DOI 10.37882/2223-2966.2024.11.24

РЕАЛИЗАЦИЯ ПРОГРАММНОЙ СИСТЕМЫ АНАЛИЗА ДАННЫХ НА ОСНОВЕ ELK-СТЕКА

IMPLEMENTATION OF A DATA ANALYSIS SOFTWARE SYSTEM BASED ON THE ELK STACK

A. Pantykhin
V. Gladun
I. Malinin
S. Molodyakov

Summary. This paper presents the development of a data analysis software system based on the ELK stack (Elasticsearch, Logstash, Kibana) designed for processing and analyzing sales data from various marketplaces. The system enables data collection, preprocessing, analysis, and visualization, allowing for the identification of market trends and demand forecasting. The focus is on the system's flexibility, scalability, and performance, making it an effective tool for informed decision-making in the digital economy.

Keywords: data analysis, ELK stack, Kafka, Logstash, Elasticsearch, Kibana, Python, BI system.

Пантюхин Андрей Максимович
Санкт-Петербургский политехнический университет Петра Великого
panandafog@gmail.com
Гладун Владимир Вадимович
Санкт-Петербургский политехнический университет Петра Великого
vladimir.gldn@gmail.com
Малинин Илья Игоревич
Санкт-Петербургский политехнический университет Петра Великого
malinin.ilja@gmail.com
Молодяков Сергей Александрович
Доктор технических наук, профессор, Санкт-Петербургский политехнический университет Петра Великого
molodyakov_sa@spbstu.ru

Аннотация. В данной работе представлена разработка программной системы анализа данных на основе ELK-стека (Elasticsearch, Logstash, Kibana), предназначенной для обработки и анализа данных о продажах товаров с различных маркетплейсов. Система обеспечивает сбор, предобработку, анализ и визуализацию данных, что позволяет выявлять рыночные тенденции и прогнозировать спрос. Основное внимание уделено гибкости, масштабируемости и производительности системы, что делает ее эффективным инструментом для принятия обоснованных решений в условиях цифровой экономики.

Ключевые слова: анализ данных, ELK стек, Kafka, Logstash, Elasticsearch, Kibana, Python, BI система.

Введение

С ростом объемов генерируемых данных возрастает необходимость в эффективном управлении и анализе информации, что делает системы бизнес-аналитики (BI — Business Intelligence) ключевыми инструментами для преобразования данных в структурированную форму и принятия обоснованных решений. BI-платформы включают инструменты для сбора, интеграции, анализа и визуализации данных, обеспечивая возможность создания отчетов и аналитических панелей для своевременного принятия решений [1].

В условиях современной цифровой экономики разработка и внедрение BI-систем требует высокой гибкости, масштабируемости и производительности. Такие системы оптимизируют бизнес-процессы, повышают качество управленческих решений и усиливают конкурентоспособность организаций, делая их важным элементом для успешного функционирования и развития бизнеса.

Особую актуальность приобретает создание BI-платформ для анализа данных с маркетплейсов и других платформ для продажи товаров [2]. Популярность маркетплейсов в последние годы активно растет, как показано на рис. 1.

Эти платформы позволяют эффективно собирать, обрабатывать и анализировать данные, выявлять рыночные тенденции и прогнозировать спрос, что способствует более глубокому пониманию рынка и улучшению стратегической и операционной деятельности компаний [3].

Анализ существующих решений

Системы бизнес-аналитики (BI) играют важную роль в стратегическом управлении и анализе данных в различных отраслях. Рассмотрим основные BI-решения и их преимущества и недостатки.

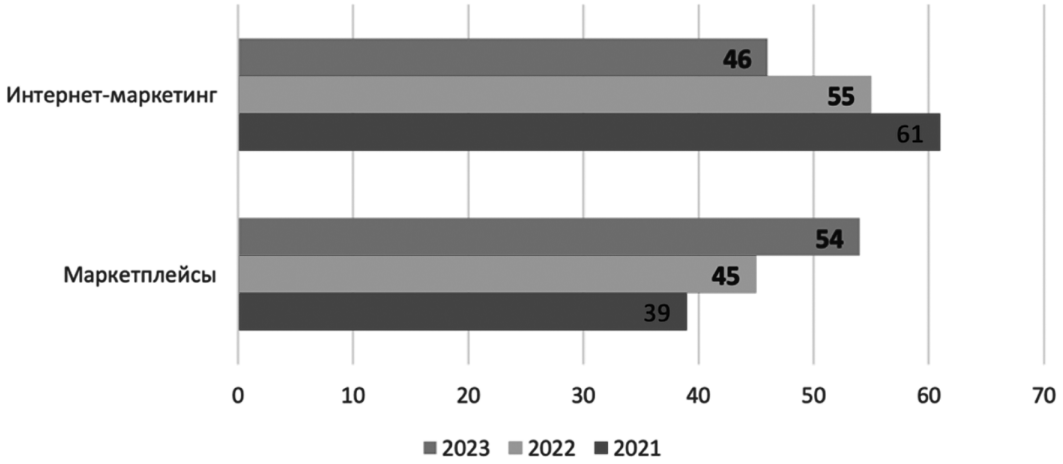

Рис. 1. Структура российского рынка электронной коммерции по каналам продаж, %

Tableau [4] отличается мощной визуализацией и интуитивно понятным интерфейсом, но его высокая стоимость и ограниченные возможности обработки данных делают его менее привлекательным для крупных проектов. Power BI [5], интегрированный с продуктами Microsoft, предлагает доступные решения и удобную визуализацию, однако может страдать от ограниченных возможностей кастомизации и зависимости от облачных сервисов.

Qlik Sense [6] предоставляет высокую производительность и ассоциативную модель данных, что облегчает исследование связей, но требует больше времени на обучение. SAP BusinessObjects [7] обладает мощными аналитическими функциями и гибкостью, но его сложность и высокая стоимость делают его подходящим лишь для крупных предприятий, использующих SAP. Looker [8] и Domo [9] предлагают мощные инструменты для анализа и работы с данными в облачных средах, но их стоимость и сложность настройки могут стать препятствием для небольших компаний.

Хотя все эти системы предлагают широкий спектр возможностей для анализа данных, они не включают специализированные инструменты для работы с данными о продажах товаров и системами сбора данных с объявлений. Это создает нишу для разработки решений, ориентированных на анализ данных о рынке товаров, которые могут эффективно конкурировать в этой области.

Архитектура системы

Для обеспечения надежного и эффективного хранения данных и работы с ними архитектура системы включает несколько ключевых модулей. Основной модуль хранения данных отвечает за масштабируемое и надежное постоянное хранение информации, обеспечивая быстрый доступ и поиск.

Второй важный компонент — модуль анализа данных, выполняющий задачи по обработке и анализу информации. Он считывает данные из хранилища, проводит необходимые вычисления и возвращает обработанные данные. Задачи этого модуля включают удаление дубликатов, фильтрацию невалидных данных и анализ содержимого объявлений.

Для представления результатов анализа необходим модуль визуализации данных, который создает интерактивные дашборды и визуализации, помогая пользователям легко интерпретировать данные. Также архитектура включает message-брокер для буферизации данных и модуль предобработки для начальной очистки и трансформации данных перед их загрузкой в хранилище. Модуль-коннектор предоставляет API для интеграции с внешними источниками данных и настройки схемы данных. Этот многоуровневый подход обеспечивает гибкость, масштабируемость и эффективность системы анализа данных.

Компоненты системы представлены на рис. 2.

Выбор технических средств

Для реализации проекта был выбран ELK стек (Elasticsearch [10], Logstash [11], Kibana [12]) благодаря его гибкости, масштабируемости и мощным возможностям визуализации данных. Эта комбинация технологий является open-source, что делает ее экономически эффективным решением. ELK стек обеспечивает надежное хранение данных, их анализ и наглядное представление результатов, что особенно важно для проектов с большими объемами информации.

Для первичной обработки данных используется Logstash, который позволяет собирать, фильтровать и трансформировать данные перед их отправкой в хранилище. В качестве message-брокера выбран Apache Kafka [13], который обеспечивает надежную и быструю

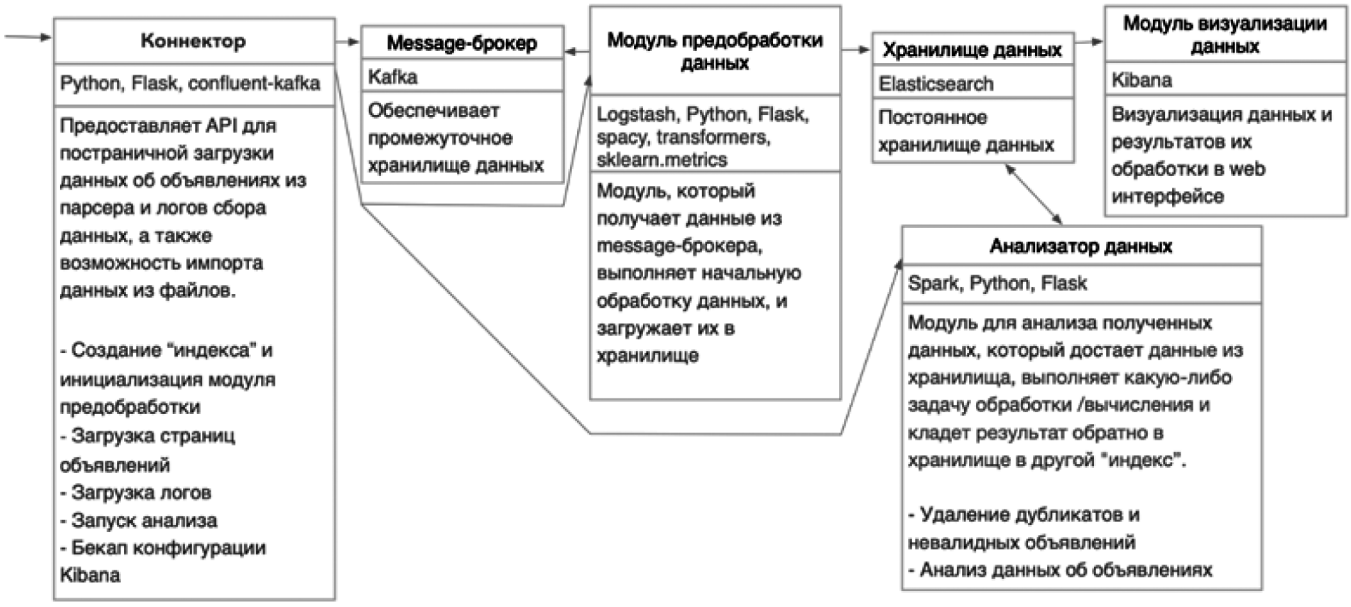

Рис. 2. Компоненты системы

передачу данных в системе. Для анализа данных применяется Apache Spark, интегрированный с Python [14], что обеспечивает высокую производительность и удобство разработки аналитических приложений.

Контейнеризация всех компонентов системы реализована с использованием Docker [15], что позволяет легко развертывать, масштабировать и управлять системой. Этот подход обеспечивает консистентность среды на всех этапах разработки и эксплуатации, а также упрощает интеграцию новых компонентов.

Реализация коннектора

Для упрощения развертывания и управления проектом используется Docker Compose [16], позволяющий определить и настроить все необходимые сервисы в одном конфигурационном файле. Сервисы включают основные компоненты системы, такие как анализатор, базы данных, инструменты визуализации и message-брокеры. Это обеспечивает гибкость в управлении зависимостями между сервисами, настройке портов и сетей, а также упрощает интеграцию и масштабирование системы.

Сборка и распространение контейнеров осуществляется через Docker Hub, что позволяет создавать универсальные образы под разные архитектуры и легко распространять их среди пользователей. Для этого используются скрипты, автоматизирующие процесс сборки и загрузки образов на Docker Hub. Масштабирование системы обеспечивается через настройку компонентов, таких как Elasticsearch и Kafka, для работы в распределенной среде с поддержкой шардинга и репликации данных, что позволяет эффективно управлять большими объемами данных в реальном времени.

На схеме (рис. 3) представлена архитектура системы коннектора.

Предобработка данных

Модуль предобработки данных представляет собой пакет Python, который интегрируется с Logstash через REST API для обработки поступающих данных. Основная задача этого модуля заключается в автоматическом заполнении недостающих данных в описаниях товаров на русском языке с помощью методов машинного обучения. Модуль использует предобученные модели на основе архитектуры BERT и другие инструменты обработки естественного языка, такие как spaCy и mBART, для обработки и анализа текстовых данных.

Основные этапы предобработки данных включают лингвистический разбор текста с использованием spaCy, перевод названий признаков с английского на русский с помощью mBART и извлечение признаков и их значений на основе синтаксического анализа текста. Модуль также применяет модель ruBERT для поиска синонимов в текстах, что позволяет более точно определять и заполнять значения признаков на основе контекста.

Этот модуль обеспечивает высокую точность и эффективность при работе с текстовыми данными, позволяя автоматически заполнять недостающие поля в описаниях товаров и улучшать качество данных перед их дальнейшей обработкой и анализом.

Модуль анализа данных

Модуль анализа данных играет ключевую роль в обработке и интерпретации информации, собранной си-

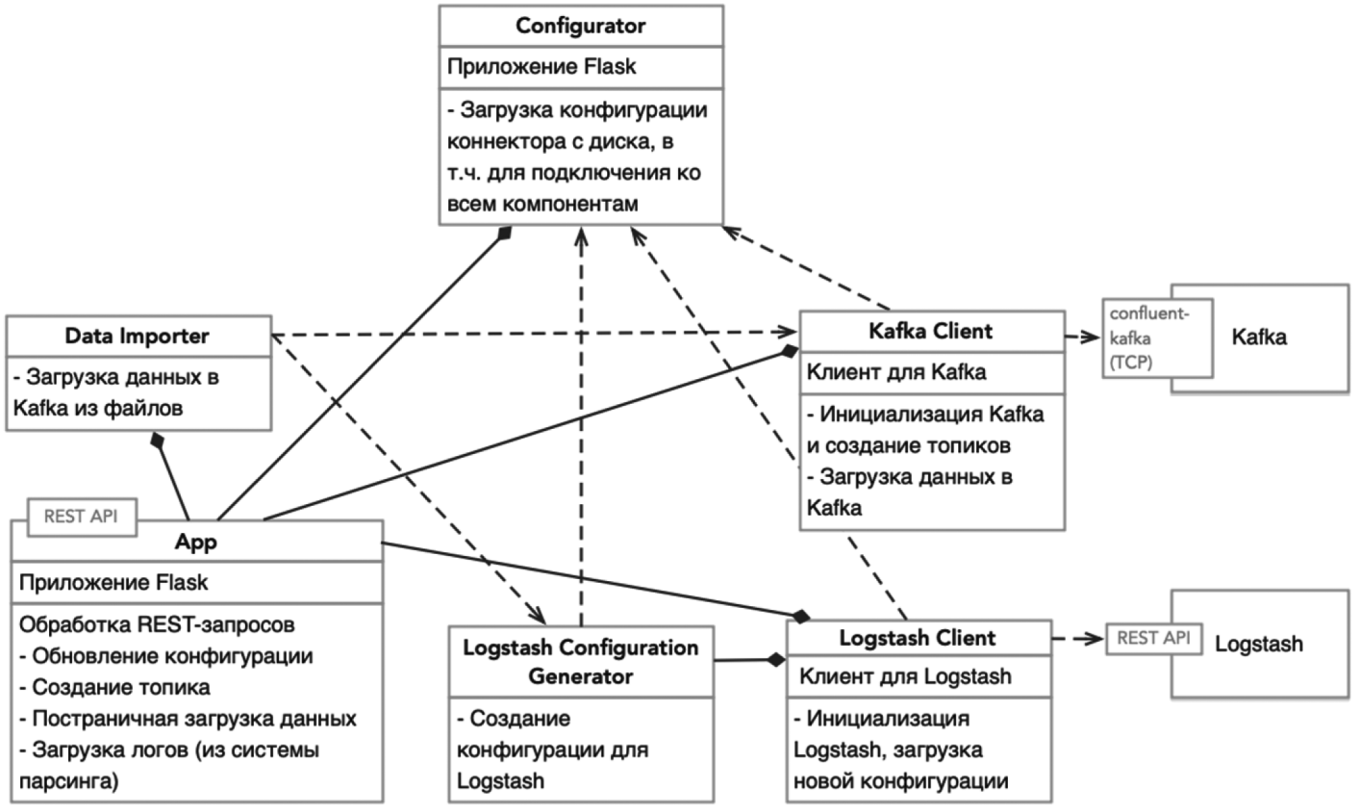

Рис. 3. Компоненты системы коннектора

стемой. После завершения процесса сбора данных или по необходимости, этот модуль запускает процедуры анализа, что позволяет извлекать из данных полезные инсайты и формировать аналитические отчеты. Для реализации аналитических процессов используется платформа Apache Spark, которая предоставляет мощные инструменты для обработки больших объемов данных в распределенной среде.

Одним из главных преимуществ использования Apache Spark является возможность выполнения SQL-запросов для анализа данных. В проекте принято решение использовать SQL, так как это упрощает создание и изменение запросов без необходимости вмешательства в код, а также обеспечивает понятность и доступность работы системы для других разработчиков. SQL-запросы позволяют гибко взаимодействовать с данными, загруженными из Elasticsearch, и выполнять сложные операции по их обработке и анализу. В результате каждый запрос на анализ представляет собой JSON-файл, содержащий параметры, такие как входной и выходной индексы Elasticsearch и сам SQL-запрос.

Процесс анализа данных включает несколько ключевых шагов: от инициации запроса через REST API до выполнения SQL-запросов и сохранения результатов в выходной индекс Elasticsearch. Важнейшие компоненты системы, такие как Analyser и SparkSessionConfigurator, обеспечивают конфигурацию и создание сессий Spark, которые затем используются для выполнения запросов и обработки данных. В дополнение к SQL-запросам, в системе реализованы базовые функции анализа для упрощения разработки и тестирования, предоставляющие дополнительные возможности для работы с данными и создания отчетов. Например, такие методы как groupBy, min, avg и max позволяют агрегировать и анализировать данные, создавая краткие отчеты по ключевым показателям целевых групп объявлений.

Схема компонентов этого модуля представлена на рис. 4.

Оценка результатов работы

Для оценки результатов работы было проведено тестирование и сбор необходимых метрик. Процесс тестирования включает в себя создание модульных и интеграционных тестов, а также тестирование всей системы целиком (End-to-end тесты). Несмотря на наличие автоматических тестов, в ходе работы над проектом все еще применялось и ручное тестирование ввиду удобства применения для отладки.

При оценке качества предобработки данных основное внимание уделяется трем ключевым характеристикам: проценту правильного распознавания отсут-

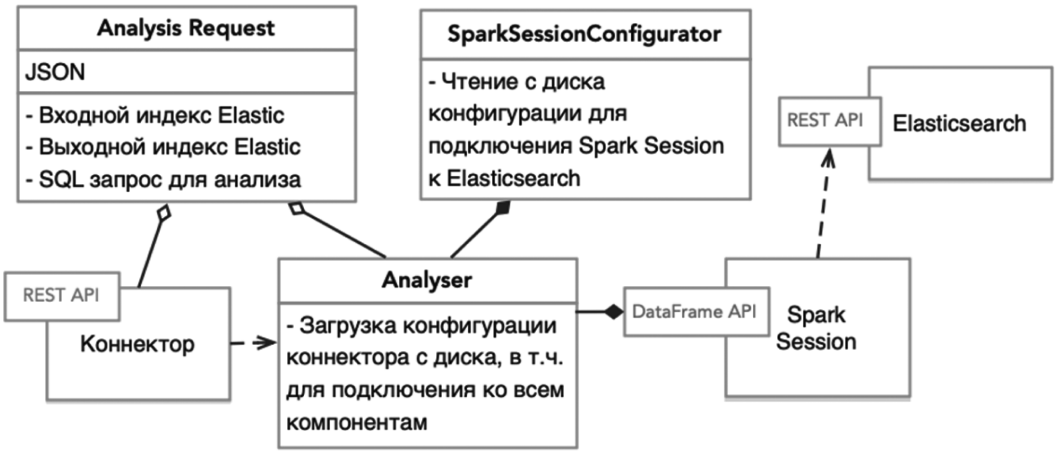

Рис. 4. Схема компонентов модуля анализа данных

ствующих признаков, проценту ложных срабатываний, и скорости работы алгоритма. Правильное распознавание недостающих признаков важно для точности анализа, но еще более критичным является минимизация ложных срабатываний, так как они могут значительно исказить данные и повлиять на результаты анализа. Скорость работы алгоритма также играет важную роль, так как она напрямую влияет на общую производительность системы.

Тестирование модуля предобработки показало, что система успешно распознает около 70 % числовых признаков, с минимальными ложными срабатываниями. Однако при работе с текстовыми признаками, такими как цвет, возникает около 20 % ложных срабатываний, что привело к решению временно отключить их распознавание до улучшения системы. Эти результаты подчеркивают важность настройки алгоритма для достижения баланса между точностью распознавания и минимизацией ошибок.

Система была протестирована на реальных данных, собранных с интернет-ресурса Авито Авто [17], где было собрано 886039 объявлений о продаже автомобилей, что заняло 48 часов и привело к объему данных в 2,4 ГБ в формате JSON. Во время испытаний было выявлено, что самым медленным компонентом системы является модуль предобработки данных, который в среднем затрачивает около трех секунд на обработку одного объявления. Хотя это время обработки не критично на текущем этапе, поскольку большинство объявлений имеют корректно заполненные поля, данная задержка может стать проблемой при увеличении объемов данных.

Скорость работы модуля анализа данных также была оценена в ходе испытаний. Несмотря на то, что она не является столь критичной, как скорость предобработки, она все же существенно влияет на пользовательский опыт. На анализ собранного объема данных с использованием базового набора запросов потребовалось около 90 секунд, что является приемлемым показателем. Для демонстрации возможностей анализа с использованием Kibana были созданы визуализации, подтверждающие эффективность работы системы в реальных условиях.

Заключение

В ходе работ была успешно спроектирован и разработан модульный сервис, который не только обладает гибкостью в плане модификаций, но и демонстрирует высокую эффективность благодаря использованию Python, MongoDB и MongoEngine. Эти технологии были выбраны с учетом их сильных сторон, таких как гибкость схем данных и простота масштабирования, что обеспечило надежную основу для проекта.

Проведено тестирование сервиса и был определен наиболее эффективный алгоритм хеширования и размера сегмента данных. Были выбраны алгоритм хеширования MD5 и размер сегмента в 32 байта. В ходе тестирования не было выявлено ошибок после дедупликации и процесса обратного восстановлении исходного файла.

Эти результаты подтверждают, что предложенный подход к дедупликации обеспечивает не только общую производительность системы хранения данных, но и высокий уровень надежности и целостности данных. На основе полученных данных можно сделать вывод, что разработанный сервис подходит для предприятий, стремящихся минимизировать расходы на хранение больших объемов информации.

ЛИТЕРАТУРА

1. Abu-AlSondos I.A. The impact of business intelligence system (BIS) on quality of strategic decision-making // International Journal of Data and Network Science, Jul 2023.
2. Окунева Е.С. Влияние маркетплейсов на развитие цифровой торговли в Российской Федерации. // Мир студенческой науки, 2023.
3. Fruhwirth M., Rachinger M., Prlja E. Discovering business models of data marketplaces. // Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.
4. Tableau Public: Resources to Learn Tableau [Электронный ресурс] // Tableau Software: [сайт]. URL: https://public.tableau.com/en-us/s/resources (дата обращения: 09.06.2024).
5. Power B.I. Documentation [Электронный ресурс] // Power BI: [сайт]. URL: https://docs.microsoft.com/en-us/power-bi/ (дата обращения: 09.06.2024).
6. Qlik Data Integration, Data Quality, and Analytics Solutions [Электронный ресурс] // Qlik Sense: [сайт]. URL: https://www.qlik.com/us (дата обращения: 09.06.2024).
7. SAP BusinessObjects Business Intelligence suite [Электронный ресурс] // SAP: [сайт]. URL: https://www.sap.com/products/technology-platform/bi-platform.html (дата обращения: 09.06.2024).
8. Looker business intelligence platform embedded analytics [Электронный ресурс] // Google Cloud: [сайт]. URL: https://cloud.google.com/looker (дата обращения: 09.06.2024).
9. Discover the Domo Data Experience Platform [Электронный ресурс] // Domo: [сайт]. URL: https://www.domo.com (дата обращения: 09.06.2024).
10. Elasticsearch: The Official Distributed Search & Analytics Engine [Электронный ресурс] // Elastic: [сайт]. URL: https://www.elastic.co (дата обращения: 09.06.2024).
11. Logstash: Collect, Parse, Transform Logs [Электронный ресурс] // Elastic: [сайт]. URL: https://www.elastic.co/logstash (дата обращения: 09.06.2024).
12. Kibana: Explore, Visualize, Discover Data [Электронный ресурс] // Kibana: [сайт]. URL: https://www.elastic.co/kibana (дата обращения: 09.06.2024).
13. Apache Kafka Use Cases [Электронный ресурс] // Apache Kafka: [сайт]. URL: https://kafka.apache.org/documentation/ (дата обращения: 09.06.2024).
14. Documentation [Электронный ресурс] // Python: [сайт]. URL: https://www.python.org/doc/ (дата обращения: 09.06.2024).
15. Docker Documentation [Электронный ресурс] // Docker: Accelerated Container Application Development: [сайт]. URL: https://docs.docker.com (дата обращения: 09.06.2024).
16. Docker Compose overview [Электронный ресурс] // Docker Docs: [сайт]. URL: https://docs.docker.com/compose/ (дата обращения: 09.06.2024).
17. Купить автомобиль в Санкт-Петербурге [Электронный ресурс] // Авито: [сайт]. URL: https://www.avito.ru/sankt-peterburg/avtomobili (дата обращения: 09.06.2024).