1)Алгоритм, свойства алгоритма, способы описания алгоритма, алгоритмические структуры (типы алгоритмов), шаблон описания алгоритма, этапы решения задач на ПК.

Алгоритм (лат. algorithmi — от имени среднеазиатского математика Аль-Хорезми) — конечная совокупность точно заданных правил решения некоторого класса задач или набор инструкций, описывающих порядок действий исполнителя для решения определённой задачи.

Свойства алгоритмов:

1. универсальность
2. дискретность
3. конечность
4. результативность
5. выполнимость (эффективность)
6. детерминированность (определенность)
7. последовательность

Способы описания алгоритма:

1. описательный
2. графический

Алгоритмические структуры (типы алгоритмы):

1. следование
2. ветвление
3. цикл
4. функция

Шаблон описания алгоритмов:(она ссылалась на этот документ)

Требования к структуре описания алгоритма (проектной процедуры) по ГОСТ 34 устанавливаются РД 50-34.698-90. В общем случае документ должен состоять из следующих разделов:

1 Назначение и характеристика

1.1 Назначение алгоритма (его части)

1.2 Обозначение документа (документов) «Описание постановки задачи», для решения которой он предназначен

1.3 Обозначение документа «Описание алгоритма», с которым связан данный алгоритм (при необходимости)
1.4 Краткие сведения о процессе (объекте), при управлении которым используют алгоритм, а также воздействия на процесс с точки зрения пользователя, осуществляемые при функционировании алгоритма
1.5 Ограничения на возможность и условия применения алгоритма и характеристики качества решения (точность, время решения и т. д.)
1.6 Общие требования к входным и выходным данным (форматам, кодам и т. д.), обеспечивающие информационную совместимость решаемых задач в системе

2 Используемая информация

2.1 Массивы информации, сформированные из входных сообщений (документов плановой, учетной и нормативно-справочной информации, сигналов и т. д.)
2.1.1 Наименование, обозначение и максимальное число записей в массиве
2.1.2 Перечень наименований и обозначений используемых (или неиспользуемых) реквизитов и (или) входных переменных задачи или ссылка на документы, содержащие эти данные
2.2 Массивы информации, полученные в результате работы других алгоритмов и сохраняемые для реализации данного алгоритма
2.2.1 Наименование, обозначение и максимальное число записей в массиве
2.2.2 Перечень наименований и обозначений используемых (или неиспользуемых) реквизитов и (или) входных переменных задачи или ссылка на документы, содержащие эти данные

3 Результаты решения

3.1 Массивы информации и (или) сигналов, формируемые для выдачи выходных сообщений (документов, видеокадров, сигналов управления и т. д.)
3.1.1 Наименование, обозначение, максимальное число записей в массиве
3.1.2 Перечень наименований и обозначений реквизитов и (или) выходных переменных, используемых для формирования выходных сообщений или ссылка на документы, содержащие эти данные
3.2 Массивы информации, сохраняемой для решения данной и других задач АС
3.2.1 Наименование, обозначение, максимальное число записей в массиве
3.2.2 Перечень наименований и обозначений реквизитов и (или) выходных переменных, используемых для формирования выходных сообщений или ссылка на документы, содержащие эти данные

4 Математическое описание

4.1 Математическая модель или экономико-математическое описание процесса (объекта)
4.2 Перечень принятых допущений и оценки соответствия принятой модели реальному процессу (объекту) в различных режимах и условиях работы (например, для АСУ ТП-стационарные режимы, режимы пуска и остановки агрегатов, аварийные ситуации и т. д.)
4.3 Сведения о результатах научно-исследовательских работ, если они использованы для разработки алгоритма

5 Алгоритм решения

5.1 Описание логики алгоритма и способа формирования результатов решения с указанием последовательности этапов счета, расчетных и (или) логических формул, используемых в алгоритме

5.2 Указания точности вычисления

5.3 Соотношения, необходимые для контроля достоверности вычислений

5.4 Описание связей между частями и операциями алгоритма

5.5 Указания о порядке расположения значений или строк в выходных документах (например, по возрастанию значений кодов объектов, по группам объектов и т. д.)

этапы решения задач на ПК:

1. постановка задачи
2. формальная постановка модели задачи
3. построение математической модели задачи
4. выбор и обоснование метода решения
5. построение алгоритма
6. составление программы
7. отладка программы
8. решение задачи на компьютере и анализ решения

2)Система, основные понятия (целостность, структура, элемент, подсистема, цели), классификация систем.

Система – множество элементов, находящихся в отношениях и связях друг с другом, образующих определенную целостность, единство и характеризующихся определенной функцией.

Целостность – свойство системы не являться суммой свойств элементов системы или частей, т.е. целое не может быть сведено к простой сумме частей; свойство системы зависит от свойств компонентов, элементов, т.е. изменение в одной части вызывает изменение во всех остальных частях и во всей системе.

Структура – отражает наиболее существенные взаимоотношения между элементами и их группами, которые мало изменяются при модернизации системы и обеспечивают существенные свойства и существование системы.

–теоретико-множественное описание: матрицы, графы и языки моделирования.

Элемент системы – простейшая и неделимая часть системы. Элемент – это предел деления системы с точки зрения решения конкретной задачи и поставленной цели.

Подсистема – если элемент обладает внутренней структурой, то его называют подсистемой. Система может быть разделена на элементы не сразу, а последовательно на такие части, которые представляют собой компоненты, более крупные, чем элементы и в тоже время более детальные, чем система в целом.

Цель – сохранение и воспроизведение самой системы и взаимодействие с другими элементами.

Классификация систем:

1) по области
2) по свойствам
3) по сложности и размерам

3)Модель, моделирование, виды моделирования, математическое моделирование, методология.

Моделирование – процесс замещения объектной сферы некоторой моделью и приведения исследований на модели с целью получения информации об объекте. Т.е. под моделированием понимается процесс построения и использования модели.

Модель (от лат. modulus- мера, образец, норма) – физический или абстрактный образ моделируемого объекта, удобный для проведения исследований и позволяющий адекватно изображать физические свойства и характеристики объекта. Другим словами, объект (материально или мысленно представляемый), который замещает в процессе изучения объект-оригинал, сохраняя его физические свойства и характеристики.

Различают моделирование материальное и идеальное.

Материальное моделирование – моделирование с использованием материального аналога, воспроизводящего физические, геометрические, динамические и функциональные характеристики объекта.

Натурное (физическое) моделирование (макеты в архитектуре, модели судов, модели самолетов, испытываемые в аэродинамических трубах.)

Аналоговое моделирование (электрические схемы, с помощью которых можно изучать механические колебания, и наоборот.)

Идеальное моделирование – это моделирование, носящее теоретический характер и основанное на аналогии идеальной (не материальной), мысленной.

Интуитивное моделирование – это моделирование, основанное на интуитивном представлении об объекте исследования. Интуитивным следует считать эмпирические (полученные на основе эксперимента или в процессе наблюдения) знания без объяснения причин и механизмов наблюдаемого явления.

Научное моделирование – это моделирование, использующее минимальное число предположений, принятых в качестве гипотез.

Математическое моделирование – это научное знаковое моделирование, при котором описание объекта осуществляется на языке математики, а исследование модели проводится с использованием различных математических методов.

АСД

06.09

Алгоритм Аль-Хорезми
Свойства алгоритма:
1) универсальность
2) дискретность
3) конечность
4) результативность
5) комплитность (эффективность)
6) детерминированность (определенность)
7) последовательность

Структура и содержание док. ГОСТ 34
№ 50-34.698-90

Этапы решения:
1) Постановка задачи
2) Формальное построение модели задачи
3) Построение мат. модели задачи
4) Выбор и обоснование метода решения
5) Построение алгоритма
6) Составление программы
7) Отладка программы
8) Решение и анализ

(Web of science, ecopm) Свойства системы:
1) Целостность
2) Структура
3) Элемент
4) Подсистема
5) Цель анализа (сохранение и совершенствование свойств системы и взаимообусловленные с другими системами)

Алгоритм
- Описательный — нат. язык
- Графический — компактный спец. язык

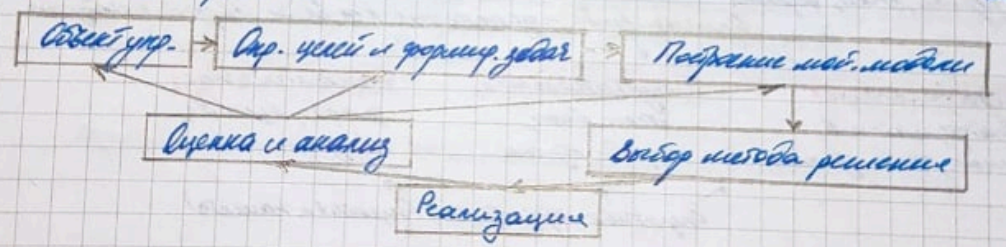
Типы алгоритмов
- Лин.
- Разв.
- Цикл.
- Слож.

Система
- Техн.
- Физ.
- Хим.
- Соц.
- Искусств.

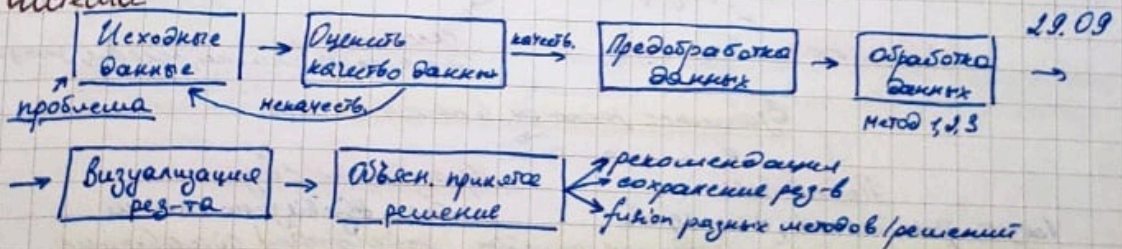
Система
- Детер.
- Лин.
- Стохаст.
- Стацион.

Система → Модель
Вход.
Рез. функ.
Выход. пар.
Конц. опис. модели

Мат. моделирование



Система



4)Уровень абстрагирования, данные, структура данных/тип данных, представление структуры данных (логическое, физическое), элемент данных, отношениями между данными, элемент отношений.

Уровень абстрагирования – множество данных, представляющих предметную область решаемой задачи.

*(Перед тем как решать любую задачу, нам нужно выбрать этот уровень абстрагирования, руководствуясь особенностями решаемой задачи и способами представления информации. Необходимо ориентироваться на те средства, которые предоставляют среды программирования и вычислительная техника, на которой будут выполняться программы)

Любые данные в памяти компьютера представляются последовательностью двоичных разрядов, или битов, а их значениями являются соответствующие двоичные числа.

Данные, рассматриваемые в виде последовательности битов, имеют очень простую организацию или, другими словами, слабо структурированы.

Организация более крупных и содержательных данных получается на основе понятия “структуры данных”.

Структура данных-организационная схема записи или массива, в соответствие с которой упорядочены данные, с тем чтобы их можно было интерпретировать и выполнять над ними определенные операции.

Структура данных - совокупность элементов данных и отношений между ними.

Пример структуры данных: одномерный массив, двумерный массив.

Тип данных - множество значений и операций над этими значениями

В языках программирования тип данных определяет:

- Возможные значения переменных, констант, функций, выражений, принадлежащих к данному типу
- Внутреннюю форму представления данных в компьютере
- Операции и функции, которые могут выполняться над величинами, принадлежащими к данному типу.

Структура данных определяет набор переменных, возможно разных типов, объединенных определенным образом. Структура данных состоит из простых данных.

Физическое/логическое представление структуры данных (уровни представления данных):

- Информация в компьютере представляется человеку в каком-то виде (в абстрактном виде), который на его взгляд упорядочивает данные и придает им смысл.

- Компьютер отводит поле для поступающей информации и задает какой-то адрес хранения

Таким образом

- структура данных, которая рассматривается без учета ее представления в машинной памяти называется абстрактной или логической. Соответственно структура, которая отражает способ физического представления данных в памяти машины называется структурой хранения, внутренней структурой или структурой памяти.

Элемент данных – неделимый информационный элемент, являющийся минимальной структурной единицей информации. Вид элемента данных определяется характером содержащихся в нем сведений и особенностью его организации или записи. Частный случай структуры данных.

Отношения между данными – функциональные связи между данными и указателями на то, где находятся эти данные.

Элемент отношений – совокупность всех связей элемента с другими элементами данных, рассматриваемой структуры.

5)Уровни представления данных, абстрактный тип данных, классификация структур данных, базовые структуры данных, операции над структурами данных.

Уровни представления данных:

- Информация в компьютере представляется человеку в каком-то виде (в абстрактном виде), который на его взгляд упорядочивает данные и придает им смысл.

- Компьютер отводит поле для поступающей информации и задает какой-то адрес хранения

Таким образом

- человек обрабатывает данные на логическом уровне(абстрактно), компьютер обрабатывает данные на физическом уровне.

Абстрактный тип данных – это набор, включающий данные и выполняемые над ними операции, набор данных и методов, служащих одной цели

Абстрактный тип данных - это тип данных, который предоставляет для работы с элементами этого типа определенный набор функций, а также возможность создавать элементы этого типа при помощи специальных функций

Примеры АТД: список, очередь, стек, дек.

Классификация структур данных:

Простые – структуры данных, которые не могут быть разделены на составные части, большие, чем бит. С точки зрения физической структуры в данной системе программирования всегда можно заранее сказать, каков будет размер данного простого типа и какова структура его размещения в памяти. С логической точки зрения простые данные являются неделимыми единицами.

Интегрированные – структуры данных, составными частями которых являются другие структуры данных(простые или в свою очередь интегрированные). Интегрированные структуры данных конструируются программистом с использованием средств интеграции данных, предоставляемых языками программирования.

В зависимости от отсутствия или наличия явно заданных связей между элементами данных следует различать:

- **несвязные структуры** (векторы, массивы, строки, стеки, очереди)
- **связные структуры** (связные списки).

Структуры данных по принципу изменчивости (изменение числа элементов):

- **статические (например, массив)**
- **полустатические (например, вектор)**
- **динамические** (например, списки, очередь, деревья, дек, стек)

Базовые структуры данных, статические, полустатические и динамические характерны для оперативной памяти и часто называются структурами.

Файловые структуры соответствуют структурам данных для внешней памяти.

По характеру упорядоченности элементов структуры данных делятся на:

- **линейные**
- **нелинейные** (многосвязные списки, деревья, графы)

В зависимости от характера взаимного расположения элементов в памяти линейные структуры можно разделить на структуры с:

- **последовательным** распределением элементов в памяти (векторы, строки, массивы, стеки, очереди)
- **произвольным связным** распределением элементов в памяти (односвязные, двусвязные списки)

Операции над структурами данных

Операции обязательные для всех структур и типов данных (выполняются над любыми структурами данных): **создание**, **уничтожение**, **выбор**, **обновление**.

Для каждой структуры могут быть определены специфические операции, работающие только с данной структурой.

Создание – выделение памяти для структуры данных в процессе выполнения программы или на этапе компиляции. Программист может сам выделять память, если захочет.

Уничтожение – понятно.

Выбор – операция доступа к данным внутри самой структуры. Метод доступа – важное свойство структуры.

Обновление – изменение значения данных в структуре данных (присваивание, передача параметров).

6)Массив, многомерные массивы, свободные массивы, описание, характеристики, их использование.

Массив — структура данных, хранящая набор значений (элементов массива), идентифицируемых по индексу или набору индексов, принимающих целые (или приводимые к целым) значения из некоторого заданного непрерывного диапазона.

Массивы состоят из элементов одного базового типа -> структура массива однородна.

Базовый тип может быть как скалярным, так и структурированным (то есть элементами массива могут быть числа, символы, строки, структуры, в том числе и массивы).

Число элементов массива фиксировано -> объем занимаемой памяти остается неизменным. Также массив в памяти хранится в виде вектора, т.е. все элементы размещаются в смежных участках памяти подряд, начиная с адреса, соответствующего началу массива

В зависимости от числа индексов различают:

- **одномерные** массивы

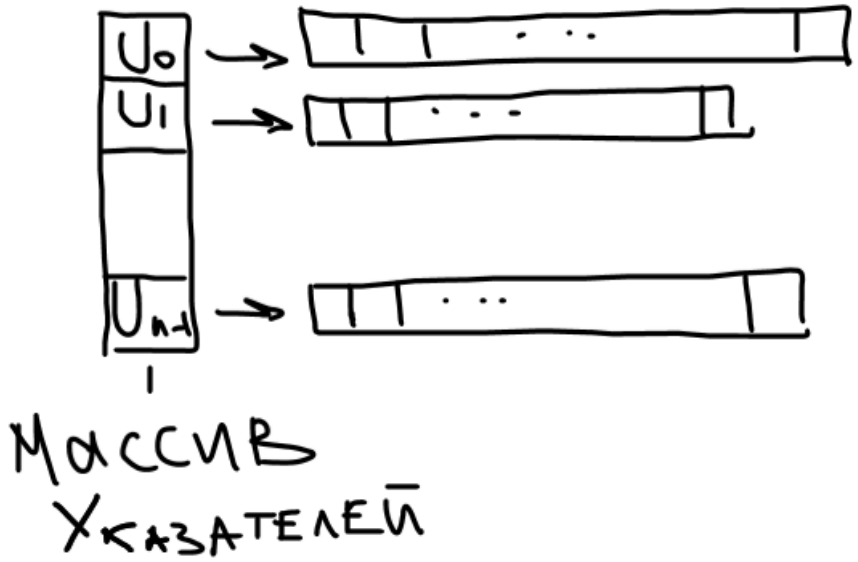
- **многомерные** массивы

Дескриптор – управляющий блок массива, который содержит следующие данные: тип структуры, адрес начала массива, тип элемента, нижняя граница индекса, верхняя граница индекса. В случае многомерного массива для каждого индекса задаются нижняя и верхняя граница или для каждой строки создается свой дескриптор.

Фиксированный размер массива может являться ограничивающим фактором: в случаях, когда мы не знаем, сколько памяти выделять под массив, мы можем построить массив в **динамической памяти**, получаемой с помощью средств управления памятью ОС. Управление доступом к элементам таких массивов осуществляется самой программой по вычисляемым индексам (адресам) элементов с использованием **указателя**.

Свободные массивы – двумерные массивы, в которых размер строк может быть различным.

Структура хранения свободного массива с n строками:



Массив указателей (U) имеет n элементов, каждый из которых содержит адреса векторов-строк массива.

Недостатки свободных массивов:

- Потребность в дополнительной памяти под массив указателей и многократное обращение к ОС для получения памяти

Преимущества свободных массивов:

- Более эффективное хранение данных (например, треугольная матрица)
- В некоторых случаях сортировки массива, можно просто менять указатели на строки, а не перемещать сами строки.
- Возможность получения памяти под каждую строку

Обработка больших объемов памяти путем размещения в оперативной памяти отдельных строк, над которыми производятся вычисления

Области применения массивов:

- Числовые массивы в вычислительных задачах
- Матричная алгебра, экстраполяция, интерполяция
- Таблица – массив с элементами типа “запись”
- Представление графов (в частном случае деревьев) в удобной для работы виде
- Управляющие и информационные таблицы в операционных системах управления СУБД (н и ч е г о н е п о н я т н о)

7)Записи, их хранение, Операции над Записями.

Запись представляет собой совокупность ограниченного числа логических связных компонентов, принадлежащих к разным типам.

Компоненты записи называются *полями*, каждое из которых определяется именем.

Мощность составляющего типа есть произведение мощностей составляющих его типов.

Каждый элемент записи имеет уникальное для этой записи *имя*. (т.е. такое же имя может встречаться только в другой таблице)

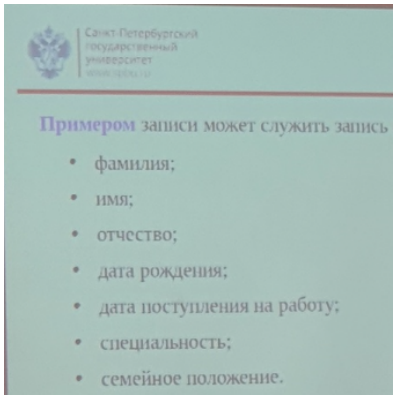
Записи как и массивы состоят из фиксированного числа элементов, но между ними имеются несколько различий:

1) В отличие от массивов где данные состоят из однокипных элементов, элементы записи могут быть разных типов.

2) В массивах доступ к элементам осуществятся по индексам, а в записях по именам. Отдельные поля записи могут служит в качестве ключей записи.

Записи используют в след. Структурах:

1. В таблицах
2. В файлах
3. В базах данных



Хранение записи:

Запись храниться в одной сплошной области памяти. Причём её элементы располагаются в памяти последовательно друг за другом, в том порядке в котором они перечислены в записи.

Операции над записями:

1. Важнейшей операцией для записи является операция доступа к выбранному записи *(Операция квалификации)*

<имя переменной записи><имя поля> == <условно имя строки><условно имя столбика>

Над выбранным полем допустимы любые операции допустимые для типа данного поля.

2) Операция присваивания – одной записи присваиваем запись состоящею из тех же типов.

3) Операция сравнения – записи состоящие из одних и тех же типов можно сравнить. (В некоторых языках это возможно сделать только поэлементно С)

8)Множества, их хранение, основные операции, их частные случаи

С точки зрения структур данных **множество** можно рассматривать как совокупность данных, над которыми выполняются некоторое число операций, образующих функциональную спецификацию структуры этого множества.

Пример: Пусть определён некоторый тип данных T. Определим другой тип данных, элементами которого является множество объектов типа T.

Основные операции над множествами:

1) создать множество – создаётся пустое множество возвращается его адрес

2) включить элемент – формируется новое множество добавлением одного элемента к уже существующему множеству

3) найти элемент – проверить, есть ли элемент в множестве, если есть определить его адрес

4) удалить элемент – формируется новое множество с удалением одного элемента если, если он есть; в противном случае множество остаётся без изменений

5) пусто – проверить есть ли какие-то элементы в множестве

Хранение множеств:

Множество в памяти хранится как массив битов, в котором каждый бит указывает является ли элемент принадлежащим объявленному множеству или нет.

Число байтов, выделяемых для данных типа множества, вычисляется по формуле:

ByteSize = (max div 8) - (min div 8) + 1,

где max/min это верхняя и нижняя границы базового типа данного множества

Номер байта для конкретного элемента E вычисляется по формуле:

ByteNumber = (E div 8) - (min div 8)

Номер бита внутри этого байта по формуле:

BitNumber = (E mod 8)

Частные случаи:

Как я понял из презентаций она хочет услышать про понятие множества в математике.

9)Линейные структуры данных

По характеру упорядочивания элементов структуры данных делятся на *линейные и нелинейные*. В зависимости от характера расположения элементов в памяти линейные структуры можно разделить на структуры с **последовательным распределением элементов** в памяти (векторы, строки, массивы, стеки, очереди) и структуры с **произвольным связным распределением** в памяти (односвязные, двусвязные списки)

Пример нелинейной структуры – многосвязные списки, деревья, графы.

Примеры линейных структур:

Массив – это поименованная совокупность однотипных элементов, упорядоченных по индексам, определяющих положение элементов в массиве.

Строка – это последовательность символов.

Структуры (запись) – это агрегат, составляющие которого (поля и функции) могут иметь имя и могут быть различного типа.

Множество – это совокупность каких-либо однородных элементов, объединенных общим признаком и представляемых как единое целое.

Таблица – представляет собой одномерный массив, элементами которого являются записи.

Ключ таблицы – поле, значение которого может быть использовано для доступа к записи.

Список – это структура данных, представляющая собой логически связанную последовательность элементов.

10)Списки, основные операции, характеристики (линейный однонаправленный, циклический однонаправленный, циклический двунаправленный):

Список – структура данных, представляющая собой логически связанную последовательность элементов.

Линейный однонаправленный список – любой элемент хранит собственно данные, а также ссылку, указывающую на следующий элемент в списке или является пустым у последнего элемента.

Рисунок 1 Линейный однонаправленный список

Основные операции, осуществляемые с линейным однонаправленным списком:

- Три операции **добавления** объекта в список (в начало, конец или после любого элемента);
- **просмотр**;
- **поиск**;
- Три операции **удаления** объекта из списка (начало, конец или после любого элемента);
-

Добавление узла: Три разные операции, описана только для произв. элемента.

Функция добавления узла в список принимает два аргумента: Указатель на узел, после которого происходит добавление, Данные для добавляемого узла.

Рисунок 2 Вставка элемента

Добавление узла включает в себя следующие этапы:

- создание добавляемого узла и заполнение его поля данных;
- переустановка указателя узла, предшествующего добавляемому, на добавляемый узел;
- установка указателя добавляемого узла на следующий узел (тот, на который указывал предшествующий узел).

Удаление узла опять же три операции:

В качестве аргументов функции удаления элемента передаются указатель на удаляемый узел, а также указатель на корень списка. Функция возвращает указатель на узел, следующий за удаляемым.

Рисунок 3 Удаление

Удаление узла ОЛС включает в себя следующие этапы: установка указателя предыдущего узла на узел, следующий за удаляемым; освобождение памяти удаляемого узла.

Следует обратить особое внимание на то, что при выполнении любых операций с линейным однонаправленным списком **необходимо обеспечивать позиционирование** какого-либо **указателя на первый элемент**. В противном случае часть или весь список будет недоступен.

Линейный однонаправленный список имеет только один указатель на элемент. Это позволяет минимизировать расход памяти на организацию такого списка. Одновременно, это позволяет осуществлять переходы между элементами только в одном направлении, что зачастую увеличивает время, затрачиваемое на обработку списка. Например, для перехода к предыдущему элементу необходимо осуществить просмотр списка с начала до элемента, указатель которого установлен на текущий элемент.

Характеристика: Достоинство – возможность **изменять** размер, **простота** реализации. Благодаря наличию ссылок, **каждый элемент** в списке, в отл. от массива, может занимать **разный объём памяти**. Адрес первого элемента однозначно определяется адресом самого списка.
https://studfile.net/preview/7278897/page:7/

Про **Циклические списки**: Линейные списки характерны тем, что в них можно выделить первый и последний элементы (имеющие пустые указатели), причем для однонаправленного линейного списка обязательно нужно иметь указатель на первый элемент. Это приводило к тому, что алгоритмы вставки и удаления крайних и средних элементов списка отличались друг от друга, что, естественно, усложняло соответствующие операции. Основное **отличие циклического списка** состоит в том, что в этом списке нет элементов, содержащих пустые указатели, и, следовательно, нельзя выделить крайние элементы. Таким образом, все элементы являются «средними».

Циклический однонаправленный список - похож на линейный однонаправленный список, но его последний элемент содержит указатель, связывающий его с первым элементом.

Для полного обхода такого списка достаточно иметь указатель на произвольный элемент, а не на первый, как в линейном однонаправленном списке. Понятие «первого» элемента здесь достаточно условно и не всегда требуется. Хотя иногда бывает полезно выделить некоторый элемент как «первый» путем установки на него специального указателя. Это требуется, например, для предотвращения «зацикливания» при просмотре списка.

Рисунок 4 Циклический однонаправленный список

Основные операции, осуществляемые с циклическим однонаправленным списком:

- вставка элемента;
- просмотр
- поиск;
- удаление элемента.
-

Операция добавления как описана выше.

Операция удаления элемента циклического однонаправленного списка осуществляет удаление элемента, на который установлен указатель текущего элемента. После удаления указатель текущего элемента устанавливается на следующий за удаляемым элемент списка. Поскольку список циклический, нет необходимости передавать указатель на корень списка. Здесь не требуется отдельных алгоритмов удаления для крайних элементов списка, как это было в линейных списках, но **в случае удаления «первого» элемента**, необходимо соответствующий указатель переместить на следующий элемент.

Циклический однонаправленный список, так же как и линейный однонаправленный список **имеет только один указатель** в элементах, что позволяет **минимизировать расход памяти** на организацию списка, но обеспечивает переходы между элементами только в одном направлении. Одновременно, здесь упрощены операции вставки и удаления элементов. Переход элементов только в одном направлении увеличивает время затрачиваемое на его обработку.

В билете **не стоит,** но напишу:

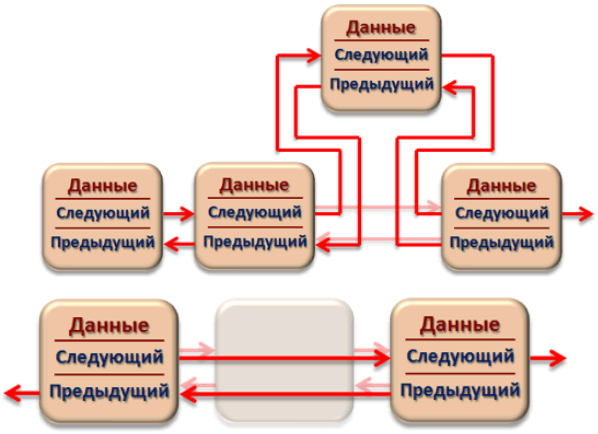
Линейный двунаправленный список:

В этом линейном списке любой элемент имеет два указателя, один из которых указывает на следующий элемент в списке или является пустым указателем у последнего элемента, а второй указывает на предыдущий элемент в списке или является пустым указателем у первого элемента.

Рисунок 5 Двусвязный линейный список

Основные операции, осуществляемые с линейным двунаправленным списком те же, что и с однонаправленным линейным списком.

Следует обратить внимание на то, что в отличии от однонаправленного списка здесь **нет необходимости** обеспечивать **позиционирование** какого-либо указателя именно **на первый элемент списка**, так как благодаря **двум указателям** в элементах можно получить **доступ к любому элементу** списка из любого другого элемента осуществляя переходы в прямом или обратном направлении. Однако часто бывает полезно иметь указатель на заголовок списка.



Характеристика:

Использование **двух указателей** в линейном двунаправленном списке позволяет **ускорить операции**, связанные **с передвижением по списку** за счет двунаправленности этого движения. Однако**, элементы** списка за счет дополнительного поля **занимают больший объем** памяти. Кроме того, **усложнились** операции **вставки и удаления** элементов за счет необходимости манипулирования большим числом указателей.

Циклический двунаправленный список:

В этом циклическом списке любой элемент имеет два указателя, один из которых указывает на следующий элемент в списке, а второй указывает на предыдущий элемент.

Рисунок 6 Двусвязный циклический список

Основные операции, осуществляемые с циклическим двунаправленным списком такие же, как и у циклического однонаправленного списка.

Для описания алгоритмов этих основных операций используем те же объявления данных, что и для линейного двунаправленного списка.

Вставка элемента в список, как уже говорилось, проще, чем для линейного двунаправленного списка и реализуется с помощью одной единственной процедуры

Операция удаления элемента также осуществляется во многом аналогично удалению из циклического однонаправленного списка.

Операции вставки и удаления элементов здесь осуществляются **проще**, чем в линейном двунаправленном списке, но **сложнее**, чем в циклическом однонаправленном списке (за счет необходимости манипулирования **большим** числом указателей).

Полезные ссылки:

https://prog-cpp.ru/category/struct-posts/

https://studfile.net/preview/7278897/page:7/ ЛОС

https://studfile.net/preview/7278897/page:8/ ЛДС

https://studfile.net/preview/7278897/page:9/ ЦЛС

https://studfile.net/preview/7278897/page:10/ ЦДС

11)Стек. Принцип LIFO, очередь. FIFO, дек.

Стек (стопка) – это *структура данных*, в которой новый элемент всегда записывается в ее начало (вершину) и очередной читаемый элемент также всегда выбирается из ее начала. В стеках используется *метод доступа* к элементам *LIFO* (*Last Input – First Output*, "последним пришел – первым вышел"). Чаще всего принцип работы стека сравнивают со стопкой тарелок: чтобы взять вторую сверху, нужно сначала взять верхнюю.

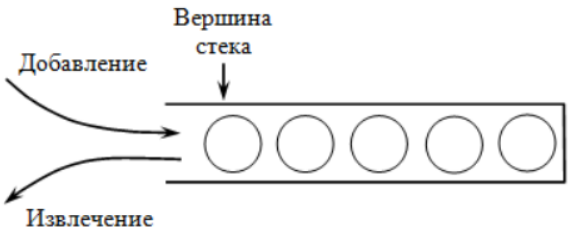


Рисунок 7 СТЕК ТАРЕЛОК

Стек можно реализовывать как **статическую структуру** данных в виде одномерного массива, а можно как динамическую структуру – в виде линейного списка. При реализации стека в виде **статического массива** необходимо **резервировать** массив, **длина** которого равна **максимально** возможной **глубине** стека, что приводит к **неэффективному** использованию **памяти**. Одновременно, работать с такой реализацией **проще** и **быстрее**. При такой реализации дно стека будет располагаться в первом элементе массива, а рост стека будет осуществляться в сторону увеличения индексов. Одновременно, необходимо отдельно хранить значение индекса элемента массива, являющегося вершиной стека.

Стек как **динамическую структуру** данных легко организовать на основе **линейного списка**. Поскольку работа всегда идет с заголовком стека, то есть **не требуется** осуществлять **просмотр** элементов, **удалению** и **вставку** элементов в **середину** или **конец** списка, то достаточно использовать экономичный по памяти **линейный однонаправленный список**. Для такого списка достаточно хранить указатель вершины стека, который указывает на первый элемент списка. Если стек пуст, то списка не существует и указатель принимает значение nil.

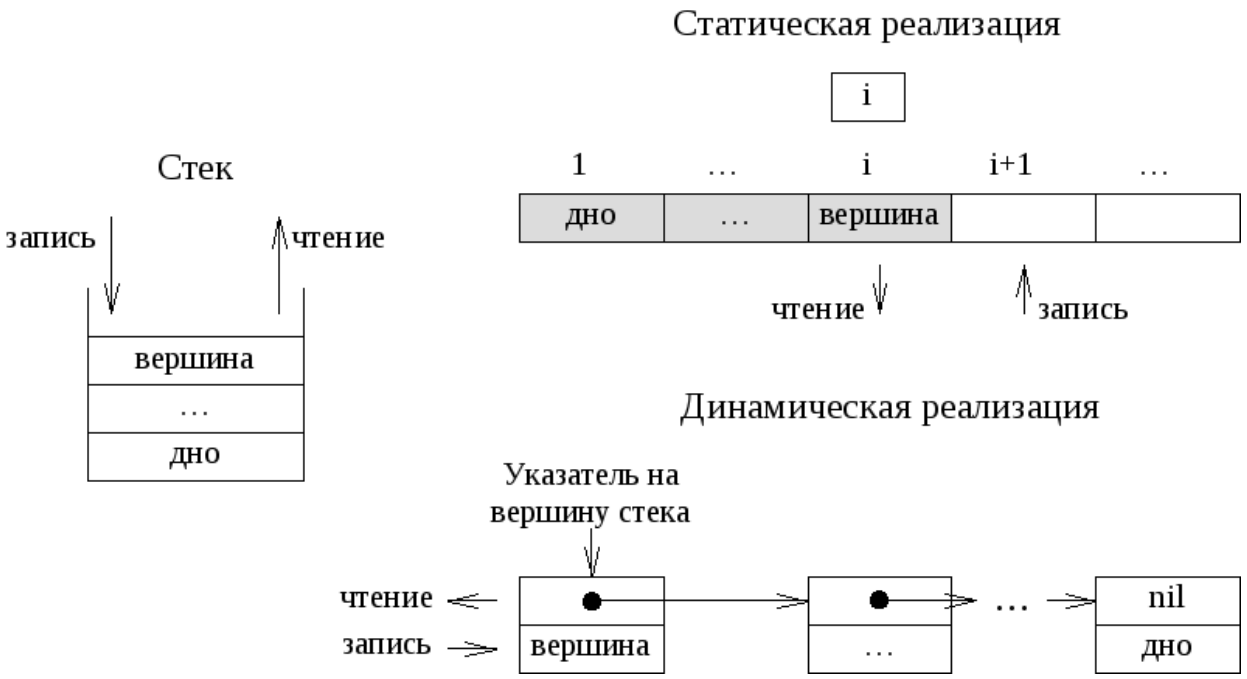


Рисунок 8 реализации стека

Основные операции, производимые со стеком:

записать (положить в стек);
прочитать (снять со стека);
очистить стек;
проверка пустоты стека.

Очередь – это *структура данных*, представляющая собой последовательность элементов, образованная в порядке их поступления. Каждый новый элемент размещается в конце очереди; элемент, стоящий в начале очереди, выбирается из нее первым. В очереди используется принцип доступа к элементам *FIFO* (*First Input – First Output*, "первый пришёл – первый вышел"). В очереди доступны два элемента (две позиции): *начало очереди* и *конец очереди*. Поместить элемент можно только в конец очереди, а взять элемент только из ее начала. Примером может служить обыкновенная *очередь* в магазине.

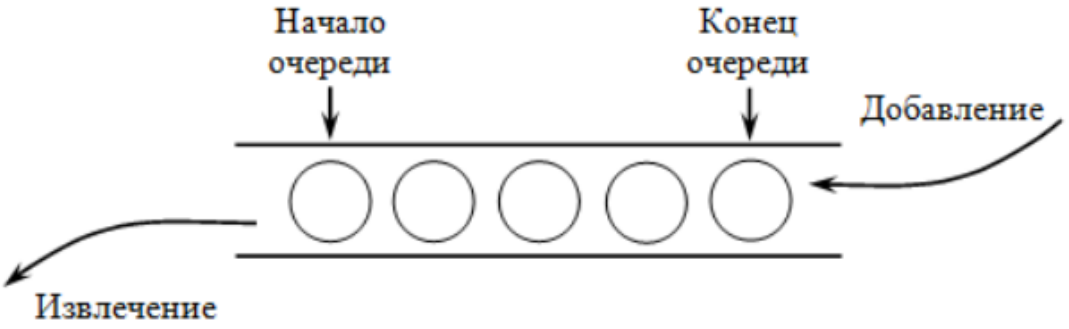


Рисунок 9 очередь

Очередь можно реализовывать как **статическую структуру** данных в виде **одномерного массива**, а можно как **динамическую структуру** – в виде **линейного списка.**

При реализации очереди в виде статического массива необходимо резервировать массив, длина которого равна максимально возможной длине очереди, что приводит к неэффективному использованию памяти.

При такой реализации **начало** очереди будет располагаться в **первом** элементе массива, а **рост** очереди будет осуществляться в сторону **увеличения индексов**. Однако, поскольку **добавление** элементов происходит в **один конец**, а **выборка** – из **другого конца** очереди, то с течением времени будет происходить **миграция** элементов очереди из **начала** массива в **сторону** его **конца**. Это может привести к быстрому **исчерпанию** массива и **невозможности добавлении** новых элементов в очередь при наличии **свободных** мест в начале массива. **Предотвратить** это можно двумя способами: осуществлять **сдвиг** всей очереди на один элемент к **началу** массива или представить массив в виде **циклической структуры**, где первый элемент массива следует за последним.

Очередь как динамическую структуру данных легко организовать на основе линейного списка. Поскольку работа идет с обоими концами очереди, то предпочтительно будет использовать линейный двунаправленный список. Если очередь пуста, то списка не существует, и указатели принимают значение nil.

Основные операции, производимые с очередью:

- добавить элемент;
- извлечь элемент;
- очистить очередь;
- проверка пустоты очереди.

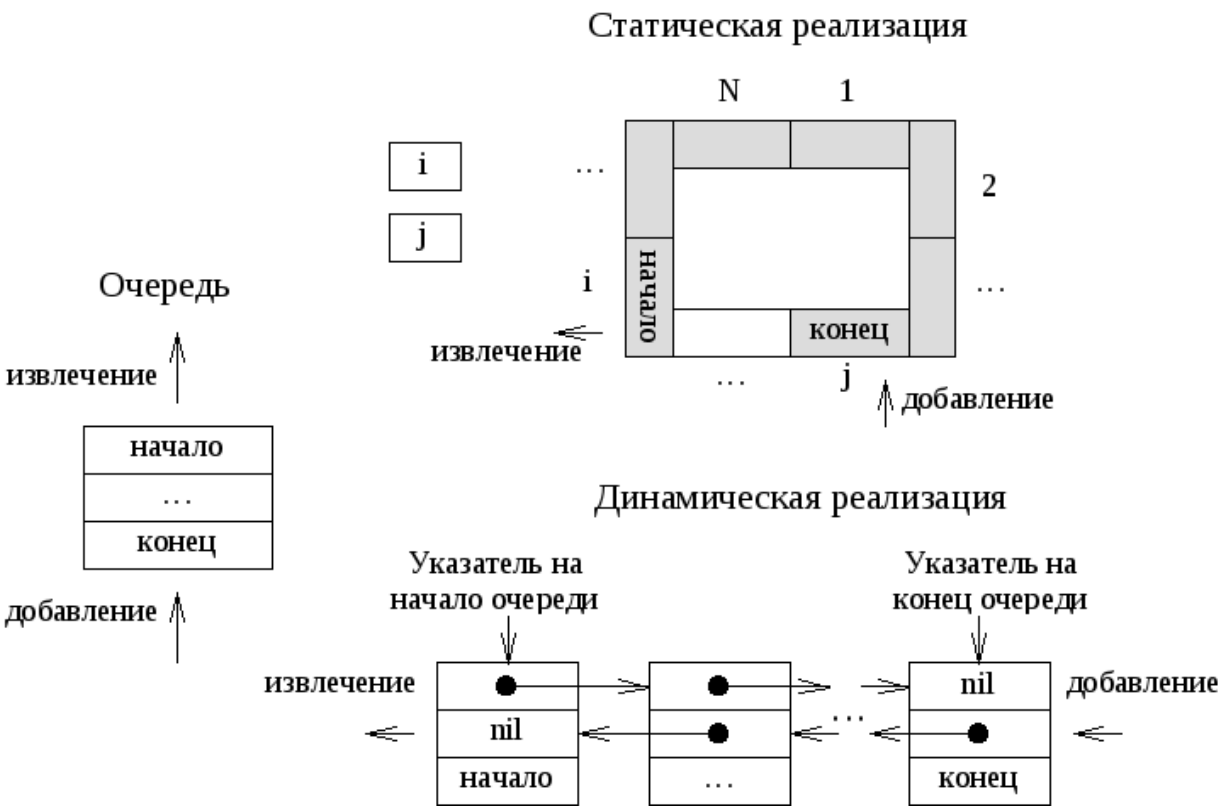


Рисунок 10 реализации очереди

Дек – это структура данных, представляющая собой последовательность элементов, в которой можно добавлять и удалять элементы с двух сторон. Первый и последний элементы дека соответствуют входу и выходу дека.

Данная структура является наиболее **универсальной** из рассмотренных выше линейных структур. Накладывая дополнительные **ограничения** на операции с **началом** и/или **концом** дека, можно осуществлять **моделирование стека** и **очереди**. Дек также можно реализовывать как **статическую структуру** данных в виде одномерного массива, а можно как **динамическую структуру** – в виде линейного списка.

Поскольку в деке, как и в очереди, осуществляется работа с **обоими концами** структуры, то **целесообразно** использовать те же подходы к организации дека, что применялись и для очереди

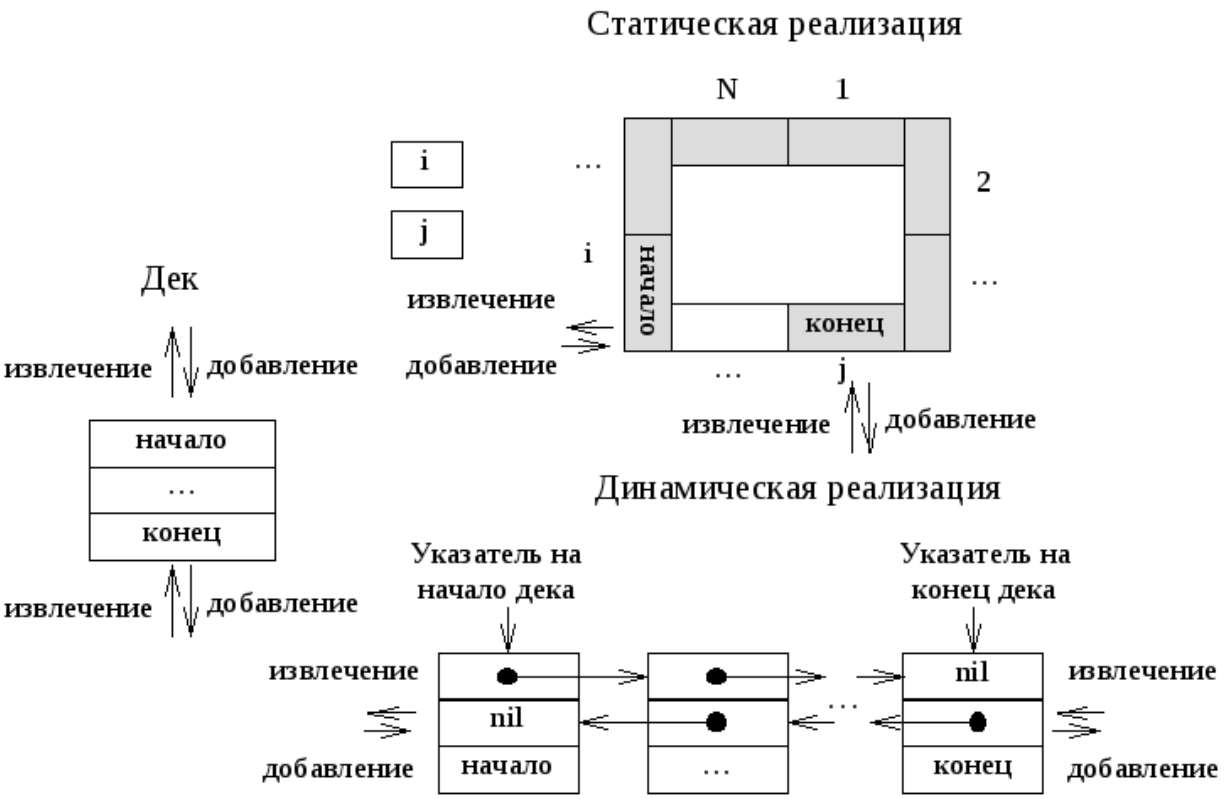


Рисунок 11 реализации дека

Основные операции, производимые с деком:

- добавить элемент в начало;
- добавить элемент в конец;
- извлечь элемент из начала;
- извлечь элемент из конца;
- очистить дек;
- проверка пустоты дека.

https://studfile.net/preview/7278897/page:11/ Стек

https://studfile.net/preview/7278897/page:12/ Очередь

https://studfile.net/preview/7278897/page:12/ Дек

12)Большие данные, определение, качество данных, показатели качества данных (точности, точности, согласованности, полноты, очистка данных.

Организации создают большое количество неструктурированных данных, вся информация хранится в различных хранилищах, компании могут иметь доступ к огромному массиву данных и не иметь необходимых инструментов для связи между данными, традиционные методы анализа не успевают за объемами растущих данных, что приводит к БИГДАТА. Ибо из простого хранения данных нельзя извлечь пользы.

Big data – набор подходов, инструментов, методов обработки больших объёмов разнородной информации.

Большие данные полезны только когда из них можно извлечь ценные для бизнеса. Чтобы анализировать множество файлов или записей Big Data, эти информационные наборы должны обладать не только определенной структурой, но и отвечать следующим характеристикам:

1. **актуальность** – соответствие данных отражают реальному состоянию целевого объекта в текущий период времени;
2. **объективность — т**очность отражения данными реального состояния целевого объекта, которая зависит от методов и процедур сбора информации, а также от плотности регистрируемых данных;
3. **целостность** – полнота отражения данными реального состояния целевого объекта, которая показывает, насколько полны, безошибочны и непротиворечивы данные по смыслу и структуре (формату) с сохранением их правильной идентификации и взаимной связанности;
4. **релевантность** – соответствие данных о реальном состоянии целевого объекта и решаемым задачам, что характеризует возможность их применения с учетом содержания, структуры и формата;
5. **совместимость** – процедурный показатель, который характеризует возможность обрабатывать и анализировать данные в дальнейшем, не только в рамках текущей задачи;
6. **измеримость** – качественные или количественные характеристики реального состояния целевого объекта и конечный объем набора цифровых данных.
7. **управляемость** – возможность целевым и осмысленным образом обработать, передать и контролировать данные о реальном состоянии целевого объекта, на основе структуры и формата датасета;

8. **привязка к источнику данных** – **с**вязанная и достоверная идентификация цепочки поставки данных, например, указание авторства, источника генерации и прочие атрибуты происхождения данных (Data Provenance);
9. **доверие к поставщику данных** – **о**ценка получателем деловых качеств поставщика публичных данных как ответственного, авторитетного, организованного и относительно независимого издателя цифровой информации высокого качества.

Совокупность количественных оценок каждого из этих показателей отражает **качество данных (Data Quality)** – характеристику, показывающую степень их пригодности к обработке и анализу, а также соответствие обязательным и специальным требованиям.

В упрощенном понимании качество данных – это степень их пригодности к использованию. В частности, стандарт ISO 9000:2015 именно так определяет качество данных по степени их удовлетворения требованиям: потребностям или ожиданиям, таким как полнота, достоверность, точность, последовательность, доступность и своевременность.

На практике оценка качества данных сильно зависит от контекста их использования. Например, для крупных маркетинговых кампаний может быть приемлемо до 3-5% дублированных или пропущенных записей, а в случае с медицинскими исследованиями такое недопустимо. Поэтому дисциплина интеллектуального анализа данных (Data Mining) выделяет целых 5 процедур подготовки информационных наборов к использованию в машинном обучении.

Метрики качества данных — это индикаторы, которые можно использовать для оценки качества данных. Они часто предсказывают качество данных и могут дать сигнал к действию. Вы можете применять показатели качества данных в различных секторах, включая здравоохранение, науку, финансы, технологии и страхование. Существует несколько типов метрик качества данных, и каждый из них предназначен для различных целей:

Показатели точности:

Точность данных — это мера того, насколько близко ваши данные соответствуют реальным фактам, которые они представляют. Чем более точными данными вы владеете, тем лучше вы сможете понять эту информацию. К таким типам показателей относятся:

- **Точность:** этот показатель определяет долю фактических элементов данных, которые вы записываете или сохраняете. Например, если у вас есть 100 сотрудников, но вы записываете данные только о 80 сотрудниках, точность составляет 80 %.
- **Отзыв**: Отзыв — это мера того, сколько элементов в списке существует и сколько из них имеет отношение к конкретным видам анализа. Вы можете рассчитать его как отношение между количеством правильных элементов и общим количеством элементов, соответствующих критериям, которые вы установили для сопоставления результатов.

- **Процентиль**: Процентиль — это мера вероятности того, что значение в вашем наборе данных достигнет целевых значений. Например, если 90 % значений данных находятся в пределах одного процентиля выше или ниже целевого значения, вы можете определить, что ваши данные точны.
- **Стандартное отклонение**: стандартное отклонение — это мера того, насколько большие или малые все значения в наборе отличаются от своего среднего или медианного значения.

Показатели целостности:

Метрики целостности помогают оценить целостность ваших данных. Высокая целостность данных позволяет выполнять более точный анализ, включают в себя:

- **Аудит транзакций**: Аудит транзакций измеряет, сколько транзакций не соответствуют всем определенным правилам. Например, если клиент заходит в страховую компанию и заполняет заявку, не указывая номер телефона, но позже получает телефонный звонок от компании, запрашивающей номер, приложение может иметь ошибку в процессе ввода данных.
- **Нулевые значения**: Нулевые значения показывают, сколько значений в вашем наборе данных являются неполными или пустыми. Эта оценка может помочь вам определить области вашего процесса, где вы можете предоставить дополнительные рекомендации и обучение, чтобы гарантировать, что значения, которые вы собираете и анализируете, являются полными и точными.
- **Отсутствующие точки данных**. Отсутствующие точки данных показывают, сколько отдельных случаев или записей не содержат всех необходимых элементов данных для определенного атрибута.

Показатели согласованности:

Метрики согласованности помогают оценить, согласуются ли значения в вашем наборе данных со значениями, которые вы ранее записали и сохранили.

- **Согласованность данных**. Согласованность дат измеряет, сколько дат в наборе данных выходит за пределы их исторического диапазона.
- **Числовая согласованность**. Числовая согласованность измеряет, сколько значений в наборе данных отличается от ожидаемого диапазона.

Показатели полноты:

. Полнота данных — это мера того, насколько данный набор данных является полным, точным и репрезентативным, включают:

- **Минимальное количество вхождений**. Минимальное количество вхождений измеряет, сколько значений в наборе данных имеют меньше заданного количества вхождений.
- **Максимальное время задержки**: Максимальное время задержки измеряет время между возникновением события и его записью в вашей системе.

Очистка данных:

Очистка данных возможна далеко не всегда и определяется конкретным этапом управления качеством данных. Для **первичных данных** эффективны системы ввода с проверкой и контролем ошибок, а также двухэтапный аудит: сначала выборочный аудит отдельных записей, а потом аудит аудита. Так сейчас работают, например, агрегаторы клинических данных. Хороший пример — Единая радиологическая информационная система (ЕРИС) с соответствующим контролем, предусматривающим аудит с проверкой протоколов и обучением персонала в первичном звене. **Однако** при очистке данных, возникающих дальше по цепочке, могут возникнуть проблемы: например, требуется обращение ко всем элементам данных, а значит, встает вопрос прав доступа к ним. **Особенно сложна ситуация с персональными данными**: не создано условий для официального использования защищенных хранилищ персональных данных для совместного использования несколькими организациями (даже на платформе ЕГАИС) и вероятность получить доступ к такому хранилищу для гармонизации персональных данных весьма мала.

https://habr.com/ru/post/321406/ Качество данных

https://buom.ru/chto-takoe-pokazateli-kachestva-dannyh-plyus-tipy-i-ispolzovanie/ Показатели качества

https://www.osp.ru/os/2020/01/13055348 Очистка данных

13)Базовые стандарты, термины.

Для исключения разночтений, при сборе простых характеристик нужной оценки качества данных применяются стандарты, начиная с ГОСТа 56215, регламентирующие понимание и применение терминов.

Стандарты группы ISO/TS 8000.

Принципы стандартов 8000:

1) Качество применимо к данным, имеющим определённое значение, учитывающихся при принятии какого-либо решения.
2) Качество данных затрагивает нужные и подходящие данные, уместные в подходящем месте в подходящее время.
3) Качество данных отвечает требованиям потребителя.
4) Качество данных предотвращает повторение дефектов данных и сокращает избыточные ресурсы.

Международные стандарты группы ISO 25000 - это три главных стандарта: 25010, 25012, 25040.

Они определяют качество программного продукта с акцентом на общей модели качества данных, представленных в структурированном виде для информационной системы, а также критерия качества "продукта данных" как специального вида программного продукта.

Термины:

Полнота и завершенность данных - качество всех имеющихся у пользователя данных, которыми он владеет на определённый момент. Полнота характеризует подтвержденную достаточность данных для достижения конкретной цели.

Авторитетный источник данных - владелец процесса, производящего данные.

Утвержденное эталонное значение - значение, применяемое в качестве согласованной ссылки при сравнении данных(реестра).

Истинное значение - значение параметров характеристики какого-либо объекта в определённых условиях.

14)Управление данными, стратегия управления качеством данных (принципы, действия: учитывать, предусматривать…), циклический конвейер управления качеством данных и его процессы, инструменты управления качеством данных.

Стратегия управления качеством данных должна:

- Учитывать политику управления данными.
- Предусматривать выполнение мероприятий по профилированию данных, их сопоставлению и гармонизации.
- Включать подготовку отчётов о качестве данных и систему управления мастер-данными.
- Предусматривать средства интеграции данных, например, о клиентах.
- Описывать интерфейсы к подсистемам управления информацией о продукте и управлению цифровыми активами.

Циклический конвейер управления качеством данных может включать следующие процессы:

- Анализ бизнес-задачи.
- Сбор метаданных и их гармонизация с мастер-данными и разделяемыми справочниками.
- Создания модели данных и описание их потоков (архитектура данных).
- Профилирование данных - сбор характеристик данных, их исследование.
- Разработка показателей, тестов и метрик.
- Контроль, мониторинг и анализ качества.
- Разработка процедур контроля и расчёта метрик.
- Устранение инцидентов, очистка данных.

- Раскрытие сведений о качестве данных.
- Анализ инцидентов, обратная связь с поставщиками данных.

Профилирование данных заключается в сборе характеристик данных и их исследовании: необходимо оценить распределение величин, выбросы, параметры выборки, пропущенные значения, нарушение целостности, несоответствие бизнес-задаче и пр.

Метрики непосредственно зависят от конкретной задачи.

Потребитель дюнных формулирует требования к их качеству, например: доля бракованных записей, отклонение от целевого значения, отклонение от исторического средства, базовое значение, агрегирование значений и пр.

Контроль и мониторинг качества выполняет эксперт по качеству данных, в идеале в одном лице совмещающий три вида экспертизы (специалист по статистике, эксперт в конкретной предметной области, специалист по методологии анализа и обработки данных).

В стратегии должен быть предусмотрен постоянный контроль данных, но это сложно и дорого, поэтому используются выборочные проверки.

Как правило, большинство специализированных систем управления качеством данных автоматизируют следующие процессы:

- Профилирование - первоначальная оценка данных, чтобы понять их текущее состояние, в том числе распределение значений.
- Стандартизация - механизм бизнес-правил, обеспечивающий соответствие данных стандартам.
- Геокодирование адресов, которое корректирует данные в соответствии с Географическими стандартами.
- Сопоставление или связывание - способ сравнения данных для выявления одинаковых по смыслу, на разных по виду представления записей. Сопоставление может использовать нечеткую логику для поиска дубликатов в данных.
- Мониторинг - отслеживание качества данных с течением времени и отчетность об изменениях.
- Пакетная и потоковая обработка - непрерывная очистка данных в пакетном режиме с последующей интеграцией в корпоративные приложения.

15)DevOps (development & operations) как методология автоматизации технологических процессов сборки, настройки и развёртывания ПО.

DevOps — методология автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения. Методология предполагает активное взаимодействие специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимную интеграцию их технологических процессов друг в друга для обеспечения высокого качества программного продукта. Предназначена для эффективной организации создания и обновления программных продуктов и услуг. Основана на идее тесной взаимозависимости создания продукта и эксплуатации программного обеспечения, которая прививается команде как культура создания продукта.

Методология фокусируется на стандартизации окружений разработки с целью быстрого переноса программного обеспечения через стадии жизненного цикла ПО, способствуя быстрому выпуску версий программного продукта.

Задача инженеров автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения (DevOps engineers) — сделать процессы разработки и поставки программного обеспечения согласованным с эксплуатацией, объединив их в единое целое с помощью инструментов автоматизации.

Поскольку DevOps — это командная работа (между сотрудниками, занимающимися разработкой, операциями и тестированием), нет единого инструмента «DevOps»: это скорее набор (или «инструментальная цепочка DevOps»), состоящий из нескольких инструментов. Как правило, инструменты DevOps вписываются в одну или несколько из этих категорий, что отражает ключевые аспекты разработки и доставки программного обеспечения:

1. Кодирование — разработка и анализ кода, инструменты контроля версий, слияние кода;

2. Сборка — инструменты непрерывной интеграции, статус сборки;

3. Тестирование — инструменты непрерывного тестирования, обеспечивающие быструю и своевременную оценку бизнес-рисков;

4. Упаковка — репозиторий артефактов, предварительная установка приложения;

5. Релиз — управление изменениями, официальное утверждение выпуска, автоматизация выпуска;

6. Настройка — конфигурация и управление инфраструктурой, Инфраструктура как инструменты кода;

7. Мониторинг — измерение производительности приложений, взаимодействие с конечным пользователем;

8. Непрерывная поставка;

9. Непрерывная интеграция.

16)Классификация данных (мастер-данные, ...), особенности работы с мастер-данными, метаданные, проблемы SOA, взаимодействие MDM и SOA на уровне сервисов данных (категории).

С точки зрения управления первичные данные обычно делят на 4 класса:

1. Мастер-данные(master-data) – определяют ключевые, представляющие особую ценность для организации или бизнеса и относительно редко изменяемые сущности.
2. Разделяемые справочники(reference data) – систематизируют и классифицируют другие данные, а также связывают между собой данные различных организаций. Сегодня на эту роль претендуют открытые данные(open data) любых уровней – от федеральных до местных.
3. Оперативные(транзакционные) данные - это данные, которые образовались в результаты выполнения предприятием каких-либо бизнес-транзакций. Например, для коммерческого предприятия: продажи продуктов и услуг, закупки, поступления/списания денежных средств, поступления на склад и т.п
4. Исторические данные(historical data) – образованы из прошлых версий мастер-данных, разделяемых справочников и транзакционных данных, возникающих после завершения соответствующих бизнес-процессов.

Еще одна классификация из презентаций:

1. Частные данные – данные принадлежащие одному человеку, ни с кем не разделяются, никто иной, кроме него самого, не осуществляет управление этими данными.
2. Распределяемые данные – данные рабочих групп или предприятия в целом.
3. Внутренние бизнес-данные – частные или распределенные данные, создаваемые и поддерживаемые работниками компании или внутренними приложениями.
4. Внешние бизнес-данные – данные, поступающие от внешних организаций и приложений.

И ещё одна:

- Структурированные данные – наборы записей с фиксированной структурой полей, определенной внешней моделью данных.
- Квазиструктурированные данные – наборы записей с переменной структурой полей, определенной внешней моделью данных.
- Неструктурированные данные – наборы данных, не имеющие внешнего описания структуры полей.
- Текущие данные – последняя версия записи.

- Исторические данные – записи предыдущих версий.

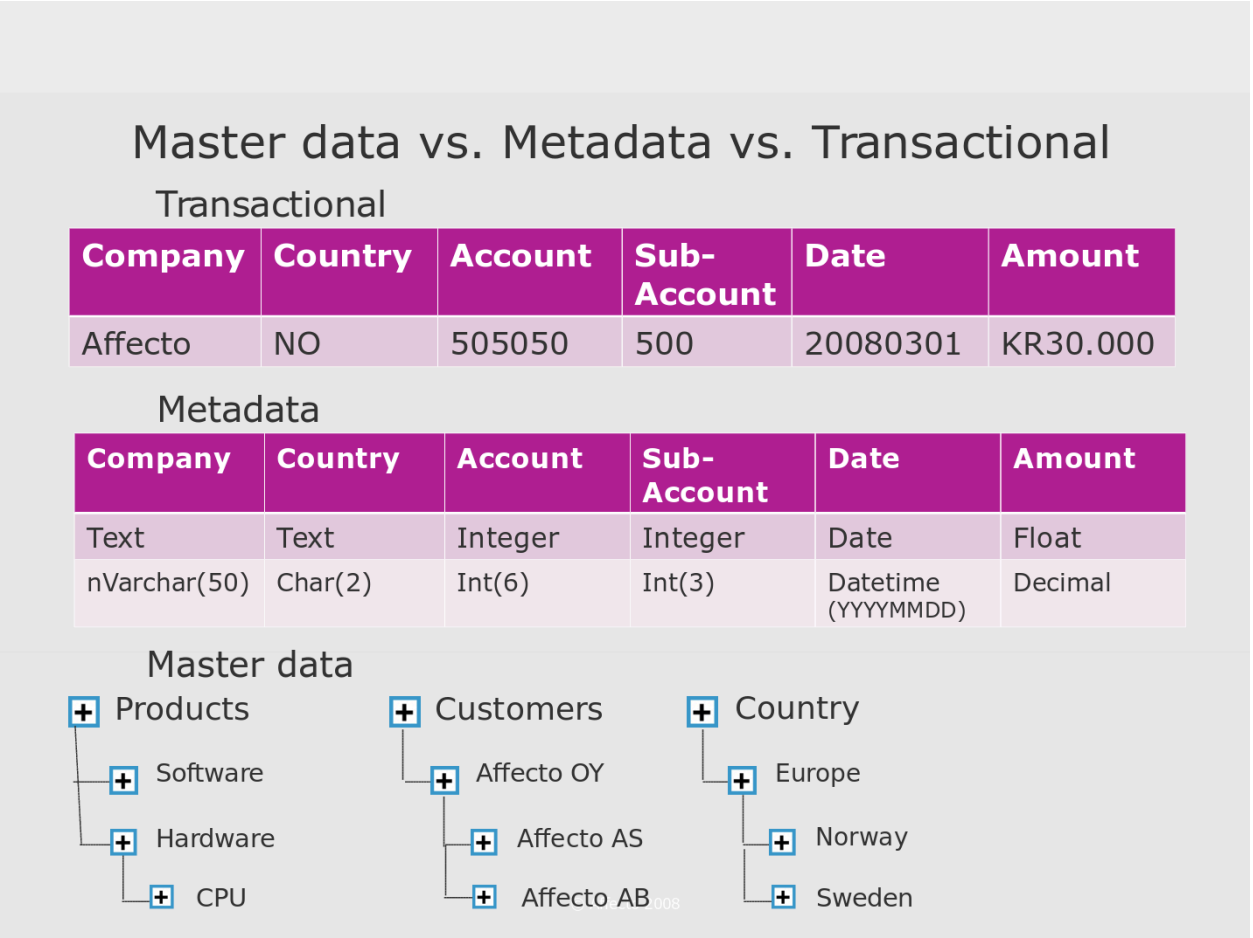
Другие определения мастер данных(в России нормативно справочная информация(НСИ)):

- Мастер-данные – это важнейшая для бизнеса информация о клиентах, продуктах, услугах, персонале, технологиях, материалах, и прочих доменных объектах, которые редко изменяются и не являются транзакционными.
- Мастер данные – это данные, содержащие ключевую информацию о бизнесе, в том числе о клиентах, о продуктах, о работниках, о технологиях и материалах. В определенных случаях мастер-данные поддерживают транзакционные процессы и операции, но в большей степени они используются для аналитической деятельности и подготовки отчетов.

Особенности работы с мастер данными:

- Проблема дубликатов(сокращенное и полное название компании)
- Мастер данные отличаюстя от обычных транзакционных данных меньшей изменяемостью и меньшими объемами.(перечни материалов не меняются ежесекундно)
- Мастер данные структурно намного сложнее, чем операционные данные.(представления о клиенте – не мат. модель, как следствие в разных подразделениях разные сведения)
- Модель мастер-данных меняется гораздл чаще, чем операционных данных(следовательно главная сложность не в них самих, а в описывающих их метаданных)

Метаданные – это данные о данных (об их составе, содержании, статусе, происхождении, местонахождении, качестве, форматах, объёме, условиях доступа, авторских правах и тп)



Главная цель управления мастер-данными – это гарантировать отсутствие пропущенных, повторяющихся, неполных и противоречивых записей об объектах бизнес-домена во всех корпоративных информационных системах.

Master Data Management – IT-дисциплина, которая включает целый ряд структурированных подходов, процессов и инструментов по эффективному управлению НСИ.

Сервис-ориентированная архитектура (SOA) – это метод разработки программного обеспечения, который использует программные компоненты, называемые сервисами, для создания бизнес-приложений. Каждый сервис предоставляет бизнес-возможности, и сервисы также могут взаимодействовать друг с другом на разных платформах и языках. Разработчики применяют SOA для многократного использования сервисов в различных системах или объединения нескольких независимых сервисов для выполнения сложных задач.

MDM означает управление основными данными, которое является ключевым компонентом сервис-ориентированной архитектуры (SOA) и упрощает внедрение SOA с чистыми и согласованными данными для предприятия. Основное намерение MDM – обеспечить надежный источник основных данных.

SOA вызывает проблемы в том случае, если сервисы обращаются к общим массивам данных, возникает необходимость в организации распределенного доступа к данным.

Категории архитектур MDM:

- Master Registry – подход к MDM предполагает наличие центрального реестра идентификаторов и ссылок на приложения, владеющие данными, попадающими в категорию мастер данных.
- Master Repository – в этом случае создается основной репозиторий MDM, содержащий как идентификаторы и ссылки, так и собственно данные.
- Master Hub – концентратор MDM лишает приложения права владения мастер-данными и создает собственный ресурс идентификаторов и транзакционных данных.

17)Управление мастер-данными, архитектуры управления мастер-данными, наблюдаемость данных, ключевые компоненты, инструментальные средства

Управление мастер-данными (Master Data Management, MDM) – дисциплина, которая работает с мастер-данными в целях создания «золотой записи», то есть целостного и всестороннего представления о мастер-сущности и взаимосвязях, эталона мастер-данных, который используются всем предприятием, а иногда и между предприятиями для упрощения обмена информацией.

Специализированные системы управления мастер данными (MDM-системы) автоматизируют все аспекты этого процесса и являются «авторитетным» источником мастер-данных масштаба предприятия. Часто MDM-системы управляют также и референс-данными.

Категории архитектур MDM:

- Master Registry – подход к MDM предполагает наличие центрального реестра идентификаторов и ссылок на приложения, владеющие данными, попадающими в категорию мастер данных.
- Master Repository – в этом случае создается основной репозиторий MDM, содержащий как идентификаторы и ссылки, так и собственно данные.
- Master Hub – концентратор MDM лишает приложения права владения мастер-данными и создает собственный ресурс идентификаторов и транзакционных данных.

Весь комплекс деятельности, связанной с созданием MDM, разделяется на две части — на поддержку работы с текущими данными и на поддержку аналитической работы с историческими данными. В свою очередь к первой части можно применить три подхода, создавая либо федеративную, либо интегрированную, либо гибридную систему MDM.

Рисунок 12 - федеративный

При федеративном подходе задача состоит в передаче и синхронизации изменений между элементами системы входа, с тем, чтобы обеспечить согласованность мастер-данных. Этот подход реализуется посредством концентратора мастер-данных (master data hub), который в соответствии с установленными правилами осуществляет асинхронный обмен данными. Основной предмет деятельности заключается в формулировании этих правил. Преимуществом этого подхода является невмешательство в действия существующих систем и сохранение контекста, а недостатком — то, что он эффективен лишь при относительно небольшом разнообразии мастер-данных и не выдерживает нагрузки при усложнении среды мастер-данных.

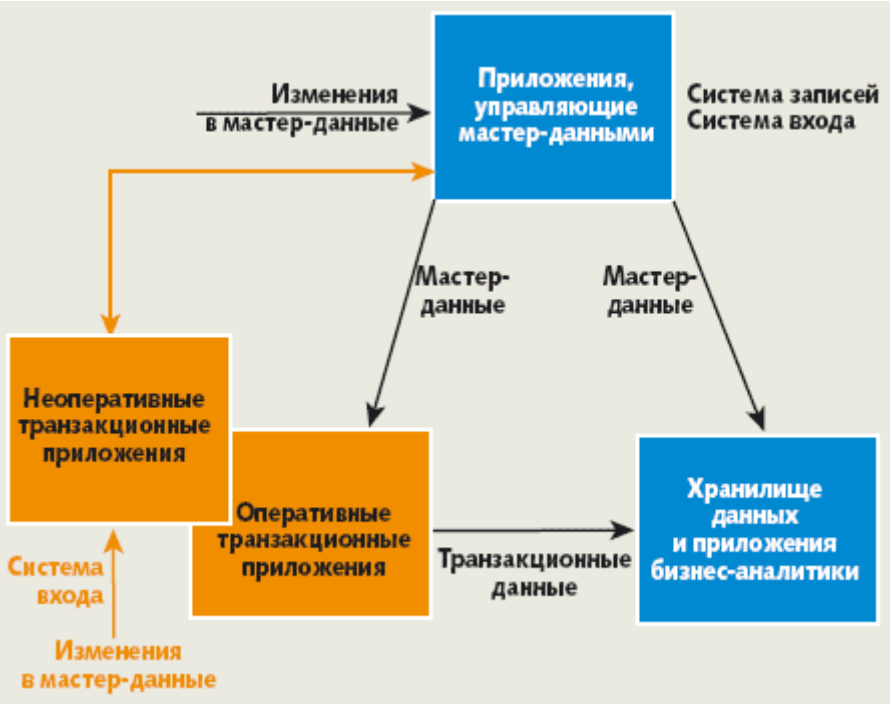


Рисунок 13 - интегрированное решение

заключается в создании автономной системы MDM, способной самостоятельно собирать изменения в мастер-данных. В таком случае все эти данные хранятся в централизованном репозитории. Данный подход обеспечивает наиболее полное и адекватное представление мастер-данных.

Гибридный подход является промежуточным между первым и вторым, его можно рассматривать как промежуточный этап развития от первого ко второму.

Наблюдаемость данных — это возможность понять работоспособность данных и систем данных, собирая и коррелируя события в таких областях, как данные, хранилище, вычисления и обработка конвейеров.

Ключевые компоненты

- Мониторинг службы платформы данных
- Мониторинг производительности конвейера данных
- Мониторинг качества данных
- Журнал преобразований данных
- Обнаружение данных

(ПРО СРЕДСТВА НИЧЕГО)

Вот ссылки на статьи которые она использовала:

https://www.osp.ru/os/2007/05/4260254 тут про виды данных, MDM, MDM и SOA

https://learn.microsoft.com/ru-ru/azure/cloud-adoption-framework/scenarios/cloud-scale-analytics/manage-observability Тут про наблюдаемость данных и компоненты, возможно средства это Azure

https://habr.com/ru/company/navicon/blog/260927/

https://habr.com/ru/post/324148/ эти про типы данных и MDM, SOA есть только в первой ссылке

18)Извлечение и получение данных, Data Mining, Этапы, Трансформация данных в процессе, преимущества и недостатки.

Извлечение данных - процесс получения данных из источника

Получение данных - это процесс идентификации и извлечения данных из базы данных на основе запроса, предоставленного пользователем или приложением.

Data mining, также называемый Обнаружение знаний в базе данных (KDD), представляет собой метод, часто используемый для анализа больших массивов данных с помощью статистических и математических методов для поиска скрытых закономерностей или тенденций и извлечения из них ценности.

Data mining — это целостный процесс сбора, отбора, очистки, преобразования и извлечения данных для оценки закономерностей и, в конечном итоге, для извлечения ценности.

Этапы:

1. Очистка данных

   В реальном мире данные не всегда очищаются и структурируются. Часто они шумные, неполные и могут содержать ошибки. Чтобы удостовериться, что результат data mining точный, сначала необходимо очистить данные. Некоторые методы очистки включают заполнение недостающих значений, автоматический и ручной контроль и т.д.

2. Интеграция данных

   Это этап, на котором данные из разных источников извлекаются, комбинируются и интегрируются. Источниками могут быть базы данных, текстовые файлы, электронные таблицы, документы, многомерные массивы данных, интернет и так далее.

3. Выборка данных

   Обычно не все интегрированные данные необходимы в data mining. Выборка данных — это этап, в котором из большой базы данных выбираются и извлекаются только полезные данные.

4. Преобразование данных

   После выбора данных они преобразуются в подходящие для добычи формы. Этот процесс включает в себя нормализацию, агрегирование, обобщение и т.д.

5. Интеллектуальный анализ данных

   Здесь наступает самая важная часть data mining — использование интеллектуальных методов для поиска закономерностей в них. Процесс

включает регрессию, классификацию, прогнозирование, кластеризацию, изучение ассоциаций и многое другое.

6. Оценка модели

Этот этап направлен на выявление потенциально полезных, простых в понимании шаблонов, а также шаблонов, подтверждающих гипотезы.

7. Представление знаний

На заключительном этапе полученная информация представлена в привлекательном виде с применением методов представления знаний и визуализации.

Трансформация данных - это важный метод предварительной обработки данных, который должен быть выполнен перед анализом данных для получения более понятных закономерностей. Преобразование данных изменяет формат, структуру или значения данных и преобразует их в чистые, пригодные для использования данные.

Преимущества:

- Помогает компаниям собирать достоверную информацию.
- Помогает предприятиям вносить выгодные производственные и операционные коррективы.
- При поиске данных используются как новые, так и устаревшие системы.
- Помогает предприятиям принимать обоснованные решения.

Недостатки:

- Большие вложения времени и труда
- Инструменты добычи данных сложны и требуют обучения для использования
- Методы добычи данных не являются безошибочными
- Приватность и безопасность данных
- Для добычи данных требуются большие базы данных

19)Data Extraction, категории, этапы, преимущества и недостатки, различия между Data Mining и Data Extraction.

Извлечение данных — это акт или процесс извлечения данных из (обычно неструктурированных или плохо структурированных) источников данных для дальнейшей обработки данных или хранения данных (миграции данных). Таким образом, импорт в промежуточную систему извлечения обычно сопровождается преобразованием данных и, возможно, добавлением метаданных перед экспортом на другой этап рабочего процесса данных.

Обычно термин извлечение данных применяется, когда экспериментальны данные впервые импортируются в компьютер из первичных источников, таких как измерительные или записывающие устройства.

Также известное как «извлечение веб-данных» и «веб-скрепинг», этот процесс представляет собой акт извлечения данных из (обычно неструктурированных или плохо структурированных) источников данных в централизованные места и централизацию в одном месте для хранения или дальнейшей обработки. В частности, к неструктурированным источникам данных относятся веб-страницы, электронная почта, документы, файлы PDF, отсканированный текст, отчеты мейнфреймов, катушечные файлы, объявления и т.д. Централизованные хранилища могут быть локальными, облачными или гибридными. Важно помнить, что извлечение данных не включает в себя обработку или другой анализ, который может произойти позже.

Что можно сделать с помощью Data Extraction?

В основном цели извлечения данных делятся на 3 категории.

- **Архивация**
  Извлечение данных может преобразовать данные из физических форматов: книг, газет, счетов-фактур в цифровые форматы, например, базы данных для хранения или резервного копирования.
- **Изменение формата данных**
  Когда вы хотите перенести данные с вашего текущего сайта на новый, находящийся в стадии разработки, вы можете собрать данные с вашего собственного сайта, извлекая их.
- **Анализ данных**
  Распространен дополнительный анализ извлеченных данных для получения представления о них. Это может показаться похожим на анализ данных при data mining, но учтите, что анализ данных — это цель их извлечения, но не его часть. Более того, данные анализируются иначе. Один из примеров: владельцы интернет-магазинов извлекают информацию о продукте с сайтов электронной коммерции, таких как Amazon, для мониторинга стратегий конкурентов в режиме реального времени. Как и data mining, data extraction —

это автоматизированный процесс, имеющий множество преимуществ. Раньше люди копировали и вставляли данные вручную из одного места в другое, что занимало очень много времени. Извлечение данных ускоряет сбор и значительно повышает точность извлекаемых данных.

Применение:

Подобно data mining, извлечение данных широко используется в различных отраслях промышленности. Помимо мониторинга цен в электронной коммерции, извлечение данных может помочь в собственном исследовании, агрегировании новостей, маркетинге, в работе с недвижимостью, путешествиях и туризме, в консалтинге, финансах и во многом другом.

- **Лидогенерация**
  Компании могут извлекать данные из каталогов: Yelp, Crunchbase, Yellowpages и генерировать лидов для развития бизнеса. Вы можете посмотреть видео ниже, чтобы узнать, как извлечь данные из Yellowpages с помощью шаблона веб-скрепинга.
- **Агрегация контента и новостей**
  Агрегирующие контент веб-сайты могут получать регулярные потоки данных из нескольких источников и поддерживать свои сайты в актуальном состоянии.
- **Анализ настроений**
  После извлечения обзоров, комментариев и отзывов из социальных сетей, таких как Instagram и Twitter, специалисты могут проанализировать лежащие в их основе взгляды и получить представление о том, как воспринимается бренд, продукт или некое явление.

Этапы:

1. **Выбор источника данных**
   Выберите источник, данные из которого вы хотите извлечь, например, веб-сайт.
2. **Сбор данных**
3. **Хранение данных**
   Сохраните данные в своей локальной базе данных или в облачном хранилище для будущего использования.

Недостатки Data Extraction

- **Сбой сервера**
  При извлечении данных в больших масштабах веб-сервер целевого сайта может быть перегружен, что может привести к поломке сервера. Это нанесет ущерб интересам владельца сайта.
- **Бан по IP**
  Когда человек слишком часто собирает данные, веб-сайты могут заблокировать его IP-адрес. Ресурс может полностью запретить IP-адрес или ограничить доступ, сделав данные неполными. Чтобы извлекать данные и избегать блокировки, нужно делать это с умеренной скоростью и применять некоторые методы антиблокировки.

- **Проблемы с законом**
  Извлечение данных из веба попадает в серую зону, когда дело касается законности. Крупные сайты, такие как Linkedin и Facebook, четко заявляют в своих условиях использования, что любое автоматическое извлечение данных запрещено. Между компаниями было много судебных исков из-за деятельности ботов.

Ключевые различия между Data Mining и Data Extraction

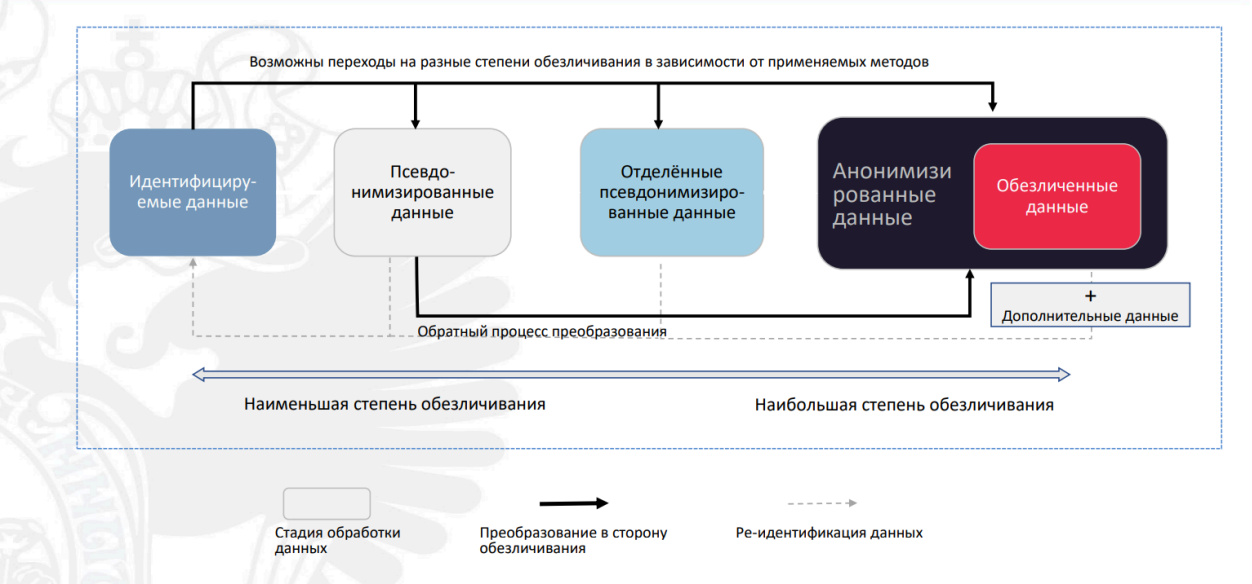
1. Data mining также называется обнаружением знаний в базах данных, извлечением знаний, анализом данных/шаблонов, сбором информации. Data extraction используется взаимозаменяемо с извлечением веб-данных, сканированием веб-страниц, сбором данных и так далее.
2. Исследования data mining в основном основаны на структурированных данных, тогда как при извлечении данных они обычно извлекаются из неструктурированных или плохо структурированных источников.
3. Цель data mining — сделать данные более полезными для анализа. Data extraction — это сбор данных в одно место, где они могут быть сохранены или обработаны.
4. Анализ при data mining основан на математических методах выявления закономерностей или тенденций. Data extraction базируется на языках программирования или инструментах извлечения данных для обхода источников.
5. Цель data mining — найти факты, которые ранее не были известны или игнорировались, тогда как data extraction имеет дело с существующей информацией.
6. Data mining сложнее и требует больших вложений в обучение людей. Data extraction при использовании подходящего инструмента может быть чрезвычайно простым и экономичным.

20)Обработка персональных данных, обезличивание и анонимизация. Определение и общий пример полного цикла базового подхода к обработки персональных данных.

Обработка персональных данных - любое действие (операция) или совокупность действий (операций), совершаемых с использованием средств автоматизации или без использования таких средств с персональными данными, включая сбор, запись, систематизацию, накопление, хранение, уточнение (обновление, изменение), извлечение, использование, передачу (распространение, предоставление, доступ), обезличивание, блокирование, удаление, уничтожение персональных данных. (подп.3) ст.3 ФЗ «О персональных данных»)

Обезличивание - действия, в результате которых становится невозможным без использования дополнительной информации определить принадлежность персональных данных конкретному субъекту персональных данных

Анонимизация (необратимое обезличивание) - действия, в результате которых персональные данные необратимо изменяются таким образом, что субъект персональных данных больше не может быть идентифицирован прямо или косвенно, в т. ч. и с использованием дополнительной информации, разумно доступной для анализа с использованием ограниченных ресурсов и за ограниченное время




БАЗОВЫЙ ПОДХОД
Атаки
Риск
Риск
Персональные Данные (содержащие идентифицирующие атрибуты - PII)
Псевдонимизированные данные
Анонимные данные
Полезность информации
Полезность информации
Измерение уровня анонимизации
Измерение уровня анонимизации
ПРОЦЕСС ОБЕЗЛИЧИВАНИЯ
ПСЕВДОНИМИЗАЦИЯ
Дополнительная информация
ПРОЦЕСС АНОНИМИЗАЦИИ
13

21)Идентификаторы. Примеры прямых идентификаторов, квази-идентификаторов. Чувствительные данные

Прямой идентификатор - атрибут персональных данных, который позволяет идентифицировать субъекта персональных данных без привлечения дополнительной информации, других атрибутов

Косвенный идентификатор - атрибут персональных данных, который позволяет идентифицировать субъекта персональных данных только при привлечении дополнительной информации

Чувствительные данные - биометрические данные, данные о здоровье

22)Основные принципы обезличивания данных. Свойства персональных данных и методов обезличивания.

ОСНОВНЫЕ ПРИНЦИПЫ

- Обезличивание персональных данных должно быть встроено в общую систему управления данными (Privacy By Design)
- Результаты обезличивания должны быть измеряемыми (полезность и риски)
- Обезличивание, как мера защиты информации должна включать в себя не только технические, но и организационные меры
- Данные должны быть категорированы по степени чувствительности, для разных категорий – разные методы
- Применение обезличивания должно быть ориентировано на практическое использование и быть корреспондировано с распространенными архитектурными шаблонами
- Применяемые методы обезличивания не должны нарушать основные атрибуты персональных данных

СВОЙСТВА ПЕРСОНАЛЬНЫХ ДАННЫХ И МЕТОДОВ ОБЕЗЛИЧИВАНИЯ

Свойства, которые должны сохраняться после обезличивания должны учитывать вопросы качества данных (Методические рекомендации по применению приказа Роскомнадзора от 5 сентября 2013 года N 996 "Об утверждении требований и методов по обезличиванию персональных данных"):

1. полнота (сохранение всей информации о конкретных субъектах или группах субъектов, которая имелась до обезличивания);
2. структурированность (сохранение структурных связей между обезличенными данными конкретного субъекта или группы субъектов, соответствующих связям, имеющимся до обезличивания);
3. семантическая целостность (сохранение семантики персональных данных при их обезличивании);
4. анонимность (невозможность однозначной идентификации субъектов данных, полученных в результате обезличивания);

Методы обезличивания также обладают рядом свойств:

- вариативность (возможность внесения изменений в параметры метода);
- стойкость (стойкость метода к атакам на идентификацию субъекта персональных данных);
- возможность оценки качества данных (возможность проведения контроля качества обезличенных данных и соответствия применяемых процедур обезличивания установленным для них требованиям).

23)Классификация методов обезличивания данных. Таксономия Обезличенных данных.

Предлагается следующая **классификация** (таксономия) в рамках РЕКОМЕНДАЦИЙ:

- Техники анонимизации, разбитые на категории: статистические методы, криптографические методы, методы подавления, методы псевдонимизации, техники обобщения, техники рандомизации, синтетические данные
- Модели приватности (метрики анонимизации)
- Модели дифференциальной приватности
- Модели оценки рисков

ФРЕЙМВОРК ОБЕЗЛИЧИВАНИЯ

ТАКСОНОМИЯ ОБЕЗЛИЧИВАНИЯ ДАННЫХ

ТЕХНИКИ ОБЕЗЛИЧИВАНИЯ

СТАТИСТИЧЕСКИЕ МЕТОДЫ
- Семплинг (sampling)
- Агрегация (aggregation)

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ
- Deterministic encryption
- Order-preserving encryption
- Format-preserving encryption
- Homomorphic encryption
- Homomorphic secret sharing

МЕТОДЫ ПОДАВЛЕНИЯ
- Local suppression
- Локальное подавление
- Record suppression

МЕТОДЫ ПСЕВДОНИМИЗАЦИИ
- Selection of attributes
- Creation of pseudonyms

АНАТОМИЗАЦИЯ

ТЕХНИКИ ОБОБЩЕНИЯ
- Rounding
- Top and bottom coding
- Combining a set of attributes into a single attribute
- Local generalization

ТЕХНИКА РАНДОМИЗАЦИИ
- Noise addition
- Microaggregation
- Permutation

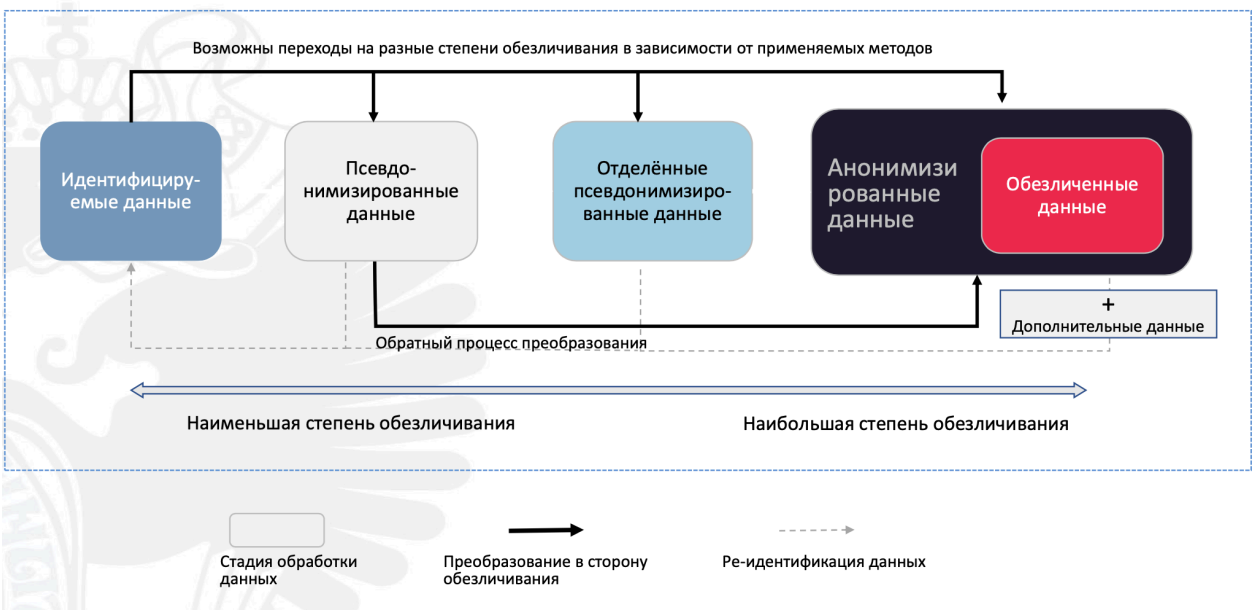
СИНТЕТИЧЕСКИЕ ДАННЫЕ

МОДЕЛИ ПРИВАТНОСТИ
- k-anonymity
- T-closeness
- L-diversity
- β-likeness
- δ-precense
- δ-disclosure

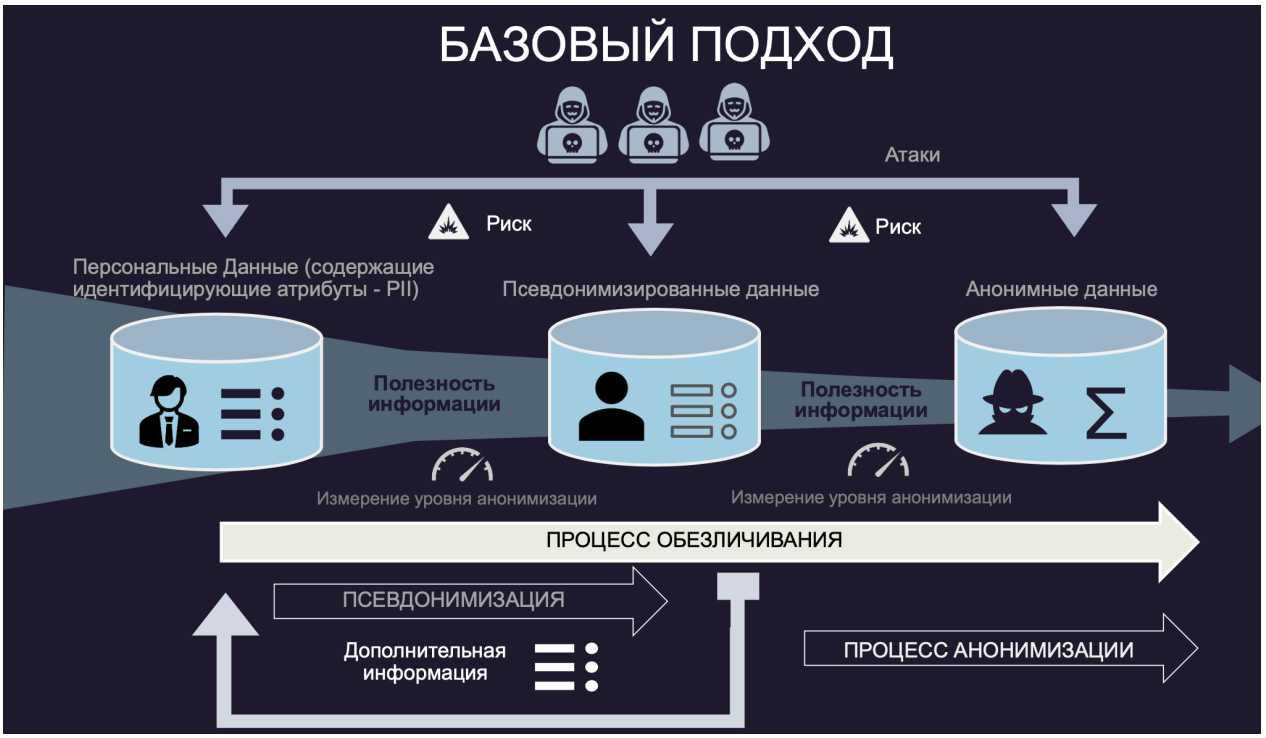
МОДЕЛИ ДИФФЕРЕНЦИАЛЬНОЙ ПРИВАТНОСТИ
- Server mode
- Local model
- Key considerations for a Differentially Private System

МЕТРИКИ ПОЛЕЗНОСТИ
- *Discernibility (различимость)*
- *Classification (классификация)*
- *Ambiguity (неоднозначность)*
- *Entropy (энтропия)*
- *Normalized average equivalence class size (размер класса эквивалентности)*
- *Domain generalization hierarchy distance (расстояние обобщения)*

24)Процесс обезличивания. Количественный и качественный подход к обезличиванию. Полезность, риск и анонимность после обезличивания персональных данных.



Ре-идентификация данных - повторная идентификация данных или деанонимизация-это **практика сопоставления анонимных данных (также известных как деидентифицированные данные) с общедоступной информацией или вспомогательными данными с целью выявления лица, которому принадлежат эти данные**.



ПОЛЕЗНОСТЬ, РИСК И АНОНИМНОСТЬ

Снижение риска за счет использования методов, приводящих к потере информации, известно как компромисс между риском и полезностью. Это можно сформулировать как задачу оптимизации.
На практике различные методы можно оценивать и сравнивать, рассматривая значения альтернативных показателей риска и полезности. Иногда может быть полезно построить карту «риск-полезность», например, как на диаграмме.

Поскольку как оценка риска раскрытия информации, так и оценки полезности данных могут включать в себя некоторые субъективные решения, структура «риск-полезность» может помочь в выборе оптимальной стратегии обезличивания.

25)Примеры обезличивания и деобезличивания персональных данных. Сценарии распространения данных. Техники обезличивания.

ПРИМЕР ОБЕЗЛИЧИВАНИЯ И ДЕОБЕЗЛИЧИВАНИЯ

Идентификатор | Квази - идентификатор | Конфиденциальные атрибуты

ФИО	Возраст	Семейное положение	Почтовый индекс	Доход	Сексуальная ориентация
Сергей Петров	32	⚭	198000	450 000 ₽	⚥
Дмитрий Иванов	34	○	198023	350 000 ₽	⚥
Татьяна Сергеева	33	⚯	198101	150 000 ₽	⚥
Джек Румор	43	○	103126	700 000 ₽	⚥
Пелагея Ильина	47	⚭	103173	550 000 ₽	⚥
Николай Борисов	45	⚭	103139	600 000 ₽	⚥

Обобщенные квази-идентификаторы | Конфиденциальные атрибуты

Возраст	Семейное положение	Почтовый индекс	Доход	Сексуальная ориентация
30-35	0.33	198***	450 000 ₽	⚥
30-35	0.33	198***	350 000 ₽	⚥
30-35	0.33	198***	150 000 ₽	⚥
40-45	0.67	1031**	700 000 ₽	⚥
40-45	0.67	1031**	550 000 ₽	⚥
40-45	0.67	1031**	600 000 ₽	⚥

Класс эквивалентности

Информация, известная злоумышленнику

ФИО | Адрес | Пол | Дата Рождения | ..

Повторная идентификация

Пол | Возраст | .. | Доход | .. | ..

Обезличенный набор данных

СЦЕНАРИИ РАСПРОСТРАНЕНИЯ ДАННЫХ

Разделы	Сценарий “Публичные Данные”	Сценарий “Межорганизационное взаимодействие”	Сценарий “Частные данные”
Права доступа	Каждый имеет свободный доступ к опубликованным данным	Доступ к опубликованным данным (или их части) предоставляется ограниченному кругу лиц или организаций	Доступ к опубликованным данным имеет узкий круг лиц или организаций
Использование данных	Неограниченный доступ к данным через веб-портал, то есть свободный доступ для всех	• Обеспечение безопасности на своей территории • Предоставляемый доступ • Дистанционный виртуальный доступ • Доступ через аналитический сервер	Обмен внутри и между организациями
Права использования	Неограниченные права на многократное использование и распространение данных	Имеются у уполномоченного лица или организации	Повторное использование, тиражирование и распространение данных запрещены
Попытка повторной идентификации	Демонстрационная атака или использование как дополнительных данных	•Умышленная внутренняя атака •Непреднамеренное опознавание конкретного лица в наборе данных знакомым •Утечка данных	
Оценка допустимого риска	|Ě| ≥ 20	|Ě| ≥ 10	|Ě| ≥ 3

Методы обезличивания - техники, применяемые к ПДн с целью устранения (снижения риска) идентификации субъекта персональных данных. В результате применения соответствующего метода из набора данных удаляется идентифицирующая личность человека информация.

Оценка риска обезличенных данных производится для каждого набора данных и каждого метода.

Все методы являются параметризуемыми, то есть содержат настраиваемые характеристики (такие как набор квази-идентификаторов), определяющие конкретный алгоритм в отношении конкретного набора данных в заданных условиях (сценариях и контексте).

26)Модели приватности. Модели дифференциальной приватности, K-anonymity.

Модели приватности

МОДЕЛИ ПРИВАТНОСТИ	
k-anonymity	T-closeness
L-diversity	β-likeness
δ-precense	δ-disclosure

L-diversity

Считается, что набор данных удовлетворяет ℓ-разнообразию, если для каждой группы записей, имеющих общую комбинацию ключевых атрибутов, существует не менее ℓ "хорошо представленных" значений для каждого существенного атрибута. Считается, что таблица имеет l-разнообразие, если каждый класс эквивалентности таблицы имеет l-разнообразие

Новые ℓ -разнообразные классы эквивалентности требуют, чтобы анонимная запись была "хорошо представлена" в записях. Например, если у нас есть 4 типа диагностики заболеваний, то для 2-разнообразия записи эквивалентности должны представлять чувствительную запись не более чем в двух из четырех возможных записей.

Таблица с 2-разнообразием будет выглядеть следующим образом:

PatientName	Postcode	AgeGroup	Disease	
*	476**	Under 30	Heart Disease	$\{QI\ Group\ 1\}$
*	476**	Under 30	Heart Disease	
*	476**	Under 30	Flu	
*	476**	Under 30	Cancer	
*	479**	Over 40	Heart Disease	$\{QI\ Group\ 2\}$
*	479**	Over 40	Heart Disease	
*	479**	Over 40	Flu	
*	479**	Over 40	Pneumonia	

δ-precense

Эта модель может быть использована для защиты данных от раскрытия членства. Набор данных является (δmin, δmax)-присутствующим, если вероятность того, что индивид из популяции содержится в наборе данных, лежит между δmin и δmax. Чтобы иметь возможность рассчитать эти вероятности, пользователям необходимо указать таблицу популяций.

Мы представляем метрику, δ-присутствие, которая четко связывает качество анонимизации с риском, возникающим при неадекватной анонимизации. Мы показываем, что существующие методы анонимизации не подходят для ситуаций,

когда δ-присутствие является хорошей метрикой (в частности, когда знание того, что человек находится в базе данных, представляет риск для конфиденциальности), и представляем алгоритмы для эффективной анонимизации в соответствии с δ-присутствием. Алгоритмы оцениваются в контексте реального сценария, демонстрируя практическую применимость подхода.

t-closeness

Эта модель конфиденциальности также может быть использована для защиты данных от раскрытия атрибутов. Она требует, чтобы распределения значений чувствительного атрибута в пределах каждого класса эквивалентности имели расстояние не более t до распределения значений атрибута во входном наборе данных. Для этого используется кумулятивная абсолютная разница между частотными распределениями, которая рассчитывается с помощью расстояния земного шара (EMD). Для переменных с различными типами данных были предложены различные варианты: (1) равное расстояние до земли считает, что все значения одинаково удалены друг от друга, (2) иерархическое расстояние до земли использует иерархии обобщения значений для определения расстояния между значениями и (3) упорядоченное расстояние до земли рассчитывает расстояния на основе порядка значений.

β-Likeness

Эта модель конфиденциальности связана с t-близостью и δ-раскрытием конфиденциальности и может также использоваться для защиты данных от раскрытия атрибутов. Она направлена на преодоление ограничений предыдущих моделей путем ограничения относительного максимального расстояния между распределениями значений чувствительных атрибутов, также учитывая положительный и отрицательный выигрыш в информации.

δ-Disclosure privacy

Эта модель конфиденциальности также может быть использована для защиты данных от раскрытия атрибутов. Она также накладывает ограничение на расстояния между распределениями чувствительных значений, но использует мультипликативное определение, которое является более строгим, чем определение, используемое в t-closeness

Дифференциальная приватность — это математическое определение понятия «наличия приватности». Это не какой-то конкретный процесс, а, скорее, свойство, которым может обладать процесс. Например, можно рассчитать (доказать), что данный конкретный процесс удовлетворяет принципам дифференциальной приватности.

Попросту говоря, для каждого человека, чьи данные входят в анализируемый набор, дифференциальная приватность гарантирует, что **результат анализа на**

дифференциальную приватность будет практически неотличим вне зависимости от того, есть ли ваши данные в наборе или нет.

Анализ дифференциальной приватности часто называют механизмом, и мы обозначим его как M

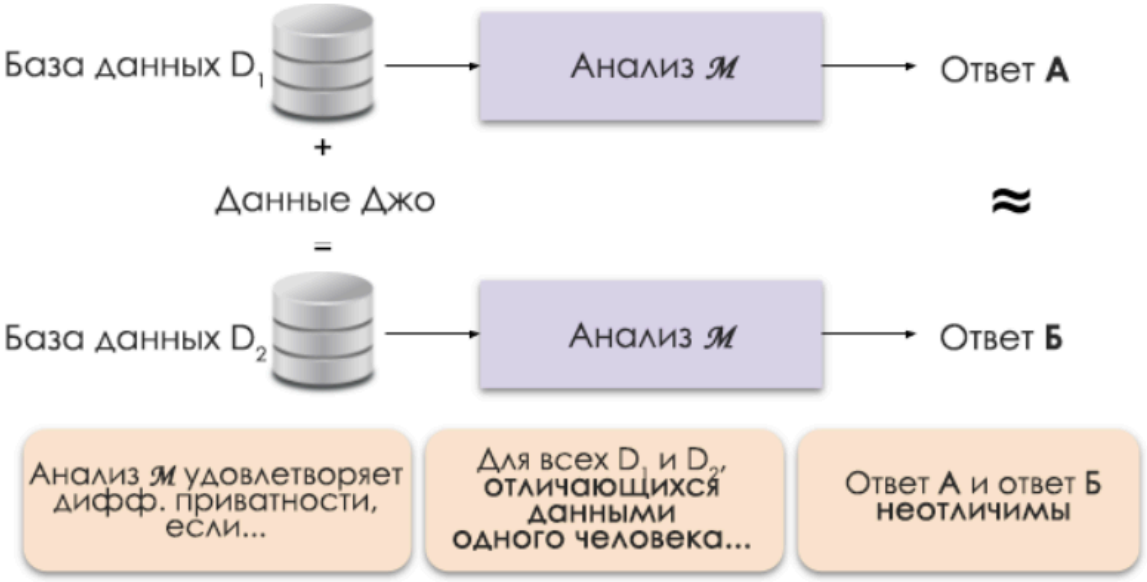


Рисунок 1: Схематичное представление дифференциальной приватности.

Мы контролируем требуемый уровень приватности через *изменение параметра приватности ε*, который также называют *потерей приватности (privacy loss)* или *бюджетом приватности (privacy budget)*. Чем меньше значение ε, тем менее различимы результаты и тем больше защищены данные отдельных людей.

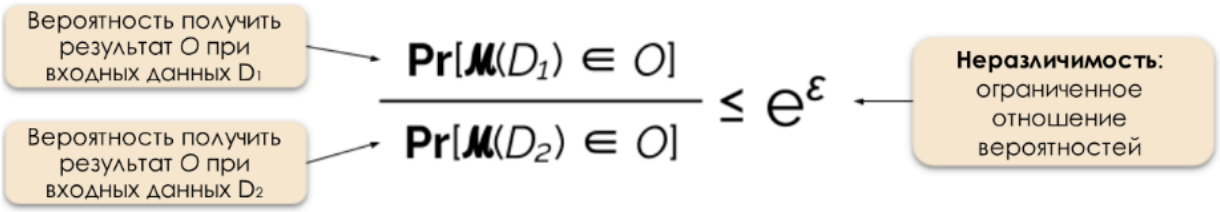


Рисунок 2: Формальное определение дифференциальной приватности.

МОДЕЛИ ДИФ. ПРИВАТНОСТИ.

Центральная модель

Главный компонент — **доверенное хранилище данных** (trusted data curator). Каждый источник передаёт ему свои конфиденциальные данные, а оно собирает их в одном месте (например, на сервере). Хранилище является *доверенным*, если мы предполагаем, что оно обрабатывает наши чувствительные данные самостоятельно, никому их не передает и не может быть никем скомпрометировано. Иными словами, мы считаем, что сервер с конфиденциальными данными не может быть взломан.

В рамках центральной модели мы обычно добавляем шум к ответам на запросы. Преимуществом этой модели является возможность добавить минимально возможное значение шума, таким образом сохранив максимальную точность, допустимую по принципам дифференциальной приватности. Ниже показана схема процесса. Мы поместили «барьер приватности» между доверенным хранилищем данных и аналитиком так, чтобы вовне могли попасть только результаты, удовлетворяющие заданным условиям дифференциальной приватности. Таким образом, аналитик не обязан быть доверенным.

Рисунок 1: Центральная модель дифференциальной приватности.

Недостаток центральной модели в том, что она требует наличия именно доверенного хранилища, а многие из них на самом деле таковыми не являются. На самом деле отсутствие доверия к потребителю данных обычно является главной причиной для использования принципов дифференциальной приватности.

Локальная модель

Локальная модель дифференциальной приватности позволяет избавиться от доверенного хранилища данных: каждый источник (или владелец данных) сам добавляет шум в свои данные, прежде чем передать их в хранилище. Это значит, что в хранилище никогда не окажется чувствительной информации, а значит и нет необходимости в его доверенности. Рисунок ниже демонстрирует устройство локальной модели: в ней барьер приватности находится между каждым владельцем данных и хранилищем (которое может быть как доверенным, так и нет).

Рисунок 2: Локальная модель дифференциальной приватности.

Локальная модель дифференциальной приватности позволяет избежать основной проблемы центральной модели: если хранилище данных будет взломано, то хакеры получат доступ только к зашумленным данным, которые уже соответствуют требованиям дифференциальной приватности

В конечном счете, такой подход оправдан только для запросов с очень стойким трендом (сигналом). Apple, например, использует локальную модель для оценки популярности эмодзи, но результат оказывается полезен только для наиболее популярных эмодзи (где тренд наиболее выраженный). Обычно эту модель не применяют для более сложных запросов, вроде тех, что используются в Бюро переписи США или машинном обучении.

Гибридные модели

Центральная и локальная модели имеют как преимущества, так и недостатки, и сейчас основные усилия направлены на то, чтобы взять у них всё самое лучшее.

Например, можно использовать *модель с перемешивателем* (shuffling model). Она содержит недоверенное хранилище данных, много отдельных владельцев данных и несколько частично доверенных *перемешивателей*. Каждый источник сначала добавляет небольшое количество шума в свои данные, а затем отправляет их перемешивателю, который добавляет еще шума перед отправкой в хранилище данных. Суть в том, что перемешиватели вряд ли «вступят в сговор» (или будут взломаны одновременно) с хранилищем данных или друг с другом, так что небольшого шума, добавленного источниками, будет достаточно для гарантии приватности. Каждый перемешиватель может работать с несколькими источниками, как и в центральной модели, так что небольшое количество его шума будет гарантировать приватность и для результирующего массива данных.

Модель с перемешивателем является компромиссом между локальной и центральной моделями: она позволяет добавить меньше шума, чем локальная, но

больше, чем центральная.

Также можно комбинировать дифференциальную приватность с криптографией, как в проколе конфиденциального вычисления (secure multiparty computation, MPC) или в полностью гомоморфном шифровании (fully homomorphic encryption, FHE). FHE позволяет выполнять вычисления с зашифрованными данными без их предварительной расшифровки, а MPC позволяет группе участников безопасно выполнять запросы по распределенным источникам без раскрытия их данных. Вычисление *дифференциально приватных функций* с помощью криптобезопасных (или просто безопасных) вычислений — многообещающий путь по достижению точности центральной модели со всеми преимуществами локальной. Причем в этом случае использование безопасных вычислений снимает необходимость иметь доверенное хранилище. Недавние работы [5] демонстрируют обнадеживающие результаты от комбинирования MPC и дифференциальной приватности, вбирая в себя большинство достоинств обоих подходов. Правда, в большинстве случаев безопасные вычисления на несколько порядков медленнее, чем локально выполненные, что особенно важно для больших массивов данных или сложных запросов. Сейчас безопасные вычисления находятся в активной фазе разработки, так что их производительность стремительно возрастает.

K-anonimity

k-анонимность - это свойство, которым обладают некоторые анонимизированные данные.

Считается, что набор данных обладает свойством k-анонимности, если информация о каждом человеке, содержащаяся в релизе, не может быть отличима по крайней мере от

k - 1 человек, информация о которых также содержится в наборе.

Чтобы настроить k-анонимность, необходимо рассмотреть следующие вопросы:

- Какие поля в наборе данных содержат чувствительную, идентифицирующую или квазиидентифицирующую информацию?
- Как можно обобщить квазиидентифицирующие поля в иерархические группы?
- Сколько человек должно быть в толпе, чтобы она считалась анонимной?

Квазиидентификаторы должны быть обработаны таким образом, чтобы каждая отдельная комбинация квазиидентификаторов обозначала не менее k записей.

Чувствительная информация - это то, что человек не хочет, чтобы о нем знали, например, о его болезни или зарплате. Идентифицирующая информация - это то, что непосредственно идентифицирует человека, например, имя или номер социального страхования. Квазиидентифицирующая информация не может

однозначно идентифицировать человека, но в сочетании с другими квазиидентифицирующими данными может.

Name	Gender	Location	Weight	Illness
Paul	m	Paris	50 kg	Bronchitis
Martin	m	Nice	100 kg	Angina
Nils	m	Munich	38 kg	Flu
Annika	f	Munich	130 kg	Broken leg

Identifier | Quasi-Identifier | Sensitive data

Sensitive, identifying, and quasi-identifying information

27)Обобщение. Локальное обобщение и Агрегация.

Локальное обобщение:

В рамках данной техники предполагается уменьшение специфичности атрибута за счет подмены точного значения атрибута его общим значением.

Применяется для гео-данных, временных интервалов, финансовых параметров.

Основные характеристики техники:

– снижает риск повторной идентификации за счет увеличения размеров классов;
– эффективен для профиля злоумышленника типов “прокурор”, “журналист;
– расчет потери информации на основе метрик энтропии.

Пример метода Локальное обобщение

ID	Возраст	Доход
1	43	1345
2	34	2543
3	23	1643
4	28	3434
5	45	6045
6	60	2365
7	51	7123
8	25	2854
9	31	1300
10	54	1312

⇨

ID	Возраст	Доход
1	40-50	Низкий
2	30-40	Средний
3	20-30	Низкий
4	20-30	Средний
5	40-50	Высокий
6	50-60	Средний
7	50-60	Высокий
8	20-30	Средний
9	30-40	Низкий
10	50-60	Низкий

Другие характеристики метода

1	Практика применения метода	Метод применяется как для непрерывных значений, так и для категориальных данных. Соответствует «методу изменения состава или семантики» [3]
2	Возможность применения в комплексе с другими методами	Метод может применяться в комплексе с методами подавления, агрегации, маскеризация (при определенных установках – аналогичным по смыслу и применению)
3	ПО, реализующее метод	Пакет Arx, *sdcMicro* (см. 4.8)
4	Сохранение связи с исходными полями	Непосредственно метод не поддерживает связь с исходным полем, такая связь может устанавливаться через другие QID или псевдонимы (см. 4.5).
5	Параметрический объем	Требует обработки данных для выделения групп аналогично методу микро-агрегации, но более детерминировано.
6	Рекомендации по применению метода	Широко используется для получения обезличенных наборов в сочетании с другими методами.

Агрегация:

Техника основана на построении обобщений по одному или нескольким атрибутам (групповые операции). Представляет из себя построение обобщенных данных по выделенным атрибутам.

Метод применяется в основном для статистических целей.

Основные характеристики техники:

– наиболее вероятная атака;
– снижает риск повторной идентификации за счет перехода к агрегирующим функциям;
– эффективен для всех профилей злоумышленника типов: “прокурор”, “журналист”, “маркетолог” ;
– расчет потери информации на основе метрик энтропии.

Пример метода Агрегация

ID	Доход	Возраст
1	23456	28
2	16879	18
3	340000	45
4	23700	14
5	60000	32

⇨

Агрегат	Значение
Максимальное значение	340000
Минимальное значение	16879
Среднее значение	92807
Общее число	5

Несмотря на использование агрегированных функций злоумышленник может использовать атаки восстановления для повторной идентификации

Другие характеристики метода

1	Практика применения метода	Метод применяется как для непрерывных значений, соответствует «методу изменения состава или семантики» [3]
2	Возможность применения в комплексе с другими методами	Метод редко применяется совместно с иными техниками из-за специфики агрегирующих функций
3	ПО, реализующее метод	Пакет Statistica, *sdcMicro* (см. 4.8)
4	Сохранение связи с исходными полями	Непосредственно метод не поддерживает связь с исходным полем, такая связь может устанавливаться через метаинформацию (см. 4.5).
5	Параметрический объем	Обработка по всем записям, значительные затраты на выполнение статистических вычислений больших наборов требует сохранения частичных сумм

28)Рандомизация. Возмущение, Микро-агрегация и Перемешивание.

- **возмущение**

Настоящая техника предполагает внесение шума в данные, которые перестают быть точными или правдивыми, но сохраняют основные статистические закономерности. Метод применим для бинарных данных (да/нет, например, пола), цифровых BLOBS (фотографии), статистически частых наборов данных. Позволяет сохранить статистическую ценность набора данных при незначительных потерях информации.

Основные характеристики техники:

− наиболее вероятная атака – на атрибуты и через связывание;

− исключает квази-идентификатор или увеличивает класс эквивалентности;

− эффективен для профиля злоумышленника типа “прокурор” – расчет вероятности риска;

− расчет потери информации на основе мер взаимной полезности.

Пример

	ФЛ	GPS координаты	Потратил
1	Иван Иванов	59.9980108685809 - 30.5633864413586	4000
2	Сергей Корнеев	55.8228843815701 - 37.7631040317474	3000
3	Михаил Звягинцев	55.8332478598453 - 37.4127809953384	2000
4	Дмитрий Дмитриев	60.0329101735402 -30.564230721555	10000
5	Алексей Иноземцев	55.6625519620301 - 37.4349304594488	10000

⇨

	GPS’	Потратил
1	59.9980128907 43-30.5633124673021	4000
2	55.8228863825180-37.7631028567894 0	3000
3	55.8332413874970-37.4127838459459	2000
4	60.0329100236789-30.5642303526745	10000
5	55.6625503572846-37.4349302678998	10000

Другие характеристики:

- практика применения метода

в основном для непрерывных значений (носящих численный характер). Соответствует “методу изменения состава или семантики)

- возможность применения в комплексе с другими методами

обезличивания: обобщение, перемешивание, микро-агрегация

- ПО, реализующие метод

  пакет sdcMicro

- Сохранение связи с исходными полями

Непосредственно метод не поддерживает связь с исходным полем, такая связь может устанавливаться через другие QID

- Параметрический объем

Выполнение метода требует проведения вычислений по всему объему данных (поиск нормального среднего). При сохранении произведенных вычислений объем необходимой информации - $N^*|\varepsilon|$

- Рекомендации по применению метода для наборов, используемых в статической обработке без учета корреляционных зависимостей
  - **микро-агрегация**

    Техника предполагает формирование групп записей, для которых вместо точных значений указывается диапазон. Метод рекомендуется к применению для параметров возраста, географических локаций, финансовых транзакций, биометрических данных. Один из самых широко применяемых методов.

    Основные характеристики техники:

    − наиболее вероятная атака – через связывание;

    − снижает риск повторной идентификации за счет увеличения размеров классов эквивалентности– рекомендуется к использованию метрики;

    − неэффективен для малых наборов информации;

    − эффективен для профиля злоумышленника типа “маркетолог” – расчет вероятности риска;

    − расчет потери информации на основе метрик на основании энтропии.

    Пример:

ID	Группа	Доход	Траты	Накопления
1	1	2310	1684	5.3
2	1	2435	1827	7.4
3	1	2111	1756	6.3
4	2	2314	1950	8.0
5	2	6,045	4569	9.2
6	2	2,345	1923	7.8

⇨

Доход	Траты	Накопления
2,285.7	1,846.3	6.3
2,285.7	1,846.3	6.3
2,285.7	1,846.3	6.3
3,567.3	2,814.0	8.3
3,567.3	2,814.0	8.3
3,567.3	2,814.0	8.3

    Алгоритм применения:

    − первым шагом в микро-агрегации является формирование небольших групп лиц, которые являются однородными по отношению к значениям выбранных переменных, таких как группы с аналогичным доходом или возрастом.

    − значения выбранных переменных всех членов группы заменяются общим значением, например, средним значением этой группы.

    Другие характеристики:

- практика применения метода

в основном для непрерывных значений (носящих численный характер). Соответствует “Методу изменения состава или семантики”

- возможность применения в комплексе с другими методами

обезличивания: обобщение, перемешивание, микро-агрегация

- ПО, реализующие метод

Пакеты Arx, sdcMicro

- Сохранение связи с исходными полями

Непосредственно метод не поддерживает связь с исходным полем, такая связь может устанавливаться через другие QID

- Параметрический объем

Выполнение метода требует проведения вычислений по всему объему данных (поиск нормального среднего). При сохранении произведенных вычислений объем необходимой информации - N*δ, где δ - размер полей после микро-агрегации

- Рекомендации по применению метода

наиболее подходит для непрерывных переменных, но в некоторых случаях может быть распространена на категоричные переменные. Также может использоваться для категориальных данных при том, что существует возможность форматирования групп и может быть рассчитана агрегированная замена значений в группе. Это касается порядковых переменных.

- **перемешивание**

Техника направлена на перемешивание аналогичных записей при сохранении статической значимости всего набора. Наиболее полезна при обезличивании временных интервалов, например, информации об отпусках. Метод применяется, как правило, в совокупности с другими методами обезличивания.

Основные характеристики техники:

− наиболее вероятна атака дифференциации;

− снижает риск повторной идентификации за счет увеличения размеров классов эквивалентности;

− эффективен для профиля злоумышленника типа “журналист”;

− расчет потери информации на основе метрик классификации.

Пример:

ID	Доход	Ранг
1	2300	2
2	2434	6
3	2123	1
4	2312	3
5	6045	10
6	2345	4
7	2543	7
8	2854	9
9	2365	5
10	2643	8

⇨

Доход (расчет по регрессии)	Ранг	Перемешанные значения
2466.56	4	2345
2583.58	7	2543
2594.17	8	2643
2530.97	6	2434
5964.04	10	6045
2513.45	5	2365
2116.16	1	2123
2624.32	9	2854
2203.45	2	2300
2358.29	3	2312

Перемешивание может базироваться на определенном сценарии (например, использовать регрессию по атрибуту).

Другие характеристики:

1. практика применения метода
2. как для непрерывных значений, так и для категориальных данных. Соответствует “методу перемешивания”
3. возможность применения в комплексе с другими методами
4. обезличивания: обобщение, перемешивание, микро-агрегация. В то же время метод является разрушающим.
5. ПО, реализующие метод
6. пакет sdcMicro
7. Сохранение связи с исходными полями
8. Непосредственно метод не поддерживает связи с исходным полем, такая связь может устанавливаться через другие QID или псевдонимы.
9. Параметрический объем
10. Перетасовка требует полного ранжирования данных, которые могут быть вычислительно очень интенсивными для больших наборов данных с несколькими переменными.
11. Рекомендации по применению метода
12. наиболее подходит для непрерывных переменных так как в случае категорированных атрибутов серьезно разрушает корреляционные связи

29)Псевдонимизация. Создание псевдонимов, Маскеризация.

- **Создание псевдонимов**

Для чувствительного атрибута проводится замена идентификаторов на условные обозначение или цифровые ключи. Как правило, проводится для частных данных. Используется в совокупности с другими методами обезличивания

Основные характеристики техники:

− эффективен для атак типа “связывание”;

− расчет риска производится по формулам модель прокурора в связи с угрозами утечек – наибольший риск;

− имеет незначительные потери информации в связи с направленностью на точные идентификаторы.

Создание псевдонимов может производиться следующими способами:

− через счетчик (использование в качестве ID значений 1,2,3, 4, …);

− генерацией случайного номера (например, 0.234, 289, 10256.91);

− использованием криптографической Hash – функции или MAC (Message Authentication Code);

− использованием разных случайных псевдонимов для одинакового значения ID, при его повторном нахождении в наборе (такой набор называется полностью рандомизированным).

Пример:

ID	Имя	Online Time
1	Сергей	02:48
2	Алиса	03:11
3	Ирина	01:45
4	Виктория	04:14
5	Николай	05:32

⇨

Имя	Online Time
3	02:48
18	03:11
24	01:45
11	04:14
7	05:32

Другие характеристики:

- практика применения метода

как для непрерывных значений, так и для категориальных данных. Соответствует “методу изменения состава или семантики”

- возможность применения в комплексе с другими методами

подавления, агрегации, маскеризация (при определенных установках - аналогичным по смыслу и применению) - для других атрибутов

- ПО, реализующие метод

пакет sdcMicro

- Сохранение связи с исходными полями

Непосредственно метод не поддерживает связь с исходным полем, такая связь может устанавливаться через другие QID или сохраненную метаинформацию

- Параметрический объем

Требует сохранения метаинформации по исключению атрибута

- Рекомендации по применению метода

Широко используется для получения обезличенных наборов в сочетании с другими методами. Не рекомендуется для публичного распространения информации.

- **Маскеризация**

В рамках данной техники проводится замена части записи заполнителями по определенному шаблону. Хотя в результате исходной строке сопоставляется строка определенного шаблона, что позволяет отнести этот метод к псевдонимизации, по ряду характеристик этот метод может быть сопоставлен микро-агрегации: в результате применения шаблонов возникают обобщенные группы. Применим для идентификаторов (номера паспортов, телефоны, номера карт)

Основные характеристики техники:

− эффективен для всех видов атак;

− имеет незначительные потери информации в связи с направленностью на точные идентификаторы.

Пример:

ID	IP	Телефон	Сайт
1	192.168.1.1	+7 950 303-4451	youtube.com
2	71.101.3.10	+7 903 092 1415	ok.ru
3	71.101.100.4	+7 901 234 1588	facebook.com
4	192.168.1.11	7 950 303-7729	vk.ru
5	170.190.42.91	+7 903 030 5170	yandex.ru
6	192.168.2.201	+ 7 950 148 1456	redporno.com
7	170.190.40.10	+7 901 145 5317	wiki.org

⇨

IP	Телефон	Сайт
192.168.X.X	950.XXX.XXXX	youtube.com
71.101.X.X	903.XXX.XXXX	ok.ru
71.101.X.X	901.XXX.XXXX	facebook.com
192.168.X.X	950.XXX.XXXX	vk.ru
170.190.X.X	903.XXX.XXXX	yandex.ru
192.168.X.X	950.XXX.XXXX	redporno.com
170.190.X.X	901.XXX.XXXX	wiki.org

Другие характеристики метода:

1) Практика применения метода

как для непрерывных значений, так и для категориальных данных. Соответствует “методу изменения состава или семантики”

2) Возможность применения в комплексе с другими методами

с подавлением, агрегацией, маскеризацией (при определенных установках - аналогичным по смыслу и применению) - для других атрибутов

3) ПО, реализующие метод

Пакет Arx, sdcMicro

4) Сохранение связи с исходными полями

Непосредственно метод не поддерживает связь с исходным полем, такая связь может устанавливаться через другие QID или псевдонимы.

5) Параметрический объем

Требует сохранения метаинформации по исключению атрибута

6) Рекомендации по применению метода

Широко используется для получения обезличенных наборов в сочетании с другими методами.

30)Шифрование. Детерминированное шифрование, Гомоморфное шифрование.

Детерминированное шифрование

Шифрование отдельного атрибута, как правило, на базе алгоритмов RSA или систем с парами ключей. Метод применяется для данных вида пароли, медицинской информации, больших текстовых атрибутов. Применяется к небольшим наборам данных.

Основные характеристики техники:

- наибольшее распространение имеет для рисков частных данных в совокупности с другими методами обезличивания;
- эффективен для атак типа “логических выводов”;
- расчет риска проводится по модель прокурор) в связи с угрозами утечек – наибольший риск;
- по выделенному атрибуту – полностью потерянная информация, в отношении других квази-идентификаторов.

Пример:

ID	Имя	Online Time
1	Сергей	02:48
2	Алиса	03:11
3	Ирина	01:45
4	Виктория	04:14
5	Николай	05:32

⇨

Имя	Online Time
d6c4575df3a246fe76b975dec9201a84	02:48
3f77956a1f0e2710efd7b595af23463d	03:11
cf6d74f84dd326c489a850a69459c073	01:45
2daa7d6326bc2918c6dd46e35ace6b6d	04:14
63d508b0eeacbf9a4daa07eb17ea613a	05:32

Шифрование может использоваться как вид псевдонимизации и проводиться по одной из следующих техник:

- симметричное шифрование;
- асимметричное шифрование позволяет иметь две различных сущности, участвующих в процессе псевдонимизации
- использования HASH-функций для нескольких атрибутов (changing mode) и формирование из них составного ключа: PK = Hash(attribute1) & Hash(attribute2) & Hash(attribute3);
- использование псевдонимов, построенных на деревьях меркла;
- децентрализованные сценарии обмена ключами (например, через блокчейн системы) или использование протоколов вида Multiparty Computation (MPC).

Другие характеристики:

Практика применения метода

как для непрерывных, так и для категориальных данных. Соответствует “методу изменения состава или семантики”

Возможность применения в комплексе с другими методами

с подавлением, агрегацией, маскеризацией (при определенных установках - аналогичным по смыслу и применению) - для других атрибутов

ПО, реализующие метод

—

Сохранение связи с другими полями

При использовании хэшей позволяет получать одинаковые значения для исходных ID

Параметрический объем

Требует сохранения метаинформации по используемым алгоритмам шифрования и/или приватным ключам(для ассиметричного шифрования)

Рекомендации по применению кода

Не рекомендуется для использования с открытыми наборами данных, наибольшую эффективность имеет для защиты от инсайдера.

Гомоморфное шифрование

Шифрование, при котором сохраняются результаты над зашифрованными объектами, как если бы они применялись к начальным текстам. Относительно редкий метод из-за ограничений в классах гомоморфного шифрования. Основное применение – для биометрических данных.

Основные характеристики техники:

− из-за сложности метод применяется в ограниченных случаях публичных данных для практических приложений рассматриваются результаты голосования;
− эффективен для атак типа “восстановление”;
− расчет риска проводится по модель прокурора из сценария распространения данных;
− по выделенному атрибуту – полностью потерянная информация, в отношении других квази-идентификаторов применяются модели 3.5.x.

Практическое развертывание гомоморфного шифрования осложняется также рядом проблем, поскольку полностью гомоморфная система шифрования еще не разработана. Ограничения в основном основаны на ограничениях классов математических функций, которые поддерживают необходимый спектр атрибутов (типов данных).

одной из существенных проблем известных полностью гомоморфных криптосистем является их крайне низкая производительность. В настоящее время существует два основных пути повышения производительности: использование «ограниченного гомоморфизма» и использование так называемого «метода упаковки шифро-текстов». Первый подразумевает криптосистему, которая может выполнять операции сложения и умножения, но в ограниченном количестве. Суть второго в том, что в один шифро-текст записывается сразу несколько открытых текстов, и при этом в процессе одиночной операции такого пакетного шифро-текста происходит одновременная обработка всех входящих в него шифро-текстов.

31)Подавление. Локальное подавление. Удаление атрибутов.

Локальное подавление и удаление атрибутов – методы подавления (вида техник).

Локальное подавление. Техника предполагает удаление или перекодирование относительно редких записей данных. В отличии от метода удаления атрибутов нацелен на удаление строк или значений для заданных атрибутов в выделенных строках. Наибольшее применение имеет для медицинской и биометрической информации, а также больших наборов статистических данных с аномалиями.

Основные характеристики техники:

− используется как дополнительный метод при неудачном применении других методов обезличивания;

− эффективен для всех видов атак;

− снижает риск повторной идентификации за счет удаления прямых идентификаторов;

− ведет к существенной потери качества данных, *расчет потери информации эффективно проводить на основе метрик энтропии.*

Пример

ID	Пол	Район	Образование
1	Ж	сельский	Высшее
2	М	сельский	Высшее
3	М	сельский	Высшее
4	М	сельский	Высшее
5	Ж	сельский	Среднее
6	Ж	сельский	Среднее
7	Ж	сельский	Среднее

⇨

Пол	Район	Образование
Исключенная строка		
М	сельский	Высшее
М	сельский	Высшее
М	сельский	Высшее
Ж	сельский	Среднее
Ж	сельский	Среднее
Ж	сельский	Среднее

Отдельные строки могут существенно влиять на показатели k-anonymity (а следовательно, на вероятность повторной идентификации). Поэтому, если количество таких строк не велико, соответствующие записи могут быть удалены или заменены на общие значения

Другие характеристики

Практика применения	Как для непрерывных значений, так и для категориальных данных. Соответствует «методу изменения состава ил семантики»
Возможность применения в комплексе с другими	Может применяться с методами подавления, агрегации, маскеризации (при определенных установках – анологичным по смыслу и применению) – для других атрибутов
ПО, реализующее метод	Пакет Arx, sdcMicro
Сохранение связи с исходными данными	Непосредственно метод не поддерживает, но связь может устанавливаться через другие QID или псевдонимы
Параметрический объём	Требует сохранения метаинформации по исключению атрибута
Рекомендации по применению	Широко используется для получения обезличенных наборов в сочетании с другими

	методами. Метод применяется для исключения редких записей.

Удаление атрибутов. Под данной техникой понимается удаление чувствительного контента без добавления замен. Метод используется для удаления прямых идентификаторов, а также удаления избыточных квази-идентификаторов.

Основные характеристики техники:

− эффективен для всех видов атак;

− снижает риск повторной идентификации за счет удаления прямых идентификаторов;

− приводит к наибольшей потере информации. *Расчет потери информации эффективно проводить на основе метрик энтропии.*

Пример

ID	Пол	Район	Образование	З/П
1	Ж	сельский	Высшее	Высокая
2	М	сельский	Высшее	Высокая
3	М	сельский	Высшее	Высокая
4	М	сельский	Начальное	Высокая
5	Ж	сельский	Среднее	Высокая
6	Ж	сельский	Среднее	Низкая
7	Ж	сельский	Среднее	Низкая

⇨

Пол	Район	Образование	
Ж	сельский	Исключенный атрибут	Высокая
М	сельский		Высокая
М	сельский		Высокая
М	сельский		Высокая
Ж	сельский		Высокая
Ж	сельский		Низкая
Ж	сельский		Низкая

Другие характеристики

Практика применения	Как для непрерывных значений, так и для категориальных данных. Соответствует «методу изменения состава ил семантики»
Возможность применения в комплексе с другими	Может применяться с методами подавления, агрегации, маскеризации (при определенных установках – анологичным по смыслу и применению) – для других атрибутов
ПО, реализующее метод	Пакет Arx, sdcMicro
Сохранение связи с исходными данными	Непосредственно метод не поддерживает, но связь может устанавливаться через другие QID или псевдонимы
Параметрический объём	Требует сохранения метаинформации по исключению атрибута
Рекомендации по применению	Широко используется для получения обезличенных наборов в сочетании с другими методами. Метод применяется, когда остальные методы не повлияли на результат.

32)Семплинг, Синтетические данные, метод декомпозиции.

Семплинг. Публикация части набора данных с примерными данными. Производятся для выбранной группы данных на основании анализа имеющихся зависимостей (репрезентативности данных)

Основные характеристики техники:

− метод применяется для относительно небольших публикуемых данных, преимущественно в сценариях публичных или полу-публичных данных;

− формулы расчета вероятности риска напрямую не применимы, но могут быть оценены как низкие;

− потеря данных оценивается по исходному набору на основании метрик взаимной полезности.

Пример

ID	Имя	Online Time
1	Сергей	02:48
2	Алиса	03:11
3	Ирина	01:45
4	Виктория	04:14
5	Николай	05:32

⇨

Имя	Online Time
Асильдер	02:21
Зоммара	03:16
Коприй	01:35
Людвина	03:57
Неофантий	05:50

Другие характеристики

Практика применения	Как для непрерывных значений, так и для категориальных данных. Соответствует «методу изменения состава ил семантики»
Возможность применения в комплексе с другими	Может применяться с методами подавления, агрегации, маскеризации
ПО, реализующее метод	Data Melt, MacAnova
Сохранение связи с исходными данными	При использовании хэшей позволяет получать одинаковые значения для исходных ID
Параметрический объём	Требует сохранения метаинформации по используемым алгоритмам шифрования и/или приватным ключам (для ассиметричного шифрования)
Рекомендации по применению	Используется в основном для методов машинного обучения.

Синтетические данные. Данные создаются на основании случайных правил или с помощью искусственного интеллекта. Метод аналогичен семплингу, но нацелен на создание больших наборов данных, схожих по внутренним зависимостям с исходным набором. Эффективен при моделировании бинарных данных, финансовых данных, данных с большим количеством атрибутов.

Основные характеристики техники:

− метод применяется для публичных или полу-публичных данных;

− формулы расчета вероятности риска напрямую не применимы, но могут быть оценены как низкие;

− потеря данных оценивается по исходному набору на основании метрик взаимной полезности.

Пример

Синтетический набор данных							
ID	ФИО	Пол	Секс. предпочтения	День рождения	Возраст	Заболевание	Рост
1	Заира Хауовна Пацкина	Ж	♀	28.08.1957	64	Синдром Жильбера	157
2	Моудин Савриевна Малявкина	Ж	♀	31.05.1945	76	НПВС-гастропатия	169
3	Меги Антариоовна Лепаева	Ж	♀	20.01.1970	51	Рак мочевого пузыря	163
4	Зиронька Егизовна Чупкина	Ж	♀	23.03.1944	77	Ком в горле	165
5	Наргиза Вочабиевна Окуняка	Ж	♀	30.11.1955	66	Аденома околощитовидной железы	156
6	Ониска Абачиевна Сименникова	Ж	♀	02.05.1970	51	Полипы толстой кишки	169
7	Элана Латиновна Мшенина	Ж	♀	12.06.1975	46	Тонзиллит	172
8	Джэйлин Сотериховна Шпалецкая	Ж	♀	04.05.1969	52	Гепатит	160
9	Авруца Енисьевна Бижова	Ж	♀	30.11.1947	74	Гипотиреоз	164

Метод декомпозиции. Разделение множества атрибутов на несколько подмножеств с последующим раздельным хранением. Метод не является собственно техникой обезличивания, но позволяет формировать наборы данных, к которым могут применяться методы обезличивания.

Основные характеристики техники:

− метод применяется для частных данных с акцентом на хранение;

− формулы расчета вероятности риска применяются в зависимости от набора квази-идентификаторов по формулам;

− потеря данных оценивается по исходному набору на основании метрик классификации.

Пример

ID	Пол	Район	Образование
1	Ж	сельский	Высшее
2	М	сельский	Среднее
3	М	сельский	Высшее
4	М	сельский	Высшее

⇨

Пол	Район
Ж	сельский
М	сельский
М	сельский
М	сельский

Пол	Образование
Ж	Высшее
М	Среднее
М	Высшее
М	Высшее

Основные выводы по применению методов обезличивания:

− прямые идентификаторы **подавляются** или псевдонимизируются;

− методы обезличивания применяются в отношении квази-идентификаторов, для разных видов квази-идентификаторов применяются разные методы;

− для каждого набора возможно применение нескольких методов обезличивания в зависимости от типа квази-идентификатора, каждый метод имеет набор параметров, влияющих на степень “размытия”;

− специальные методы (такие как гомоморфное шифрование, дифференциальная приватность, **генерация синтетических данных**) требуют проработки архитектуры всей системы и имеют ряд ограничений.

− Классические методы обезличивания (**подавление**, обобщение) – эффективны при формировании отчуждаемых наборов данных (непосредственно работа с данными) и при нечетких гипотезах по анализу данных с помощью искусственного интеллекта;

− обычное шифрование может применяться для защиты прямых идентификаторов и не вполне является методом обезличивания.