ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Автоматизация сверки и устранения дубликатов в персональных данных

Автор: Абубакиров А.Р.
Научный руководитель: ст. преп. Севрюков С.Ю.
Рецензент: к.ф.-м.н. Корхов В.В.

Введение в проблематику

1. Что такое дубликаты?
2. Почему они возникают?
3. Какие сферы деятельности это затрагивает?
4. Зачем нужно устранять дубликаты?
5. Почему нельзя положиться только на номер паспорта?

Цели и задачи

1. Повышение качества данных
2. Разработка алгоритма поиска дубликатов
3. Применение на реальных данных
4. Оценка результатов алгоритма
5. Разработка прототипа библиотеки для интеграции решения в ИС МИАЦ

Существующие решения

Решения

1. Сервис “Dadata.ru”
2. Сервис “dedupe.io”
3. Сервис “Мастер адресов”
4. Решение “Индекс пациентов” от компании Нетрика

Недостатки

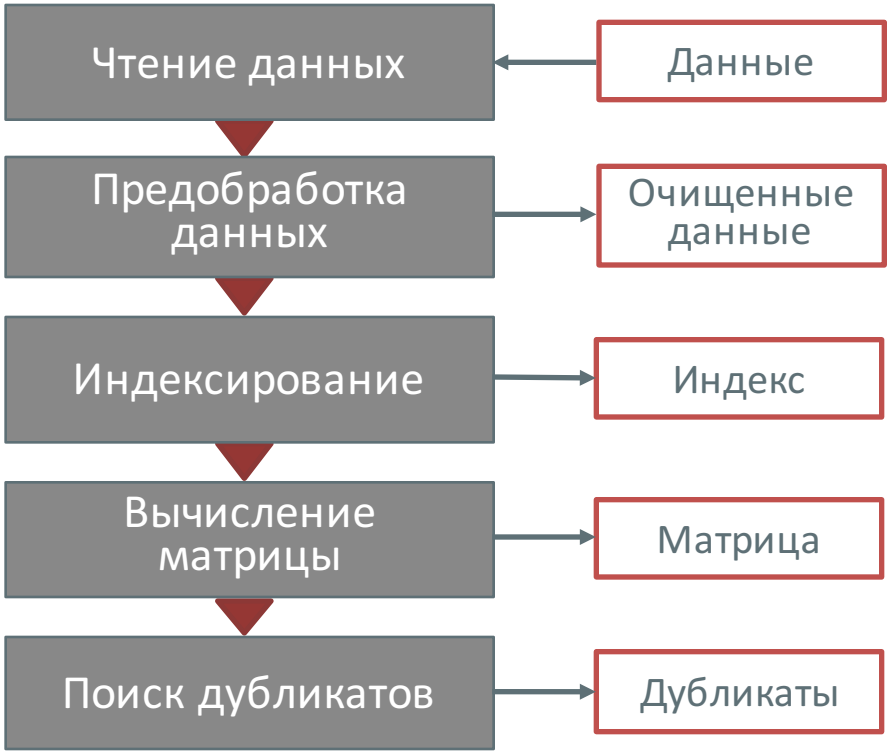
1. Необходимость передачи персональных данных третьим лицам
2. Закрытость решения по административным и формальным причинам

Обзор данных

	Фамилия	Имя	Отчество	Дата рождения
Всего значений	34 406	34 405	31 995	34 406
Уникальные значения	16 304	1 315	2 006	16 508
Пропущенные значения	0	1	2 411	0

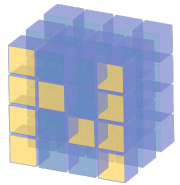
Табл.1. Статистика по данным.

Структура библиотеки



Используемые технологии

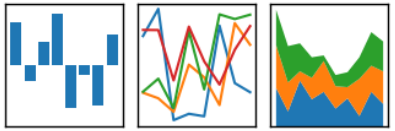
python



NumPy

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Примеры ошибок

Фамилия	Имя	Отчество	Дата рождения
Иванова (дев. Шереметьева)	Ольга	Викторовна	12/06/1990
Шереметьева	Ольга	-	12.06.1990 00:00:00
Иванова	Ольга	неизвестно	12/06/1990
Амиров	Амир	Арслан Оглы	12-08-1980
Амиров	нет	Арсланович	12.08.1980
Иванов Сергей		Vladimirovich	12/04/1999

Табл.2. Пример данных с различными видами ошибок.

Алгоритм поиска дубликатов: индексирование

Задача: уменьшить количество рассматриваемых записей.

Решение: выделить группы потенциальных дубликатов.

Критерий: на основе подстроки или общих символов.

index — ассоциативный массив, в котором:

ключ — номер записи,

значение — список номеров потенциальных дубликатов.

Алгоритм поиска дубликатов: вычисление матрицы

Матрица расстояний:

$$x[i][j][k] = dist(row_i[k], row_j[k]), \text{ где}$$

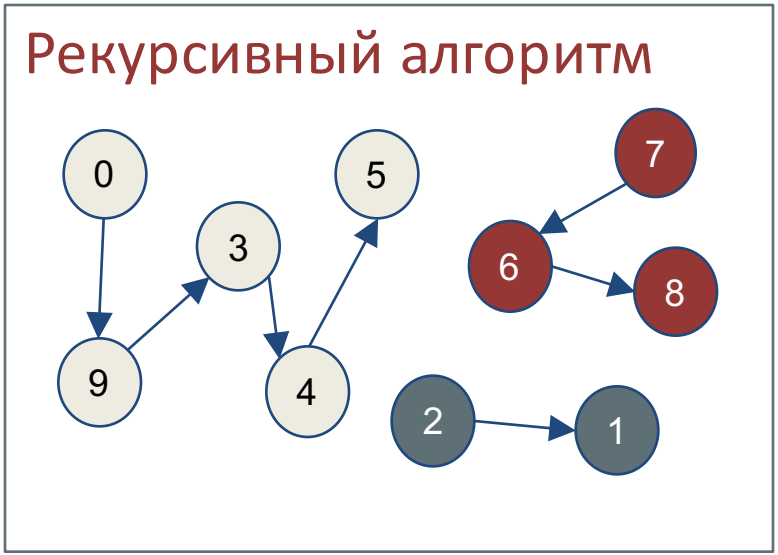
$dist$ — редакционное расстояние,

row_i — i-ая запись,

$row_i[k]$ — k-ое поле i-ой записи,

$$k \in [1, count_{fields}],\ i \in [1, count_{rows}],\ j \in index[i].$$

Алгоритм поиска дубликатов



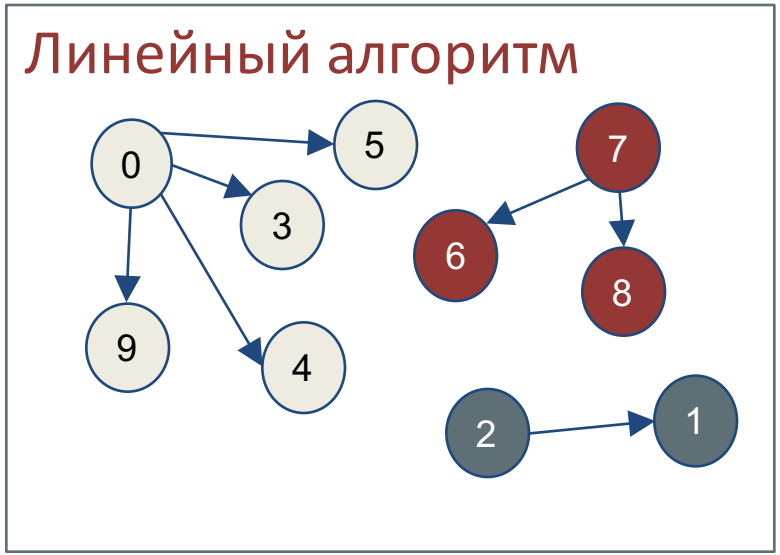


Рис.2. Варианты алгоритма поиска дубликатов

Устранение дубликатов: подходы

1. Оставить наиболее полную запись
2. Оставить самую свежую запись
3. Объединить все записи в одну
4. Предоставить выбор пользователю

Полученные результаты

	Тестовая выборка			Полная выборка
Всего записей	1000	1000	1000	34406
Уникальные записи	990	987	988	33987
Кластеры по 1 записи	981	976	976	33575
Кластеры по 2 записи	8	9	12	405
Кластеры по 3 и более записей	1	2	0	7
Точность	0.998	0.997	0.998	Неизвестно

Табл.2. Полученные результаты. Алгоритм поиска: линейный.
Пороговые значения: 0.8, 0.8, 0.8, 0.9.

Заключение

- Решить задачу без участия человека нельзя.
- Идеи, на которых построен алгоритм, заслуживают дальнейшего развития.
- Реализован алгоритм поиска дубликатов.
- Реализован прототип библиотеки для применения решения в сторонних проектах.

Спасибо за внимание

Типы ошибок и методы их исправления

Типы ошибок

1. Ошибки на уровне поля
    a. Опечатки
    b. Различные стандарты написания
2. Ошибки на уровне записи
    a. Пропущенные значения
    b. Несоответствие поля и значения

Методы исправления

1. Регулярные выражения
2. Эвристические правила
3. Полуавтоматическое исправление

Временные затраты

Типы индексация	Время	Точность
Без индексации	7 мин 30 с	0.998
На основе подстроки	17 с	0.8
На основе общих символов	30 с	0.997

Табл.3. Сравнение различных методов индексации.