этих данных без значительной потери их полезности. Этот метод вносит небольшие случайные изменения во время или дату, сохраняя общую релевантность информации.

Атрибуты, такие как количество товаров и стоимость, также требуют особого подхода. Например, количество товаров в корзине и их стоимость можно обработать с помощью микро-агрегации, когда записи объединяются в группы, и для каждой группы вычисляется агрегированное значение. Это уменьшает количество уникальных значений, сохраняя при этом общие тенденции.

Местоположение магазина — это ключевой атрибут, который нужно сохранить в минимально обезличенном виде. Для этого используется метод локального обобщения: названия улиц заменяются на районы или округа, что сохраняет полезность данных и значительно уменьшает количество уникальных значений, повышая степень защищенности.

Применение этих методов к датасету о покупках позволяет достичь высокой степени защиты данных при сохранении их полезности для последующего анализа покупательского поведения.

# **Описание работы программы**

1. Задаются квази идентификаторы по которым будем происходить обезличивание данных и имена файлов, в которых находится датасета и куда необходимо записать обезличенный датасет.
2. Проверка на использование хотя бы одного атрибута в качестве квази идентификатора и правильность названий файлов.
3. Обезличивание датасета:
   1. Атрибуты «Название магазина», «Количество», «Товар», «Производитель» обезличиваются по методу локального обобщения, если они выбраны в качестве квази идентификаторов.
   2. Атрибуты «Дата и Время», «Карта оплаты» обезличиваются по методу маскеризации, если они выбраны в качестве квази идентификаторов.
   3. Атрибуты «Координаты» обезличиваются по методу возмущения, если они выбраны в качестве квази идентификаторов.
   4. Атрибут «Цена» обезличивается по методу микро-агрегации, если он выбран в качестве квази идентификатора.
4. Запись обезличенных данных в csv файл

# **Блок-схема программы**

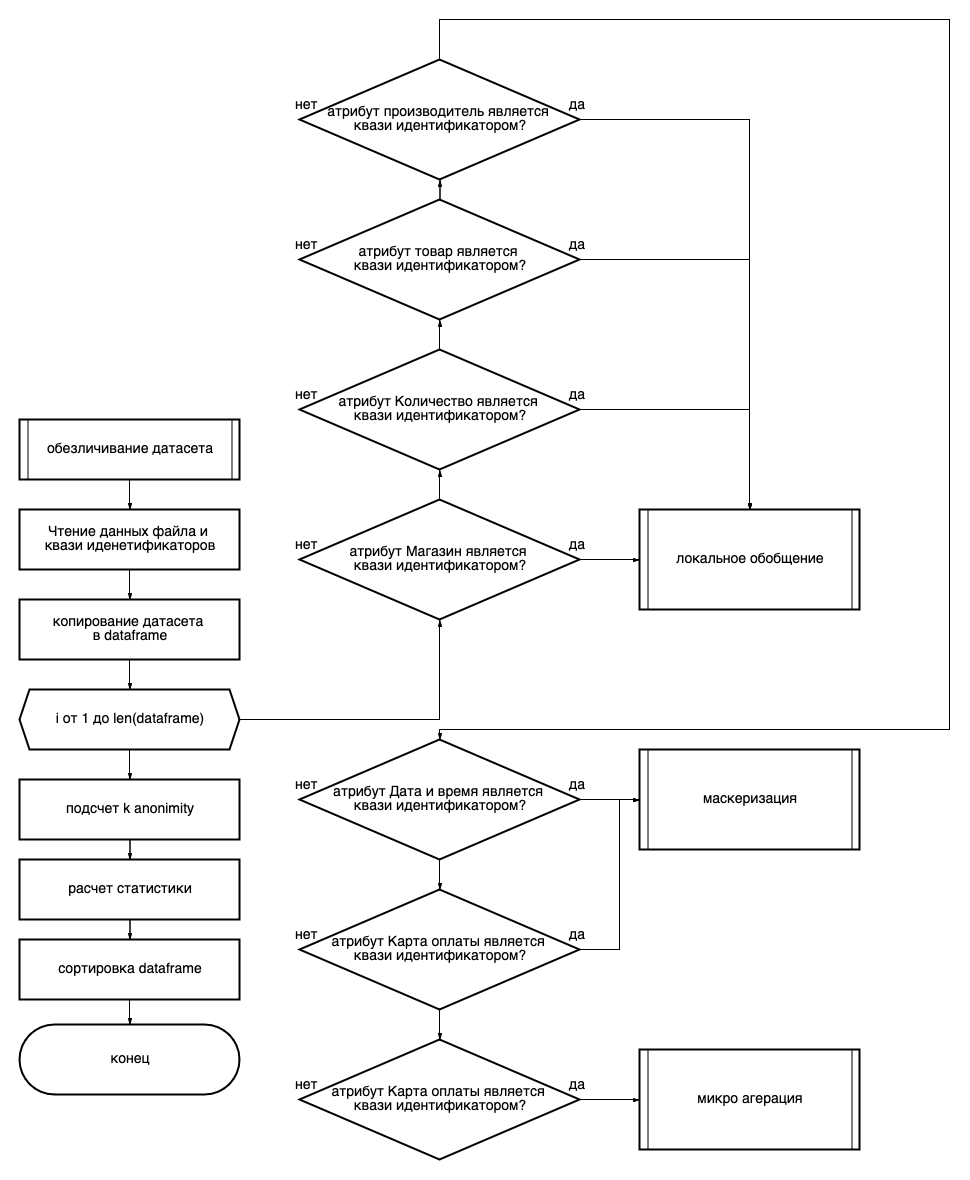


Рисунок 1 Основной алгоритм программы

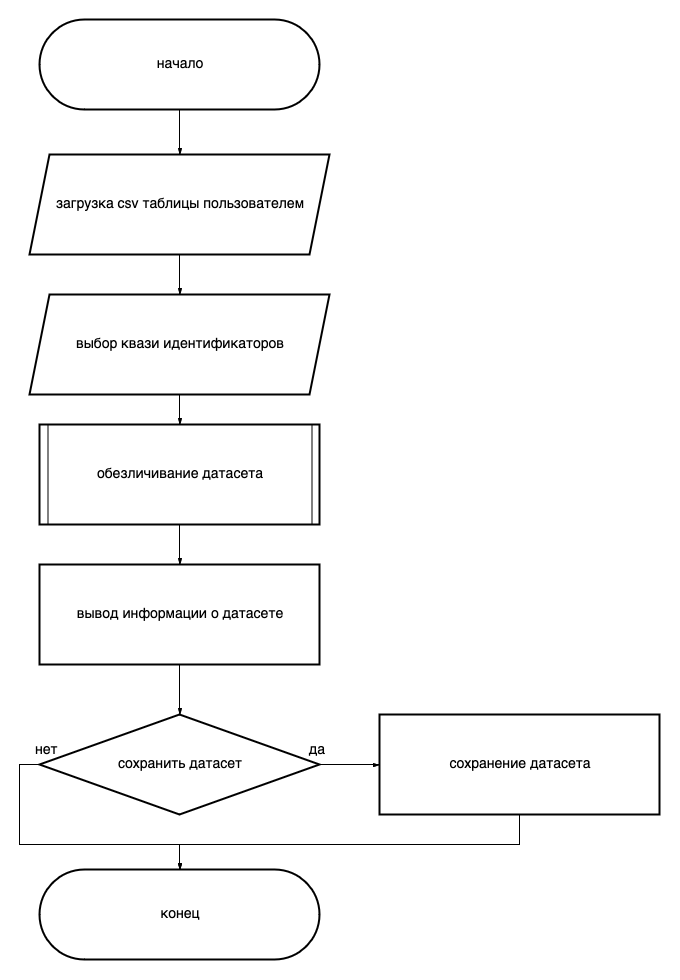


Рисунок 2. Алгоритм работы интерфейса

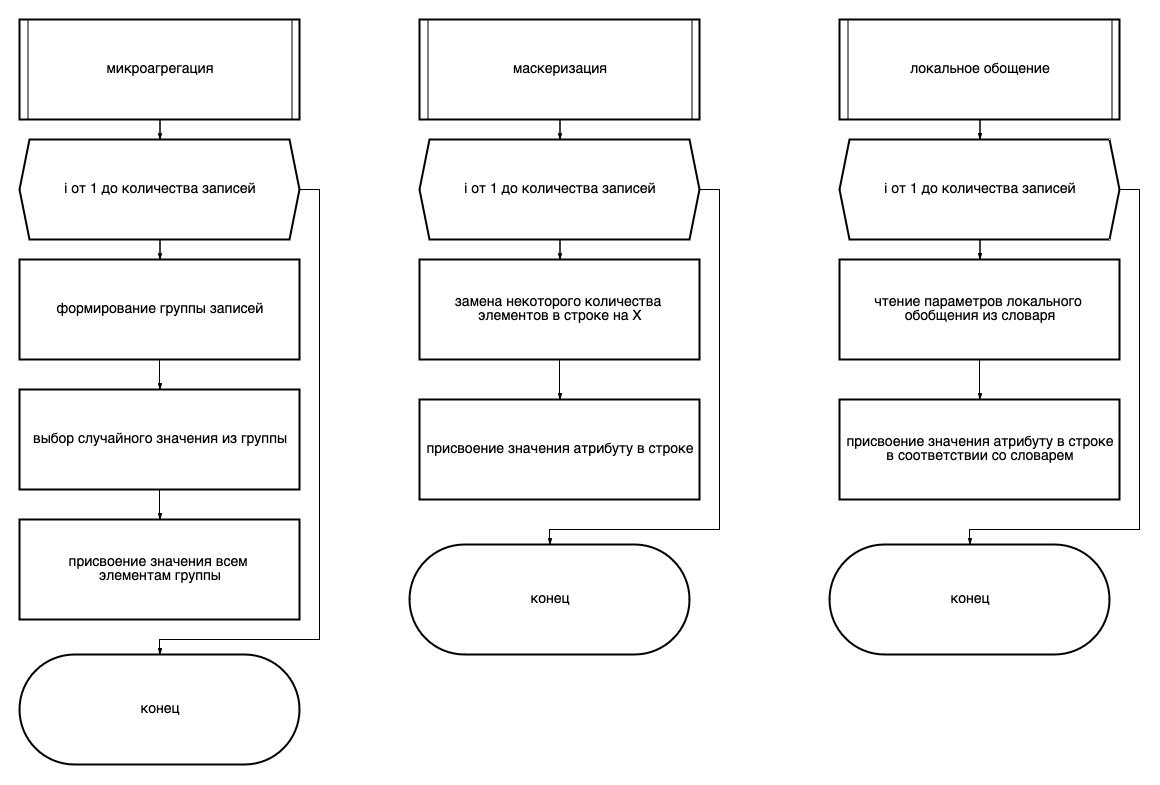


Рисунок 3. Алгоритм работы функций микроагрегация, маскеризация и локальное обобщение параметров

# **Описание программы**

Программа реализована при помощи языка python 3.11.4, с использованием библиотек: os, random, pandas, tkinter. В программе использовались 22 функции и 7 структур данных. В таблице 1 представлено описание функций

Таблица 1. Описание функций

|  |  |  |
| --- | --- | --- |
| Имя функции | Тип возвращаемого значения | Описание функции |
| read\_csv | DataFrame | Читает csv таблицу |
| item\_mask | None | заменяет название товара на одну из категорий |
| brand\_mask | None | заменяет название бренда на общую категорию (clothes food electronics) |
| coordinates\_mask\_alternative | None | если у сети магазинов с одним названием много точек по городу, то меняет их общие координаты на коордианыт одного магазина |
| date\_mask | None | сохраняет только год покупки |
| get\_shop\_category | string | возваращет категорию магазина (например М.Видео -> electronics, Zara -> clothes) |
| shop\_mask | None | заменяет название магазина на общую категорию (clothes food electronics) |
| amount\_mask | None | Локальное обобщение данных |
| price\_mask | None | Обобщение |
| even\_stronger\_anonymization | None | Еще более сильные методы анонимизации |
| add\_or\_update\_k\_anonymity\_column | None | Функция для добавления или обновления k-anonymity к каждой строке датасета  На основе квазиидентификаторов |
| remove\_worst\_k\_anonymity\_rows | Dataframe | Функция для удаления строк с самым низким показателем k-anonymity,  но не более max\_percent от общего количества строк. |
| k\_anonymity\_statistics | dict | Функция для подсчета количества строк, k-anonymity которых меньше указанного порога,  а также для вычисления среднего, медианного, максимального и минимального значений k-anonymity. |
| anonimize | DataFrame | Анонимизация на основе выбранных квази-идентификаторов |
| savefile | CSV | Сохранение таблицы |
| load\_file | DataFrame | Загрузка таблицы |
| show\_results | None | Показ результатов |
| process\_file | None | Обработка файла |
| interface | None | Отрисовка интерфейса |

В таблице 2 представлено описание структур, используемых в программе.

Таблица 2. Описание структур

|  |  |  |  |
| --- | --- | --- | --- |
| Имя структуры | Тип структуры | Тип хранимых данных | Описание |
| df\_global | DataFrame | string | Хранит csv таблицу |
| df\_original | DataFrame | string | Хранит csv таблицу (резервная копия) |
| stats | dict | string | Хранит статистику датасета |
| columns | list | string | Хранит атрибуты |
| bad\_locations | list | string | Хранит названия названия трудно обезличиваемых локаций |
| item\_dict | dict | string | Хранит словарь, для метода обощения |
| file\_path | string | string | Хранит путь к csv таблице |

# **Рекомендации пользователя**

Для запуска программы необходимо наличие устройства с установленной операционной системой Linux, macOS или Windows, а также среды разработки, поддерживающей запуск python версии 3.11.4.

Необходимо установить python библиотеки: pandas, tkinter. Для этого необходимо открыть терминал и прописать следующие команды:

*pip* install -r requirements.txt

В полях ввода, расположенных под строками «загузить csv файл», выберите файл из которого необходимо получить датасет (рисунок 1)

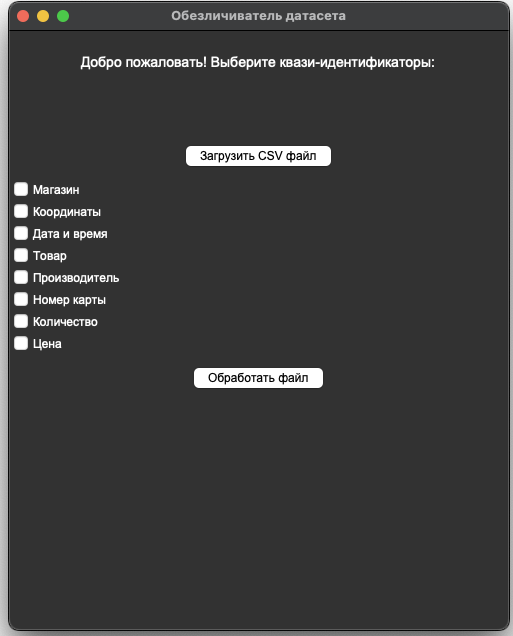


Рисунок 4. Интерфейс программы

**Ниже кнопки “Загрузить CSV файл” необходимо поставить галочки возле названия тех атрибутов, которые необходимо считать квази идентификаторами. (рисунок 4)**

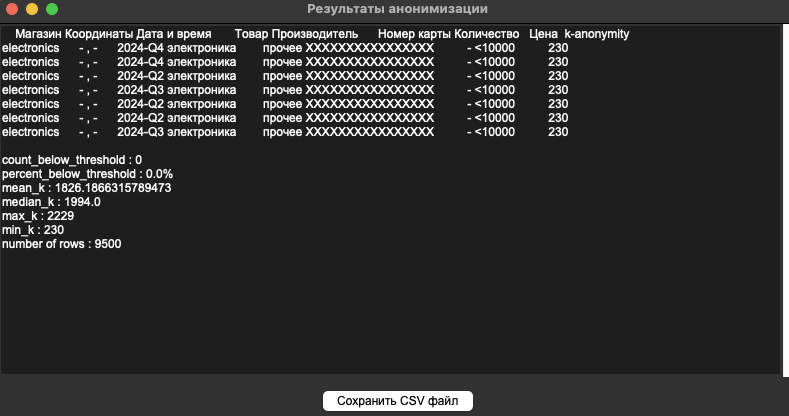
****

Рисунок 5. Результат работы программы

После процесса анонимизации откроется окно, в котором можно посмотреть статистику, а так же строки с наихудшими показателями анонимизации. (рис 5)

В таблице 3 приведены названия кнопок и соответствующие им выполняемые операции.

Таблица 3. Описание операций, выполняемых при нажатии на кнопку

|  |  |
| --- | --- |
| Название кнопки | Описание операции, которая выполняется при нажатии на кнопку |
| «Загрузить CSV файл» | Загружает csv таблицу в программу |
| «выбор квази идентификаторы» | Позволяет выбрать квази идентификаторы |
| «Обработать файл» | Обезличивает датасет. Выводит уникальные строки датасета для квази идентификаторов |
| «Сохранить CSV файл» | Сохраняет датасет в выходной файл с расширением csv |

# **Рекомендации для программиста**

Для внесения изменений в исходный код программы необходимо наличие устройство с установленное операционной системой Linux, macOS или Windows, а также среды разработки, поддерживающей запуск python версии 3.11.4

Исходный код программы доступен по ссылке ниже:

<https://github.com/alice3e/spbu-data-anonymization>

# **Контрольный пример**

В этом разделе представлен пример работы программы. Запускается программа с помощью команды в терминале (из директории с файлом main.cpp):

$ python main.cpp

Далее открывается графический интерфейс, где пользователь выбирает файл, и выбирает квази идентификаторы. Далее пользователь нажимает кнопку “обработать файл” и ждет необходимое количество времени, пока не откроется окно программы с результатом выполнения. В этом окне пользователь видит худшие по k-anonimity строки датасета, а так же подробную статистику. Далее пользователь может закрыть окно программы, либо сохранить файл в нужную ему директорию.

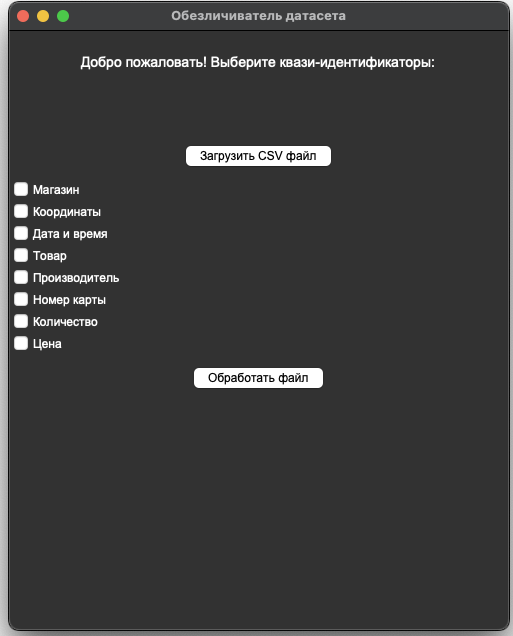


Рисунок 6. Стартовый интерфейс программы

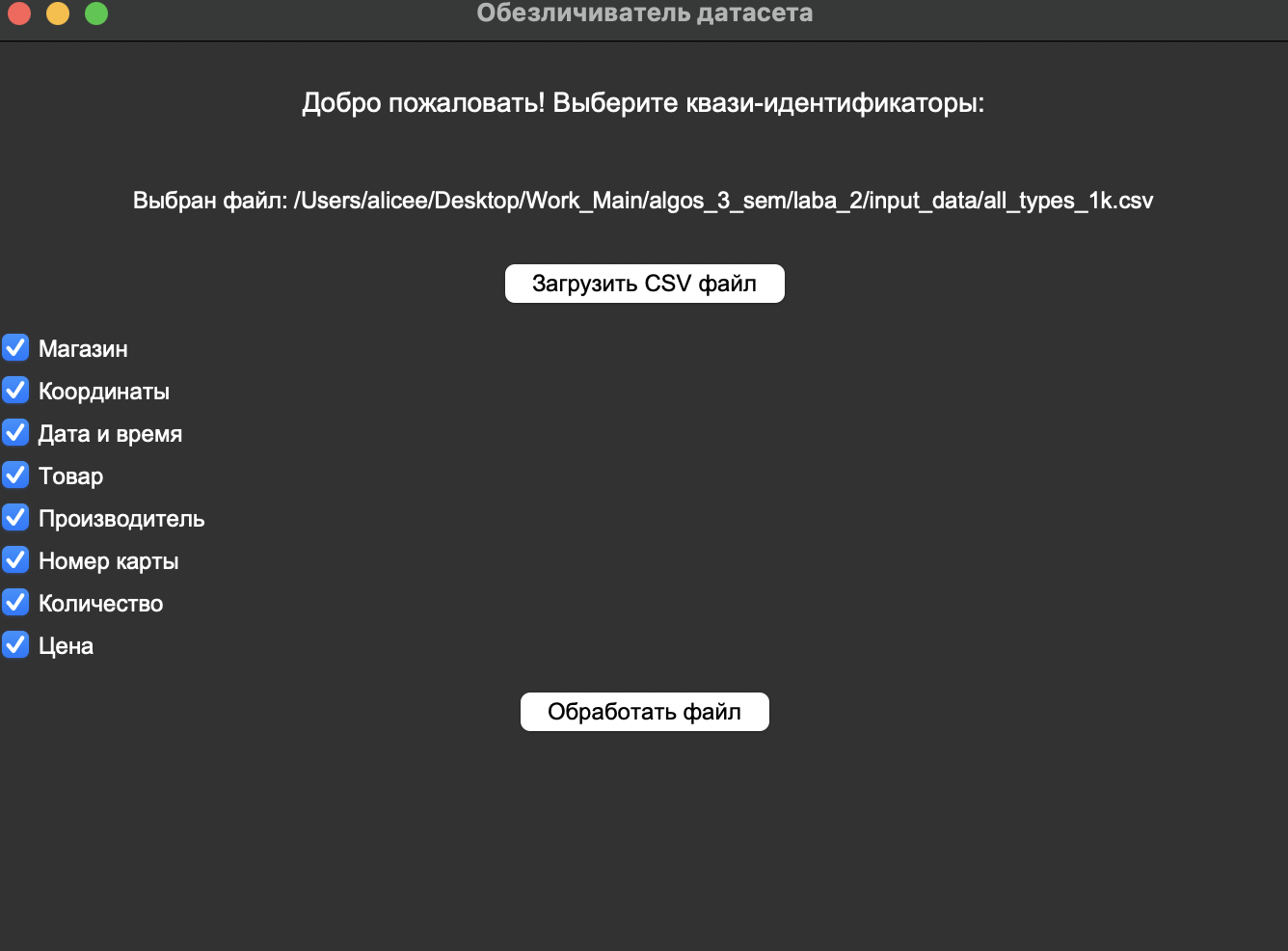


Рисунок 7. Вид программы после загрузки файла

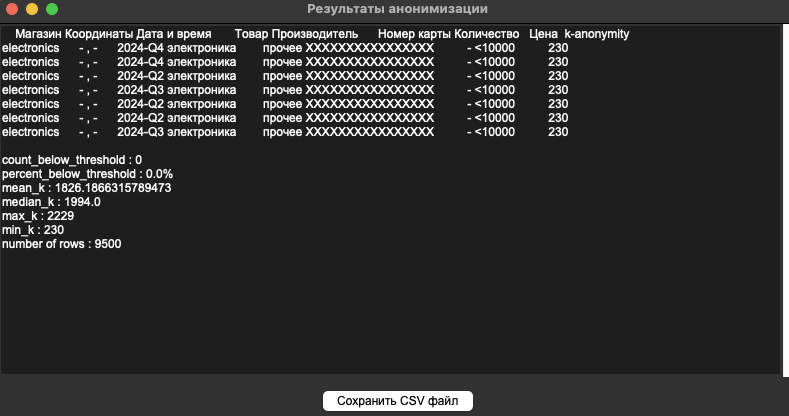
****

Рисунок 8. Результат работы программы

# **Вывод**

В ходе выполнения данной работы были изучены принципы и методы обезличивания данных. Разработан алгоритм, который обезличивает датасет, содержащих информацию о покупках на территории Санкт-Петербурга с необходимой, для количества данных в датасете, степенью защиты от деобезличивания. Разработан алгоритм, который рассчитывает k anonymity. Реализована программа на языке python, которая по выбранным квази идентификаторам обезличивает датасет и рассчитывает k anonymity.

# Список использованной литературы

[1] Статья о методах обезличивания персональных данных [(datasec.ru)](https://data-sec.ru/personal-data/depersonalization/)

[2] Статья по k anonymity и методам обезличивания ([Wikipedia](https://en.wikipedia.org/wiki/K-anonymity))

[3] Статья о том, как правильно обезличивать персональные данные [(habr.com)](https://habr.com/ru/companies/rubda/articles/688116/)