

**РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ПОИСКА ДУБЛИКАТОВ
ПЕРСОНАЛЬНЫХ ДАННЫХ КЛИЕНТОВ НА ОСНОВЕ МЕТОДОВ
МАШИННОГО ОБУЧЕНИЯ**

**DEVELOPMENT OF SOFTWARE FOR SEARCHING FOR DUPLICATES
OF CLIENT PERSONAL DATA BASED ON MACHINE LEARNING
METHODS**

ВАРТАНЯН АРТЁМ АЛЕКСАНДРОВИЧ,
студент,

МИРЭА – Российский технологический университет.

ХАБАРОВА МАРИЯ АНТОНОВНА,
студентка,

МИРЭА – Российский технологический университет.

РЕДЬКО ЕГОР АНДРЕЕВИЧ,
студент,

МИРЭА – Российский технологический университет.

РУСАКОВ АЛЕКСЕЙ МИХАЙЛОВИЧ,
старший преподаватель,

МИРЭА – Российский технологический университет.

VARTANYAN ARTEM ALEXANDROVICH,
student,

MIREA – Russian Technological University.

KHABAROVA MARIA ANTONOVNA,
student,

MIREA – Russian Technological University.

REDKO EGOR ANDREEVICH,
student,

MIREA – Russian Technological University.

RUSAKOV ALEKSEJ MIHAJLOVICH,
senior lecturer,

MIREA – Russian Technological University.

В данной статье приводится обзор и анализ современных методов дедубликации информации в базах персональных данных клиентов на основе методов машинного обучения. Предметом исследования являются этапы дедубликации данных, алгоритмы вычисления схожести пар данных, их классификации и обнаружения дубликатов. Исследуется реализация данных алгоритмов с помощью технологий машинного обучения и нейронных сетей. Рассматриваются различные способы их применения; анализируются преимущества и недостатки существующих алгоритмов и нескольких программных решений, созданных на их основе, а также выделяются функции, которые необходимо реализовать в программном обеспечении для возможности эффективной дедубликации данных с помощью рассмотренных методов.

This article provides an overview and analysis of modern methods of deduplication of information in databases of personal data of clients based on machine learning methods. The subject of the research is the stages of data deduplication, algorithms for calculating the similarity of data pairs, their classification and detection of duplicates. The implementation of these algorithms using machine learning technologies and neural networks is investigated. Various ways of their application are considered; the advantages and disadvantages of existing algorithms and several software solutions based on them are analyzed, as well as the functions that need to be implemented in the software for the possibility of effective data deduplication using the methods considered are highlighted.

Ключевые слова: дедупликация, персональные данные, анализ схожести данных, классификация данных, машинное обучение.

Key words: deduplication, personal data, data similarity analysis, data classification, machine learning.

Введение. Учитывая постоянно растущий объем данных, собираемых предприятиями и государственными учреждениями, в последнее десятилетие наблюдается большой интерес к новым методам, позволяющим эффективно обрабатывать, управлять и анализировать большие массивы данных. Способность своевременно проводить аналитику может обеспечить конкурентное преимущество коммерческому предприятию, привести к повышению эффективности государственных учреждений, что важно для национальной безопасности [1; 2].

Одной из задач сопоставления данных является обнаружение дубликатов клиентов – идентификации и сопоставления записей, которые относятся к одним и тем же людям в пределах одной или нескольких баз данных. Решение этой задачи поможет отделу маркетинга эффективнее взаимодействовать с клиентами, имея всю информацию об уже предоставленных услугах, банкам – лучше учитывать риски при кредитном скоринге, идентифицировать подозрительные операции, государственным учреждениям – повысить качество предоставления государственных услуг, консолидировав разрозненные данные гражданина.

Характеристика предметной области

Во многих крупномасштабных информационных системах данные хранятся в нескольких разрозненных источниках. Анализ данных, собранных из них как внутри организации, так и между различными организациями, может принести гораздо больше преимуществ по сравнению с анализом баз данных в изоляции. В таких случаях данные из нескольких источников необходимо интегрировать и дедуплицировать, а затем обогатить существующие источники данных.

Задачей дедупликации данных является идентификация, сопоставление и слияние записей, которые относятся к одной и той же сущности в пределах одной или нескольких баз данных. Рассматриваемые сущности обычно являются людьми, например, клиентами компании, налогоплательщиками, пациентами клиники. За последнее десятилетие различные прикладные области и области исследований разработали свои собственные решения проблемы, и в результате эта задача теперь известна под разными названиями – «связывание записей» (recordlinkage), «разрешение сущностей» (entityresolution), «сопоставление данных» (datamatching)

Основной проблемой в решении данной задачи является отсутствие общих надежных идентификаторов, по которым данные можно сопоставить. Как правило, в базах данных содержатся естественные ключи – например, данные о документе, удостоверяющем личность, или индивидуальный номер налогоплательщика – однако, для сопоставления данных только на основе этих идентификаторов необходимо быть абсолютно уверенным в их точности, полноте и надежности, потому что любая ошибка в таком идентификаторе приведет к некор-

ректному слиянию записей. Таким образом, при решении задачи поиска дубликатов необходимо использовать как можно более полный набор имеющихся данных о человеке [3;4].

Еще одной проблемой является необходимость учитывать возможные ошибки и опечатки, связанные с переносом письменной информации в базу данных. Для ее решения используют различные алгоритмы предобработки данных – начиная от сравнения строк по их звучанию (Soundex и его вариации) и до использования метрик сходства (расстояние Левенштейна).

Процесс дедупликации данных состоит из 4 основных шагов, представленных на рис. 1.

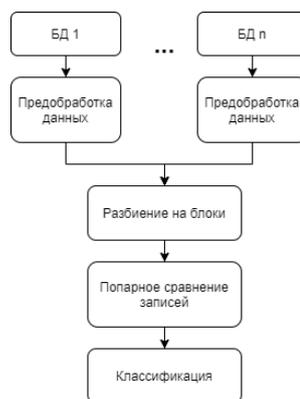


Рис. 1. Основные этапы дедупликации данных.

Первым шагом обычно является этап предварительной обработки, которая необходима для приведения данных к одному виду. Кроме того, большинство реальных баз данных содержат зашумленные, неконсистентные и ошибочные данные. Существует много различных способов предварительной обработки и очистки данных, но самыми популярными из них являются: удаление нежелательных символов и токенов (стоп-слов), стандартизация, токенизация и сегментация.

Очищенные и стандартизированные данные уже готовы для сравнения. Потенциально, каждая запись из базы должна быть сравнена с каждой такой же записью, таким образом, будет осуществлено операций, где n – количество записей в БД.

$$c = \frac{n * (n - 1)}{2} \quad (1)$$

Соответственно, на больших объемах данных, каждый последующий блок данных будет требовать все больше времени на вычисление. Однако специфика дедупликации данных заключается в том, что подавляющее число сравнений не приведут к обнаружению совпадения. Чтобы решить данную проблему, на 2 шаге, называемом разбиение на блоки, используются несколько различных техник, направленных на уменьшение количества пар, потенциально подходящих для сравнения.

Хотя на 1 шаге данные были очищены и стандартизированы, в них все еще могут содержаться ошибки или вариации. Например, 2 записи «Москва, Героев Панфиловцев» и «Россия, г. Москва, ул. Героев-Панфиловцев» относятся к 1 и тому же адресу. Для решения этой проблемы на 3 шаге для всей записи целиком или для каждого конкретного поля используют различные метрики сходства значений. Обычно, значения схожести нормализованы и находятся в пределах от 0 до 1, где 1 – полное совпадение, 0 – полное несовпадение, а

значения внутри диапазона характеризуют схожесть значения. Для разных атрибутов применяются разные метрики, которые подходят для атрибутов разных типов.

По сути, вся задача дедупликации сводится к бинарной классификации полученного на 3 шаге вектора схожести. Полученный результат впоследствии может быть обработан любым образом в соответствии с потребностями стейкхолдеров.

Основным документом, регулирующим обработку персональных данных в Российской Федерации, является Федеральный закон от 27 июля 2006 года № 152-ФЗ «О персональных данных». Целью данного Федерального закона является обеспечение защиты прав и свобод человека и гражданина при обработке его персональных данных, в том числе защиты прав на неприкосновенность частной жизни, личную и семейную тайну.

Для обеспечения требований российского законодательства необходимо выполнить 6 предусмотренных законом шагов:

1. Определение ответственных лиц.
2. Формирование перечня ПДн и ИСПДн.
3. Определение уровня защищенности ПДн, классификация ИСПДн. На данном этапе так же происходит определение угроз безопасности ПДн при их обработке в ИСПДн, разработка и применение технических мер по обеспечению безопасности ПДн, оценка эффективности принимаемых мер защиты ПДн в ИСПДн.
4. Формирование перечня должностей, списка лиц допущенных к обработке ПДн.
5. Подготовка политики обработки ПДн.
6. Уведомление Роскомнадзора.

Обзор программных решений в предметной области

В настоящее время существуют как открытые, так и коммерческие продукты, направленные на решение данной задачи. В данном разделе описаны некоторые программные продукты для сопоставления данных, большинство из которых создано в рамках научных работ и исследований новых алгоритмов сопоставления данных.

Единый клиент

На российском рынке представлено лишь одно решение, которое в данный момент используется крупнейшими банками («Сбербанк», «ВТБ» и другие), страховыми компаниями («Росгосстрах», «Альфа страхование», «ВТБ страхование»), предприятиями из ИТ и телеком секторов. Оно представляет собой коробочный продукт, устанавливаемый на оборудовании заказчика.

Решение основано на наборах правил, созданных специалистами компании-разработчика. Основная проблема данного подхода заключается в сложности поддержки и добавлении новых правил поиска дубликатов. Например, если в специфичной доменной области появится новый атрибут, по которому можно идентифицировать человека, потребуется доработка программного продукта под нужды заказчика

FEBRL

FreelyExtensibleBiomedicalRecordLinkage – система с открытым исходным кодом, разрабатываемая с 2003 года в Австралийском национальном университете совместно с Министерством здравоохранения Австралии. Данный программный продукт имеет графический интерфейс и поддерживает работу с CSV и TSV файлами.

Данный программный продукт позволяет проводить гибкую настройку параметров сравнения данных и классификации. Однако это же является и основной проблемой – для настройки требуются глубокие теоретические знания в области дедупликации данных, а для использования в промышленной среде необходима масштабная переработка архитектуры.

PythonRecordLinkageToolkit

«PythonRecordLinkageToolkit» – набор библиотек и компонентов для языка Python, реализующих основные технические и математические техники дедупликации данных. Для манипуляции данными эта библиотека использует numpy и pandas

Недостатком данной библиотеки является то, что она разработана для дедупликации небольших и средних наборов данных, а также то, что для использования данной библиотеки в промышленной среде потребуется разработка собственного сервиса на основе данной библиотеки.

Обзор существующих математических и технологических методов решения задачи

Стандартным подходом в данном случае является простой поиск записей, имеющих что-то общее с обрабатываемой записью. Количество записей, найденных в процессе поиска, зависит от частоты распределения используемых ключей поиска. В качестве упрощенного примера возьмем ключ, имеющий равномерное распределение. Тогда, общее число записей-кандидатов для сравнения, c , будет равно:

$$c = \frac{b * \left(\frac{n * n - 1}{b}\right)}{2} = \frac{n * (n - 1)}{2b}, \tag{2}$$

где n – общее количество записей, b – количество возможных значений используемого ключа.

Если b будет достаточно большим, это существенно повысит скорость дедупликации. Однако следует отметить, что оптимальный выбор ключей требует определенных знаний в предметной области

Другой подход, основанный на уменьшении количества возможных пар, «сортированное соседство» (sortedneighborhood) был представлен в 1995 году в работе Маурисио Эрнандеса и Сальватора Стольфо. Основная идея данного подхода заключается в сортировке базы данных определенным образом, а затем выборке фиксированного количества w ($w > 1$) записей [5; 6]. Сравнение данного и основанного на поиске методов представлено на рис. 2.

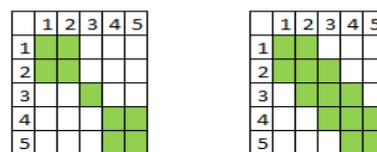


Рис. 2. Сравнение методов уменьшения количества пар.

Альтернативой данному семейству методов будет кластеризация записей алгоритмом, не требующим существенных вычислительных затрат, например, сапору. И хотя количество сравнений будет равно количеству сравнений без кластеризации, общее время работы будет все равно меньше.

Меры схожести значений

Самым простым примером может являться компаратор на основе полного совпадения значений. Формализовано он может быть представлен следующим образом:

$$d_{exact}(s1, s2) = \begin{cases} 1.0, & \text{если } s1 = s2 \\ 0.0, & \text{если } s1 \neq s2 \end{cases} \tag{3}$$

Одним из основных применяемых методов являются меры, основанные на расстоянии редактирования. Самым известным из них является расстояние Левенштейна – минимальное число односимвольных операций (вставки, удаления, замены), необходимое для преобразования одной последовательности символов в другую. Пусть s_1 и s_2 – две строки (длиной M и N соответственно), тогда редакционное расстояние (расстояние Левенштейна) $d(s_1, s_2)$ можно подсчитать по следующей рекуррентной формуле:

$$D(M, N) = \begin{cases} d(s_1, s_2) = D(M, N), \text{ где} \\ \quad 0, \text{ если } i = 0, j = 0 \\ \quad i, \text{ если } j = 0, i > 0 \\ \quad j, \text{ если } i = 0, j > 0 \\ \min \left(\begin{matrix} D(i, j - 1) + 1, \\ D(i - 1, j) + 1, \\ D(i - 1, j - 1) + m(s_1[i], s_2[j]) \end{matrix} \right), \text{ если } i > 0, j > 0 \end{cases}, \quad (4)$$

где $m(a, b)$ равна нулю, если $a=b$ и единице в противном случае; $\min(a, b, c)$ возвращает наименьший из аргументов. Для нормализации значений расстояния редактирования используется следующая формула:

$$d_{levenshtein}(s_1, s_2) = 1.0 - \frac{d(s_1, s_2)}{\max(|s_1|, |s_2|)} \quad (5)$$

Другое семейство эвристических алгоритмов было разработано Мэтью Джаро и Вильямом Винклером. Их методы лучше расстояния Левенштейна подходят для сравнения именованных сущностей. В своих работах [13] они представили следующие формулы:

$$d_{jaro}(s_1, s_2) = \frac{1}{3} \left(\frac{c}{|s_1|} + \frac{c}{|s_2|} + \frac{c-t}{c} \right) \quad (6)$$

$$d_{jaro_winkler}(s_1, s_2) = d_{jaro}(s_1, s_2) + \left(1.0 - d_{jaro}(s_1, s_2) \right) \frac{p}{10},$$

где c – количество совпадающих символов, стоящих не дальше, чем друг от начала строки

$$\frac{\max(|s_1|, |s_2|)}{2} - 1 \quad (7)$$

t – половина числа транспозиций (замены мест двух соседних символов, в результате которой получается одинаковая подстрока – «ao» и «oa»), $p = \min(4, z)$, z – наибольшая общая начальная подстрока.

Еще один метод условного сравнения был разработан американскими учеными в области информационных технологий Альваро Монжем и Чарльзом Элканом специально для вычисления сходства строк, состоящих из нескольких строк [15]. На первом шаге строки разбиваются на отдельные токены, а затем, используя внешнюю метрику сходства, происходит поиск среднего значения из максимально совпадающей пары токенов из 2 строк:

$$d_{monge_elkan}(s_1, s_2) = \frac{1}{A \vee \sum_{i=1}^{AVBV} \max_{j=1 \sim t(A_i B_j)} d} \quad (8)$$

Классификация записей

Самый простой вариант реализации бинарного классификатора – использование порогового значения. Полученный на предыдущем шаге вектор схожести суммируется, а затем сравнивается с заданным пороговым значением. Формально:

$$\begin{aligned} \sum (w) \geq t &\rightarrow \text{дубликат} \\ \sum (w) < t &\rightarrow \text{недубликат} \end{aligned} \quad (9)$$

где w – вектор схожести, t – заданное пороговое значение. Похожим образом так же работает классификация на основе взвешенной суммы схожести.

Другой подход к классификации – использование предварительно заданных правил, которые задаются экспертом в конкретной предметной области [11]. Набор правил может быть применен как к вектору схожести, так и к исходным данным. Формально правила могут быть представлены в виде булевого предиката:

$$P = (\text{утверждение1} \cup \text{утверждение 2}) \cap \text{утверждение 3} \quad (10)$$

Большое количество задач не может быть решено заранее определенными методами или алгоритмами. Это происходит по той причине, что заранее не известны механизмы порождения исходных данных или же известная нам информация недостаточна для построения модели источника, генерирующего поступающие к нам данные.

В этих условиях ничего не остается, как только изучать доступную нам последовательность исходных данных и пытаться строить предсказания, совершенствуя нашу схему в процессе предсказания. Подход, при котором прошлые данные или примеры используются для первоначального формирования и совершенствования схемы предсказания, называется методом машинного обучения [12].

Различают 2 основных подхода к машинному обучению: обучение с использованием предварительно размеченных данных или «обучение с учителем» (supervised learning), обучение без этих данных, или «обучение без учителя» (unsupervised learning). Применительно к задаче классификации обычно рассматривают именно первый подход, второй же используется в основном для кластеризации данных.

В последнее десятилетие применение различных подходов из данной области начало активно изучаться в контексте задачи дедубликации данных. Рассмотрим некоторые из них.

Один из наиболее популярных методов обучения в задачах классификации – метод опорных векторов (support vector machine). Основная идея заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным образом. Допустим, имеется выборка предварительно размеченных данных $(x_1, y_1), \dots, (x_m, y_m)$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$ (в зависимости от того, какому классу принадлежит точка). Метод опорных векторов строит классифицирующую функцию F в виде:

$$F(x) = \text{sign}(w, x > +b), \quad (11)$$

где \langle, \rangle – скалярное произведение, w – нормальный вектор к разделяющей гиперплоскости, b – вспомогательный параметр. Те объекты, для которых $F(x) = 1$ попадают в один класс, а объекты с $F(x) = -1$ – в другой. Выбор именно такой функции неслучаен: любая гиперплоскость может быть задана в виде $\langle w, x \rangle + b = 0$ для некоторых w и b .

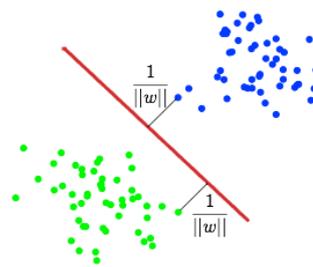


Рис. 3. Иллюстрация метода опорных векторов.

Далее, мы хотим выбрать такие w и b которые максимизируют расстояние до каждого класса. Можно подсчитать, что данное расстояние равно $\frac{1}{\|w\|}$. Проблема нахождения максимума $\frac{1}{\|w\|}$ эквивалентна проблеме нахождения минимума $\|w\|^2$. Запишем все это в виде задачи оптимизации:

$$\|w\|^2 \rightarrow \min_{i=1, \dots, m} (wx_i > +b) \geq 1, \quad (12)$$

Алгоритм построения оптимальной разделяющей гиперплоскости, предложенный в 1963 году Владимиром Вапником и Алексеем Червоненкисом — алгоритм линейной классификации. Однако в 1992 году Бернхард Босер, Изабель Гийон и Вапник предложили способ создания нелинейного классификатора, в основе которого лежит переход от скалярных произведений к произвольным ядрам, так называемый kerneltrick (предложенный впервые М. А. Айзерманом, Э. М. Браверманом и Л. И. Розоноэром для метода потенциальных функций), позволяющий строить нелинейные разделители. Результирующий алгоритм крайне похож на алгоритм линейной классификации, с той лишь разницей, что каждое скалярное произведение в приведённых выше формулах заменяется нелинейной функцией ядра (скалярным произведением в пространстве с большей размерностью). В этом пространстве уже может существовать оптимальная разделяющая гиперплоскость.

Еще одним методом в обучении с учителем являются деревья решений. По сути, деревья решений являются объединением логических утверждений в структуру данных типа «дерево». Отличием от классификации на основе правил является то, что они строятся автоматически на основе принципа уменьшения информационной энтропии – меры неопределенности некоторой системы. Для независимых случайных событий x с n возможными состояниями, распределенных с вероятностями $p_i (i=1, \dots, n)$ энтропия высчитывается по формуле Шеннона:

$$S = - \sum_{i=1}^N p_i \log_2 p_i \quad (13)$$

где p_i – вероятность нахождения системы в i -ом состоянии.

Основным достоинством данного подхода является то, что он хорошо поддается интерпретации человеком. Однако его недостаток заключается в том, что построенное дерево ре-

шений может плохо работать для новых объектов, поскольку оно идеально подстроено под обучающие данные. Для решения этой проблемы ограничивают глубину дерева [14].

Упоминания так же заслуживает метод поиска к ближайших соседей. Идея данного алгоритма заключается в поиске к ближайших к классифицируемому объекту точек. Если среди них преобладает 1 класс, наиболее вероятно, что классифицируемый объект принадлежит к нему же [10].

Последним упомянутым методом будут нейронные сети. Это понятие возникло при попытке смоделировать процессы, происходящие в мозге. В настоящее время существует большое количество различных архитектур нейронных сетей, но в качестве примера будет рассмотрен только перцептон – искусственная нейронная сеть, состоящая из 2 и более слоев – слоя входных данных и скрытых слоя. Данные слоя умножаются на определенные веса, затем суммируются, и передаются на вход следующему слою. Веса нейронной сети являются аналогом синапсов, суммирующий блок представляет собой нейрон. К результирующим данным из скрытого слоя обычно так же применяется какая-либо функция активации, например, линейный выпрямитель(ReLU):

$$f(x) = \begin{cases} 0, & x < 0, \\ x, & x \geq 0 \end{cases} \quad (14)$$

Математически, этот процесс может быть представлен как произведение вектора признаков на некую матрицу, к которому затем применена функция активации. Количество столбцов матрицы равно количеству признаков вектора, а количество строк – количеству нейронов. Обучение нейронной сети представляет собой процесс подбора весов таким образом, чтобы значение на выходе соответствовало предварительно известному результату. Обычно для этой цели используют стохастический градиентный спуск и алгоритм обратного распространения ошибки [9].

Выводы

Была дана характеристика предметной области. Произведен обзор и анализ существующих программных решений в исследуемой предметной области. Большинство из них являются не готовыми программными продуктами, а библиотеками, позволяющими решить поставленную задачу, что требует высокого уровня теоретической подготовки программиста. Выполнен обзор и анализ существующих математических и технологических методов решения задачи. Также предложена концепция программы, которая сможет решить поставленную задачу без специальных знаний в обозреваемых областях математики и программирования.

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон «О персональных данных»: федер. закон: [принят Гос. Думой 8 июля 2006 года: одобр. Советом Федерации 14 июля 2006 года]: офиц. текст с изм. и доп. на 2020 г. URL: https://www.consultant.ru/document/cons_doc_LAW_61801.
2. Вовченко А.Е., Калиниченко Л.А., Ковалев Д.Ю. Методы разрешения сущностей и слияния данных в ETL-процессе и их реализация в среде Hadoop // Информатика и её применения. 2014. Т. 8. №. 4. С. 94-109.
3. Вьюгин В.В. Математические основы машинного обучения и прогнозирования. М.: МЦНМО, 2013. 484 с.
4. Гильманшина А.Л. Анализ литературы и нормативно-правовой базы в области защиты персональных данных // Вестник науки. 2019. Т. 2. №. 3. С. 13-16.
5. Гребнев К.Н. Машинное обучение с помощью библиотеки scikit-learn языка Python // Математический вестник педвузов и университетов Волго-Вятского региона. 2017. №. 19. С. 277-281.

6. Дивенкова Д.В., Мигунова Л.Е., Филимоненкова Н.В. Использование метрик при распознавании словоформ // Неделя науки СПбПУ. 2018. С. 217-221.
7. Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академии наук. Российская академия наук. 1965. Т. 163. №. 4. С. 845-848.
8. Миронов А.М. Машинное обучение. Часть 1. М.: МАКС Пресс. 2018. 84 с.
9. Решетников А.Д. О подходах для определения меры несходства в текстовых данных // Вестник Воронежского Института Высоких Технологий. 2019. №3. С.35-38
10. Съедин Д.Ю. Разработка и реализация алгоритма связывания данных в государственной информационной системе гражданского назначения // Научно-техническая информация. Серия 2: Информационные процессы и системы. 2018. №. 7. С. 32-39.
11. Торвальдсен Г. Связывание записей в историческом регистре населения Норвегии // Историческая информатика. 2019. № 2. С.212-231.
12. Christen P. Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection // Springer Science & Business Media. 2012. 272 p.
13. Jaro M.A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida // Journal of the American Statistical Association. 1989. Т. 84. №. 406. С. 414-420.
14. Haug A., Zachariassen F., Van Liempd D. The costs of poor data quality // Journal of Industrial Engineering and Management (JIEM). 2011. Т. 4. №. 2. С. 168-193.
15. Monge A.E. et al. The Field Matching Problem: Algorithms and Applications /, A.E Monge, C.P. Elkan // Kdd. 1996. Vol. 2. P. 267-270.

© *Вартанян А.А., Хабарова М.А., Редько Е.А., Русаков А.М., 2023.*