УДК 004.9
doi:10.21685/2227-8486-2023-1-9

АРХИТЕКТУРА СИСТЕМЫ КЛАСТЕРИЗАЦИИ ПАССАЖИРОВ

А. Д. Столяров[1], В. В. Гордеев[2], В. И. Абрамов[3]

[1, 3] Национальный исследовательский ядерный университет МИФИ, Москва, Россия
[2] АЭРОЛАБС, Москва, Россия
[1] mr.alexst@gmail.com, [2] v.gordeev@aerolabs.aero, [3] viabramov@mephi.ru

Аннотация. *Актуальность и цели.* В работе поднимается проблема выстраивания архитектуры программного обеспечения для первичной обработки данных о пассажирах авиакомпании, их структурирования и дальнейшей кластеризации с учетом отраслевой специфики. *Материалы и методы.* Для решения поставленной задачи были изучены лучшие практики построения нагруженных систем работы с большими данными, выделены наиболее перспективные с точки зрения развития и взаимной интеграции. *Результаты*. Была успешно реализована на практике выбранная в результате работы архитектура построения системы кластеризации пассажиров, которая показала себя эффективной и надежной. *Выводы*. Описанный подход рекомендуется к применению как хорошо себя зарекомендовавший в работе с большими массивами отраслевых данных. Подход позволит гибко масштабировать систему обработки данных и подключать к ней в дальнейшем иные модули.

Ключевые слова: рекомендательные сервисы, кластеризация данных, машинное обучение, цифровая трансформация, цифровые технологии, структурирование данных, архитектура программного обеспечения

Финансирование: работа была выполнена при поддержке Федерального государственного бюджетного учреждения «Фонд содействия развитию малых форм предприятий в научно-технической сфере» по договору (соглашению) № 147ГРЦТС10-D5/61890 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ.

Для цитирования: Столяров А. Д., Гордеев В. В., Абрамов В. И. Архитектура системы кластеризации пассажиров // Модели, системы, сети в экономике, технике, природе и обществе. 2023. № 1. С. 136–148. doi:10.21685/2227-8486-2023-1-9

ARCHITECTURE OF THE PASSENGER CLUSTERING SYSTEM

A.D. Stolyarov[1], V.V. Gordeev[2], V.I. Abramov[3]

[1, 3] National Research Nuclear University MEPhI, Moscow, Russia
[2] Aerolabs LLC, Moscow, Russia
[1] mr.alexst@gmail.com, [2] v.gordeev@aerolabs.aero, [3] viabramov@mephi.ru

Abstract. *Background.* The paper raises the problem of building a software architecture for the primary processing of airline passenger data, data structuring and further clustering, taking into account industry specifics. *Materials and methods.* To solve this problem, the best practices of building loaded systems for working with big data were studied, the most promising from the point of view of development and mutual integration were identified. *Results.*

The architecture of the passenger clustering system chosen as a result of the work was successfully implemented in practice and proved to be effective and reliable. *Conclusions.* The described approach is recommended for use as well-proven in working with large arrays of industrial data. The approach will allow flexible scaling of the data processing system and connection of other modules to it in the future.

Keywords: recommendation services, data clustering, machine learning, digital transformation, digital technologies, data structuring, software architecture

Acknowledgments: the work was carried out with the support of the Federal State Budgetary Institution "Fund for Assistance to the Development of Small Forms of Enterprises in the Scientific and Technical Sphere" under contract (agreement) № 147GRTS10-D5/61890 on the provision of a grant for research and development work.

For citation: Stolyarov A.D., Gordeev V.V., Abramov V.I. Architecture of the passenger clustering system. *Modeli, sistemy, seti v ekonomike, tekhnike, prirode i obshchestve = Models, systems, networks in economics, technology, nature and society*. 2023;(1):136–148. (In Russ.). doi:10.21685/2227-8486-2023-1-9

Введение

В условиях BANI-мира (акроним от английских слов: хрупкий, тревожный, нелинейный и непонятный), сложившихся в результате развития Индустрии 4.0, и становления шестого технологического уклада требуются новые подходы к управлению компаниями, основанные на активном использовании новых бизнес-моделей и цифровых технологий. Антироссийские санкции внесли дополнительную неопределенность в развитие событий, и для России в сложившихся геополитических условиях и особенно при возрастающем санкционном давлении проведение политики импортозамещения должно быть направлено на развитие экономической деятельности и повышение инновационной активности предприятий [1]. Задача цифровой трансформации экономики и увеличения темпов экономического развития страны актуальна как никогда, поэтому требуются иные подходы к управлению с использованием инновационных цифровых технологий, дающих новые способы наращивания эффективности работы предприятий. Важным условием и фактором успешного проведения цифровой трансформации является повышение цифровой зрелости, которое выражается в степени готовности предприятия к запланированным переменам [2].

Показано, что стратегическая задача цифровой трансформации компании заключается в построении практически жизнеспособного цифрового двойника, который будет описывать взаимосвязь между цифровыми активами и видами деятельности, моделируя взаимодействие между различными источниками данных в организации [3]. Разработка рекомендательной системы для повышения эффективности работы с клиентами является важным этапом при цифровизации бизнес-процессов компании. Анализ данных, которыми располагает авиакомпания, и кластеризация клиентов на основе этих данных позволяет получить богатый «маркетинговый» портрет пользователя и сформировать уникальную рекомендательную систему, потенциально способную значительно увеличить объем продаж дополнительных сервисов самого авиаперевозчика и его партнеров. В связи с этим актуальна задача превращения большого объема неструктурированных данных в потенциально обозримый набор кластеров, значимых с точки зрения потребительских характеристик, входящих в них клиентов. Наиболее релевантными подходами для решения данной

задачи являются методы машинного обучения, а именно семейство методов обучения без учителя. Подобные методы широко используются в задачах, не имеющих четкой разметки данных и требующих поиска закономерностей в больших объемах данных. Для успешной реализации системы кластеризации пассажиров необходима разработка хранилища данных и системы резервирования данных.

Материалы и методы

Основной задачей системы кластеризации является формирование на основе предоставленных данных групп пользователей, значимых с точки зрения потребительских характеристик, входящих в них клиентов. Разработанная система кластеризации, реализованная в виде отдельного модуля, является первым блоком в создании комплексного продукта и для полноценной работы будет требовать дополнительный данных из разрабатываемых на следующих этапах систем. Для этого на текущем этапе разработки система кластеризации должна обладать архитектурой, которая бы позволила с легкостью ее в будущем интегрировать с другими модулями программного обеспечения, поэтому решено было ее исполнить в виде серверного приложения, наполняющего свою БД с использованием данных, полученных с других серверов в текстовом виде.

Используемые для кластеризации данные по умолчанию не структурированы и не полны, ввиду чего требуется их предварительная обработка и структурирование. Эти функции выполняет дополнительно разработанное серверное приложение. Приложения подобного класса принято называть «парсерами» (от англ. parser – «анализатор»). Парсер, используя данные, полученные с удаленного сервера, структурирует и связывает их, после чего записывает в базу данных.

Записанные в базу структурированные данные уже поступают на вход непосредственно модуля кластеризации, выполняющего поиск оптимального их разбиения. Схематичное представление описанной структуры изображено на рис. 1.

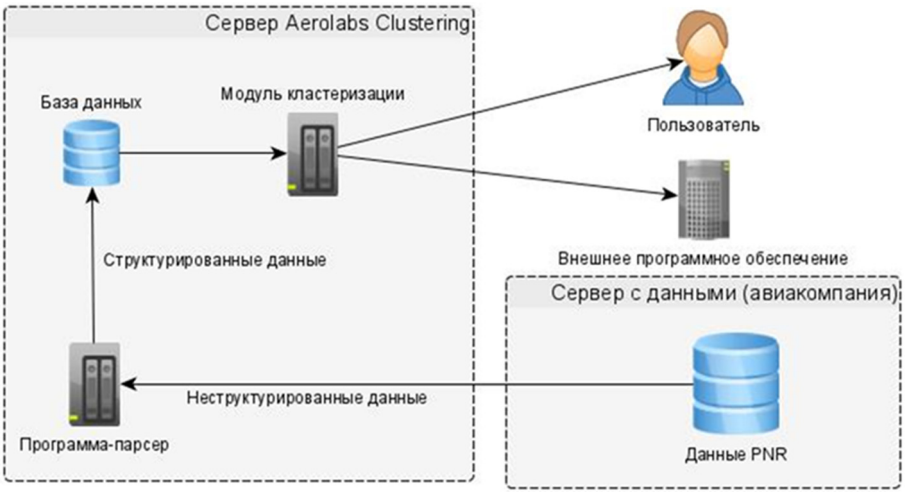

Рис. 1. Место модуля кластеризации в программном обеспечении

В качестве основного метода кластеризации использовался метод K-средних. Для определения оптимального количества кластеров использовался «метод локтя» с автоматизированным подбором оптимального количества кластеров, который осуществлялся по следующему алгоритму:

– последовательное разбиение массива данных на кластеры от 1 до 20;

– расчет на каждом шаге внутрикластерной суммы квадратов остатков (WCSS);

– построение графика («локтя»), где по оси абсцисс откладывалось число кластеров, а по оси ординат – WCSS;

– строилась касательная к графику, проходящая под углом 135°, определялась координата точки касания по оси абсцисс;

– в качестве оптимальных бралось число кластеров, равное ближайшим целым числам справа и слева от полученной координаты.

Полученные количества кластеров далее проверялись на предмет качества кластеризации с помощью ключевых метрик:

– наполненность кластеров – не менее 10 % (не должно быть кластеров, в которые вошло бы менее 10 % совокупности);

– четкость кластеров не менее 0,8 (отношение пограничных точек к общему количеству точек);

– устойчивость кластеров не менее 0,9, вычисляемая методом Кауфмана.

В случае если ни одна из двух взятых значений кластеров не проходила проверку по данным критериям, брались следующие по удаленности числа кластеров.

В литературе применительно к кластеризации клиентов встречаются различные подходы к кластеризации. Множество работ посвящено не столько алгоритмам кластеризации, сколько выявлению релевантных и технических параметров для дальнейшей логической сегментации. Нас же интересуют работы, связанные именно с методологией кластеризации в маркетинговой сфере. Так, описан подход из близкой к нам сферы кластеризации участников программы лояльности [4]. В данном подходе не используются автоматизированные средства и выполняется скорее сегментация клиентской базы на основе формальных правил, что не применимо для обучающихся рекомендательных систем. Больше всего работ по кластеризации клиентов посвящено банковскому сектору. В найденных работах для кластеризации применяется также алгоритм K-means [5, 6], однако в данных подходах не применяется автоматизированный подбор количества кластеров, равно как не разрабатываются критерии качества кластеризации, из-за чего они ограничиваются только ручным применением. Для кластеризации применяется также метод DBSCAN [6] с аналогичными методическими недостатками, результаты кластеризации которого на практике не уступают методу K-means. Применяются для кластеризации клиентских баз карты Кохонена [7]. Данный метод нами тоже исследовался, однако дал распределение данных по кластерам близкое к случайному. Это объясняется тем, что карты Кохонена предназначены прежде всего для работы с количественными данным, тогда как множество данных о пассажирах авиакомпании являются качественными.

Ключевая новизна метода заключается в том, что он позволяет автоматизировано проводить перекластеризацию данных без участия пользователя. Так как клиентская база растет и обогащается новыми транзакциями и в том числе данными по уже существующим в ней пользователям, разбиение клиентов на кластеры делается ежедневно, что оптимально достижимо с разработанным подходом. При этом используемые методы достаточно просты с точки зрения проводимых вычислений и не создают большой нагрузки на оборудование. Кластеризация, как правило, используется для исследования аналитиком существующего набора данных, и поэтому необходимость проведения регулярной кластеризации обновляемого набора данных достаточно редка. Однако для автоматизации кластеризации существуют определенные подходы, например на основе карт Кохонена [8], которые, как уже было сказано, в нашем случае (небольшое число количественных параметров) слабо применимы. В подавляющем же количестве случаев задача автоматизации кластеризации применяется для работы с текстовой информацией (документами) [9–14]. Однако подходов, которые были бы применимы для автоматизированной кластеризации в целях маркетинга, найдено не было.

Результаты и обсуждение

Система была разработана на основе библиотеки машинного обучения Scikit-learn на языке Python. Некоторые алгоритмы написаны на Cython для повышения производительности [15]. Scikit-learn хорошо интегрируется со многими другими библиотеками Python, такими как matplotlib и plotly для построения графиков и представлении визуализации, numpy для векторизации массивов, pandas dataframes, scipy и др. [16]. Первым делом требуется обработка данных, поступающих из базы данных с помощью SQL-запросов. Это возможно благодаря использованию библиотеки sqlalchemy, которая открывает данную возможность [17].

Далее для стандартизации (представлении в числовом формате) и нормализации (расположении численных значений в диапазоне от 0 до 1) данных использовался пакет «preprocessing» библиотеки Scikit-learn, которая уже имеет реализованные тесты, обрабатывающие информацию о процессах работы реализованных методов[1]. Благодаря этому возможно детектировать неточности в работе алгоритмов, а также выявлять ошибки.

Сама кластеризация реализована с использованием пакета K-means, которая включает в себя полное тестирование работы одноименного итерационного алгоритма кластеризации. Отслеживается качественное содержание батчей (набора данных, передаваемых для присвоения кластеров), а также этапы миграции центров кластеров и количество проведенных итераций для достижения оптимальных центров кластеров [18].

Основываясь на формате входных данных, была разработана следующая инфологическая модель данных, содержащая три основных аспекта – бронирование, пассажир, рейс (рис. 2).

[1] Документация к Scikit Learn: Preprocessing data. URL: https://scikit-learn/stable/modules/preprocessing.html (дата обращения: 12.09.2022).

Рис. 2. Инфологическая модель данных

Ввиду наличия ряда стандартного вида связей (one-to-many, many-to-many) и необходимости учитывать их при выборках данных в качестве типа хранилища данных была выбрана реляционная база данных.

Задача серверной части – формирование базы данных и структурирование информации и обеспечения к ней доступа по REST API (от англ. Application Programming Interface – «программный интерфейс приложения»). API представляет собой описание способов, используя которые, одна компьютерная программа может взаимодействовать с другой. REST (от англ. Representational State Transfer – «передача состояния представления») – наиболее популярный и признанный во всем мире подход к проектированию взаимодействия распределенных приложений в сети Интернет. Для внедрения данного подхода обязательно выполнение следующих принципов [19]:

– клиент-серверная модель;

– разграничение состояний (состояния, относящиеся к клиенту хранятся на стороне клиента);

– кэширование (сохранение ответов на запросы у клиента);

– многослойность (возможность добавления в систему посредников, выполняющих разные функции);

– код по запросу (некоторые приложения могут скачиваться и выполняться на стороне клиента для снижения нагрузки на сервер).

В качестве основного языка разработки серверной части выбран Python ввиду популярности, высокоуровневости и наличия большого количества библиотек.

Поскольку само по себе создание веб-сервера с REST API на Python является стандартной задачей, была использована хорошо зарекомендовавшая

себя связка flask (фреймворк) – gunicorn (веб-сервер) – nginx (прокси) [20]. Взаимодействие фреймворка и веб-сервера происходит по протоколу WSGI, обеспечивающему унификацию интерфейсов фреймворков и веб-серверов и позволяющему распараллеливать запросы, доверяя их обработку предварительно созданным процессам-рабочим. Фреймворк flask, в свою очередь, был выбран, с одной стороны, из-за простоты создания прототипа приложения, с другой – в связи с обилием возможностей по расширению функционала с использованием множества внешних модулей при необходимости.

В качестве СУБД выбрана PostgreSQL ввиду ее высокой производительности и богатства функционала и расширений [21]. Система развернута в сервисе Amazon Web Services (AWS), на машине конфигурации t2.micro (1 виртуальное ядро, 1 ГБ ОЗУ). ОС – Red Hat Enterprise Linux 8. База данных развернута в RDS.

Для удобства работы с данными используется ORM (англ. Object-Relational Mapping, объектно-реляционное отображение) SQLAlchemy. Она представляет собой слой абстракции, позволяющий сосредоточиться на логике обработки данных, а не на правильном построении SQL-запросов – как правило, запросы типичны и повторяют друг друга, а для особых случаев, например когда необходимо особое связывание, join, таблиц, без которого работа алгоритма замедлится в сотни раз, всегда можно задать особые правила построения запроса [22].

Для надежности отслеживания и применения миграций (изменения структуры) БД используется инструмент Alembic. Он позволяет вести контроль всех изменений структуры БД и применять эти изменения. При этом Alembic позволяет генерировать широкий спектр типов миграций прямо из изменений кода объектов ORM [23].

Данный набор технологий позволит эффективно хранить данные в базе данных и обеспечивать оптимальную обработку поступающих на сервер запросов. Созданная структура также позволит гибко дорабатывать и далее масштабировать разработанное программное обеспечение.

Вторым аспектом разработки является преобразование данных, получаемых от клиентов в формат, соответствующий архитектуре.

В результате анализа информационных баз данных клиентов-авиакомпаний было принято решение остановиться на информации, содержащийся в записях о PNR (Passenger Name Record). PNR представляют собой записи о маршруте пассажира или группы пассажиров и содержат полные данные о бронировании, включая класс полета, данные о питании и т.д.

Записи PNR жестко регламентированы отраслевым стандартом Advance Passenger Information (API) Guidelines[1], разработанным и поддерживаемым:

- Международной ассоциацией воздушного транспорта (IATA);
- Международной организацией гражданской авиации (ICAO);
- Всемирной таможенной организацией (WCO).

Данный стандарт унифицирует передаваемые авиакомпаниями сообщения о PNR, обеспечивая взаимную интегрированность всех CRS систем (Computer Reservation System).

[1] Guidelines on advance passenger information (API) / WCO/IATA/ICAO. 2014.

Данные поступают на вход в виде наборов текстовых файлов – по одному файлу каждого типа для каждого дня. Название всех файлов стандартное и может быть записано маской PNR_A_BC.txt, где:

– PNR – постоянно присутствующая в имени всех файлов часть, относящая их к данным PNR;

– A – сюда подставляется код авиалиний;

– B – дата, записи которой содержит данный файл в формате ГГГГММДД (без пробелов);

– C – код, характеризующий тип данных, содержащихся в файле. Пишется слитно с датой без пробела.

Таким образом, например, файл, содержащий информацию о пассажирах на дату 01.02.2016, имеет имя «PNR_9U_20160201PAX.TXT».

Файлы, относящиеся к одному типу данных, всегда имеют одинаковую структуру данных. В качестве основных используются следующие файлы данных (код «C»):

– PAX – файл, содержащий базовую информацию о пассажирах – о группе в одной брони (поле «PAX_NMBR»), фамилию/имя с указанием пола или статус «ребенок» (PAX_NAME);

– SSRS – файл, содержащий дополнительную информацию, связанную с билетами пассажиров;

– RES – файл, содержащий информацию о бронированиях. Ключевой интересующей нас характеристикой является BOOKING_CITY_CD – город, в котором приобретен билет (либо регион сайта-агента);

– RES_CONTACTS – файл, содержащий контактную информацию пассажиров – номер телефона (H) и e-mail (E). Особенностью этих данных является то, что они зачастую могут быть заполнены турагентом неверно или заполнены контактами самого турагента, поэтому не всегда могут быть использованы для надежной идентификации записей;

– RES_INF_TKT – файл, содержащий информацию о младенцах на рейсе. Информация важна тем, что пассажиру с младенцем, скорее всего, понадобится, например, такая услуга, как такси с детским креслом;

– RES_LEGS – файл, содержащий информацию о «плечах», т.е. единичных перелетах в составе маршрута. Кроме данных о самих перелетах (время и места вылета/прилета) данный файл содержит важную информацию относительно класса бронирования – эконом/бизнес;

– FARE_QUOTE – файл, содержащий качественную информацию о билетах, включая их стоимость;

– HISTORY_LEGS – файл, содержащий информацию о предыдущих полетах;

– PAX_SEATS – файл, содержащий места, на которые приобретены билеты;

– PAX_SERVICES – файл, содержащий информацию о дополнительных сервисах.

Всего в дневной выгрузке присутствует 21 тип файлов PNR, но перечисленные выше для наших целей являются основными.

Данные из всех таблиц были увязаны между собой, для чего использовалось поле PNR, которое являлось ID брони, а в некоторых случая дополнительно имена пассажиров в формате «ФАМИЛИЯ/ИМЯ (ОБРАЩЕНИЕ)».

В итоге разработанная структура базы данных имеет вид, представленный на рис. 3.

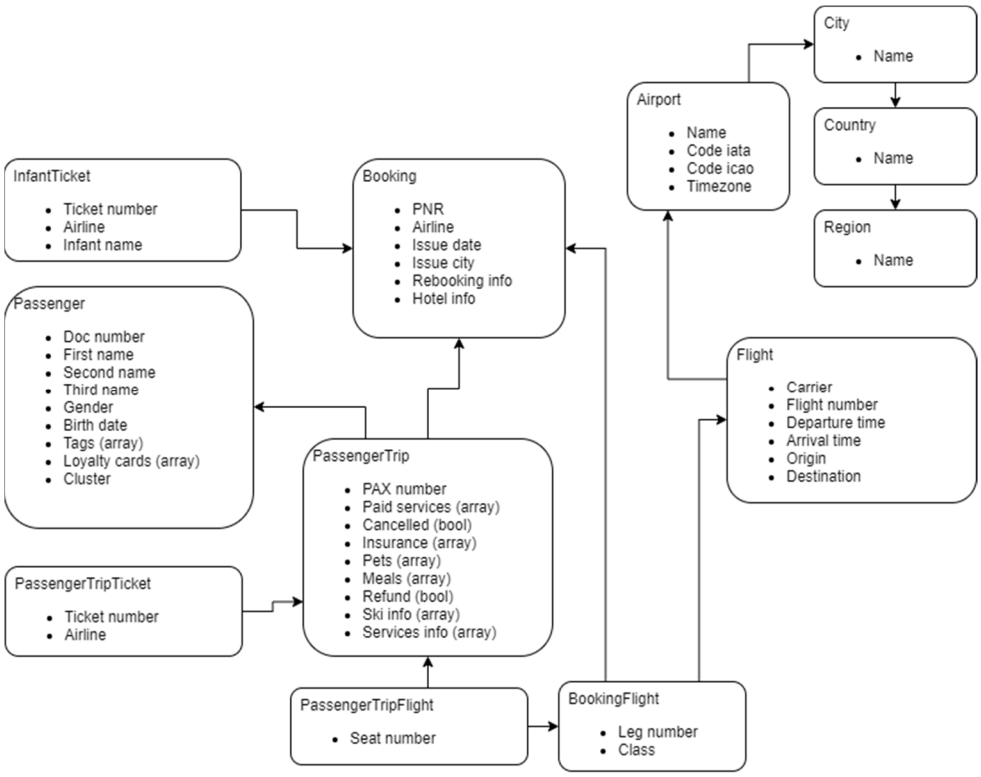

Рис. 3. Структура данных системы кластеризации

Очевидно, центральной моделью данных является пассажир (модель Passenger) – его характеристики вычисляются путем обработки относящихся к нему бронирований и всего, что с ними связано. Две другие «корневые» (ни на что не ссылающиеся) модели – это Booking (бронирование, уникально характеризуется по совокупности PNR и имени авиакомпании) и Flight (рейс, уникально характеризуется по авиакомпании, номеру рейса и дате вылета). Просто так связать пассажира и бронирование нельзя – в брони может быть несколько человек, поэтому вводится модель PassengerTrip, которая указывает на пассажира в брони (PNR + PAX_NMBR). Рейс связывается с бронированием через модель BookingFlight (PNR + LEG_NMBR), и в конце концов PassengerTrip связывается с BookingFlight через PassengerTripFlight (PNR + + PAX_NMBR + LEG_NMBR). Билеты привязываются к PassengerTrip или к Booking.

В процессе обработки данных также была обнаружена проблема с наполнением PAX_SEATS – файла, который содержит информацию о местах пассажиров. Хотя эти файлы почти всегда присутствуют, записей в них крайне мало – иногда на порядок меньше, чем в других файлах, поэтому было принято решение связывать BookingFlight и PassengerTrip записями-заглушками (имеющими ссылки на другие записи, но не имеющими информации в своем теле) при отсутствии данных для заполнения «правильного» PassengerTripFlight.

В результате проведенной кластеризации в качестве оптимальных разбиений, соответствующих заданным формальным критериям, были разбиения на четыре и на пять кластеров. По завершении кластеризации осуществлялась также формально-логическая проверка данных, включенных в кластеры на предмет наличия в них неких определяющих данные кластеры признаков. В разбиении на пять кластеров таких признаков обнаружено не было – кластеры внешне были похожи на случайное распределение, поэтому в качестве оптимального было выбрано разбиение на четыре кластера. Данное разбиение характеризовалось следующими ключевыми метриками:

- наполненность кластеров более 20 %;
- коэффициент четкости 0,82;
- устойчивость кластеров 0,91.

Разработанная система кластеризации тестировалась в составе программного обеспечения для генерации персональных предложений. Разработанная архитектура позволила успешно интегрировать систему кластеризации с иными модулями программного обеспечения. Тестирование скорости работы системы кластеризации не проводилось, так как она работает асинхронно с другими модулями и по расписанию, не вызывая задержек в работе других модулей программного обеспечения. Отсутствие задержек в работе говорит об оптимальной скорости работы системы кластеризации.

Заключение

В результате проведенных исследований была реализована и испытана архитектура, предлагаемая для реализации промышленного программного обеспечения для структурирования, обработки и кластеризации данных пассажиров авиакомпаний. Используемая архитектура позволяет программному обеспечению данные о пассажирах из PNR-записей, формируемых внутренними системами авиакомпании в текстовом виде, преобразовывать в структурированную базу данных, позволяющую выполнять дальнейшую обработку штатными методами. Разработанная архитектура позволяет работать с большими объемами данных (стандартная дневная выгрузка составляет несколько гигабайт текстовых записей), позволяя гибко масштабировать нагрузку. Описанное решение может выступать как самостоятельное программное обеспечение для предварительной обработки данных, а также работать в составе комплекса программных модулей, обеспечивая структурирование входящего потока информации.

Реализованная с помощью разработанного программного обеспечения архитектура дала соответствующее всем заданным критериям разбиение данных на четыре кластера.

Список литературы

1. Абрамов В. И., Маркина Ю. В., Камынин Д. А. Реализация политики импортозамещения как фактор активизации инновационных процессов // Экономика и предпринимательство. 2017. № 12-1. С. 134–137.
2. Абрамов В. И., Борзов А. В., Семенков К. Ю. Оценка готовности малых и средних предприятий к цифровой трансформации // Вопросы инновационной экономики. 2022. Т. 12, № 3. doi:10.18334/vinec.12.3.115000

3. Абрамов В. И., Бобоев Д. С., Гильманов Т. Д., Семенков К. Ю. Теоретические и практические аспекты создания цифрового двойника компании // Вопросы инновационной экономики. 2022. Т. 12, № 2. С. 967–980. doi:10.18334/vinec.12.2.114890
4. Белоцерковская М. Г. Кластеризация клиентской базы участников программы лояльности // Московский экономический журнал. 2017. № 2. С. 112–119.
5. Задворная И. А., Ромакина О. М. Применение алгоритма «кластеризация» для анализа данных потенциальных клиентов банка // Ученые записки Брянского государственного университета. 2019. № 2. С. 7–15.
6. Кудашкин А. В., Мохов А. С. Кластеризация клиентов банка на основе их персональных данных и банковских транзакций // Информационные системы и технологии ИСТ-2020. 2020. С. 780–785.
7. Ляховец А. В. Кластеризация с помощью нейронной сети Кохонена и модифицированного алгоритма иерархической кластеризации Хамелеон в различных предметных областях // Реєстрація, зберігання і обробка даних. 2013.
8. Сеньковская И. С., Сараев П. В. Автоматическая кластеризация в анализе данных на основе самоорганизующихся карт Кохонена // Вестник Магнитогорского государственного технического университета им. Г. И. Носова. 2011. № 2. С. 78–79.
9. Серебряная Л. В., Чебаков С. В. Методы автоматической классификации и кластеризации текстовой информации // Информатизация образования. 2011. № 2. С. 52–61.
10. Киселев М. Метод кластеризации текстов, основанный на попарной близости термов, характеризующих тексты, и его сравнение с метрическими методами кластеризации // Интернет-математика. 2007. С. 74–83.
11. Киселев М. В., Пивоваров В. С., Шмулевич М. М. Метод кластеризации текстов, учитывающий совместную встречаемость ключевых терминов, и его применение к анализу тематической структуры новостного потока, а также ее динамики // Интернет-математика. 2005. С. 412–435.
12. Кушнарев Д. А. Классификация алгоритмов кластеризации текстовых документов // Карповские научные чтения : сб. науч. ст. Вып. 5 : в 2 ч. Ч. 1 / под ред. А. И. Головня. Минск : Белорусский Дом печати, 2011. С. 179–183.
13. Кан А. В., Козловская Я. Д., Кадушкин Н. А., Хорошилов А. А. Автоматическая кластеризация документов СМИ на основе анализа их смыслового содержания // Моделирование и анализ данных. 2020. Т. 10, № 3. С. 24–38.
14. Сажок Н. Н. Кластеризация слов при построении лингвистической модели для автоматического распознавания речевого сигнала // Кибернетика и вычислительная техника. 2012. № 4. С. 59–66.
15. Сравнительный анализ эффективности работы Cython и Python. URL: https://habr.com/ru/post/676426/?ysclid=l8mq7cv0zb688214357 (дата обращения: 12.09.2022).
16. Рашка С., Мирджалили В. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2 : пер. с англ. 3-е изд. СПб. : Диалектика, 2020. 848 с.
17. Myers J., Copeland R. Essential SQLAlchemy: Mapping Python to Databases. O'Reilly Media, Inc., 2015.
18. Wu B. K. K-means clustering algorithm and Python implementation // IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE). 2021. P. 55–59.
19. Masse M. REST API Design Rulebook. 1st ed. Beijing Köln : O'Reilly Media, 2011. 112 p.
20. Relan K. Deploying Flask Applications // Building REST APIs with Flask: Create Python Web Services with MySQL. Berkeley, CA : Apress, 2019. P. 159–182.
21. Jung M.-G., Youn S.-A., Bae J., Choi Y.-L. A study on data input and output performance comparison of mongodb and postgresql in the big data environment // 8th international conference on database theory and application (DTA). 2015. P. 14–17.

22. Fredstam M., Johansson G. Comparing database management systems with SQLAlchemy: A quantitative study on database management systems. 2019.
23. Holt B., Briggs P., Ceze L., Oskin M. Alembic: automatic locality extraction via migration // Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications. 2014. P. 879–894.

References

1. Abramov V.I., Markina Yu.V., Kamynin D.A. Implementation of import substitution policy as a factor of activation of innovation processes. *Ekonomika i predprinimatel'stvo = Economics and entrepreneurship*. 2017;(12-1):134–137. (In Russ.)
2. Abramov V.I., Borzov A.V., Semenkov K.Yu. Assessment of readiness of small and medium-sized enterprises for digital transformation. *Voprosy innovatsionnoy ekonomiki = Issues of innovative economy*. 2022;12(3). (In Russ.). doi:10.18334/vinec.12.3.115000
3. Abramov V.I., Boboev D.S., Gil'manov T.D., Semenkov K.Yu. Theoretical and practical aspects of creating a digital double of the company. *Voprosy innovatsionnoy ekonomiki = Issues of innovative economy*. 2022;12(2):967–980. (In Russ.). doi:10.18334/vinec.12.2.114890
4. Belotserkovskaya M.G. Clustering of the customer base of loyalty program participants. *Moskovskiy ekonomicheskiy zhurnal = Moscow Economic Journal*. 2017;(2):112–119. (In Russ.)
5. Zadvornaya I.A., Romakina O.M. Application of the clusterization algorithm for analyzing data of potential bank customers. *Uchenye zapiski Bryanskogo gosudarstvennogo universiteta = Scientific notes of the Bryansk State University*. 2019;(2):7–15. (In Russ.)
6. Kudashkin A.V., Mokhov A.S. Clusterization of bank customers based on their personal data and bank transactions. *Informatsionnye sistemy i tekhnologii IST-2020 = Information systems and technologies IST-2020*. 2020:780–785. (In Russ.)
7. Lyakhovets A.B. Clusterization using the Kohonen neural network and the modified algorithm of hierarchical clustering Chameleon in various subject areas. *Reestratsiya, zberigannya i obrobka danikh*. 2013. (In Russ.)
8. Sen'kovskaya I.S., Saraev P.V. Automatic clustering in data analysis based on self-organizing Kohonen maps. *Vestnik Magnitogorskogo gosudarstvennogo tekhnicheskogo universiteta im. G.I. Nosova = Bulletin of Magnitogorsk State Technical University named after G. I. Nosov*. 2011;(2):78–79. (In Russ.)
9. Serebryanaya L.V., Chebakov S.V. Methods of automatic classification and clustering of textual information. *Informatizatsiya obrazovaniya = Informatization of education*. 2011;(2):52–61. (In Russ.)
10. Kiselev M. A text clustering method based on pairwise proximity of terms characterizing texts and its comparison with metric clustering methods. *Internet-matematika = Internet mathematics*. 2007:74–83. (In Russ.)
11. Kiselev M.V., Pivovarov V.S., Shmulevich M.M. The method of clustering texts, taking into account the joint occurrence of key terms, and its application to the analysis of the thematic structure of the news stream, as well as its dynamics. *Internet-matematika = Internet mathematics*. 2005:412–435. (In Russ.)
12. Kushnarev D.A. Classification of clustering algorithms of text documents. *Karpovskie nauchnye chteniya: sb. nauch. st. Vyp. 5: v 2 ch. Ch. 1 = Karpov scientific readings : collection of scientific articles. Iss. 5 : in 2 parts. Part 1*. Minsk: Belorusskiy Dom pechati, 2011:179–183. (In Russ.)
13. Kan A.V., Kozlovskaya Ya.D., Kadushkin N.A., Khoroshilov A.A. Automatic clustering of media documents based on the analysis of their semantic content. *Modelirovanie i analiz dannykh = Modeling and data analysis*. 2020;10(3):24–38. (In Russ.)
14. Sazhok N.N. Clustering of words in the construction of a linguistic model for automatic speech signal recognition. *Kibernetika i vychislitel'naya tekhnika = Cybernetics and computer engineering*. 2012;(4):59–66. (In Russ.)

15. *Sravnitel'nyy analiz effektivnosti raboty Cython i Python = Comparative analysis of the effectiveness of Cython and Python*. (In Russ.). Available at: https://habr.com/ru/post/676426/?ysclid=l8mq7cv0zb688214357 (accessed 12.09.2022).
16. Rashka S., Mirdzhalili V. *Python i mashinnoe obuchenie: mashinnoe i glubokoe obuchenie s ispol'zovaniem Python, scikit-learn i TensorFlow 2 : per. s angl. 3-e izd. = Python and machine learning: machine and deep learning using Python, scikit-learn and TensorFlow 2 : trans. from English 3rd ed.* Saint Petersburg: Dialektika, 2020:848. (In Russ.)
17. Myers J., Copeland R. *Essential SQLAlchemy: Mapping Python to Databases*. O'Reilly Media, Inc., 2015.
18. Wu B.K. K-means clustering algorithm and Python implementation. *IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)*. 2021:55–59.
19. Masse M. *REST API Design Rulebook. 1st ed.* Beijing Köln: O'Reilly Media, 2011:112.
20. Relan K. Deploying Flask Applications. *Building REST APIs with Flask: Create Python Web Services with MySQL*. Berkeley, CA: Apress, 2019:159–182.
21. Jung M.-G., Youn S.-A., Bae J., Choi Y.-L. A study on data input and output performance comparison of mongodb and postgresql in the big data environment. *8th international conference on database theory and application (DTA)*. 2015:14–17.
22. Fredstam M., Johansson G. *Comparing database management systems with SQLAlchemy: A quantitative study on database management systems*. 2019.
23. Holt B., Briggs P., Ceze L., Oskin M. Alembic: automatic locality extraction via migration. *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications*. 2014:879–894.

Информация об авторах / Information about the authors

Александр Дмитриевич Столяров
аспирант,
Национальный исследовательский ядерный университет МИФИ
(Россия, г. Москва, Каширское шоссе, 31)
E-mail: mr.alexst@gmail.com

Aleksandr D. Stolyarov
Postgraduate student,
National Research Nuclear University MEPhI
(31 Kashirskoe highway, Moscow, Russia)

Владимир Владимирович Гордеев
генеральный директор,
АЭРОЛАБС
(Россия, г. Москва, ул. Котляковская, 3)
E-mail: v.gordeev@aerolabs.aero

Vladimir V. Gordeev
Chief executive officer,
Aerolabs LLC
(3 Kotlyakovskaya street, Moscow, Russia)

Виктор Иванович Абрамов
доктор экономических наук, кандидат физико-математических наук, доцент, профессор кафедры управления бизнес-проектами,
Национальный исследовательский ядерный университет МИФИ
(Россия, г. Москва, Каширское шоссе, 31)
E-mail: viabramov@mephi.ru

Viktor I. Abramov
Doctor of economical sciences, candidate of physical and mathematical sciences, associate professor, professor of the sub-department of business project management,
National Research Nuclear University MEPhI
(31 Kashirskoe highway, Moscow, Russia)

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflicts of interests.

Поступила в редакцию/Received 18.12.2022
Поступила после рецензирования/Revised 14.02.2023
Принята к публикации/Accepted 13.03.2023