**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

УДК 004.62

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО |  | УТВЕРЖДАЮ |
| Старший преподаватель |  | Академический руководитель образовательной программы «Программная инженерия»  старший преподаватель департамента программной инженерии |
| Ю. В. Силаев |  | Н.А. Павлочев |

**Отчет**

**по исследовательскому курсовому проекту**

на тему «Исследование методов обезличивания персональных данных для табличных данных»

по направлению подготовки бакалавров 09.03.04 «Программная инженерия»

|  |
| --- |
| Выполнил  студент группы БПИ227  образовательной программы  09.03.04 «Программная инженерия» |
| Шомполов Максим Андреевич |
| 12.05.2025 |

Москва 2025

# РЕФЕРАТ

Важность проблемы обеспечения конфиденциальности персональных данных значительно выросла за последнее десятилетие. Одной из причин этого стало развитие интернета и цифровизация общества. Другая причина заключается в увеличении роли анализа данных. Одним из способов решения данной проблемы является обезличивание персональных данных.

В настоящей работе проводится исследование методов и алгоритмов обезличивания табличных персональных данных. Предлагается сравнить качество подходов с точки зрения риска раскрытия персональных данных и полезности обезличенных данных.

Для достижения данной цели будет произведён анализ различных методов обезличивания, описанных исследователями в данной области, а также различных подходов к оценке качества данных методов, реализация данных методов, разработка и реализация своих методов обезличивания и оценки качества, подготовка данных для оценки качества работы алгоритмов обезличивания персональных данных, проведение экспериментов с реализованными алгоритмами и анализ результатов.

В результате были предложены несколько подходов к обезличиванию и к оценке качества обезличивания, разработан пакет на языке программирования Python для обезличивания и оценки качества обезличивания. Также было проведено подробное сравнение известных и предложенных алгоритмов обезличивания.

Данная работа состоит из 65 страниц, 21 таблица, 6 приложений. Использовано 20 источников.

Ключевые слова: обезличивание, персональные данные, табличные данные, агрегация, рандомизация, подавление, обобщение, распределение данных, перемешивание данных, k-анонимность, l-разнообразие, t-близость.

# СОДЕРЖАНИЕ

[РЕФЕРАТ 2](#_Toc197960646)

[СОДЕРЖАНИЕ 3](#_Toc197960647)

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 6](#_Toc197960648)

[ВВЕДЕНИЕ 8](#_Toc197960649)

[1. ОБЗОР И АНАЛИЗ ИСТОЧНИКОВ 9](#_Toc197960650)

[1.1. Основные нормативные документы 9](#_Toc197960651)

[1.2. Основные источники, связанные с методами оценки качества обезличивания персональных данных 10](#_Toc197960652)

[1.2.1. Основные источники, связанные с критериями оценки рисков деобезличивания 10](#_Toc197960653)

[1.2.2. Основные источники, связанные методами оценки полезности обезличенных персональных данных 11](#_Toc197960654)

[1.3. Основные источники, связанные с алгоритмами обезличивания 11](#_Toc197960655)

[Выводы главы: 13](#_Toc197960656)

[2. МЕТОДЫ, ОЦЕНКИ КАЧЕСТВА ОБЕЗЛИЧИВАНИЯ 14](#_Toc197960657)

[2.1. Особенности типов столбцов и отдельных полей в наборах данных 14](#_Toc197960658)

[2.1.1. Рассматриваемые типы столбцов и отдельных полей в наборах данных 14](#_Toc197960659)

[2.1.2. Приведение данных к воспринимаемому моделями машинного обучения формату 14](#_Toc197960660)

[2.2. Способы оценки рисков деобезличивания 15](#_Toc197960661)

[2.2.1. k-анонимность 15](#_Toc197960662)

[2.2.2. l-разнообразие 16](#_Toc197960663)

[2.2.3. t-близость 16](#_Toc197960664)

[2.2.4. Adversarial knowledge gain (AKG) 17](#_Toc197960665)

[2.2.5. Adversarial accuracy gain (AAG) 18](#_Toc197960666)

[2.2.6. Оценка рисков с помощью градиентного бустинга 18](#_Toc197960667)

[2.3. Способы оценки полезности обезличенных данных 19](#_Toc197960668)

[2.3.1. Средний размер класса эквивалентности 19](#_Toc197960669)

[2.3.2. Отношение количества уникальных записей к общему числу записей 19](#_Toc197960670)

[2.3.3. Доля изменившихся отдельных значений в датасете 19](#_Toc197960671)

[2.3.4. Неоднородная энтропия (non-uniform entropy) 20](#_Toc197960672)

[2.3.5. Потеря информации на основе энтропии (entropy-based information loss) 20](#_Toc197960673)

[2.3.6. Оценка расстояния между исходными и получившимися данными поэлементно 20](#_Toc197960674)

[2.3.7. Оценка полезности обезличенных данных при помощи градиентного бустинга 21](#_Toc197960675)

[2.4. Выводы главы 22](#_Toc197960676)

[3. АЛГОРИТМЫ ОБЕЗЛИЧИВАНИЯ 23](#_Toc197960677)

[3.1. Методы обезличивания 23](#_Toc197960678)

[3.1.1. Подавление 23](#_Toc197960679)

[3.1.2. Обобщение 23](#_Toc197960680)

[3.1.3. Агрегация 23](#_Toc197960681)

[3.2. Общие алгоритмы обезличивания квази-идентификаторов 24](#_Toc197960682)

[3.2.1. Алгоритм перебора 24](#_Toc197960683)

[3.2.2. Алгоритм жадного выбора на основе расстояний 25](#_Toc197960684)

[3.2.3. Алгоритм на основе последовательного объединения групп 25](#_Toc197960685)

[3.2.4. Модифицированный Datafly 27](#_Toc197960686)

[3.2.5. Алгоритм разбиения столбцов на равные группы с обобщением 27](#_Toc197960687)

[3.2.6. Алгоритм рандомизации 28](#_Toc197960688)

[3.3. Алгоритмы обезличивания, применяемые для специальных целей 29](#_Toc197960689)

[3.3.1. Алгоритмы перемешивания 29](#_Toc197960690)

[3.3.2. Алгоритм хеширования 30](#_Toc197960691)

[3.3.3. DPP2GA (объединение распределённого горизонтально датасета) 30](#_Toc197960692)

[Выводы главы 32](#_Toc197960693)

[4. ЭКСПЕРИМЕНТЫ НАД АЛГОРИТМАМИ ОБЕЗЛИЧИВАНИЯ 33](#_Toc197960694)

[4.1. Библиотека для обезличивания 33](#_Toc197960695)

[4.2. Данные использующиеся в экспериментах 34](#_Toc197960696)

[4.2.1. Датасет с Kaggle 34](#_Toc197960697)

[4.2.2. Сгенерированный датасет 35](#_Toc197960698)

[4.3. Методика проведения экспериментов 36](#_Toc197960699)

[4.4. Результаты проведения экспериментов 38](#_Toc197960700)

[4.4.1. Эксперименты с сгенерированным датасетом 38](#_Toc197960701)

[4.4.2. Эксперименты с датасетом с Kaggle 38](#_Toc197960702)

[4.4.3. Эксперименты с коротким датасетом 38](#_Toc197960703)

[4.5. Выводы из результатов проведения экспериментов 38](#_Toc197960704)

[Выводы главы 41](#_Toc197960705)

[ЗАКЛЮЧЕНИЕ 42](#_Toc197960706)

[СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ 43](#_Toc197960707)

[ПРИЛОЖЕНИЕ 1 КАЛЕНДАРНЫЙ ПЛАН РАБОТ 45](#_Toc197960708)

[ПРИЛОЖЕНИЕ 2 ФУНКЦИИ ПАКЕТА CP2025, РЕЗЛИЗУЮЩИЕ МЕТОДЫ ОЦЕНКИ КАЧЕСТВА ОБЕЗЛИЧИВАНИЯ 46](#_Toc197960709)

[ПРИЛОЖЕНИЕ 3 КЛАССЫ ПАКЕТА CP2025, РЕЗЛИЗУЮЩИЕ АЛГОРИТМЫ ОБЕЗЛИЧИВАНИЯ 49](#_Toc197960710)

[ПРИЛОЖЕНИЕ 4 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ НА СРЕНЕРИРОВАННОМ ДАТАСЕТЕ 54](#_Toc197960711)

[ПРИЛОЖЕНИЕ 5 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ НА ДАТАСЕТЕ С KAGGLE 58](#_Toc197960712)

[ПРИЛОЖЕНИЕ 6 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ НА КОРОТКОМ ДАТАСЕТЕ 62](#_Toc197960713)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

***Необезличенные персональные данные*** – персональные данные в виде, в котором они были получены от их владельца.

***Персональная идентификационная информация (персональные идентификаторы, идентификаторы)*** – любые данные, напрямую указывающие на их владельца.

***Обезличенные персональные данные*** – данные, полученные в результате обработки необезличенных персональных данных, деобезличивание которых затруднено в результате данной обработки.

***Деобезличиавние персональных данных*** – преобразование обезличенных персональных данных с целью целиком или частично установить соответствие между владельцем персональных данных и самими персональными данными.

***Чувствительная информация*** – персональные данные не позволяющие идентифицировать их владельца, раскрытие которых несёт угрозу его анонимности.

***Нечувствительная информация*** - персональные данные не позволяющие идентифицировать их владельца, раскрытие которых не несёт угрозу его анонимности.

***Квази-идентификаторы*** – любые части персональных данных, которые позволяют определить, что какие-либо данные из данного набора не относятся к конкретному субъекту персональных данных.

***Уровень анонимности обезличенных персональных данных*** – оценка рисков раскрытия владельца персональных данных.

***Уровень полезности обезличенных персональных данных*** – свойство обезличенных персональных данных, позволяющее решать прикладные задачи с их использованием.

***Класс эквивалентности (корзина)*** – набор строк в датасете, имеющих одинаковые значения квази-идентификаторов

***Метод обезличивания*** – способ приведения всех квази-идентификаторов в классе эквивалентности к одному и тому же значению

***Алгоритм обезличивания*** – способ преобразования данных, в результате которого их деобезличивание затрудняется

***Value Generalization Hierarchy (VGH)*** — это иерархия обобщения значений, которая указывает на возможные преобразования, которые может претерпевать атрибут

# ВВЕДЕНИЕ

В настоящее время роль анализа данных значительно выросла. Он позволяет находить различные закономерности и проверять гипотезы в различных сферах: бизнесе, медицине, биологии, социальной сфере, экономике, финансах и других. Однако при обработке информации часто присутствует потребность работы с персональными данными. В этом случае возникает проблема получения согласия владельца персональных данных на их предоставление специалисту, который будет осуществлять анализ данных.

Другой проблемой, уже непосредственно в сфере защиты информации, является обеспечение безопасности информации, а именно её конфиденциальности, которая может быть нарушена в результате утечки данных. Актуальность этого вопроса в данный момент стала значительно выше из-за цифровизации общества.

Одним из подходов к решению данных проблем является обезличивание персональных данных. Оно представляет собой такую их обработку, что восстановление какого-либо соответствия между обработанными персональными данными и субъектом этих персональных данных затрудняется или становится невозможным. В результате данного преобразования часть информации теряется, но данные всё ещё в какой-то степени остаются содержательными для работы и анализа. При этом в зависимости от алгоритма обезличивания персональных данных степень сложности деобезличивания и степень полезности данных может меняться. Основные положения об обезличивании данных в России закреплены в Законе №152-ФЗ «О персональных данных» [[1]](#а1).

На данный момент исследователями предложено достаточно большое количество алгоритмов обезличивания и оценки его качества. Большинство из них имеет недостаток, заключающийся в том, что они осуществляют обезличивание, основываясь на его вариантах, предложенных пользователем, а не определяют наилучшую стратегию обезличивания автоматизированно.

В настоящей работе проводится исследование различных методов обезличивания табличных персональных данных и их сравнение. Для этого проводится изучение самих алгоритмов обезличивания и подходов к оценке их качества с точки зрения рисков деобезличивания и полезности данных, разработка своих алгоритмов и подходов, а также реализация этих алгоритмов и подходов. На основе реализованных методов осуществляется сравнительный анализ способов обезличивания. Основными рассматриваемыми методами обезличивания являются агрегация, рандомизация, локальное подавление, обобщение и распределение данных.

# ОБЗОР И АНАЛИЗ ИСТОЧНИКОВ

## 1.1. Основные нормативные документы

Прежде, чем перейти к рассмотрению алгоритмов обезличивания и методов оценки его качества, необходимо рассмотреть положения законодательства, регламентирующие этот процесс, так как деятельность, связанная с обеспечением конфиденциальности персональных данных, всегда тесно связана с регламентирующими её нормативными актами.

В России основными документами, регламентирующими процесс обезличивания, являются:

1. Закон №152-ФЗ «О персональных данных» [[1]](#а1)
2. Приказ Федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций от 5 сентября 2013 г. N 996 "Об утверждении требований и методов по обезличиванию персональных данных" [[2]](#а2)

Согласно данным документам, обезличивание имеет следующие особенности:

1. Обезличенными данными считаются такие данные, над которыми невозможно произвести успешный процесс обезличивания, без использования дополнительной информации. Важно отметить, что данная классификация не соответствует классификации, используемой в данном отчёте, так как в данном отчёте допускается, что для деобезличивания может быть использована вся информация, не являющаяся чувствительной
2. Обезличенные данные являются персональными данными, если существует возможность проведения деобезличивания. То есть алгоритмы, описанные в данном отчёте, могут быть применены для того, чтобы обрабатываемые данные перестали быть персональными

В Европейском союзе действует законодательный акт GDPR (General Data Protection Regulation) [[3]](#а3), регламентирующий обезличивание. Согласно данному акту, обезличивание имеет следующие особенности:

1. Анонимизированные данные – данные, которые невозможно деобезличить (алгоритмы получения таких данных будут рассмотрены в данном отчёте)
2. Псевдоанонимизированные данные – данные, которые можно деобезличить с использованием дополнительных сведений
3. Анонимизированные не считаются персональными данными, а псевдоанонимизированные всё ещё считаются

И в России, и в Европейском союзе выделяются следующие методы обезличивания:

1. Удаление идентификаторов
2. Хеширование или шифрование идентификаторов
3. Обобщение (будет раскрыто позднее в данном отчёте)
4. Агрегация (будет раскрыта позднее в данном отчёте)
5. Искажение данных
6. Разделение данных на несколько частей

Таким образом, можно отметить, что в России и в Европейском союзе законодательство об обезличивании персональных данных является схожим, однако граница между обезличенными данными являющимися персональными и не являющимися персональными в России является в некоторой степени более размытой.

## 1.2. Основные источники, связанные с методами оценки качества обезличивания персональных данных

Теперь рассмотрим источники, являющиеся отправной точкой при исследовании методов обезличивания табличных данных. Для понимания алгоритмов обезличивания требуется сначала рассмотреть способы оценки качества данных алгоритмов, так как подходы к обезличиванию в своей реализации опираются на критерии оценки рисков деобезличивания и методы оценки полезности обезличенных персональных данных.

### 1.2.1. Основные источники, связанные с критериями оценки рисков деобезличивания

Первой статьёй, описывающей эти критерии, является [[4]](#а4). В данной статье рассматриваются существовавшие на тот момент проблемы обезличивания данных, заключавшиеся в том, что уделение персональной идентификационной информации являлось недостаточным для обеспечения конфиденциальности персональных данных. Для решения это проблемы был предложен декларативный подход к обезличиванию табличных данных и одновременно способ оценки его качества – k-анонимность. Также в статье рассматриваются конкретные способы обезличивания: обобщение и подавление. Также данный подход и его недостатки описываются в [[5]](#а5).

Как уже было сказано k-анонимность имеет некоторые недостатки, которые не позволяют ей обеспечивать достаточную конфиденциальность персональных данных в некоторых ситуациях. Часть из этих проблем решается при помощи подхода l-разнообразие, который является развитием k-анонимности и был предложен в источнике [[6]](#а6). Однако данный метод также подвержен ряду рисков в случае, если обезличиваемые данные имеют определённую структуру. Эти недостатки описываются в статье [[7]](#а7). Для их снижения в [[7]](#а7) предлагается использовать метод t-близости, который дополнительно учитывает распределение данных. Данный подход был изначально предложен в [[8]](#а8).

Другой подход к оценке рисков деобезличивания предложен в [[9]](#а9). В данной статье предложено использовать метрики Adversarial Knowledge Gain и Adversarial Accuracy Gain, которые оценивают роль предоставленных обезличенных данных для проведения деобезличивания по сравнению со случаем, когда никаких данных предоставлено не было. Данные метрики также основываются на классах эквивалентности (корзинах), рассмотренных выше.

Важной проблемой всех этих подходов является то, что они опираются на классы эквивалентности при оценке рисков, а следовательно, при помощи них невозможно оценить риски для обезличенных датасетов, в которых нет классов эквивалентности, например, датасетов, полученных добавлением к квазиидентификаторам случайной величины.

### 1.2.2. Основные источники, связанные методами оценки полезности обезличенных персональных данных

В [[9]](#а9) для оценки полезности обезличенных персональных данных предложен следующий подход: применять обезличенные данные для предсказания какого-либо признака при помощи модели машинного обучения. Результаты можно сравнить с предсказаниями на изначальном датасете и получить некоторую меру потери полезности. Недостатком данного подхода является то, что данные должны позволять достичь хорошего качества предсказаний при помощи моделей машинного обучения, так как иначе результаты оценки полезности могут быть непоказательными.

В документации программного пакета ARX [[10]](#б10) предложен ряд методов оценки полезности обезличенных персональных данных:

* Метод оценки при помощи модели машинного обучения, описанный выше
* Средний размер класса эквивалентности
* Отношение количества уникальных записей к общему числу записей (distinctness)
* Доля изменившихся отдельных значений в датасете
* Неоднородная энтропия (non-uniform entropy)
* Потеря информации на основе энтропии (entropy-based information loss)

Проблемой всех этих методов (кроме метода с применением модели машинного) обучения является тот факт, что они опираются на классы эквивалентности при оценке качества.

## 1.3. Основные источники, связанные с алгоритмами обезличивания

В качестве отправной точке в вопросе исследования алгоритмов обезличивания можно считать [[11]](#и11). В данной статье рассматриваются общие подходы к обезличиванию без рассмотрения деталей разбиения на классы эквивалентности.

В [[12]](#и12) рассматривается концепция VGH (Value Generalization Hierarchy) – дерево (граф), определяющий до каких диапазонов и значений может быть обобщено определённое значение. Например, VGH может иметь следующую структуру: каждый день с 01.12 до 31.12 и от 01.01 до 29.02 может быть обезличен до времени года «зима»; каждый день от 01.03 до 31.05 – до времени года «весна»; от 01.06 до 31.08 – «лето»; от 01.09 до 30.11 – «осень». При этом значения «зима», «весна», «лето», «осень» могут быть обезличены до значения «год». (рисунок 1).

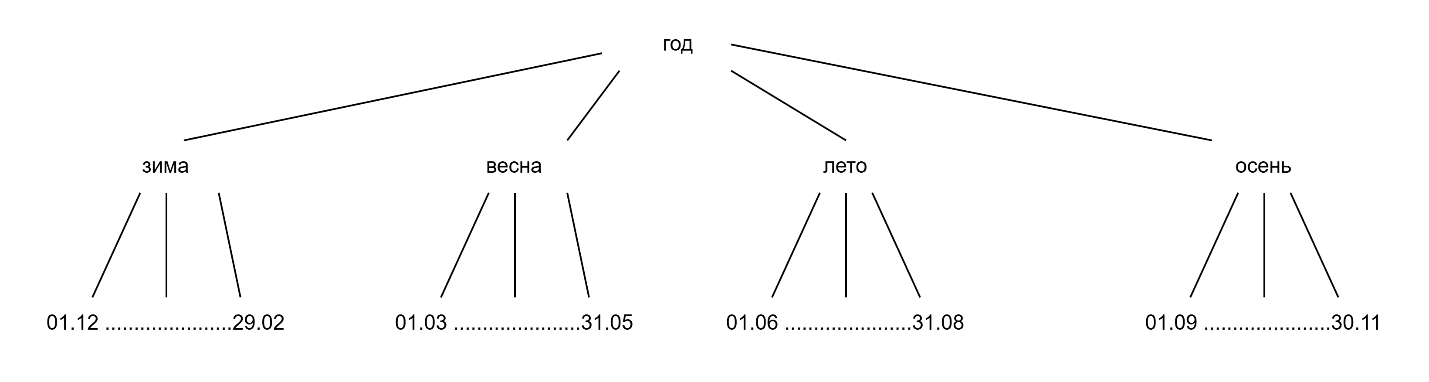


Рисунок 1 – Пример VGH

Данный подход обладает следующим недостатком: VGH должна быть задана человеком, а человек не всегда может знать, какая VGH лучше для обезличивания определённого датасета. Это лишает пользователя возможности полностью использовать современные вычислительные ресурсы, для получения оптимального обезличенного датасета.

На данном этапе важно отметить, что достижение k-анонимности и минимальной потерей полезности данных является NP-сложной задачей, что доказывается в [[13]](#и13). Поэтому для обезличивания с достижением k-анонимности, l-разнообразия и t-близости (данные метрики подробно рассмотрены в разделе 2) используются приближённые алгоритмы.

В [[14]](#и14) приведено большое количество алгоритмов, большинство из которых используют VGH для обезличивания.

В дальнейшем в работе рассматривается модификация алгоритма Datafly [[15]](#и15), в которой, в отличие от самого алгоритма Datafly, построение VGH осуществляется автоматически.

Важно отметить, что не все алгоритмы зависимы от VGH. Примером такого алгоритма является Mondrian Multidimensional K-Anonymity [[16]](#и16), который разделяет пространство признаков на части, что в некотором смысле аналогично работе решающего дерева.

Наконец в [[17]](#и17) в том числе содержатся общие сведения о алгоритмах объединения разделённых датасетов, а в [[18]](#и18) и [[19]](#и19) рассмотрены алгоритмы TDS2P и DPP2GA соответственно – алгоритмы объединения разделённых горизонтально датасетов (один владелец имеет часть столбцов, а другой – другую часть, при этом каждый имеет некоторый столбец с общим идентификатором) с сохранением анонимности в соответствии с заданной метрикой.

## Выводы главы:

Обезличивание персональных данных регламентировано нормативными актами в России и в Европе, при этом многие положения являются схожими. Исследователями описано множество подходов к обезличиванию персональных данных и оценке качества их обезличивания. Данные подходы имеют ряд недостатков при их применении в определённых ситуациях.

# МЕТОДЫ, ОЦЕНКИ КАЧЕСТВА ОБЕЗЛИЧИВАНИЯ

Рассмотрим подходы, которые будут применяться в данной работе для оценки качества обезличивания персональных данных.

## 2.1. Особенности типов столбцов и отдельных полей в наборах данных

### 2.1.1. Рассматриваемые типы столбцов и отдельных полей в наборах данных

Для того, чтобы наилучшим образом использовать особенности тек или иных данных, в данной работе рассматриваются следующие типы столбцов в наборе данных:

1. Вещественные значения – допустимо выполнение всех стандартных математических операций и хеширования
2. Порядковые значения – присутствует линейный порядок на данных значениях, допустимо сравнивать на равенство, хешировать
3. Номинальные значения – допустимо сравнивать на равенство, хешировать

В результате обезличивания выходной набор данных может содержать следующие типы отдельных значений:

1. Стандартное значение – значение одного из стандартных типов
2. Отсутствие значения – значение None (для метода подавление)
3. Диапазон значений – пара из минимума и максимума на отрезке между которыми лежит исходное значение (для вещественных и порядковых значений при использовании метода обобщение)
4. Набор значений – набор значений, среди которых лежит исходное значение (для номинальных значений при использовании метода обобщение)

### 2.1.2. Приведение данных к воспринимаемому моделями машинного обучения формату

Обезличенный набор данных может содержать:

1. None значения
2. Столбцы с вещественными значениями и диапазонами
3. Столбцы с порядковыми значениями и диапазонами
4. Столбцы с номинальными значениями и множествами

В данном подходе производится одновременная обработка обучающей и тестовой выборок для того, чтобы при one hot кодировании были созданы столбцы для всех уникальных значений порядковых и номинальных признаков.

В первую очередь необходимо избавиться от диапазонов и множеств:

1. Если в столбце есть диапазоны, то он разделяется на 2 столбца: содержащий минимумы и содержащий максимумы. Все не диапазоны принимаются и за минимумы, и за максимумы
2. Если в столбце есть множества, то все его значения принимаются за множества. Затем выполняется подобие one hot encoding по следующим правилам:
   1. Если значение равно None, то во всех столбцах стоят нули
   2. Если значение одно, то аналогично one hot encoding в соответствующем столбце стоит 1
   3. Если значение изначально представляло множество, то во всех соответствующих столбцах стоит значение, равное числу обратному мощности множества

Теперь необходимо заменить все None значения:

1. Если столбец вещественный, то None заменяется на среднее не равных None значений
2. Если столбец порядковый, то None заменяется на медиану не равных None значений (при чётной длине берётся меньшее значение)
3. Если столбец номинальный, то None заменяется на медиану не равных None значений
4. Если в столбце все значения равны None, то столбец считается константным признаком и все значения заменяются на 0

Наконец происходит one hot кодирование всех ещё не обработанных с этой точки зрения порядковых и номинальных столбцов (порядковых и тех номинальных, в которых не было множеств).

## 2.2. Способы оценки рисков деобезличивания

### 2.2.1. k-анонимность

Формально данный метод можно описать следующим образом: говорят, что датасет является k-анонимным, если для каждой его записи существует ещё как минимум k-1 запись, такая что в каждой из них все квази-идентификаторы соответственно (по колонкам) совпадают. Таким образом, в случае идентификация конкретной записи по квази-идентификаторам конкретного человека становится невозможна, так как для каждого их набора будет существовать 0 или как минимум k соответствующих записей. Будем называть это множество записей корзиной.

Проблемой данного метода является возможное неравномерное распределение данных в датасете. В случае, если атрибут, содержащий чувствительную информацию, совпадает для всех k записей деобезличивание окажется возможным.

Данный метод требует значительно меньшей потери полезности информации, чем подходы, которые будут описаны ниже. Однако из-за этого риски раскрытия владельца персональных данных растут.

### 2.2.2. l-разнообразие

Определим данный подход формально. Датасет является l-разнообразным, если в каждой корзине из метода 2.2.1. для каждого признака, содержащего чувствительную информацию, существует хотя бы l различных значений. В таком случае, проблема, описанная в пункте 2.2.1. в некоторой степени решена, так как в k-записях или более с одинаковым набором квази-идентификаторов будут содержаться хотя бы l различных значений.

Несмотря на то, что при применении l-разнообразия становится невозможным точно определить чувствительную информацию для конкретного владельца персональных данных, всё ещё можно установить некоторое соответствие между обработанными персональными данными и субъектом этих персональных. Несмотря на то, что известно только то, что чувствительная информация владельца содержится в каком-либо множестве мощности хотя бы l, может оказаться, что этого достаточно, чтобы нанести ущерб владельцу персональных данных, в случае если каждый из l вариантов содержит в себе, например, дискредитирующую информацию.

Другим недостатком данного подхода является возможность того, что данные l значений будут неравномерно распределены в корзине или это распределение будет сильно отличаться от распределения датасета. Например, в случае если k равняется 20, а l равняется 5, но при этом корзина содержит каждое из 4 значений только в одной записи, оставшееся значение в 16 записях, можно с уверенностью 80% утверждать, что запись субъекта персональных данных содержит часто встречающееся значение (содержащееся в 80% записей корзины).

Данный метод обеспечивает более высокую защищённость от деобезличивания, но как показывают исследования, упоминающиеся в [[5]](#и5), даже при малых l происходит значительная потеря полезности данных.

### 2.2.3. t-близость

Снова сначала формально определим способ оценки. Датасет соответствует критерию t-близости, если расстояние между распределением данных в каждом столбце, содержащем чувствительную информацию, в каждой корзине из пункта 2.2.2., и распределением значений этом столбце во всём датасете на превышает t. В данном отчёте предлагается расстояние между распределениями столбцов с номинальными и порядковыми значениями измерять при помощи энтропии, а между распределениями столбцов с вещественными значениями – при помощи расстояния Васерштейна. Все значения расстояний нормируются. Таким образом, t-близость решает проблемы l-разнообразия, описанные в пункте 2.2.2. Действительно в случае, если каждая корзина по своей структуре с точки зрения чувствительной информации похожа на весь датасет, то факт содержания записи пользователя в корзине не даёт больше информации, чем факт содержания записи пользователя в датасете (с точностью до t), что предполагается известным.

Основной проблемой этого метода является значительная потеря полезности данных при t, достаточных для решения проблем, описанных в пункте 2.2.2.

### 2.2.4. Adversarial knowledge gain (AKG)

Пусть противник пытается осуществить деобезличивание, при этом для каждого набора (класса эквивалентности) квази-идентификаторов он пытается брать самое часто встречающееся значение

Пусть – множество всех чувствительных значений в данном столбце во всём наборе данных, - класс эквивалентности для строки с квази-идентификаторами . – доля среди значений в данном столбце во всём датасете, - доля среди значений в данном столбце в данном классе эквивалентности.

Тогда раскрытие конфиденциальных атрибутов (sensitive attribute disclosure), представляющее собой разницу между информацией, которой владеет противник при наличии у него данного набора квази-идентификаторов и информацией при наличии у него только чувствительных значений, для данного класса эквивалентности можно вычислить по формуле (1):

*-* sensitive attribute disclosure

*-* данный класс эквивалентности

В таком случае adversarial knowledge gain, представляющее собой среднее количество информации, которое получает о чувствительных значениях данного столбца противник на основе полученных обезличенных квази-идентификаторов, можно вычислить по формуле (2):

- adversarial knowledge gain

- число строк в наборе данных

– множество классов эквивалентности

– размер класса эквивалентности

Для того, чтобы подсчитать adversarial knowledge gain для всех столбцов с чувствительными значениями одновременно в данном отчёте предлагается брать максимальный adversarial knowledge gain рассчитанный для каждого столбца отдельно, так как целью защиты информации является недопущение нарушение конфиденциальности не одного из чувствительных значений.

### 2.2.5. Adversarial accuracy gain (AAG)

Adversarial accuracy gain представляет собой рост доли правильных ответов (accuracy) противника из пункта 2.1.4. и вычисляется по формуле:

- adversarial accuracy gain

- число строк в наборе данных

– множество классов эквивалентности

– размер класса эквивалентности

*–* самое часто встречающееся чувствительное значение в данном столбце в классе эквивалентности

*–* самое часто встречающееся чувствительное значение в данном столбце во всём наборе данных

По аналогии с adversarial knowledge gain adversarial accuracy gain для всех столбцов с чувствительными значениями одновременно в данном отчёте предлагается вычислять как максимальный adversarial accuracy gain рассчитанный для каждого столбца отдельно.

### 2.2.6. Оценка рисков с помощью градиентного бустинга

Как уже было упомянуто в пункте 1.2.1. для алгоритмов обезличивания, не применяющих при своей работе разбиение на классы эквивалентности (например рандомизация), перечисленные в пунктах 2.1.1. – 2.1.5. методы оценки рисков деобезличивания неприменимы. Для таких случаев в данном отчёте предлагается другой подход.

Целью данного подхода является имея вектор изначальных (до обезличивания) квази-идентификаторов (считается, что они известны противнику) конкатенированный с вектором из обезличенного датасета (вся строка, так как обезличенный датасет целиком известен противнику) предсказать, принадлежат ли они одному человеку или разными.

В качестве модели машинного обучения для этих целей выбран градиентный бустинг, так как он является достаточно сильной моделью для работы с табличными данными и несмотря на то, что имеет много гиперпараметров, их число не так велико, как для нейросетей, для которых в том числе гиперпараметром можно считать архитектуру.

Обучение осуществляется следующим образом:

1. Первая часть датасета представляет собой квази-идентификаторы изначального датасета, конкатенированные с обезличенным датасетом. Для них предсказываемое значение равно 0
2. Вторая часть датасета датасета представляет собой квази-идентификаторы изначального датасета, конкатенированные с обезличенным датасетом, циклически сдвинутым на случайно сгенерированное значение не кратное числу строк в датасете. В этом случае предсказываемое значение равно 1
3. После получения объединённого датасета (конкатенация по вертикали дух частей датасета и преобразование в соответствии с пунктом 2.1.2.) производится обучение градиентного бустинга с перебором гиперпараметров (grid search)

Итоговая метрика представляет собой разность 1 и accuracy (доли верных ответов) на данных, отложенных для тестирования.

Можно сказать, что полученная метрика представляет собой среднее доля записей, для которых противник сможет осуществить деобезличивание. С другой стороны, для алгоритмов, применяющих классы эквивалентности для обезличивания получение данного значения значительно проще. Пусть число классов эквивалентности равно , их размеры равны . Сумма вероятностей угадать владельца каждой строки персональных данных в классе эквивалентности равна = 1. Тогда искомая средняя доля записей равна , так как классы эквивалентности не пересекаются. Получается, что искомая средняя доля записей равна числу обратному среднему размеру класса эквивалентности.

## 2.3. Способы оценки полезности обезличенных данных

### 2.3.1. Средний размер класса эквивалентности

Представляет собой число записей в обезличенном наборе данных делённое на число уникальных строк квази-идентификаторов (число классов эквивалентности).

### 2.3.2. Отношение количества уникальных записей к общему числу записей

Представляет собой число уникальных строк квази-идентификаторов (число классов эквивалентности), делённое на число записей в обезличенном наборе данных.

### 2.3.3. Доля изменившихся отдельных значений в датасете

Рассчитывается по формуле (4):

*-* доля изменившихся отдельных значений в датасете

*–* число строк в датасете

– число столбцов в датасете

– элемент изначального датасета в строке и столбце

– элемент обезличенного датасета в строке и столбце

### 2.3.4. Неоднородная энтропия (non-uniform entropy)

Рассчитывается по формуле (5):

*–* входной датасет

– набор классов эквивалентности в данном датасете

– размер данного класса эквивалентности

Формула получается подстановкой в стандартную энтропию частот классов эквивалентности

### 2.3.5. Потеря информации на основе энтропии (entropy-based information loss)

Рассчитывается по формуле (6):

– исходный датасет

*–* обезличенный ддатасет

## 2.3.6. Оценка расстояния между исходными и получившимися данными поэлементно

Так как согласно пункту 2.1. данные в столбцах могут принимать значения с резными свойствами, в данном отчёте предлагается следующий подход к расчёту расстояния между двумя элементами. Расстояние всегда лежит на отрезке от 0 до 1.

Если один из элементов None, а другой нет, то расстояние принимает значение 1. Если оба элемента None, то 0. Если элементы равны, то расстояние между ними 0.

Пусть оба элемента не None и не равны. Тогда пусть – первый элемент и он всегда не является диапазоном или множеством (такой всегда есть, так как в изначальном датасете не может быть диапазонов и множеств), а – второй и, если – диапазон, то – нижняя граница диапазона, а – верхняя. Если – множество, то - мощность множества.

Рассмотрим все возможные типы столбцов с и :

1. Вещественные значения (пусть – максимальное значение в столбце, а - минимальное):
   1. Если – значение, то расстояние вычисляется по формуле (7):
   2. Если – диапазон, то расстояние рассчитывается как математическое ожидание расстояния, указанного в пункте 1.а, если равномерно распределено по диапазону, то есть (формула (8)):
2. Порядковые значения (пусть – число строк в датасете, – функция получения ранга элемента):
   1. Если – значение, то расстояние вычисляется по формуле (9):
   2. Если – диапазон, то расстояние рассчитывается по формуле, аналогичной пункту 1.b, то есть (формула (10)):
3. Номинальные значения:
   1. Если – значение, то расстояние равно 1 при совпадении элементов и 0 при несовпадении
   2. Если – множество, то расстояние вычисляется как вероятность угадать при равновероятном выборе, то есть по формуле (11):

Результирующая метрика представляет собой среднее арифметическое из полученных расстояний.

Важно отметить, что данный подход применим к наборам данных, обезличенным любым способам, не обязательно с применением классов эквивалентности.

### 2.3.7. Оценка полезности обезличенных данных при помощи градиентного бустинга

Данный метод основан на том, чтобы применить модель машинного обучения на изначальных данных и на обезличенных данных и сравнить результаты. Так как данные из датасетов нужны для решения задач, похожих на задачи моделей машинного обучения, данный метод правдоподобно на практике оценивает полезность обезличенных данных.

Минусом данного подхода является тот факт, что не на всех данных возможно хорошо решить какую-либо задачу при помощи модели машинного обучения. Также минусом является особенность, что результат оценки полезности данных очень сильно зависит от поставленной задачи, то есть невозможно оценить потерю полезности данных в общем.

Для оценки полезности этим способом совершаются следующие действия:

1. Выбирается столбец, который будет предсказывается
2. Выборка разделяется на обучающую и тестовую
3. Выборки обезличиваются
4. Производится преобразование обезличенных данных в соответствии с пунктом 2.1.2.
5. Производится обучение при помощи градиентного бустинга по причинам аналогичным причинам, описанным в пункте 2.1.6. с применением перебора гиперпараметров (grid search)
6. По тестовой выборка оценивается метрика MAPE (как не зависящая от порядка предсказываемых данных и интерпретируемая метрика) для регрессии и accuracy для классификации

## 2.4. Выводы главы

Исследователями предложено большое количество способов оценки качества обезличивания данных. Однако общей проблемой этих подходов является их неприменимость к данным, обезличивание которых производилось не на основе классов эквивалентности. В данной главе были предложены подходы, которые позволяют устранить этот недостаток.

# АЛГОРИТМЫ ОБЕЗЛИЧИВАНИЯ

## 3.1. Методы обезличивания

Методом обезличивания назовём подход к обработка строк набора данных после разбиения на классы эквивалентности внутри класса эквивалентности.

### 3.1.1. Подавление

Подавление заключается в замене некоторых отдельных значений на отсутствующие значения (None).

Для того, чтобы осуществить подавление необходимо выполнить следующие действия:

1. Каждый раз рассматривается какой-то класс эквивалентности
2. Для каждого класса эквивалентности сравнить каждую строку с какой-либо заранее выбранной строкой из класса эквивалентности
3. Если в каком-либо столбце хотя бы 1 раз возникало несовпадение в сравнении в пункте 2, - подавить все значения в этом столбце в данном классе эквивалентности

### 3.1.2. Обобщение

Обобщение заключается в замещении отдельных значений на диапазоны и множества, описанные в пункте 2.1.

Для того, чтобы осуществить обобщение необходимо выполнить следующие действия:

1. Каждый раз рассматривается какой-то класс эквивалентности
2. Если столбец имеет вещественный или порядковый тип, то необходимо найти максимальное и минимальное значение в этом столбце в данном классе эквивалентности из заменить все значения в столбце в данном классе эквивалентности на диапазон с границами в виде найденных минимума и максимума
3. Если столбец имеет номинальный тип, то необходимо найти множество всех значений, встречающихся в данном столбце в данном классе эквивалентности и заменить все значения в столбце в данном классе эквивалентности на данное множество

### 3.1.3. Агрегация

Агрегация заключается в подсчёте функции агрегации над всеми значениями в данном столбце в данном классе эквивалентности и замене на получившееся значение всех элементов данном столбце в данном классе эквивалентности.

Для того, чтобы осуществить агрегацию необходимо выполнить следующие действия:

1. Каждый раз рассматривается какой-то класс эквивалентности
2. Если столбец имеет вещественный тип, то функцией агрегации является среднее арифметическое
3. Если столбец имеет порядковый тип, то функцией агрегации является медиана (если число элементов чётно, берётся меньшее значение)
4. Если столбец имеет номинальный тип, то функцией агрегации является мода

## 3.2. Общие алгоритмы обезличивания квази-идентификаторов

В данном пункте рассматриваются алгоритмы, которые подавляют идентификаторы и работают только с квази-идентификаторами, причём принимают на вход 1 датасет. Под алгоритмом подразумевается

В данной работе при разработке собственных алгоритмов целью было избавиться от VGH (Value Generalization Hierarchy), так как:

1. Она не позволяет достичь достаточно качественного с точки зрения сохранения полезности данных обезличивания, так как задаётся человеком (а не строится с использованием современных вычислительных возможностей)
2. Собственно, требует от человека её задание, что является ответственной задачей, так как в значительной степени определяет результат работы алгоритма
3. Не позволяет объективно оценить качество работы алгоритмов и произвести их сравнение (что является одной из задач данной работы), так как не гарантируется, что VGH для сравниваемых алгоритмов заданы оптимально

### 3.2.1. Алгоритм перебора

Так как достижение оптимальной с точки зрения полезности k-анонимности является NP-сложной задачей, был рассмотрен алгоритм перебора. Данный алгоритм осуществляет перебор всех возможных наборов отдельных значений датасета и проверяет, соответствует ли датасет заданной метрике:

1. K-анонимности
2. L-разнообразию
3. T-близости

Среди всех подходящих вариантов выбирается вариант с наименьшим количеством подавлений.

Данный алгоритм имеет асимптотику , где – число строк, – число столбцов, так как всего рассматривается случаев и для каждого случая самой алгоритмически сложной операцией является проверка на k-анонимность, которая при использовании хеш-таблицы имеет асимптотику .

Такая асимптотика делает алгоритм непригодным для практического использования.

### 3.2.2. Алгоритм жадного выбора на основе расстояний

Алгоритм осуществляет разбиение исходного датасета на группы следующей последовательностью действий:

1. Подсчитать расстояния между всеми строками квази-идентификаторов попарно. Расстояние представляет собой среднее арифметическое из расстояний между соответствующими элементами строки, рассчитываемыми по методике, описанной в пункте 2.3.6. с некоторыми особенностями:
   1. Если в конце обезличивание осуществляется обобщением или агрегацией, то подсчёт расстояния осуществляется в точном соответствии с методикой
   2. Если в конце обезличивание осуществляется подавлением, то считается, что все столбцы содержат номинальные значения, так как для подавление значение имеет только равенство элементов
2. Пока есть не отнесённые к какое-либо группе строки:
   1. Выбрать какую-либо не отнесённую к какой-либо группе строку
   2. Отсортировать расстояния от данной строки до всех остальных в порядке возрастания
   3. Начиная с самой близкой строки объединять строки в группу, пока для полученной группы не будет достигнута соответствующая метрика:
      1. K-анонимность
      2. L-разнообразие
3. Если остались нераспределённые строки, а собрать группу, соответствующую метрике, не получается, то данные строки добавляются к последней полученной группе
4. В результате набор полученных групп представляет собой набор классов эквивалентности, соответствующий заданной метрике

После получения набора классов эквивалентности осуществляется обезличивание датасета одним из рассмотренных в пункте 3.2.1. методов.

Данный алгоритм неприменим для достижения t-близости, так как увеличение размера класса эквивалентности не гарантирует достижение t-близости.

### 3.2.3. Алгоритм на основе последовательного объединения групп

Данный алгоритм предполагает последовательное объединение групп до момента достижения метрики:

1. K-анонимности
2. L-разнообразия
3. T-близости

В ходе выполнения алгоритма выполняются следующие действия:

1. Изначально каждая строка находится в своей группе
2. Пока не достигнута метрика:
   1. Выбирается случайная группа, не удовлетворяющая метрике (в любом случае её рано или поздно придётся с чем-то объединять)
   2. Между выбранной группой и остальными группами (включая группы, в которых достигнута метрика) рассчитываются расстояния по следующему принципу:
      1. При помощи способа, приведённого в пункте 2.3.6., рассчитывается расстояние между строками в данных двух группах в изначальном датасете и в датасете, который получился бы при обезличивании в текущей конфигурации групп (расстояние считается для каждой группы отдельно, а затем складывается в связи с особенностями, описанными в пунктах 2.2.3. – 2.2.5. данного алгоритма)
      2. При помощи способа, приведённого в пункте 2.3.6., рассчитывается расстояние между строками в данных двух группах в изначальном датасете и в датасете, который получился бы при обезличивании в конфигурации групп, в которой дополнительно объединены данные 2 группы
      3. Для k-анонимности расстояние остаётся в исходном виде
      4. Для l-разнообразия расстояние дополнительно делится на сумму минимумов из l и количеств уникальных чувствительных значений в каждом столбце с чувствительными значениями (сумма из значений минимумов для столбцов)
      5. Для t-близости расстояние дополнительно умножается на сумму максимумов из t и расстояний между распределением чувствительных значений в группе и чувствительных значений во всём датасете в каждом столбце с чувствительными значениями (сумма из значений максимумов для столбцов)
      6. Итоговое расстояние вычисляется как разность расстояния, полученного в пункте 2.2.2., и расстояния, полученного в пункте 2.2.1. данного алгоритма
      7. Для того, чтобы не вычислять каждый раз расстояние из пункта 2.2.1. данного алгоритма может выполняться хранение расстояний из пункта 2.2.2. данного алгоритма
      8. Датасеты могут не обезличиваться каждый раз, и в таком случает расстояние будет подсчитываться на основе вычисления минимумов, максимумов, медиан, мод и других статистик в соответствии с выбранным методом обезличивания из пункта 3.1.
   3. Берётся ближайшая к данной группе группа и производится их объединение
3. В конце производится обезличивание на основе полученных классов эквивалентности и выбранного метода из пункта 3.1.

Преимуществом данного алгоритма является возможность достигать t-близости за разумное время.

### 3.2.4. Модифицированный Datafly

Как уже было упомянуто, данный алгоритм является модификацией предложенного исследователями алгоритма Datafly, которая заключается в автоматизации построения VGH.

В данном алгоритме достигается только метрика k-анонимности.

В данном алгоритме для выполнения обезличивания выполняются следующие действия:

1. Пока количество строк, не относящихся к классу эквивалентности, содержащему хотя бы k элементов, больше указанного значения:
   1. Выбрать столбец с квази-идентификаторами с наибольшим количеством уникальных значений
   2. В данном столбце преобразовать 2 уникальных значения в одно. Значения выбираются по следующим принципам:
      1. Для номинальных значений:
         1. 2 с наименьшим количеством представителей в столбце
         2. Обобщённое значение – множество минимальной мощности, содержащее оба значения (значения могут быть множествами)
      2. Для порядковых и вещественных значений:
         1. Выбирается значение с наименьшим количеством представителей в столбце и соседнее с точки зрения порядка с ним значение, имеющее меньшее количество представителей в столбце
         2. Диапазоны сортируются по минимальному значению
         3. Обобщённое значение – диапазон минимального размера, включающий оба значения (значения могут быть диапазонами)
2. Целиком подавить квази-идентификаторы в строках, не относящихся к классу эквивалентности, содержащему хотя бы k элементов

Таким образом, VGH строится на этапе выбора объединяемых уникальных значений в столбце.

### 3.2.5. Алгоритм разбиения столбцов на равные группы с обобщением

В данном алгоритме достигается только метрика k-анонимности.

В данном алгоритме для выполнения обезличивания выполняются следующие действия:

1. Возьмём за k размер минимальной группы разбиения столбца
2. Прибавлять 1 к размеру, пока не достигнута k-анонимность следующим подходом к обезличиванию для каждого отдельного столбца с квази-идентификаторами:
   1. Если столбец вещественный или порядковый:
      1. Отсортировать столбец
      2. Добавлять в текущую группу наборы элементов, соответствующих какому-либо уникальному значению, в порядке возрастания уникальных значений пока не набрана группа хотя бы заданного размера
      3. Таким образом образуется определённое количество групп
      4. Оставшиеся строки добавляются к последней группе
      5. Производится обобщение в соответствии с разбиением на группы
   2. Если столбец номинальный:
      1. Разбиение на группы происходит аналогично пункту 2.1. данного алгоритма, но уникальные значения идут в случайном порядке

### 3.2.6. Алгоритм рандомизации

В данном алгоритме каждое значение заменяется на близкое, причём степень близости зависит от сгенерированного из стандартного нормального распределения значения и параметра scale.

Генерация нового значения квази-идентификатора происходит по следующему принципу:

1. Для номинальных значений:
   1. Из всех (с повторениями) значений данного столбца равновероятно выбирается какое-либо
2. Для порядковых значений:
   1. Рассчитывается сдвиг по формуле (12):

- сдвиг

- сгенерированное из стандартного нормального распределения

*–* число строк в датасете

*–* параметр алгоритма рандомизации

* 1. Пусть то новое значение вычисляется по формуле (13):

– старое значение

*–* новое значение от старого значения

– сдвиг

*–* число строк в датасете

1. Для вещественных значений:
   1. По значениям, содержащимся в столбце, строится гистограмма и из её опорных значений случайно выбирается число значений, равное числу значений в столбце. Назовём этот построенный отсортированный массив массивом генерации
   2. Рассчитывается сдвиг по формуле (14):

- сдвиг

- сгенерированное из стандартного нормального распределения

*–* число строк в датасете

*–* параметр алгоритма рандомизации

* 1. Рассмотрим функцию, полученную из массива генерации линейной аппроксимацией и взятием индексов массива за абсциссы, а значений – за ординаты. Обозначим её
  2. Новое значение вычисляется по формуле (15):

– старое значение

*–* новое значение от старого значения

– сдвиг

*–* число строк в датасете

Важно отметить, что данный алгоритм не опирается на классы эквивалентности при обезличивании, поэтому оценка качества его работы может производиться только при помощи подходов, описанных в пунктах 2.2.6., 2.3.6, 2.3.7.

## 3.3. Алгоритмы обезличивания, применяемые для специальных целей

В данном пункте содержатся все рассмотренные алгоритмы, не отвечающие цели, указанной в пункте 3.2.

### 3.3.1. Алгоритмы перемешивания

Данный алгоритм может быть представлен в двух версиях:

1. Перемешивание всего датасета
2. Перемешивание внутри классов эквивалентности

В ходе выполнения алгоритма для перемешивания всего датасета выполняются следующие действия:

1. Выбираются столбцы, которые будут перемешиваться
2. Случайно выбираются строки, которые будут меняться местами
3. Осуществляется перемешивание строк в выбранных столбцах

В ходе выполнения алгоритма для перемешивания внутри классов эквивалентности выполняются следующие действия:

1. Выбираются столбцы, которые будут перемешиваться
2. Случайно выбираются строки, которые будут меняться местами внутри каждого класса эквивалентности
3. Осуществляется перемешивание строк внутри каждого класса эквивалентности (значение может быть поменяно местами только со значением внутри того же класса эквивалентности) в выбранных столбцах

### 3.3.2. Алгоритм хеширования

Данный алгоритм хеширует или шифрует значения, являющиеся идентификаторами. Таким образом, эти значения теряют своё идентифицирующее свойство, но всё ещё позволяют исследователям, проводящим анализ данных сравнивать их на равенство, что может увеличить полезность обезличенных данных.

### 3.3.3. DPP2GA (объединение распределённого горизонтально датасета)

В данном подходе ставится следующая задача:

1. Имеются 2 владельца датасетов
2. Каждый имеет датасет с одинаковым количеством строк, причём для каждой строки, содержащей персональные данные какого-либо субъекта, первого датасета, присутствует строка для того же субъекта персональных данных во втором датасете
3. У каждого субъекта есть идентификатор, присутствующий в каждом датасете
4. Столбцы чувствительных значений в датасетах не совпадают
5. Столбцы квази-идентификаторов могут
6. Задачей является объединить данные датасетов в один датасет, без утечки чувствительной информации от одного владельца к другому

Данный подход опирается на k-анонимность.

Введём понятие равенства двух наборов классов эквивалентности в рамках данного алгоритма. 2 набора классов эквивалентности равны, если (формула (16)):

– множество классов эквивалентности датасета 1

– множество классов эквивалентности датасета 2

– множество идентификаторов субъектов персональных данных, которым принадлежат строки, входящие в класс эквивалентнтсти

– множество идентификаторов субъектов персональных данных, которым принадлежат строки, входящие в класс эквивалентнтсти

– параметр k-анонимности

Таким образом, данная задача решается следующей последовательность действий:

1. Идентификаторы хешируются (шифруются)
2. Датасеты обезличиваются локально при помощи какого-либо алгоритма (в данном отчёте рассматривается алгоритм, описанный в пункте 3.2.3.)
3. Каждый владелец датасета шифрует каждый отдельный идентификатор в множестве , где – номер владельца, при помощи коммутативной функции шифрования (в данной работе применяется коммутативное шифрование на эллиптических кривых)
4. Каждый владелец датасета отправляет другому владельцу зашифрованное множество
5. Каждый владелец датасета шифрует идентификаторы полученного множества при помощи своей функции шифрования
6. Так как шифрование коммутативное, каждый идентификатор для каждого владельца представлен одинаково (то есть, один и тот же идентификатор в дважды зашифрованных множествах имеет один и тот же вид)
7. Каждый владелец отправляет другому полученное им дважды зашифрованное множество
8. Каждый владелец проверяет на равенство (по методике, описанной выше) 2 полученных им набора классов эквивалентности в виде зашифрованных множеств
9. Если наборы не равны, то каждый владелец выполняет 1 шаг обезличивания при помощи какого-либо алгоритма обезличивания, работающего по пошаговой модели (например, Datafly или алгоритм, предложенный в 3.2.3.). В рамках данной работы предполагается, что используется алгоритм, предложенный в пункте 3.2.3. Затем производится переход к шагу 3 данного алгоритма
10. Если наборы равны, то производится обмен незашифрованными обезличенными датасетами без идентификаторов и их объединение на основе номеров строк, разделённых по классам эквивалентности. При этом при объединении присутствует проблема, заключающаяся в том, что неизвестно (и не может быть известно), какая строка в датасете первого владельца относится к какой строке второго владельца. В данной работе предлагается устанавливать соответствие случайно. Таким образом будет получен объединённый датасет с обезличенными строками.

В результате ни один из владельцев не получит информации о чувствительных значениях в датасете другого больше, чем позволяет k-анонимность, так как такая информация может быть получена на основе пересечения классов эквивалентности, а любое такое непустое пересечение содержит не менее k строк.

## Выводы главы

В данной главе были рассмотрены алгоритмы обезличивания, соответствующие каждому из подходов, определённых в законодательстве и перечисленных в пункте 1.1. данного отчёта.

При этом алгоритмы разрабатывались таким образом, чтобы их последующее сравнение было максимально объективным и минимально зависело от заданных параметров алгоритма. Также данные алгоритмы не требуют детального анализа имеющихся данных для проведения обезличивания.

# ЭКСПЕРИМЕНТЫ НАД АЛГОРИТМАМИ ОБЕЗЛИЧИВАНИЯ

## 4.1. Библиотека для обезличивания

В ходе выполнения курсовой работы был разработан Python пакет с реализованными перечисленными в главе 2 методами оценки качества и перечисленными в главе 3 алгоритмами обезличивания.

Пакет был загружен при помощи workflow GitHub в репозиторий библиотек PyPI. Установка пакета осуществляется командой: pip install cp2025.

Реализации методов оценки качества описаны в приложении 2.

Каждый алгоритм обезличивания представляет собой класс, которому в конструкторе передаются необходимые параметры, описанные в приложении 3. Для осуществления обезличивания осуществляется вызов метода depersonalize(df, identifiers\_ids, quasi\_identifiers\_ids, sensitives\_ids), где:

1. df – список, numpy array или pandas DataFrame
2. identifiers\_ids – индексы столбцов с идентификаторами
3. quasi\_identifiers\_ids – индексы столбцов с квази-идентификаторами
4. sensitives\_ids – индексы столбцов с чувствительными значениями

Некоторые алгоритмы требуют указания типов столбцов. В этом случае в них передаётся список длины равной количеству столбцов, для которых указываются типы, с следующими строковыми значениями:

1. “real” – для вещественного столбца
2. “ordered” – для порядкового столбца
3. “unordered” – для номинального столбца

Метод depersonalize возвращает кортеж из обезличенного датасета и статистик, собранных при обезличивании.

Диапазоны и множества для всех алгоритмов представлены в виде класса GeneralizationRange, который имеет следующие поля:

1. mn – нижняя граница диапазона
2. mx – верхняя граница диапазона
3. column\_type – тип столбца (“real”, “ordered”, “unordered”)
4. column\_values – множество значений

Для преобразования обезличенного датасета к воспринимаемому моделью машинного обучения формату в соответствии с пунктом 2.1.2. в пакете реализована функция prepare\_data, содержащаяся в модуле cp2025.utility.prepare\_data. Функция принимает следующие аргументы:

1. train – обучающую выборку
2. test – тестовую выборку
3. has\_ranges – массив булевых значений; True, если в столбце содержатся диапазоны или множества и False, если нет
4. column\_types – список типов столбцов
5. normalize – True, если необходима нормализация данных и False иначе (True по умолчанию)

При установке пакета в систему устанавливается приложение командной строки depersonalize, которое позволяет осуществить быстрое обезличивание данных (описано в приложении 4). Пример использования: depersonalize -h (вызов справки).

Ссылка на GitHub репозиторий с исходным кодом проекта: <https://github.com/ShompolovMaxim/CP_2025>

## 4.2. Данные использующиеся в экспериментах

Для проведения экспериментов были использованы 2 датасета:

1. Датасет с Kaggle [[20]](#р20) с данными клиентов банка
2. Сгенерированный при помощи ChatGpt датасет с различными квази-идентификаторами и идентификаторами: репозитория курсового проекта, data/ Bank\_Personal\_Loan\_Modelling.csv

### 4.2.1. Датасет с Kaggle

Датасет содержит 5000 строк со следующими столбцами:

* ID - идентификатор записи (идентификатор)
* Age - возраст (квази-идентификатор)
* Experience - стаж (квази-идентификатор)
* Income - заработная плата (чувствительное значение)
* ZIP Code - почтовый индекс (квази-идентификатор)
* Family - число членов семьи (квази-идентификатор)
* CCAvg - средний процент по кредитной карте (чувствительное значение)
* Education - уровень образования (квази-идентификатор)
* Mortgage - размер ипотеки (чувствительное значение)
* Personal Loan - число потребительских кредитов (чувствительное значение)
* Securities Account - наличие счёта ценных бумаг (чувствительное значение)
* CD Account - наличие CD-счёта (чувствительное значение)
* Online - использование онлайн услуг (чувствительное значение)
* CreditCard - наличие кредитной карты (чувствительное значение)

При проведении экспериментов используются:

* Первые 1000 строк
* Квази-идентификаторы: Age, Experience, ZIP Code, Family, Education
* Чувствительные значения: Income, CCAvg, Mortgage

Также для проведения экспериментов с алгоритмом перебора используется укороченный датасет с Kaggle до 12 строк и 5 столбцов (с целью уменьшить время выполнения). В этом случае квази-идентификаторами выступают Age и Experience, а чувствительные значения сохраняются. Обозначим этот датасет коротким.

### 4.2.2. Сгенерированный датасет

Датасет содержит 10000 строк со следующими столбцами:

* gender - пол (квази-идентификатор)
* birth\_date - дата рождения (квази-идентификатор)
* age - возраст (квази-идентификатор)
* city - город (квази-идентификатор)
* postal\_code - почтовый индекс (квази-идентификатор)
* country - страна (квази-идентификатор)
* loan\_purpose - цель займа (квази-идентификатор)
* education\_level - уровень образования (квази-идентификатор)
* employment\_status - занятость (квази-идентификатор)
* years\_of\_experience - стаж (квази-идентификатор)
* family\_size - число членов семьи (квази-идентификатор)
* number\_of\_children - число детей (квази-идентификатор)
* marital\_status - статус брака (квази-идентификатор)
* housing\_status - статус жилья (квази-идентификатор)
* industry - сфера деятельности (квази-идентификатор)
* disease - заболевание (чувствительное значение)
* chronic\_condition - хроническое заболевание (чувствительное значение)
* insurance\_status - страховой статус (чувствительное значение)
* monthly\_medical\_expenses\_usd - траты на лечение в месяц в долларах (чувствительное значение)
* monthly\_income\_usd - заработная плата в долларах (чувствительное значение)
* monthly\_expenses\_usd - траты в месяц в долларах (чувствительное значение)
* loan\_amount\_usd - общий объём займов в долларах (чувствительное значение)
* credit\_score - кредитный рейтинг (чувствительное значение)

При проведении экспериментов используются:

* Первые 1000 строк
* Все квази-идентификаторы
* Чувствительные значения chronic\_condition, monthly\_medical\_expenses\_usd

## Методика проведения экспериментов

Цели проведения экспериментов:

1. Оценить особенности используемых датасетов в контексте обезличивания и оценки его качества
2. Оценить, насколько точно используемые метрики отражают степень качества обезличивания
3. Провести сравнение методов обезличивания и оценить степень выполнения ими задачи обезличивания в соответствии с поставленной задачей (метрикой) с максимальным сохранением полезности данных
4. Провести сравнение алгоритмов разбиения на классы для последующего обезличивания
5. Произвести сравнение алгоритмов рандомизации и перебора с другими алгоритмами

Эксперименты проводятся над общими алгоритмами обезличивания квази-идентификаторов (пункт 3.2.), так как все они имеют одну и ту же задачу, а следовательно, сравнимы между собой.

Соответственно сравниваются следующие алгоритмы в следующих сценариях:

1. Алгоритм перебора:
   1. Рассматриваются отдельно версии алгоритма для достижения k-анонимности, l-разнообразия и t-близости
   2. Алгоритм имеет только версию для подавления
   3. Производится обезличивание только короткого датасета
2. Алгоритм жадного выбора на основе расстояний (условное название time optimal):
   1. Рассматриваются отдельно версии алгоритма для достижения k-анонимности и l-разнообразия
   2. Рассматриваются отдельно версии алгоритма, использующие агрегацию, обобщение и подавление
   3. Производится обезличивание всех датасетов (сгенерированный, Kaggle, короткий)
3. Алгоритм на основе последовательного объединения групп (условное название group join):
   1. Рассматриваются отдельно версии алгоритма для достижения k-анонимности, l-разнообразия и t-близости
   2. Рассматриваются отдельно версии алгоритма, использующие агрегацию, обобщение и подавление
   3. Производится обезличивание всех датасетов (сгенерированный, Kaggle, короткий)
4. Модифицированный Datafly (условное название Datafly):
   1. Рассматривается достижение k-анонимности
   2. Алгоритм использует одновременно обобщение и подавление
   3. Производится обезличивание всех датасетов (сгенерированный, Kaggle, короткий)
5. Алгоритм разбиения столбцов на равные группы с обобщением (условное название equal sized)
   1. Рассматривается достижение k-анонимности
   2. Алгоритм использует обобщение
   3. Производится обезличивание всех датасетов (сгенерированный, Kaggle, короткий)
6. Алгоритм рандомизации
   1. Производится обезличивание сгенерированного и Kaggle датасетов

Для сгенерированного и Kaggle датасетов используются следующие наборы параметров (scale – параметр рандомизации):

1. k=2, l=2, t=1, scale = 0,05
2. k=10, l=3, t=0,5, scale = 0,25
3. k=100, l=4, t=0.1, scale = 1

Для короткого датасета применяются следующие наборы параметров:

1. k=2, l=2, t=1
2. k=3, l=3, t=0,5
3. k=4, l=4, t=0.1

Если алгоритм не требует какого-либо параметра из набора, то этот параметр может не применяться.

Для оценки качества применяются все рассмотренные в главе 2 подходы, кроме среднего размера класса эквивалентности (то же самое, что и distinctness) и энтропии (не рассматривается относительно изначального датасета, поэтому непоказательна) с некоторыми исключениями:

1. Для короткого датасета подходы, связанные с машинным обучением, не применяются, так как данный датасет имеет недостаточный размер
2. Для всех алгоритмов, кроме рандомизации, оценка рисков с помощью градиентного бустинга не применяется по причинам, описанным в пункте 2.2.6.
3. Для рандомизации не используются подходы к оценке качества, опирающиеся на классы эквивалентности

## 4.4. Результаты проведения экспериментов

Эксперименты содержатся в Jupiter Notebook в репозитории проекта: experiments/КП2025\_эксперименты.ipynb. Также в данном файле приведена демонстрация работы не рассмотренных в данном пункте алгоритмов.

### 4.4.1. Эксперименты с сгенерированным датасетом

Для данного датасета MAPE при предсказании столбца monthly\_medical\_expenses\_usd составляет: 1,63

* Эксперименты с набором параметров k=2, l=2, t=1, scale = 0,05 – приложение 4 (таблица 4, таблица 5)
* Эксперименты с набором параметров k=10, l=3, t=0,5, scale = 0,25 – приложение 4 (таблица 6, таблица 7)
* Эксперименты с набором параметров k=100, l=3, t=0,1, scale = 1 – приложение 4 (таблица 8, таблица 9)

### 4.4.2. Эксперименты с датасетом с Kaggle

Для данного датасета accuracy при предсказании столбца Family (4 различных значения) составляет: 0,83

* Эксперименты с набором параметров k=2, l=2, t=1, scale = 0,05 – приложение 5 (таблица 10, таблица 11)
* Эксперименты с набором параметров k=10, l=3, t=0,5, scale = 0,25 – приложение 5 (таблица 12, таблица 13)
* Эксперименты с набором параметров k=100, l=3, t=0,1, scale = 1 – приложение 5 (таблица 14, таблица 15)

### 4.4.3. Эксперименты с коротким датасетом

* Эксперименты с набором параметров k=2, l=2, t=1, scale = 0,05 – приложение 6 (таблица 16, таблица 17)
* Эксперименты с набором параметров k=10, l=3, t=0,5, scale = 0,25 – приложение 6 (таблица 18, таблица 19)
* Эксперименты с набором параметров k=100, l=3, t=0,1, scale = 1 – приложение 6 (таблица 20, таблица 21)

## 4.5. Выводы из результатов проведения экспериментов

Датасеты:

1. Сгенерированный датасет содержит недостаточное количество зависимостей, поэтому метрики, применяющие машинное обучение, работают на нём плохо (даже на не обезличенном датасете модели машинного обучения показывают низкое качество)
2. Наиболее объективные результаты с использованием методов машинного обучения получены на датасете с Kaggle
3. Короткий датасет сильно подвержен потере информации при обезличивании, поэтому результаты для двух последних наборов параметров алгоритмов на этом датасете отличаются незначительно

Точность достижения заданной метрики (степень значительности её превышения результатами обезличивания):

1. Наименее точно достигается метрика t-близости, так как само её достижение является достаточно сложной задачей и часто её достижение возможно только при очень малом числе классов эквивалентности
2. Наиболее точно с достижением k-анонимности справляются методы, основанные на обобщении или агрегации, а методы, основанные на подавлении, справляются значительно хуже, так как используют расстояние Хэмминга для разбиения на классы эквивалентности
3. Алгоритм на основе последовательного объединения групп достигает несколько большего значения l, чем заданное ему, по сравнению с остальными алгоритмами
4. В целом достижение метрик подавлением значительно менее точное

Сравнение методов обезличивания:

1. С точки зрения потери информации и степени изменения датасета хуже всего себя показывает подавление, затем агрегация, а лучше всего – обобщение. Другими словами, подавление само по себе делает данные наименее информативными, а обобщение – наиболее информативными
2. Наименее длительное время выполняется обезличивания с использованием подавления, затем – с использованием обобщения, а дольше всего – с использованием агрегации
3. Наиболее большое число отдельных элементов датасета изменяется при использовании подавления и обобщения

Особенности работы алгоритмов разбиения на классы эквивалентности и значений методов оценки качества при достижении k-анонимности, l-разнообразия или t-близости

1. Adversarial knowledge gain в значительной степени зависит от достигнутого максимального l, что соответствует её определению
2. Adversarial accuracy gain в значительной степени зависит от минимального достигнутого t, что также соответствует её определению
3. К-анонимность с наименьшей потерей полезности достигается при использовании алгоритма жадного выбора на основе расстояний
4. L-разнообразие с наименьшей потерей полезности достигается при использовании алгоритма на основе последовательного объединения групп
5. T-близость за разумное время и с наименьшей потерей полезности достигается только при использовании алгоритма на основе последовательного объединения групп

Сравнение алгоритмов по времени выполнения:

1. За малое время (менее 5 секнуд) выполняются алгоритмы жадного выбора на основе расстояний, модифицированный Datafly и рандомизации
2. За среднее время (менее минуты) выполняются алгоритмы: на основе последовательного объединения групп для достижения k-анонимности или l-разнообразия и разбиения столбцов на равные группы с обобщением
3. За большое время (менее 10 минут) выполняется алгоритм на основе последовательного объединения групп для достижения t-близости
4. Алгоритм перебора для датасета средних разменов выполняется слишком большое время

Сравнение алгоритмов на основе метрик, использующих машинное обучение:

1. Как уже было упомянуто, метрики, использующие машинное обучение актуальны только для датасета с Kaggle, а метрику оценки рисков на основе градиентного бустига для рандомизации необходимо сравнивать с метрикой distinctness для остальных алгоритмов
2. При достижении k-анонимности даже при малых k предсказание целевой переменной в метрике полезности сводится к угадыванию (accuracy близка к 0,25, а всего 4 различных значения), что говорит о том, что полезности данных сильно уменьшается даже при достижении этой метрики, не говоря о l-разнообразии и t-близости
3. Для рандомизации уменьшение accuracy в метрике полезности не так значительно, как для других алгоритмов, что говорит о том, что добавление случайных значений к элементам таблицы не так значительно уменьшает полезности данных, как достижение k-анонимности
4. Если при scale равном 0,05 алгоритм рандомизации имеет оценку рисков с использованием градиентного бустинга равную 0,65 (distinctness равное 0,5 эквивалентно k-анонимности с идеальными разбиением на классы), что с точки зрения рисков обезличивания хуже, чем 2-анонимнсть, то для scale равного 0,25 модель уже практически не может предсказать владельца персональных данных (при этом оценка полезности с использованием градиентного бустинга остаётся значимо более высокой, чем для простого угадывания)

Итоговое сравнение алгоритмов и методов обезличивания:

1. Наиболее точно необходимая метрика достигается с использованием обобщения и алгоритма жадного выбора на основе расстояний
2. Наилучшее сочетание быстродействия и качества обезличивания демонстрирует алгоритм жадного выбора на основе расстояний
3. Единственный алгоритм, который может достичь t-близости за разумное время - алгоритм на основе последовательного объединения групп
4. Алгоритм рандомизации показывает лучшие результаты с точки зрения рисков деобезличивания и полезности обезличенных данных по сравнению с алгоритмами, основывающимися на классах эквивалентности, однако не даёт гарантий безопасности (какое-то какая-нибудь строка может быть недостаточно искажена и впоследствии деобезличена)
5. Алгоритмы модифицированный Datafly и разбиения столбцов на равные группы с обобщением не имеют равных преимуществ над другими алгоритмами
6. Алгоритм перебора не показывает лучшие результаты, чем другие алгоритмы

## Выводы главы

В данной главе был подробно рассмотрен подготовленный в ходе выполнения курсового проекта Python пакет с алгоритмами обезличивания и метриками оценки его качества.

Также были рассмотрены результаты проведённых с использованием данного пакета экспериментов и выявлены сильные и слабые стороны реализованных алгоритмов обезличивания.

# ЗАКЛЮЧЕНИЕ

В ходе проделанной работы:

1. Были рассмотрены положения законодательства об обезличивании персональных данных
2. Были рассмотрены основные, предложенные исследователями, методы оценки качества обезличивания персональных данных
3. Были предложены свои методы оценки качества обезличивания персональных данных как в общем случае, так и для специальных задач
4. Были предложены алгоритмы обезличивания табличных данных без использования VGH для численных, порядковых и номинальных значений
5. Все рассмотренные алгоритмы и методы оценки качества были реализованы на языке программирования Python в виде пакета, загруженного в репозиторий PyPI
6. Были проведены эксперименты над разработанным пакетов с использованием 2 наборов данных
7. На основе проведённых экспериментов был проведён сравнительный анализ реализованных алгоритмов и были выявлены их слабые стороны

Поставленные задачи были выполнены в полном объёме

Направления дальнейшей работы:

1. Адаптация других, предложенных исследователями алгоритмов для использования без задания VGH
2. Дополнение разработанного пакета ещё не рассмотренными алгоритмами
3. Оптимизация рассмотренных алгоритмов с точки зрения качества обезличивания и времени выполнения
4. Доработка разработанного пакета с точки зрения стандартов программирования на языке Python

# СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ

1. Федеральный закон от 27.07.2006 №152-ФЗ «О персональных данных»
2. Приказ Федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций от 5 сентября 2013 г. N 996 "Об утверждении требований и методов по обезличиванию персональных данных"
3. Регламент (ЕС) 2016/679 Европейского парламента и Совета от 27 апреля 2016 г. о защите физических лиц при обработке персональных данных и о свободном обращении таких данных, и об отмене Директивы 95/46/ЕС (Общий регламент о защите данных) // Официальный журнал Европейского союза. – 2016. – L 119. – С. 1–88.
4. Samarati P., Sweeney L. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. – 1998
5. k-anonymity, the parent of all privacy definitions // https://desfontain.es URL: https://desfontain.es/blog/k-anonymity.html (дата обращения: 01.12.2024).
6. Machanavajjhala A. et al. l-diversity: Privacy beyond k-anonymity //Acm transactions on knowledge discovery from data (tkdd). – 2007. – Т. 1. – №. 1. – С. 3-es.
7. l-diversity, because reidentification doesn't tell the whole story // https://desfontain.es URL: https://desfontain.es/blog/l-diversity.html (дата обращения: 01.12.2024).
8. Li N., Li T., Venkatasubramanian S. Closeness: A new privacy measure for data publishing //IEEE Transactions on Knowledge and Data Engineering. – 2009. – Т. 22. – №. 7. – С. 943-956.
9. Brickell J., Shmatikov V. The cost of privacy: destruction of data-mining utility in anonymized data publishing //Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. – 2008. – С. 70-78.
10. ARX – Data Anonymization Tool [Электронный ресурс]. – URL: https://arx.deidentifier.org (дата обращения: 12.05.2025).
11. Garfinkel S. L. De-Identification of Personal Information //National Institute of Standard and Technology. – 2015.
12. Samarati P. Protecting respondents identities in microdata release //IEEE transactions on Knowledge and Data Engineering. – 2002. – Т. 13. – №. 6. – С. 1010-1027.
13. Meyerson A., Williams R. On the complexity of optimal k-anonymity //Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. – 2004. – С. 223-228.
14. Gkoulalas-Divanis A., Loukides G., Sun J. Publishing data from electronic health records while preserving privacy: A survey of algorithms //Journal of biomedical informatics. – 2014. – Т. 50. – С. 4-19.
15. Sweeney L. Achieving k-anonymity privacy protection using generalization and suppression //International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. – 2002. – Т. 10. – №. 05. – С. 571-588.
16. LeFevre K., DeWitt D. J., Ramakrishnan R. Mondrian multidimensional k-anonymity //22nd International conference on data engineering (ICDE'06). – IEEE, 2006. – С. 25-25.
17. Borisov S. A., Bosov A. A., Ivanov D. E. APPLICATION OF SIMULATED COMPUTER SIMULATION TO THE TASK OF PERSONAL DEPERSONALIZATION DATA. MODEL AND ALGORITHM FOR DECONTAMINATION BY SYNTHESIS //Programmirovanie. – 2023. – №. 5. – С. 19-34.
18. Wang K., Fung B. C. M., Dong G. Integrating private databases for data analysis //International conference on intelligence and security informatics. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2005. – С. 171-182.
19. Jiang W., Clifton C. Privacy-preserving distributed k-anonymity //Data and Applications Security XIX: 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Storrs, CT, USA, August 7-10, 2005. Proceedings 19. – Springer Berlin Heidelberg, 2005. – С. 166-177.
20. Soroush M. B. Personal Loan Dataset [Электронный ресурс] // Kaggle. – 2024. – URL: https://www.kaggle.com/datasets/mbsoroush/personal-loan-dataset/data (дата обращения: 11.05.2025).

# ПРИЛОЖЕНИЕ 1 КАЛЕНДАРНЫЙ ПЛАН РАБОТ

Таблица 1 – Календарный план работ

|  |  |  |  |
| --- | --- | --- | --- |
| Этап проекта | Описание работ | Ожидаемые результаты | Сроки выполнения |
| 1. Анализ методов обезличивания | Рассмотреть описанные исследователями алгоритмы обезличивания и особенности их реализации.  Выбрать алгоритмы для последующей реализации. | Наличие перечня алгоритмов для реализации и знание особенностей их работы. | 06.11.2024 – 25.01.2025 |
| 2. Анализ подходов к оценке качества обезличивания | Рассмотреть описанные исследователями алгоритмы оценки качества обезличивания и особенности их реализации.  Выбрать алгоритмы для последующей реализации. | Наличие перечня алгоритмов для реализации и знание особенностей их работы. | 06.11.2024 – 25.01.2025 |
| 3. Подготовка данных | Поиск готовых данных и (или) подготовка своего датасета. | Наличие нескольких готовых датасетов, содержащих табличные персональные данные различных типов. | 03.01.2024 – 20.01.2024 |
| 4. Реализация методов оценки и обезличивания | Реализация методов, выбранных на этапах 1 и 2 на языке программирования Python. | Реализованы описанные алгоритмы. | 25.12.2024 – 10.03.2025 |
| 5. Проведение экспериментов | Проведение экспериментов с реализованными алгоритмами и подготовленными данными. | Получены результаты экспериментов, позволяющие сравнить алгоритмы между собой. | 10.01.2025 – 25.03.2025 |
| 6. Анализ результатов и написание отчёта | Проведение сравнения качества обезличивания при помощи реализованных алгоритмов и написание отчёта. | Подготовлен отчёт об исследовательском проекте. | 30.01.2025 – 15.04.2025 |

# ПРИЛОЖЕНИЕ 2 ФУНКЦИИ ПАКЕТА CP2025, РЕЗЛИЗУЮЩИЕ МЕТОДЫ ОЦЕНКИ КАЧЕСТВА ОБЕЗЛИЧИВАНИЯ

Таблица 2 – Описание функций оценки качества пакета cp2025

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Метод** | **Модуль** | **Название функции** | **Входные данные** | **Выходные данные** |
| k-анонимность | cp2025. utility. metrics | find\_k\_ anonimus | * Numpy array обезличенных квази-идентификаторов | Максимальное k, при котором датасет k-анонимен |
| l-разнообразие | cp2025. utility. metrics | find\_l\_ diverse | * Numpy array обезличенных квази-идентификаторов * Numpy array чувствительных значений | Максимальное l, при котором датасет l-разнообразен |
| t-близость | cp2025. utility. metrics | find\_t\_ close | * Numpy array обезличенных квази-идентификаторов * Numpy array чувствительных значений * Список типов столбцов с чувствительными значениями * Булевое значение, указывающее, всегда ли использовать энтропию для расчёта расстояния между распределениями | Минимальное t, при котором датасет соответствует t-близости |
| Adversarial knowledge gain | cp2025. utility. metrics | adversarial\_ knowledge\_ gain | * Numpy array обезличенных квази-идентификаторов * Numpy array чувствительных значений | Значение adversarial knowledge gain |
| Adversarial accuracy gain | cp2025. utility. metrics | adversarial\_ accuracy\_ gain | * Numpy array обезличенных квази-идентификаторов * Numpy array чувствительных значений | Значение adversarial accuracy gain |
| Оценка рисков с помощью градиентного бустинга | cp2025. utility. boosting\_ security\_ score | get\_ boosting\_ security\_ score | * Numpy array квази-идентификаторов в обучающей выборке * Numpy array обезличенных квази-идентификаторов в обучающей выборке * Numpy array квази-идентификаторов в тестовой выборке * Numpy array обезличенных квази-идентификаторов в тестовой выборке | Значение метрики в соответствии с предложенным в отчёте подходом |
| Средний размер класса эквивалентности | cp2025. utility. metrics | average\_ equivalence\_ class\_size | * Numpy array обезличенных квази-идентификаторов | Средний размер класса эквивалентности |
| Отношение количества уникальных записей к общему числу записей | cp2025. utility. metrics | distinctness | * Numpy array обезличенных квази-идентификаторов | Отношение количества уникальных записей к общему числу записей |
| Доля изменившихся отдельных значений в датасете | cp2025. utility. metrics | changed\_ proportion | * Numpy array квази-идентификаторов * Numpy array обезличенных квази-идентификаторов | Доля изменившихся отдельных значений в датасете |
| Неоднородная энтропия (non-uniform entropy) | cp2025. utility. metrics | non\_ uniform\_ entropy | * Numpy array обезличенных квази-идентификаторов | Значение non-uniform entropy |
| Потеря информации на основе энтропии (entropy-based information loss) | cp2025. utility. metrics | entropy\_ based\_ information\_ loss | * Numpy array квази-идентификаторов * Numpy array обезличенных квази-идентификаторов | Потеря информации на основе энтропии |
| Оценка расстояния между исходными и получившимися данными поэлементно | cp2025. utility. metrics | my\_by\_ element\_ distance\_ columns\_ unordered | * Numpy array квази-идентификаторов * Numpy array обезличенных квази-идентификаторов | Метрика в соответствии с отчётом |

# ПРИЛОЖЕНИЕ 3 КЛАССЫ ПАКЕТА CP2025, РЕЗЛИЗУЮЩИЕ АЛГОРИТМЫ ОБЕЗЛИЧИВАНИЯ

Таблица 3 – Описание реализаций алгоритмов обезличивания пакета cp2025

|  |  |  |  |
| --- | --- | --- | --- |
| **Модуль, cp2025. algorithms.** | **Класс** | **Аргументы конструктора** | **Описание** |
| SuppressionKAnonymityBaseline | SuppressionKAnonymityBaseline | * k – параметр k-анонимности | Алгоритм перебора для достижения k-анонимности |
| SuppressionLDiversityBaseline | SuppressionLDiversityBaseline | * k – параметр k-анонимности * l – параметр l-разнообразия | Алгоритм перебора для достижения l-разнообразия |
| SuppressionTClosenessBaseline | SuppressionTClosenessBaseline | * k – параметр k-анонимности * t – параметр t-близости * Список типов чувствительных значений | Алгоритм перебора для достижения t-близости |
| SuppressionKAnonymityTimeOptimal | SuppressionKAnonymityTimeOptimal | * k – параметр k-анонимности | Алгоритм жадного выбора на основе расстояний для k-анонимности и подавления |
| SuppressionLDiversityTimeOptimal | SuppressionLDiversityTimeOptimal | * k – параметр k-анонимности * l – параметр l-разнообразия | Алгоритм жадного выбора на основе расстояний для l-разнообразия и подавления |
| GeneralizationKAnonymityTimeOptimal | GeneralizationKAnonymityTimeOptimal | * k – параметр k-анонимности * Список типов квази-идентификаторов | Алгоритм жадного выбора на основе расстояний для k-анонимности и обобщения |
| GeneralizationLDiversityTimeOptimal | GeneralizationLDiversityTimeOptimal | * k – параметр k-анонимности * l – параметр l-разнообразия * Список типов квази-идентификаторов | Алгоритм жадного выбора на основе расстояний для l-разнообразия и обобщения |
| AggregationKAnonymityTimeOptimal | AggregationKAnonymityTimeOptimal | * k – параметр k-анонимности * Список типов квази-идентификаторов | Алгоритм жадного выбора на основе расстояний для k-анонимности и агрегации |
| AggregationLDiversityTimeOptimal | AggregationLDiversityTimeOptimal | * k – параметр k-анонимности * l – параметр l-разнообразия * Список типов квази-идентификаторов | Алгоритм жадного выбора на основе расстояний для l-разнообразия и агрегации |
| Datafly | Datafly | * k – параметр k-анонимности * Список типов квази-идентификаторов * Максимальное число подавляемых строк | Модифицированный Datafly |
| GeneralizationGreedyByOneEqualSizedGroups | GeneralizationGreedyByOneEqualSizedGroups | * k – параметр k-анонимности * Список типов квази-идентификаторов | Алгоритм разбиения столбцов на равные группы с обобщением |
| groupjoin | GroupJoinDepersonalizator | * Класс, содержащий сведения о методе обезличивания (GroupJoinSuppression, GroupJoinGeneralization, или GroupJoinAggregation) * Класс, содержащий сведения о метрике (GroupJoinKAnonymity, GroupJoinLDiversity или GroupJoinTCloseness) | Алгоритм на основе последовательного объединения групп (group join) |
| groupjoin | GroupJoinSuppression | нет | Класс, содержащий сведения о методе обезличивания подавление для алгоритма обезличивания group join |
| groupjoin | GroupJoinGeneralization | * Список типов квази-идентификаторов | Класс, содержащий сведения о методе обезличивания обобщение для алгоритма обезличивания group join |
| groupjoin | GroupJoinAggregation | * Список типов квази-идентификаторов | Класс, содержащий сведения о методе обезличивания агрегация для алгоритма обезличивания group join |
| groupjoin | GroupJoinKAnonymity | * k – параметр k-анонимности * Функция, принимающая на вход my\_by\_element\_loss рассматриваемый в данном алгоритме, размер одной группы, размер второй группы и k, и возвращающая потери при объединении (по умолчания в соответствии с описанием алгоритма в главе 3 возвращает my\_by\_element\_loss) * Сид генератора случайных чисел | Класс, содержащий сведения о метрике k-анонимность, для алгоритма обезличивания group join |
| groupjoin | GroupJoinLDiversity | * k – параметр k-анонимности * l – параметр l-разнообразия * Функция, принимающая на вход my\_by\_element\_loss рассматриваемый в данном алгоритме, размер одной группы, размер второй группы, k, l и массив минимумов из количеств уникальных чувствительных значений в каждом столбце в объединённой группе и l, и возвращающая потери при объединении (по умолчания в соответствии с описанием алгоритма в главе 3 возвращает my\_by\_element\_loss делённый на сумму элементов массива) * Сид генератора случайных чисел | Класс, содержащий сведения о метрике l-разнообразие, для алгоритма обезличивания group join |
| groupjoin | GroupJoinTCloseness | * k – параметр k-анонимности * t – параметр t-близости * Список типов чувствительных значений * Функция, принимающая на вход my\_by\_element\_loss рассматриваемый в данном алгоритме, размер одной группы, размер второй группы, k, t и массив расстояний между распределением чувствительных значений в каждом столбце в объединённой группе и во всём датасете, и возвращающая потери при объединении (по умолчания в соответствии с описанием алгоритма в главе 3 возвращает my\_by\_element\_loss умноженный на сумму элементов массива) * Сид генератора случайных чисел | Класс, содержащий сведения о метрике t-близость, для алгоритма обезличивания group join |
| RandomizationDepersonalizator.py | RandomizationDepersonalizator.py | * Сид генератора случайных чисел * Список типов квази-идентификаторов * scale | Алгоритм рандомизации |
| IdentifierHasher | IdentifierHasher | нет | Алгоритм хеширования |
| Shuffler | Shuffler | * Список столбцов для перемешивания * Сид генератора случайных чисел | Алгоритмы перемешивания всего датасета |
| ShufflerInBatches | ShufflerInBatches | * Список столбцов для перемешивания * Сид генератора случайных чисел | Алгоритмы перемешивания строк внутри классов эквивалентности |
| DistributedDataKAnonymityDepersonalization.py | DistributedDataOwnerKAnonymityDepersonalizator | * Numpy array квази-идентификаторов * Numpy array чувствительных значений * k – параметр k-анонимности * Секретный ключ шифрования * Список типов квази-идентификаторов * Сид генератора случайных чисел | Алгоритм DPP2GA (объединение распределённого горизонтально датасета).  Не имеет метода depersonalize.  Необходимо создать 2 экземпляра данного класса, олицетворяющих владельцев информации, сообщить им друг о друге при помощи метода set\_other\_data\_owner(other\_data\_owner) и вызвать у одного из них метод exchange\_data() для проведения объединения датасетов. Объединённый датасет будет находиться в поле joined\_data каждого объекта-владельца. |

# ПРИЛОЖЕНИЕ 4 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ НА СРЕНЕРИРОВАННОМ ДАТАСЕТЕ

Таблица 4 - сгенерированный датасет, полезность (k=2, l=2, t=1, scale = 0,05)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance | Boosting score |
| Time optimal, k, подавление | 0,5 | 0,53 | 0,1 | 0,53 | 1,21 |
| Time optimal, k, обобщение | 0,5 | 0,56 | 0,1 | 0,17 | 1,4 |
| Time optimal, k, агрегация | 0,5 | 0,37 | 0,1 | 0,17 | 2,29 |
| Time optimal, l, подавление | 0,43 | 0,59 | 0,13 | 0,59 | 2,14 |
| Time optimal, l, обобщение | 0,42 | 0,63 | 0,13 | 0,24 | 2,12 |
| Time optimal, l, агрегация | 0,42 | 0,42 | 0,13 | 0,2 | 1,89 |
| Group join, k, подавление | 0,17 | 0,69 | 0,31 | 0,69 | 2,1 |
| Group join, k, обобщение | 0,47 | 0,57 | 0,11 | 0,19 | 1,5 |
| Group join, k, агрегация | 0,32 | 0,45 | 0,17 | 0,22 | 2,5 |
| Group join, l, подавление | 0,16 | 0,69 | 0,3 | 0,69 | 2,56 |
| Group join, l, обобщение | 0,46 | 0,58 | 0,11 | 0,2 | 2,11 |
| Group join, l, агрегация | 0,3 | 0,46 | 0,18 | 0,23 | 1,8 |
| Group join, t, подавление | 0,002 | 1 | 0,03 | 1 | 1,85 |
| Group join, t, обобщение | 0,012 | 1 | 0,7 | 0,74 | 1,73 |
| Group join, t, агрегация | 0,019 | 0,83 | 0,69 | 0,58 | 2,14 |
| Datafly | 0,04 | 0,96 | 0,51 | 0,58 | 1,89 |
| Equal sized | 0,03 | 0,93 | 0,5 | 0,59 | 2 |
| Рандомизация | - | - | - | 0,28 | 1,56 |

Таблица 5 - сгенерированный датасет, риски обезличивания (k=2, l=2, t=1, scale = 0,05)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Boosting score | Время, с |
| Time optimal, k, подавление | 2 | 1 | 0,48 | 1 | 0,5 | - | 1,91 |
| Time optimal, k, обобщение | 2 | 1 | 0,49 | 1 | 0,5 | - | 1,87 |
| Time optimal, k, агрегация | 2 | 1 | 0,49 | 1 | 0,5 | - | 1,82 |
| Time optimal, l, подавление | 2 | 2 | 0,49 | 1 | 0,43 | - | 1,86 |
| Time optimal, l, обобщение | 2 | 2 | 0,46 | 1 | 0,42 | - | 1,88 |
| Time optimal, l, агрегация | 2 | 2 | 0,46 | 1 | 0,42 | - | 1,88 |
| Group join, k, подавление | 2 | 1 | 0,44 | 0,99 | 0,19 | - | 5,17 |
| Group join, k, обобщение | 2 | 1 | 0,48 | 1 | 0,47 | - | 10,9 |
| Group join, k, агрегация | 2 | 1 | 0,42 | 1 | 0,32 | - | 35,4 |
| Group join, l, подавление | 2 | 2 | 0,46 | 0,99 | 0,19 | - | 7,14 |
| Group join, l, обобщение | 2 | 2 | 0,48 | 1 | 0,46 | - | 12,8 |
| Group join, l, агрегация | 2 | 2 | 0,43 | 1 | 0,3 | - | 18,4 |
| Group join, t, подавление | 4 | 4 | 0,08 | 0,008 | 0,001 | - | 265,6 |
| Group join, t, обобщение | 2 | 2 | 0,21 | 0,85 | 0,011 | - | 531 |
| Group join, t, агрегация | 2 | 2 | 0,22 | 0,85 | 0,017 | - | 594 |
| Datafly | 2 | 2 | 0,27 | 0,96 | 0,08 | - | 13,4 |
| Equal sized | 23 | 4 | 0,13 | 0,97 | 0,083 | - | 58,8 |
| Рандомизация | - | - | - | - | - | 1 | 0,025 |

Таблица 6 - сгенерированный датасет, полезность (k=10, l=3, t=0.5, scale = 0,25)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance | Boosting score |
| Time optimal, k, подавление | 0,022 | 0,95 | 0,73 | 0,95 | 2,86 |
| Time optimal, k, обобщение | 0,1 | 0,96 | 0,33 | 0,63 | 1,87 |
| Time optimal, k, агрегация | 0,1 | 0,63 | 0,33 | 0,36 | 1,9 |
| Time optimal, l, подавление | 0,022 | 0,95 | 0,74 | 0,95 | 2,86 |
| Time optimal, l, обобщение | 0,1 | 0,96 | 0,33 | 0,63 | 1,97 |
| Time optimal, l, агрегация | 0,1 | 0,63 | 0,33 | 0,63 | 1,89 |
| Group join, k, подавление | 0,06 | 0,78 | 0,42 | 0,78 | 1,57 |
| Group join, k, обобщение | 0,07 | 0,88 | 0,4 | 0,72 | 2,28 |
| Group join, k, агрегация | 0,074 | 0,64 | 0,38 | 0,38 | 1,67 |
| Group join, l, подавление | 0,004 | 0,99 | 0,96 | 0,98 | 1,85 |
| Group join, l, обобщение | 0,66 | 0,89 | 0,4 | 0,62 | 2,54 |
| Group join, l, агрегация | 0,63 | 0,65 | 0,4 | 0,38 | 2,1 |
| Group join, t, подавление | 0,001 | 1 | 1 | 1 | 1,85 |
| Group join, t, обобщение | 0,01 | 1 | 0,7 | 0,75 | 1,99 |
| Group join, t, агрегация | 0,01 | 0,83 | 0,73 | 0,58 | 2,13 |
| Datafly | 0,016 | 0,97 | 0,61 | 0,6 | 2 |
| Equal sized | 0,032 | 0,93 | 0,5 | 0,58 | 2 |
| Рандомизация | - | - | - | 0,31 | 1,8 |

Таблица 7 - сгенерированный датасет, риски обезличивания (k=10, l=3, t=0.5, scale = 0,25)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Boosting score | Время, с |
| Time optimal, k, подавление | 10 | 3 | 0,18 | 0,63 | 0,05 | - | 1,67 |
| Time optimal, k, обобщение | 10 | 3 | 0,23 | 0,99 | 0,17 | - | 1,66 |
| Time optimal, k, агрегация | 10 | 3 | 0,24 | 1 | 0,17 | - | 1,68 |
| Time optimal, l, подавление | 10 | 3 | 0,18 | 0,64 | 0,05 | - | 1,66 |
| Time optimal, l, обобщение | 10 | 3 | 0,24 | 0,99 | 0,17 | - | 1,65 |
| Time optimal, l, агрегация | 10 | 3 | 0,24 | 1 | 0,17 | - | 1,69 |
| Group join, k, подавление | 10 | 3 | 0,26 | 0,98 | 0,13 | - | 5,13 |
| Group join, k, обобщение | 10 | 3 | 0,17 | 0,98 | 0,14 | - | 14,9 |
| Group join, k, агрегация | 10 | 2 | 0,26 | 0,98 | 0,12 | - | 118 |
| Group join, l, подавление | 10 | 4 | 0,11 | 0,1 | 0,003 | - | 7,2 |
| Group join, l, обобщение | 10 | 3 | 0,21 | 0,98 | 0,11 | - | 16,9 |
| Group join, l, агрегация | 10 | 4 | 0,22 | 0,98 | 0,01 | - | 41,4 |
| Group join, t, подавление | 1000 | 4 | 0 | 0 | 0 | - | 261 |
| Group join, t, обобщение | 31 | 4 | 0,016 | 0,86 | 0,009 | - | 279 |
| Group join, t, агрегация | 12 | 4 | 0,04 | 0,8 | 0,008 | - | 312 |
| Datafly | 22 | 4 | 0,11 | 0,93 | 0,05 | - | 13,3 |
| Equal sized | 23 | 4 | 0,14 | 0,97 | 0,083 | - | 58,4 |
| Рандомизация | - | - | - | - | - | 0,995 | 0,018 |

Таблица 8 - сгенерированный датасет, полезность (k=100, l=4, t=0.1, scale = 1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance | Boosting score |
| Time optimal, k, подавление | 0,001 | 1 | 1 | 1 | 1,85 |
| Time optimal, k, обобщение | 0,01 | 1 | 0,67 | 0,82 | 1,72 |
| Time optimal, k, агрегация | 0,01 | 0,76 | 0,67 | 0,49 | 1,93 |
| Time optimal, l, подавление | 0,001 | 1 | 1 | 1 | 1,86 |
| Time optimal, l, обобщение | 0,01 | 1 | 0,67 | 0,82 | 1,72 |
| Time optimal, l, агрегация | 0,01 | 0,76 | 0,67 | 0,49 | 1,93 |
| Group join, k, подавление | 0,005 | 0,94 | 0,79 | 0,94 | 1,86 |
| Group join, k, обобщение | 0,006 | 0,97 | 0,75 | 0,9 | 1,9 |
| Group join, k, агрегация | 0,009 | 0,79 | 0,74 | 0,53 | 2,1 |
| Group join, l, подавление | 0,001 | 1 | 1 | 1 | 1,85 |
| Group join, l, обобщение | 0,007 | 0,96 | 0,72 | 0,92 | 1,95 |
| Group join, l, агрегация | 0,006 | 0,79 | 0,75 | 0,52 | 1,98 |
| Group join, t, подавление | 0,001 | 1 | 1 | 1 | 1,85 |
| Group join, t, обобщение | 0,005 | 1 | 0,77 | 0,71 | 2 |
| Group join, t, агрегация | 0,003 | 0,85 | 0,58 | 0,59 | 2,04 |
| Datafly | 0,004 | 1 | 0,8 | 0,63 | 1,87 |
| Equal sized | 0,008 | 0,93 | 0,7 | 0,62 | 1,93 |
| Рандомизация | - | - | - | 0,39 | 1,6 |

Таблица 9 - сгенерированный датасет, риски обезличивания (k=100, l=4, t=0.1, scale = 1)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Boosting score | Время, с |
| Time optimal, k, подавление | 1000 | 4 | 0 | 0 | 0 | - | 1,63 |
| Time optimal, k, обобщение | 100 | 4 | 0,04 | 0,9 | 0,035 | - | 1,67 |
| Time optimal, k, агрегация | 100 | 4 | 0,04 | 0,9 | 0,035 | - | 1,67 |
| Time optimal, l, подавление | 1000 | 4 | 0 | 0 | 0 | - | 1,64 |
| Time optimal, l, обобщение | 100 | 4 | 0,04 | 0,9 | 0,035 | - | 1,64 |
| Time optimal, l, агрегация | 100 | 4 | 0,04 | 0,9 | 0,035 | - | 1,63 |
| Group join, k, подавление | 122 | 4 | 0,02 | 0,73 | 0,042 | - | 5,28 |
| Group join, k, обобщение | 102 | 4 | 0,049 | 0,8 | 0,027 | - | 15,1 |
| Group join, k, агрегация | 115 | 4 | 0,046 | 0,82 | 0,04 | - | 99,2 |
| Group join, l, подавление | 1000 | 4 | 0 | 0 | 0 | - | 7,23 |
| Group join, l, обобщение | 109 | 4 | 0,04 | 0,85 | 0,029 | - | 17,2 |
| Group join, l, агрегация | 107 | 4 | 0,028 | 0,8 | 0,02 | - | 42,3 |
| Group join, t, подавление | 1000 | 4 | 0 | 0 | 0 | - | 262 |
| Group join, t, обобщение | 141 | 4 | 0,006 | 0,78 | 0,004 | - | 555 |
| Group join, t, агрегация | 261 | 4 | 0,003 | 0,66 | 0,002 | - | 310 |
| Datafly | 196 | 4 | 0,03 | 0,74 | 0,016 | - | 13,3 |
| Equal sized | 114 | 4 | 0,03 | 0,87 | 0,032 | - | 48,6 |
| Рандомизация | - | - | - | - | - | 1 | 0,017 |

# ПРИЛОЖЕНИЕ 5 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ НА ДАТАСЕТЕ С KAGGLE

Таблица 10 – датасет c Kaggle, полезность (k=2, l=2, t=1, scale = 0,05)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance | Boosting score |
| Time optimal, k, подавление | 0,44 | 0,36 | 0,13 | 0,36 | 0,295 |
| Time optimal, k, обобщение | 0,5 | 0,5 | 0,01 | 0,016 | 0,295 |
| Time optimal, k, агрегация | 0,5 | 0,4 | 0,1 | 0,017 | 0,245 |
| Time optimal, l, подавление | 0,17 | 0,66 | 0,41 | 0,66 | 0,255 |
| Time optimal, l, обобщение | 0,27 | 0,62 | 0,21 | 0,087 | 0,275 |
| Time optimal, l, агрегация | 0,27 | 0,55 | 0,21 | 0,05 | 0,235 |
| Group join, k, подавление | 0,34 | 0,36 | 0,19 | 0,36 | 0,245 |
| Group join, k, обобщение | 0,41 | 0,51 | 0,13 | 0,06 | 0,19 |
| Group join, k, агрегация | 0,39 | 0,45 | 0,14 | 0,04 | 0,28 |
| Group join, l, подавление | 0,2 | 0,47 | 0,29 | 0,48 | 0,245 |
| Group join, l, обобщение | 0,23 | 0,59 | 0,11 | 0,2 | 0,275 |
| Group join, l, агрегация | 0,23 | 0,51 | 0,22 | 0,05 | 0,305 |
| Group join, t, подавление | 0,058 | 0,86 | 0,77 | 0,86 | 0,245 |
| Group join, t, обобщение | 0,36 | 0,54 | 0,16 | 0,078 | 0,235 |
| Group join, t, агрегация | 0,33 | 0,45 | 0,17 | 0,05 | 0,265 |
| Datafly | 0,016 | 0,85 | 0,63 | 0,2 | 0,245 |
| Equal sized | 0,012 | 1 | 0,69 | 0,22 | 0,245 |
| Рандомизация | - | - | - | 0,05 | 0,505 |

Таблица 11 - датасет c Kaggle, риски обезличивания (k=2, l=2, t=1, scale = 0,05)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Boosting score | Время, с |
| Time optimal, k, подавление | 2 | 1 | 0,61 | 0,97 | 0,47 | - | 1,69 |
| Time optimal, k, обобщение | 2 | 1 | 0,58 | 0,98 | 0,5 | - | 1,75 |
| Time optimal, k, агрегация | 2 | 1 | 0,58 | 0,98 | 0,5 | - | 1,74 |
| Time optimal, l, подавление | 2 | 2 | 0,6 | 0,73 | 0,2 | - | 1,66 |
| Time optimal, l, обобщение | 2 | 2 | 0,58 | 0,95 | 0,32 | - | 1,65 |
| Time optimal, l, агрегация | 2 | 2 | 0,58 | 0,95 | 0,32 | - | 1,65 |
| Group join, k, подавление | 2 | 1 | 0,63 | 0,94 | 0,42 | - | 2,66 |
| Group join, k, обобщение | 2 | 1 | 0,58 | 0,97 | 0,44 | - | 9,8 |
| Group join, k, агрегация | 2 | 1 | 0,58 | 0,97 | 0,44 | - | 43,7 |
| Group join, l, подавление | 2 | 2 | 0,57 | 0,88 | 0,26 | - | 5,44 |
| Group join, l, обобщение | 2 | 2 | 0,58 | 0,95 | 0,32 | - | 14,1 |
| Group join, l, агрегация | 2 | 2 | 0,58 | 0,95 | 0,33 | - | 18,4 |
| Group join, t, подавление | 2 | 1 | 0,36 | 0,28 | 0,07 | - | 153 |
| Group join, t, обобщение | 2 | 1 | 0,67 | 0,97 | 0,4 | - | 267 |
| Group join, t, агрегация | 2 | 1 | 0,68 | 0,96 | 0,38 | - | 151 |
| Datafly | 19 | 5 | 0,12 | 0,53 | 0,04 | - | 2,26 |
| Equal sized | 5 | 2 | 0,16 | 0,43 | 0,022 | - | 8,68 |
| Рандомизация | - | - | - | - | - | 0,65 | 0,012 |

Таблица 12 - датасет c Kaggle, полезность (k=10, l=3, t=0.5, scale = 0,25)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance | Boosting score |
| Time optimal, k, подавление | 0,017 | 0,85 | 0,7 | 0,84 | 0,245 |
| Time optimal, k, обобщение | 0,1 | 0,72 | 0,33 | 0,19 | 0,245 |
| Time optimal, k, агрегация | 0,1 | 0,69 | 0,33 | 0,06 | 0,275 |
| Time optimal, l, подавление | 0,017 | 0,86 | 0,7 | 0,86 | 0,245 |
| Time optimal, l, обобщение | 0,095 | 0,73 | 0,34 | 0,21 | 0,19 |
| Time optimal, l, агрегация | 0,095 | 0,69 | 0,34 | 0,064 | 0,25 |
| Group join, k, подавление | 0,03 | 0,69 | 0,53 | 0,69 | 0,255 |
| Group join, k, обобщение | 0,069 | 0,64 | 0,39 | 0,36 | 0,07 |
| Group join, k, агрегация | 0,061 | 0,61 | 0,41 | 0,078 | 0,24 |
| Group join, l, подавление | 0,023 | 0,68 | 0,58 | 0,67 | 0,245 |
| Group join, l, обобщение | 0,063 | 0,64 | 0,4 | 0,29 | 0,24 |
| Group join, l, агрегация | 0,066 | 0,64 | 0,4 | 0,08 | 0,245 |
| Group join, t, подавление | 0,001 | 1 | 1 | 1 | 0,245 |
| Group join, t, обобщение | 0,059 | 0,69 | 0,42 | 0,37 | 0,235 |
| Group join, t, агрегация | 0,056 | 0,73 | 0,45 | 0,11 | 0,245 |
| Datafly | 0,017 | 0,86 | 0,62 | 0,2 | 0,245 |
| Equal sized | 0,002 | 1 | 0,9 | 0,33 | 0,245 |
| Рандомизация | - | - | - | 0,13 | 0,375 |

Таблица 13 - датасет c Kaggle, риски обезличивания (k=10, l=3, t=0.5, scale = 0,25)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Boosting score | Время, с |
| Time optimal, k, подавление | 10 | 3 | 0,17 | 0,42 | 0,033 | - | 1,59 |
| Time optimal, k, обобщение | 10 | 1 | 0,26 | 0,9 | 0,17 | - | 1,63 |
| Time optimal, k, агрегация | 10 | 1 | 0,26 | 0,9 | 0,17 | - | 1,59 |
| Time optimal, l, подавление | 10 | 3 | 0,28 | 0,41 | 0,034 | - | 1,62 |
| Time optimal, l, обобщение | 10 | 3 | 0,26 | 0,9 | 0,17 | - | 1,6 |
| Time optimal, l, агрегация | 10 | 3 | 0,26 | 0,9 | 0,17 | - | 1,62 |
| Group join, k, подавление | 10 | 3 | 0,24 | 0,67 | 0,066 | - | 2,15 |
| Group join, k, обобщение | 10 | 2 | 0,23 | 0,86 | 0,15 | - | 12,7 |
| Group join, k, агрегация | 10 | 3 | 0,28 | 0,84 | 0,15 | - | 52,8 |
| Group join, l, подавление | 10 | 3 | 0,15 | 0,58 | 0,062 | - | 5,7 |
| Group join, l, обобщение | 10 | 3 | 0,25 | 0,84 | 0,15 | - | 15,3 |
| Group join, l, агрегация | 10 | 3 | 0,31 | 0,85 | 0,17 | - | 19,5 |
| Group join, t, подавление | 1000 | 90 | 0 | 0 | 0 | - | 154 |
| Group join, t, обобщение | 10 | 2 | 0,33 | 0,81 | 0,13 | - | 165 |
| Group join, t, агрегация | 10 | 1 | 0,27 | 0,76 | 0,12 | - | 172 |
| Datafly | 19 | 5 | 0,12 | 0,53 | 0,04 | - | 2,27 |
| Equal sized | 495 | 77 | 0,006 | 0,15 | 0,006 | - | 10 |
| Рандомизация | - | - | - | - | - | 0,95 | 0,011 |

Таблица 14 - датасет c Kaggle, полезность (k=100, l=4, t=0.1, scale = 1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance | Boosting score |
| Time optimal, k, подавление | 0,001 | 1 | 1 | 1 | 0,255 |
| Time optimal, k, обобщение | 0,01 | 1 | 0,67 | 0,75 | 0,255 |
| Time optimal, k, агрегация | 0,01 | 1 | 0,67 | 0,75 | 0,25 |
| Time optimal, l, подавление | 0,001 | 1 | 1 | 1 | 0,245 |
| Time optimal, l, обобщение | 0,01 | 1 | 0,67 | 0,75 | 0,255 |
| Time optimal, l, агрегация | 0,01 | 1 | 0,67 | 0,75 | 0,25 |
| Group join, k, подавление | 0,006 | 0,84 | 0,75 | 0,84 | 0,26 |
| Group join, k, обобщение | 0,006 | 0,94 | 0,74 | 0,69 | 0,255 |
| Group join, k, агрегация | 0,006 | 0,93 | 0,75 | 0,18 | 0,25 |
| Group join, l, подавление | 0,006 | 0,82 | 0,75 | 0,82 | 0,26 |
| Group join, l, обобщение | 0,006 | 0,93 | 0,74 | 0,33 | 0,25 |
| Group join, l, агрегация | 0,007 | 0,92 | 0,72 | 0,17 | 0,245 |
| Group join, t, подавление | 0,001 | 1 | 1 | 1 | 0,245 |
| Group join, t, обобщение | 0,003 | 1 | 0,89 | 0,62 | 0,255 |
| Group join, t, агрегация | 0,003 | 1 | 0,85 | 0,26 | 0,245 |
| Datafly | 0,004 | 1 | 0,8 | 0,26 | 0,245 |
| Equal sized | 0,002 | 1 | 0,9 | 0,33 | 0,245 |
| Рандомизация | - | - | - | 0,3 | 0,39 |

Таблица 15 - датасет c Kaggle, риски обезличивания (k=100, l=4, t=0.1, scale = 1)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Boosting score | Время, с |
| Time optimal, k, подавление | 1000 | 90 | 0 | 0 | 0 | - | 1,55 |
| Time optimal, k, обобщение | 100 | 24 | 0,12 | 0,46 | 0,026 | - | 1,65 |
| Time optimal, k, агрегация | 100 | 24 | 0,12 | 0,46 | 0,026 | - | 1,55 |
| Time optimal, l, подавление | 1000 | 4 | 0 | 0 | 0 | - | 1,56 |
| Time optimal, l, обобщение | 100 | 24 | 0,12 | 0,46 | 0,026 | - | 1,59 |
| Time optimal, l, агрегация | 100 | 24 | 0,12 | 0,46 | 0,026 | - | 1,55 |
| Group join, k, подавление | 108 | 28 | 0,06 | 0,33 | 0,017 | - | 3,19 |
| Group join, k, обобщение | 107 | 10 | 0,14 | 0,34 | 0,018 | - | 13 |
| Group join, k, агрегация | 105 | 28 | 0,12 | 0,34 | 0,016 | - | 17,1 |
| Group join, l, подавление | 102 | 22 | 0,08 | 0,35 | 0,015 | - | 5,78 |
| Group join, l, обобщение | 111 | 32 | 0,13 | 0,35 | 0,027 | - | 15,7 |
| Group join, l, агрегация | 115 | 35 | 0,1 | 0,4 | 0,019 | - | 19,7 |
| Group join, t, подавление | 1000 | 90 | 0 | 0 | 0 | - | 154 |
| Group join, t, обобщение | 106 | 33 | 0,03 | 0,16 | 0,005 | - | 309 |
| Group join, t, агрегация | 188 | 54 | 0,01 | 0,19 | 0,006 | - | 172 |
| Datafly | 128 | 30 | 0,09 | 0,28 | 0,014 | - | 2,27 |
| Equal sized | 495 | 77 | 0,006 | 0,15 | 0,006 | - | 8,24 |
| Рандомизация | - | - | - | - | - | 0,975 | 0,01 |

# ПРИЛОЖЕНИЕ 6 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ НА КОРОТКОМ ДАТАСЕТЕ

Таблица 16 – короткий датасет, полезность (k=2, l=2, t=1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance |
| Time optimal, k, подавление | 0,17 | 0,92 | 0,82 | 0,92 |
| Time optimal, k, обобщение | 0,5 | 0,92 | 0,28 | 0,09 |
| Time optimal, k, агрегация | 0,5 | 0,92 | 0,28 | 0,09 |
| Time optimal, l, подавление | 0,17 | 0,92 | 0,81 | 0,92 |
| Time optimal, l, обобщение | 0,5 | 0,92 | 0,28 | 0,92 |
| Time optimal, l, агрегация | 0,5 | 0,92 | 0,28 | 0,09 |
| Group join, k, подавление | 0,17 | 0,92 | 0,82 | 0,92 |
| Group join, k, обобщение | 0,33 | 1 | 0,45 | 0,13 |
| Group join, k, агрегация | 0,33 | 1 | 0,45 | 0,06 |
| Group join, l, подавление | 0,17 | 0,92 | 0,82 | 0,92 |
| Group join, l, обобщение | 0,33 | 1 | 0,45 | 0,13 |
| Group join, l, агрегация | 0,33 | 1 | 0,45 | 0,062 |
| Group join, t, подавление | 0,083 | 1 | 1 | 1 |
| Group join, t, обобщение | 0,33 | 0,92 | 0,46 | 0,14 |
| Group join, t, агрегация | 0,33 | 0,92 | 0,46 | 0,065 |
| Datafly | 0,17 | 1 | 0,72 | 0,25 |
| Equal sized | 0,33 | 1 | 0,44 | 0,08 |
| Перебор, k, подавление | 0,17 | 0,92 | 0,81 | 0,92 |
| Перебор, l, подавление | 0,17 | 0,92 | 0,81 | 0,92 |
| Перебор, t, подавление | 0,17 | 0,92 | 0,81 | 0,92 |

Таблица 17 - короткий датасет, риски обезличивания (k=2, l=2, t=1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Время, с |
| Time optimal, k, подавление | 2 | 2 | 0,29 | 0,28 | 0,08 | < 1е-6 |
| Time optimal, k, обобщение | 2 | 2 | 0,41 | 0,83 | 0,42 | 0,000998 |
| Time optimal, k, агрегация | 2 | 2 | 0,41 | 0,83 | 0,42 | < 1е-6 |
| Time optimal, l, подавление | 2 | 2 | 0,29 | 0,27 | 0,083 | < 1е-6 |
| Time optimal, l, обобщение | 2 | 2 | 0,41 | 0,83 | 0,42 | < 1е-6 |
| Time optimal, l, агрегация | 2 | 2 | 0,41 | 0,83 | 0,42 | < 1е-6 |
| Group join, k, подавление | 2 | 2 | 0,35 | 0,28 | 0,08 | 0,001 |
| Group join, k, обобщение | 2 | 2 | 0,26 | 0,74 | 0,25 | 0,001 |
| Group join, k, агрегация | 2 | 2 | 0,26 | 0,74 | 0,25 | 0,004 |
| Group join, l, подавление | 2 | 2 | 0,35 | 0,27 | 0,083 | 0,001 |
| Group join, l, обобщение | 2 | 2 | 0,26 | 0,74 | 0,25 | 0,001 |
| Group join, l, агрегация | 2 | 2 | 0,26 | 0,74 | 0,25 | 0,001 |
| Group join, t, подавление | 12 | 9 | 0 | 0 | 0 | 0,005 |
| Group join, t, обобщение | 2 | 2 | 0,35 | 0,72 | 0,25 | 0,015 |
| Group join, t, агрегация | 2 | 2 | 0,35 | 0,72 | 0,26 | 0,006 |
| Datafly | 5 | 5 | 0,19 | 0,49 | 0,08 | 0,002 |
| Equal sized | 3 | 3 | 0,26 | 0,75 | 0,25 | < 1е-6 |
| Перебор, k, подавление | 2 | 2 | 0,29 | 0,28 | 0,083 | 125 |
| Перебор, l, подавление | 2 | 2 | 0,29 | 0,28 | 0,083 | 343 |
| Перебор, t, подавление | 2 | 2 | 0,29 | 0,28 | 0,083 | 466 |

Таблица 18 – короткий датасет, полезность (k=3, l=3, t=0,5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance |
| Time optimal, k, подавление | 0,083 | 1 | 1 | 1 |
| Time optimal, k, обобщение | 0,33 | 1 | 0,44 | 0,09 |
| Time optimal, k, агрегация | 0,33 | 0,96 | 0,44 | 0,07 |
| Time optimal, l, подавление | 0,083 | 1 | 1 | 1 |
| Time optimal, l, обобщение | 0,33 | 1 | 0,44 | 0,09 |
| Time optimal, l, агрегация | 0,33 | 0,96 | 0,44 | 0,07 |
| Group join, k, подавление | 0,083 | 1 | 1 | 1 |
| Group join, k, обобщение | 0,25 | 1 | 0,56 | 0,24 |
| Group join, k, агрегация | 0,25 | 1 | 0,57 | 0,09 |
| Group join, l, подавление | 0,023 | 0,68 | 0,58 | 0,68 |
| Group join, l, обобщение | 0,25 | 1 | 0,57 | 0,24 |
| Group join, l, агрегация | 0,25 | 1 | 0,57 | 0,09 |
| Group join, t, подавление | 0,083 | 1 | 1 | 1 |
| Group join, t, обобщение | 0,25 | 1 | 0,56 | 0,17 |
| Group join, t, агрегация | 0,25 | 1 | 0,56 | 0,09 |
| Datafly | 0,17 | 1 | 0,72 | 0,25 |
| Equal sized | 0,33 | 1 | 0,44 | 0,08 |
| Перебор, k, подавление | 0,083 | 1 | 1 | 1 |
| Перебор, l, подавление | 0,083 | 1 | 1 | 1 |
| Перебор, t, подавление | 0,083 | 1 | 1 | 1 |

Таблица 19 - короткий датасет, риски обезличивания (k=3, l=3, t=0,5)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Время, с |
| Time optimal, k, подавление | 12 | 9 | 0 | 0 | 0 | 0,001 |
| Time optimal, k, обобщение | 3 | 3 | 0,26 | 0,75 | 0,25 | < 1e-6 |
| Time optimal, k, агрегация | 12 | 9 | 0 | 0 | 0 | 0,001 |
| Time optimal, l, подавление | 12 | 9 | 0 | 0 | 0 | 0,001 |
| Time optimal, l, обобщение | 3 | 3 | 0,26 | 0,75 | 0,25 | < 1е-6 |
| Time optimal, l, агрегация | 3 | 3 | 0,26 | 0,75 | 0,25 | < 1е-6 |
| Group join, k, подавление | 12 | 9 | 0 | 0 | 0 | 0,001 |
| Group join, k, обобщение | 3 | 3 | 0,26 | 0,65 | 0,17 | 0,001 |
| Group join, k, агрегация | 3 | 3 | 0,26 | 0,65 | 0,17 | 0,004 |
| Group join, l, подавление | 12 | 9 | 0 | 0 | 0 | 0,001 |
| Group join, l, обобщение | 3 | 3 | 0,26 | 0,65 | 0,17 | 0,001 |
| Group join, l, агрегация | 3 | 3 | 0,26 | 0,65 | 0,17 | 0,001 |
| Group join, t, подавление | 12 | 9 | 0 | 0 | 0 | 0,006 |
| Group join, t, обобщение | 4 | 3 | 0,17 | 0,67 | 0,17 | 0,015 |
| Group join, t, агрегация | 4 | 3 | 0,17 | 0,67 | 0,17 | 0,007 |
| Datafly | 5 | 5 | 0,19 | 0,48 | 0,08 | 0,001 |
| Equal sized | 3 | 3 | 0,26 | 0,75 | 0,25 | 0,001 |
| Перебор, k, подавление | 12 | 9 | 0 | 0 | 0 | 136 |
| Перебор, l, подавление | 12 | 9 | 0 | 0 | 0 | 321 |
| Перебор, t, подавление | 12 | 9 | 0 | 0 | 0 | 352 |

Таблица 20 – короткий датасет, полезность (k=4, l=4, t=0,1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | Distinctness | Доля изменений | Потеря информации | My distance |
| Time optimal, k, подавление | 0,083 | 1 | 1 | 1 |
| Time optimal, k, обобщение | 0.25 | 1 | 0.56 | 0.13 |
| Time optimal, k, агрегация | 0,25 | 1 | 0,56 | 0,1 |
| Time optimal, l, подавление | 0,083 | 1 | 1 | 1 |
| Time optimal, l, обобщение | 0,025 | 1 | 0,56 | 0,13 |
| Time optimal, l, агрегация | 0,25 | 1 | 0,56 | 0,1 |
| Group join, k, подавление | 0,083 | 1 | 1 | 1 |
| Group join, k, обобщение | 0,25 | 1 | 0,56 | 0,17 |
| Group join, k, агрегация | 0,25 | 1 | 0,56 | 0,09 |
| Group join, l, подавление | 0,083 | 1 | 0 | 1 |
| 1Group join, l, обобщение | 0,17 | 1 | 0,74 | 0,23 |
| Group join, l, агрегация | 0,17 | 1 | 0,74 | 0,1 |
| Group join, t, подавление | 0,083 | 1 | 1 | 1 |
| Group join, t, обобщение | 0,17 | 1 | 0,72 | 0,3 |
| Group join, t, агрегация | 0,17 | 1 | 0,72 | 0,18 |
| Datafly | 0,17 | 1 | 0,73 | 0,25 |
| Equal sized | 0,17 | 1 | 0,72 | 0,17 |
| Перебор, k, подавление | 0,083 | 1 | 1 | 1 |
| Перебор, l, подавление | 0,083 | 1 | 1 | 1 |
| Перебор, t, подавление | 0,083 | 1 | 1 | 1 |

Таблица 21 - короткий датасет, риски обезличивания (k=4, l=4, t=0,1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Алгоритм, метрика (k, l, t), метод | max k | max l | min t | AKG | AAG | Время, с |
| Time optimal, k, подавление | 12 | 9 | 0 | 0 | 0 | < 1e-6 |
| Time optimal, k, обобщение | 4 | 4 | 0.26 | 0.67 | 0.17 | 0.001 |
| Time optimal, k, агрегация | 4 | 4 | 0,26 | 0,67 | 0,17 | < 1е-6 |
| Time optimal, l, подавление | 12 | 9 | 0 | 0 | 0 | 0,001 |
| Time optimal, l, обобщение | 4 | 4 | 0,26 | 0,67 | 0,17 | 0,001 |
| Time optimal, l, агрегация | 4 | 4 | 0,26 | 0,67 | 0,17 | < 1е-6 |
| Group join, k, подавление | 12 | 9 | 0 | 0 | 0 | 0,001 |
| Group join, k, обобщение | 4 | 3 | 0,17 | 0,67 | 0,17 | 0,001 |
| Group join, k, агрегация | 4 | 3 | 0,17 | 0,67 | 0,17 | 0,001 |
| Group join, l, подавление | 12 | 9 | 0 | 0 | 0 | < 1е-6 |
| Group join, l, обобщение | 4 | 4 | 0,17 | 0,44 | 0,083 | 0,002 |
| Group join, l, агрегация | 4 | 4 | 0,17 | 0,44 | 0,083 | 0,002 |
| Group join, t, подавление | 12 | 9 | 0 | 0 | 0 | 0,005 |
| Group join, t, обобщение | 6 | 5 | 0,07 | 0,5 | 0,083 | 0,016 |
| Group join, t, агрегация | 6 | 5 | 0,07 | 0,5 | 0,083 | 0,006 |
| Datafly | 5 | 5 | 0,19 | 0,49 | 0,083 | 0,003 |
| Equal sized | 6 | 6 | 0,19 | 0,5 | 0,083 | 0,001 |
| Перебор, k, подавление | 12 | 9 | 0 | 0 | 0 | 135 |
| Перебор, l, подавление | 12 | 9 | 0 | 0 | 0 | 233 |
| Перебор, t, подавление | 12 | 9 | 0 | 0 | 0 | 237 |