МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА
Рецензент
_____
«___»_____ 2022 г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____Д.В. Топольский
«___»_____ 2022 г.

Система поддержки аналитического мониторинга профайлов

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К МАГИСТЕРСКОЙ ДИССЕРТАЦИИ
ЮУРГУ-090401.2022.213 ПЗ МГ

Руководитель работы,
к.т.н., доцент каф. ЭВМ
_____ И.Л. Кафтанников
«___»_____2022 г.

Автор работы,
студент группы КЭ-222
_____ М.М. Топалов
«___»_____2022 г.

Нормоконтролер,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
«___»_____2022 г.

Челябинск-2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____Д.В. Топольский
«___»_____2022 г.

ЗАДАНИЕ

на выполнение магистерской диссертации

студенту группы КЭ-222
Топалову Михаилу Михайловичу
обучающемуся по направлению
09.04.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Система поддержки аналитического мониторинга профайлов» утверждена приказом по университету от № 697-13/12(приложение № 73) от 25.04.2022
2. **Срок сдачи студентом законченной работы:** 8 июня 2022 г.
3. **Исходные данные к работе:**

– разработать систему мониторинга для автоматизированного сбора и обработки данных с последующим детектированием аномальных значений;
– язык программирования Python;
– СУБД PostgreSQL;
– открытое программное обеспечение оркестрации сценариев Apache Airflow;
– платформа контейнерной виртуализации Docker;
– среда разработки Colaboratory;
– выборка данных логов рекламных событий.

4. **Перечень подлежащих разработке вопросов:**

– обзор отечественной и зарубежной литературы;

– изучение особенностей и структуры мониторинга профайлов;

– проведение сравнительного анализа существующих методов мониторинга профайлов;

– выявление специфичных особенностей мониторинга профайлов;

– разработка системы мониторинга профайлов для автоматизации и получения результатов;

– верификация и визуализация разработанного метода на основе данных.

5. **Дата выдачи задания:** 1 декабря 2021 г.

Руководитель работы _____ /*И.Л. Кафтанников*/

Студент _____/*М.М. Топалов*/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	1.03.2022	
Изучение существующих методов мониторинга профайлов	1.04.2022	
Обработка и агрегирование данных	25.04.2022	
Разработка дата-пайплайна	1.05.2022	
Разработка алгоритма для мониторинга профайлов	10.05.2022	
Компоновка текста работы и сдача на нормоконтроль	31.05.2022	
Подготовка презентации и доклада	7.06.2022	

Руководитель работы _____/*И.Л. Кафтанников*/

Студент _____/*М.М. Топалов*/

Аннотация

М.М. Топалов. Система поддержки аналитического мониторинга профайлов. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2022, 57 с., 16 ил., библиогр. список – 17 наим.

В рамках магистерской диссертации была разработана система поддержки аналитического мониторинга профайлов с возможностью оповещения.

В ходе реализации работы были выполнены такие задачи, как анализ предметной области, анализ существующих решений, определение функциональных требований, выбор инструментов и инфраструктуры разрабатываемой системы, проектирование системы, реализация системы и проведение верификации.

Проведение верификации мониторинга профайлов доказало эффективность разрабатываемой системы.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Получение точной информации является важной составляющей для развития компании. Характерная особенность данных заключается в том, что они бывают структурированные или неструктурированные. Данные показывают больше деталей о поведении, деятельности и событиях, которые происходят вокруг, поэтому аналитика этих данных дает доступ к разнообразным и различным типам из огромных ресурсов с меньшим временем отклика. Их обрабатывают при помощи специальных автоматизированных инструментов, чтобы использовать для статистики, анализа, прогнозов и принятия решений.

Важным этапом при анализе является мониторинг профайлов. Сбор данных в крупных компаниях производится ежесекундно и для отслеживания «Здоровья бизнеса» нужно мгновенно выявлять различные аномалии в получаемых данных. У компаний появляется необходимость в разработке методов, которые будут автономно находить аномалии с целью выявить значимые зависимости в данных, обрабатывать данные и предупреждать о проблемах, связанных с получением информации о поведении клиентов или о проблемах с продуктом.

Цель и задачи работы

1. Произвести подбор публикаций по тематике разрабатываемой программно-аппаратной системы.

2. Выполнить обзор существующих аналогичных решение.

3. Изучение особенностей и структуры представления данных.

4. Выявление специфических особенностей мониторинга данных.

5. Разработка методов мониторинга данных для автоматизации и исследование для получения значимых результатов.

6. Верификация разработанного метода на основе тестовых данных.

Структура и объем работы

Работа состоит из введения, трёх глав, заключения, библиографического списка, приложения.

Работа составляет 57 страниц, в библиографическом списке указано 17 источников, объем приложения – 3 страницы.

Содержание работы

В первой главе проводится обзор публикаций по тематике мониторинга и анализу данных, выполнен анализ предметной области, приведен обзор существующих аналогичных решений.

Во второй главе описаны основные принципы построения дата-пайплайна, основная структура базы данных, проведена работа по автоматизации сбора, обработки и анализу данных.

В третьей главе описан выбор, подходящей математической модели, проведена разработка функции для мониторинга профайлов, реализована оркестрация данных.

В заключении описаны результаты реализации разработанной системы, полученные в ходе выполнения работы.

Актуальность работы

Обработка и анализ данных стали важным событием последнего десятилетия, многие компании уделяют особое внимание анализу данных. В настоящее время нельзя представить крупную компанию, которая не собирает и не хранит данные. В современном мире данные имеют ценность, которую можно сравнить с нефтью или золотом.

1. АНАЛИЗ ДАННЫХ ДЛЯ ПРОФАЙЛОВ КОМПАНИЙ

1.1. Описание анализа данных

Аналитика данных - это процесс изучения больших массивов данных с целью выявления скрытых закономерностей, неизвестных корреляций, тенденций рынка, предпочтений клиентов и другой полезной бизнес-информации.

Аналитика данных показывает новые связи между данными, может выявить невидимые ранее тенденции и способствует созданию новых знаний, которые затем могут быть использованы для повышения эффективности и увеличения прибыльности компании. В долгосрочной перспективе они могут компенсировать затраты, связанные с покупкой специализированного программного обеспечения.

При анализе данных нужно сосредоточиться на поиске корреляций и закономерностей, которые указывают на то, «что происходит», также нужно найти объяснении причин, «почему это происходит».

Большие данные обрабатывают постоянно поступающие данные из окружающей среды и изнутри компании. Таким образом, аналитика данных основана на данных, собранных в режиме реального времени, и именно поэтому результаты анализа точны и формируются без задержек.

Аналитические приложения универсальны и обладают широкой функциональностью, полезны как в крупных компаниях, так и в небольших компаниях из сектора малого и среднего бизнеса. Они позволяют интегрировать сложные бизнес-процессы и быстро реагировать на любые изменения как на операционном уровне, так и в бизнес-среде.

Благодаря им компании могут регулярно отслеживать состояние каждого процесса и быстро реагировать на события путем гибкой модификации процессов.

Анализ данных включает в себя различные аналитические методы, аналитическую архитектуру для данных и программное обеспечение, используемое для анализа данных. При анализе данных применяется множество аналитических методов, таких как: обучение ассоциативным правилам,

направленное на выявление связей в базах данных; A/B тестирование, позволяющее сравнивать контрольную группу с тестовой; кластерный анализ, позволяющий классифицировать объекты, разделенные на более мелкие группы; краудсорсинг, позволяющий собирать данные, генерируемые сообществами; объединение и интеграция данных, при которых анализируются данные, поступающие из разных источников; генетические алгоритмы, основанные на процессе естественной эволюции и применяемые в основном для оптимизации; машинное обучение и обработка естественного языка, создающие область искусственного интеллекта; нейронные сети, основанные на функциональности нервной системы человека и находящие применение в оптимизации и распознавании образов; анализ узлов в сетях; прогнозное моделирование и анализ регрессии на основе математических моделей; пространственный анализ и моделирование; контролируемое и неконтролируемое обучение и доски визуализации, включая облачные метки, историю и пространственные потоки информации и управленческие панели.

Этот длинный список методов анализа данных, вероятно, не является полным, поскольку постоянно появляются новые методы извлечения информации и знаний из наборов данных. Предприятия практически всех отраслей развивают концепцию информации и данных как стратегического развитие активов для бизнеса.

1.2. Анализ данных как эффективный инструмент для достижения бизнес-цели компании

Менеджеры все чаще принимают стратегии, основанные на получении, обработке и использовании высококачественных данных для принятия решений (Data driven подход).

Исследования, проведенные EMC Forum, показывают, что:

- 39% предпринимателей считают, что анализ данных обеспечивают успех бизнеса;

- 19% предпринимателей придерживаются мнения, что с помощью анализа данных они добились конкурентного преимущества;

- 36% предпринимателей считают, что использование Data driven подхода повысит безопасность и сохранность их данных.

Исследователи из опроса Economist Intelligence Unit [26], искали ответ на вопрос "Какие из следующих бизнес-процессов, по вашему мнению, являются наиболее приоритетными для применения больших данных сейчас, а какие будут наиболее важными через три года?". Результаты их исследования представлены на рисунке 1.

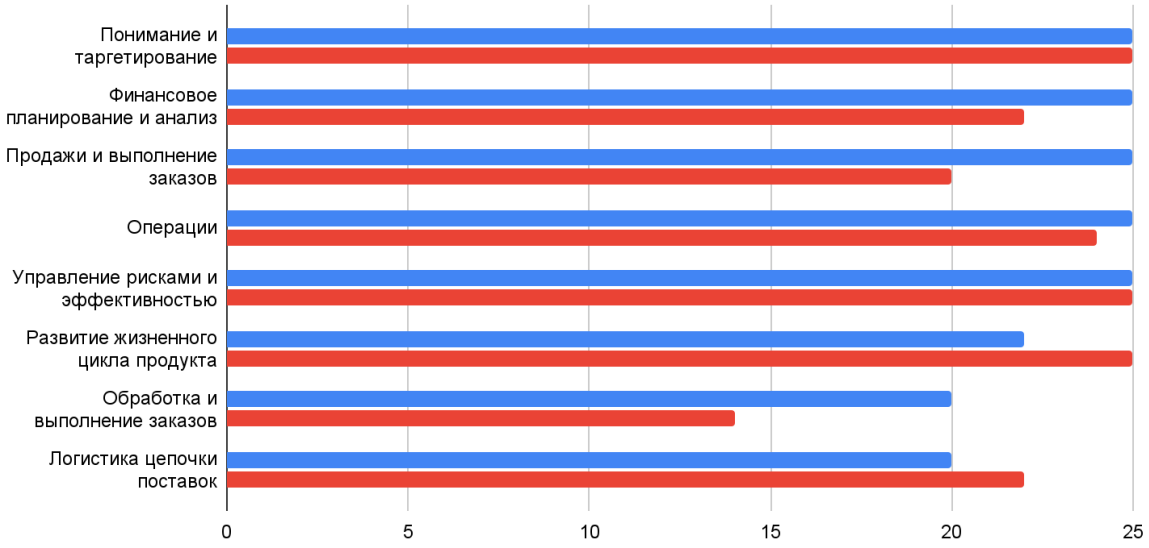

Рисунок 1 - Приоритеты применения анализа данных

Процессы работы с клиентами в настоящее время являются приоритетными для применения анализа данных, по мнению 42% руководителей высшего звена. Финансовое планирование с 32% и продажи с 29% являются следующими показателями для аналитики данных. Среди других приоритетов - операционная деятельность, управление рисками и управление эффективностью, а также эволюция жизненного цикла продукта и другие. Длинный список приоритетов предполагает широкие возможности для решений на основе анализа данных в масштабах всего предприятия. Через три года разделить эти приоритеты станет еще сложнее. По мнению респондентов, понимание и таргетирование клиентов

останется главным приоритетом, однако в относительном выражении он будет снижаться по мере появления нескольких других.

Данные создаются в каждом сегменте производства. Для производительности предприятия анализ данных помогает определить потребности в продукции, производительность и эффективность в соответствии с различными бизнес-целями. Для производства анализ данных позволяет обнаружить причины, вызывающие проблемы.

Анализ данных используется в управлении предприятием в следующих областях и видах деятельности:

- трансформация ключевых организационных бизнес;
- важные процессы;
- поддержка принятия стратегических решений;
- определение наиболее экономически эффективных поставщиков при своевременной поставке продукции;
- разработка продукции;
- выявление отклонений в работе оборудования и процессов, которые могут быть индикаторами проблем с качеством;
- анализ дебиторской задолженности, предвидение платежей;
- управление активами;
- определение того, какие маркетинговые акции и кампании наиболее эффективны для увеличения трафика, вовлеченности и продаж;
- прогнозирование поведения клиентов;
- управление взаимоотношениями с клиентами;
- оптимизация маркетинг-миксов с учетом маркетинговых целей;
- оптимизация распределения ресурсов сбыта;
- переопределение продукта;
- совместная фильтрация;
- анализ спроса и предложения.

1.3. Основной стек технологий для анализа данных и архитектура для хранения данных

Аналитика данных находится на пересечении трех направлений - математики, программирования и понимания бизнес-процессов. Последний пункт очень важен: опытный аналитик разбирается в устройстве конкретного бизнеса и знает все про его продукты или предоставляемые сервисы и услуги. Именно это залог того, что принимаемые им решения будут выгодными для компании. Помимо этого, аналитик смыслит в визуализации данных, то есть может представить их наглядно - в виде графиков, диаграмм и схем, чтобы они были понятными для коллег.

Основной стек технологий:

1. Язык программирования Python - является одним и ключевых, так как благодаря ему проводится основной анализ, проверка гипотез, визуализаций и машинного обучения. Основные библиотеки (Pandas, Numpy, Matplotlib, Seaborn, Sklearn и др.).

2. Чтобы анализировать данные их нужно получит из различных БД, для это аналитику необходимо знать SQL.

3. Любой аналитик должен уметь представить результат своей работы, для это используют популярные BI-инструменты, такие как Tableau или Power BI.

Для эффективного использования Данных компаниям необходимы новые ИТ-архитектуры, то есть конфигурация аппаратного и программного обеспечения таким образом, чтобы обеспечить эффективную обработку Данных. Технологии облачных вычислений могут предоставить неограниченные ресурсы по запросу. Это может стать решением проблемы растущего объема данных и позволит эффективно управлять данными. Популярно архитектурой для хранения Данных является GreenPlum.

GreenPlum - это, по сути, инфраструктура распределенных данных: Он распределяет массивные коллекции данных по нескольким узлам в кластере товарных серверов, что означает, что вам не нужно покупать и обслуживать

дорогостоящее специализированное оборудование. Она также индексирует и отслеживает эти данные, позволяя обрабатывать и анализировать Большие Данные гораздо эффективнее, чем это было возможно ранее.

GreenPlum быстро стал основой для задач обработки данных, таких как научная аналитика, планирование бизнеса и продаж, обработка огромных объемов сенсорных данных, в том числе от датчиков Интернета вещей.

GreenPlum используется такими крупными корпорациями, как Yandex, Tinkoff.

1.4. Структура анализа данных

План анализа данных рисунке 2, наверное, одно из наиболее важных действий, которое должны предпринять аналитики компании. Как и любое планирование, четкое понимание процесса будущего анализа поможет избежать ошибок, сделать сбор информации и последующий ее анализ прозрачным и понятным всем участникам. Составление плана анализа данных поможет компании сэкономить значительные финансовые и человеческие ресурсы.

Рисунок 2 - План анализа данных

Предобработка данных - Получаемые аналитиком данные не всегда соответствуют ожидаемому уровню качества. Человеческий фактор, ошибки системы или процесса выгрузки могут «испортить» их, то есть сделать непригодными для анализа. На данном этапе проходит работа с пропусками. Определение аномальных значений. Преобразование типов данных. Основные методы поиска дубликатов. Работа с несовершенными реальными наборами данных.

Исследовательский анализ данных - изучение срезов данных. Нахождение взаимосвязей разных параметров в данных. Объединение таблиц. Получение выводов по сгруппированным данным. На подготовительном этапе они нужны для оценки качества данных, а затем - для выдвижения гипотез, поисков закономерностей и подкрепления выводов в отчетах.

Статистический анализ - изучение объектов и их взаимосвязей методами статистики. Выборки и статистическая значимость. Выявление и обработка

аномалий. В работе с продуктом возникает множество гипотез. Их проверяют статистическими методами. Это важно для подтверждения результатов исследования.

Сбор и хранение данных - это процесс целенаправленного извлечения и анализа информации о предметной области, в роли которой может выступать тот или иной процесс, объект. Цель сбора - обеспечение готовности данных к дальнейшему продвижению в процессе. Поскольку эта фаза начинает цикл обращения данных, она очень важна, от качества ее исполнения во многом зависит от качества данных, которая будет использоваться потребителем при решении целевых задач информационной технологии.

Анализ бизнес-показателей - рассчитывать поведенческие и финансовые метрики. Строить воронки метрик и делать выводы на основе данных в этих воронках. Применять когортный анализ для изучения поведения клиентов. Расчета LTV и окупаемости бизнеса. Считать юнит-экономику, определять прибыльность или убыточность бизнеса. Принимать правильные решения на основе их результатов.

Принятие решений в бизнесе на основе данных - распознавать важнейшие бизнес-метрики. Выдвигать гипотезы и использовать правильные методы для их проверки. Использовать методы для приоритезации гипотез. Проведение A/B тестирования.

Как рассказать история с помощью графиков - презентация результатов аналитического исследования. Способы наглядного представления данных. Создание отчетов, объясняющих выводы аналитика.

Автоматизация - настройка пайплайнов процессов анализа данных. Потоковые аналитические решения. Регистрация событий в логах, создание регулярных отчетов. Дашборды. Мониторинг. Система метрик для мониторинга продуктовой эффективности.

Прогнозы и предсказания - это предсказание, сделанное путем изучения статистических данных и прошлых моделей. Компании используют программные инструменты и системы для анализа больших объемов данных, собранных за

длительный период. Затем ПО прогнозирует будущий спрос и тенденции, помогая компаниям принимать более точные финансовые, маркетинговые и операционные решения.

1.5. Мониторинг профайлов

Мониторинг профайлов - это критически важная IT-функция, которая имеет широкий спектр преимуществ для предприятий любого размера. Он может сэкономить время на производительности труда сотрудников и затратах на инфраструктуру, и он гораздо более стратегический, чем предполагает его название. Мониторинг включает в себя наблюдение и отчетность о проблемах 24 часа в сутки 7 дней в неделю, а также оптимизацию потока данных и доступа к ним в сложной и меняющейся среде.

Система мониторинга может помочь найти решения широкого спектра проблем, включая медленную загрузку веб-страниц, потерю электронной почты, сомнительную активность пользователей и доставку файлов, вызванную перегрузкой, сбоем серверов и проблемами с сетевыми соединениями.

Частая проблема в 21 веке, является утечка данных. Мониторинг, позволяет отслеживать в режиме реального времени о том, кто, когда и где получает доступ к данным, позволяют информировать ИТ-отделы и владельцев данных о том, как используются их данные, чтобы у них был аудиторский след всех случаев доступа. Это помогает им убедиться, что данные используются должным образом авторизованными пользователями. Такая практика защищает от утечек информации и интеллектуальной собственности и незаконного присвоения.

Система мониторинга данных сообщает вам о месте загрузки данных, о том, кто был пользователем, соответствовало ли извлечение данных шаблону использования данных, ожидаемому для этого пользователя, и где было произведено извлечение. В качестве превентивной меры безопасности и криминалистики, знание того, где находятся ваши данные, кто их использует, как они используются и когда они используются, является важным аспектом

управления данными, который не полностью покрывают сетевые брандмауэры, проверки на уязвимости и обнаружение вторжений.

Проблема мониторинга данных вызывает широкий интерес у исследователей таких, как Гасанова И. С. [1], Алискаров С. Ж. [2], Мартышов М. И. [3], Константинов О. Г. [4], Бекенева Я. А. [5], Антонова, В. М. [6], Полтавцева М. А. [7], Дубровин М. Г. [8], Колесников И. А. [9], Горбунов П. Н. [10].

Исследование методов и анализа данных посвящены работы, Фиофанова О. А. [11], Сущенко Н. А. [12], Павлов А. Н. [13], Прохорова М. М. [14], Пиотровская К. Р. [15], Зунина Н. В. [16], Бова В. В. [17]. В работах предложена общая концепция метода мониторинга и анализ данных.

В качестве примера рассмотрим случай, когда компания покупает 100 рекламных объявлений в день и 3000 в месяц. В какой-то из дней по техническим причинам из рекламного кабинеты начинает отправляться по 100 объявлений в минуту, явно, что такое развитие событий не входило в планы маркетологов. Если у компании есть система мониторинга, то в тот самый момент, когда что-то идет не по сценарию, сотрудник получит оповещение о происходящем и сможет избежать данной проблемы.

Мониторинг фактически представляют собой отчеты о том, что происходит в настоящее время. Обычно они обеспечивают конкретные данные в рамках тщательно разработанных показателей. К сожалению, как и отчеты, они не сообщают, почему наблюдается рост загрузки ЦП, и не говорят, что следует предпринять прямо сейчас для решения проблемы, то есть они не дают важного контекста.

Мониторинги на самом деле отчеты о том, что происходит в данный момент. Обычно они также предоставляют конкретные данные в рамках тщательно разработанных индикаторов. К сожалению, они не говорят вам, почему растет загрузка ЦП, как и отчеты, и они не говорят вам, что вы должны сделать прямо сейчас, чтобы решить проблему они не дают вам важного контекста.

Нет причинно-следственного объяснения. Это момент, когда системные администраторы или инженеры по эксплуатации начинают изучать журнал регистрации событий, чтобы понять, что происходит,

почему и как это исправить: сделать откат назад, раскрутить дополнительные серверы, перенастроить выравниватель нагрузки и так далее.

На рисунке 3 приведен пример загрузки сервера. С небольшими вариациями на протяжении дня очередь выполнения составляет 0,5 или меньше. В час ночи загрузка начинает расти и за 30 минут увеличивается до пяти и выше, в десять раз по сравнению с «нормой». Ситуация нестандартная. Что происходит? Возможно, требуется вмешательство?

В данном случае это всего лишь еженедельное резервное копирование данных. Оно осуществляется каждый четверг в час ночи. Это абсолютно штатная ситуация. Мы имеем четкие данные и ясно представленные показатели. Нет только контекста: что причина повышения загрузки - резервное копирование данных, что оно ожидаемо и запланировано происходит в определенное время и что сервер спокойно справляется с этой загрузкой.

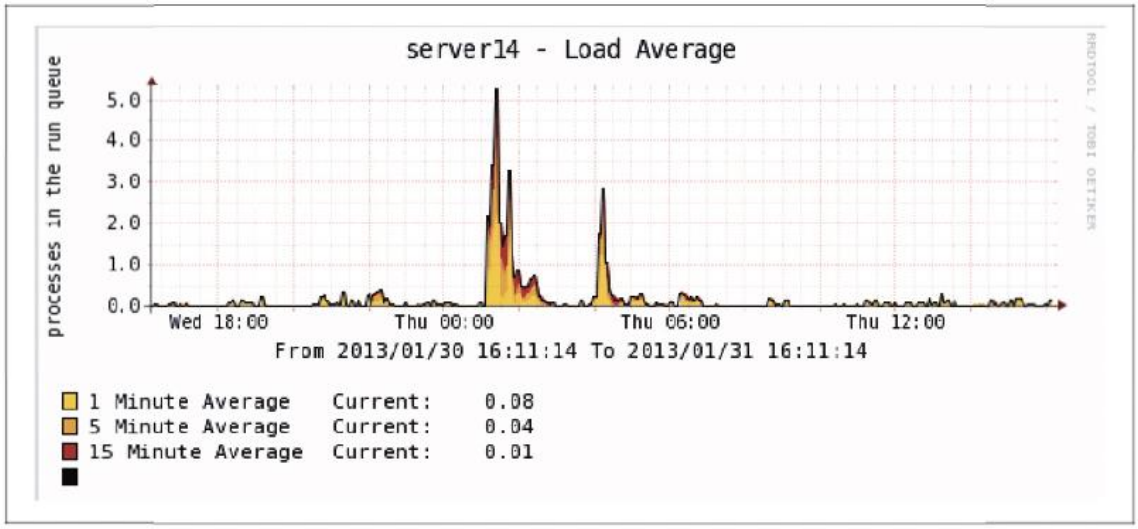

Рисунок 3 - Загрузка сервера

Исходя из вышесказанного, разработка методов мониторинга и обработки информации, позволяет быстро получать статистически значимые и точные результаты исследования, является актуальной проблемой.

Объектом исследования являются данные о логах рекламных событий включающие в себя различные сведения, по которым будет разработан алгоритм системы мониторинга.

Предметом исследования являются методы мониторинга и анализа данных.

Основной целью выпускной квалификационной работы является разработка алгоритма для создания системы мониторинга профайлов. Данная система мониторинг позволит мгновенно получать оповещение о происходящем в базе данных, что в свою очередь в дальнейшем позволит принимать правильные решения для компании. Такой метод позволит снять нагрузку на специалистов, повысить производительность и эффективность всего отдела аналитики.

1.5.1. Пример мониторинга профайлов в компаниях

В успешных компаниях практика применения мониторинга является неотъемлемой частью.

На рисунке 4 представлен мониторинг профайлов для верхнеуровневый аналитики, которые позволяет отслеживать важные метрики для SaaS-компании.

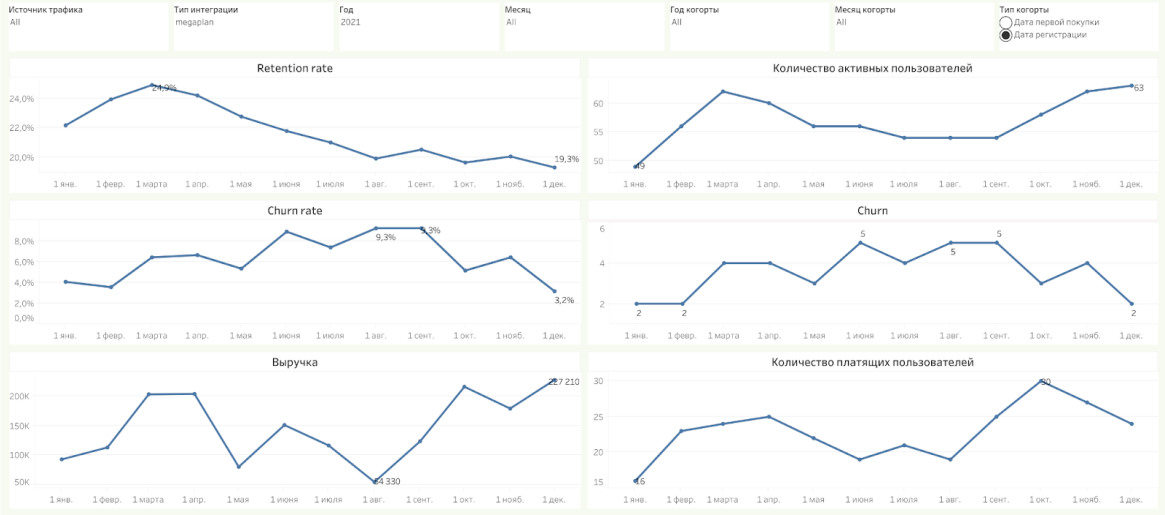

Рисунок 4 - Мониторинг профайлов для верхнеуровневый аналитики

Ярким примером мониторинга является отслеживание дохода компании, самая важная часть бизнеса, представлена на рисунок 5.

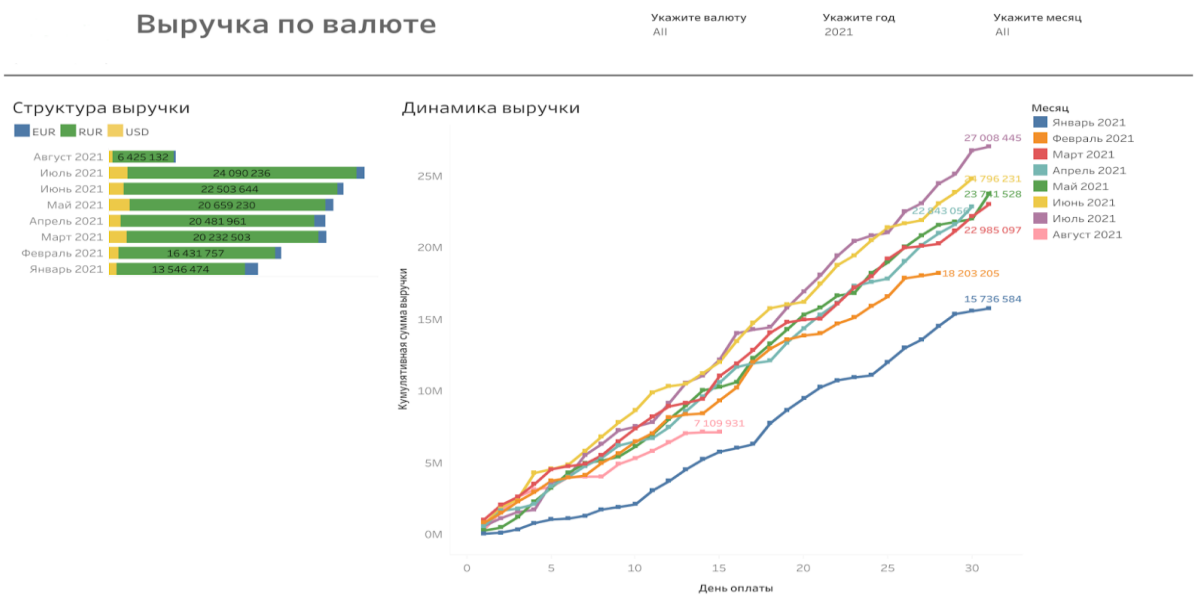

Рисунок 5 – Мониторинг выручки

1.6. От мониторинга к анализу

Составление отчетов и проведения мониторинга - необходимые факторы управления на основе данных, но этого недостаточно. Хотя не стоит недооценивать важность двух этих видов деятельности. Подготовка отчетов чрезвычайно важна для управления на основе данных: компания не сможет быть эффективной без этого элемента. А вот обратное не обязательно верно: существует множество организаций, сосредоточенных на отчетности, у которых может не быть качественного анализа. Составление отчетности может быть вызвано официальными требованиями.

Данные отчетов информируют, что произошло в прошлом. Кроме того, они могут быть тем фундаментом, с которого можно наблюдать за изменениями и тенденциями. Они могут представлять интерес для инвесторов и акционеров, но в целом это ретроспективный взгляд на ситуацию. Для управления на основе данных нужно двигаться дальше. Необходимо прогнозировать развитие ситуации, на основе анализа стараться понять, почему меняются показатели, и, где возможно, проводить эксперименты для сбора данных, которые могут помочь понять причины.

Если сравнивать эти понятия:

Отчетность - процесс организации данных в информационные сводки для отслеживания того, как функционируют разные сферы бизнеса.

Мониторинг - процесс наблюдения и регистрации данных о каком-либо объекте на неразрывно примыкающих друг к другу интервалах времени, в течение которых значения данных существенно не изменяются.

Анализ - преобразование данных в выводы, на основе которых будут приниматься решения и осуществляться действия с помощью людей, процессов и технологий.

В таблице 1 суммированы отличия между этими понятиями. Теперь должно быть очевидно, почему анализ и управление на основе данных - настолько важный компонент ведения бизнеса. Это факторы, способные дать компании новые направления развития или вывести ее на новый уровень эффективности.

Таблица 1 - Основные характеристики мониторинга и анализа

Мониторинг	**Анализ**
Описательный	Дает рекомендации
Что происходит?	Почему?
Ретроспективный	Перспективный
Поднимает вопросы	Отвечает на вопросы
Данные → информация	Данные + инофрмация → выводы
Отчеты, дэшборды, мониторинги	Наблюдения, рекомендации, прогнозы
Отсутствие контекста	Контекст + история

Таблица 2 - Гипотетические основные вопросы, на которые отвечает аналитика, по Дэвенпорту (на основе работы Дэвенпорта)

	Прошлое	**Настоящее**	**Будущее**
Информация	A) Что случилось? Отчет	B) Что происходит сейчас? Оповещение	C) Что произойдет? Экстраполяция

Выводы	D) Как и почему это произошло? Моделирование, экспериментальное планирование	E) Какой следующий оптимальный шаг? Рекомендации	F) Что самое хорошее/плохое может произойти? Прогноз, оптимизация, симуляция

Пункт D представляет собой ценную аналитику, пункты E и F обеспечивают управление на основе данных, если эта информация стимулирует конкретные действия.

В нижнем ряду таблицы отражены действия, приводящие к выводам. Как уже отмечалось ранее, составление отчетов (A) и мониторинг (B) - не управление на основе данных: они отмечают, что уже произошло или что необычное или нежелательное происходит сейчас, но при этом не дают объяснений, почему это произошло или происходит, и не дают рекомендаций по улучшению ситуации. Предвестником управления на основе данных служит дальнейшее изучение причинно-следственных связей с помощью моделей или экспериментов (D). Только понимая причины произошедшего, можно сформулировать план действий или рекомендации (E). Пункты E и F обеспечивают управление на основе данных, но только если полученная информация стимулирует конкретные действия.

Пункт C представляет собой опасную зону, поскольку слишком велик соблазн распространить существующий тренд на будущее. Даже при обдуманном выборе функциональной формы модели может быть множество причин, почему этот прогноз ошибочен. Для уверенности в прогнозах следует использовать модель учета причинно-следственных связей.

Итак, в нижнем ряду таблицы отражены перспективные виды деятельности, включающие элементы причинно-следственного объяснения.

Основные результаты и выводы по первой главе

Проведено исследование по структуре анализа данных. Где были рассмотрены основные аспекты.

Из вышесказанного следует, что мониторинг в совокупности с анализом являются важным инструментом для бизнеса.

Далее в исследование будет представлен метод для эффективного мониторинга профайлов по логам событий.

2. АВТОМАТИЗАЦИЯ СБОРА, ОБРАБОТКИ И АНАЛИЗА ДАННЫХ

2.1. Дата-пайплайны для автоматизации

Основа автоматизации - дата-пайплайны. Их еще называют конвейерами данных. Дата-пайплайн - специальная программа. Она вызывается по расписанию: собирает, объединяет, трансформирует и сохраняет данные автоматически. собираются из яндекс метрики

Пайплайны позволяют:

- парсить данные в Интернете и сохранять результат в базу данных;
- собирать информацию о визитах и покупках пользователей из корпоративных систем и формировать отчеты для когортного анализа;
- отслеживать аномалии в поведении пользователей;
- анализировать A/B-тесты.

Элементы пайплайна рисунок 6 описывает аббревиатура ETL (от англ. extract, transform, load):

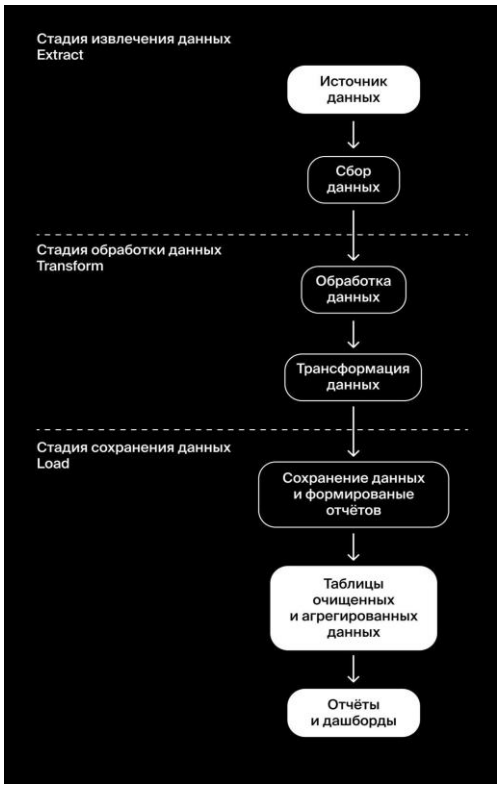

Рисунок 6 - Элементы пайплайна ETL

На стадии извлечения данных (Extract) пайплайн собирает информацию из разных источников: веб-сайтов, баз данных компании и партнеров, внешних API.

На этапе обработки данных (Transform) информация приводится к единому стандарту. Например, текст преобразуется в числа или даты. На этом же этапе данные категоризируют. Затем происходит трансформация данных: они приводятся к формату, удобному для построения отчетов или хранения агрегированных сведений. На этом этапе, например, данные о посещениях и продажах могут трансформировать в таблицы с рассчитанным LTV для когортного анализа.

На последнем шаге (Load) пайплайн сохраняет агрегированные данные в таблицы БД и, если нужно, формирует отчеты и рассылки.

Дата-пайплайн проектируют так, чтобы все его этапы можно было перезапускать снова, всякий раз получая стабильный, повторяющийся результат. Дело в том, что у систем, откуда добывают информацию, случаются аварии. Если такое произойдет, в пайплайн попадут неполные данные. Потому каждая программа внутри пайплайна должна иметь систему мониторинга. Это нужно, чтобы выявить проблему в консистентности данных, решить ее и получить правильные результаты обработки данных.

На практике пайплайны строят в готовых библиотеках, например Luigi, Bubbles или Airflow. Они позволяют строить целые системы пайплайнов, собирающие данные из множества источников, со взаимными зависимостями и по сложному расписанию.

2.2. Агрегация данных и создание таблиц в БД

На практике исходные данные для анализа хранят в базах. Чаще всего, это необработанные логи — журналы событий.

Объемы логов огромны. Постоянные запросы к ним нагружают базу, замедляют работу систем и занимают очень много времени. Чем чаще будет

обращение к сырым логам напрямую, тем чаще у администраторов базы данных будет возникать больше не нужной работы.

Аналитику сырые данные нужны редко: для отчетов и выводов хватает агрегированных.

Агрегирование или агрегация — процесс преобразования данных с высокой степенью детализации к более обобщенному представлению. Заключается в вычислении так называемых агрегатов — значений, получаемых в результате применения данного преобразования к некоторому набору фактов, связанных с определенным измерением. При этом чаще всего используется простое суммирование, вычисление среднего или медианы, выбор максимального или минимального значений.

Чтение больших объемов данных может занимать часы! Лучше завести новую таблицу, в которую сохранять данные, уже сгруппированные по признаку. При данном подходе количество данных уменьшается в миллион раз - подготовка отчета займет секунды вместо часов, снижается нагрузка на сервера.

2.2.1. Формирование агрегированных данных

```
CREATE TABLE имя_таблицы (
    первичный_ключ тип_данных,
    имя_столбца1 тип_данных,
    имя_столбца2 тип_данных,
    ...);
```

Рисунок 7 - Пример создания таблицы в SQL

В SQL много типов данных. Рассмотрим базовые:

- int — целое число;

- JSON;

- varchar(n) — строка, где n — ее максимальная длина. Например, varchar(128) задает поле, содержащее строку не длиннее 128 символов;

- timestamp — дата и время;

- serial — специальный тип данных для значений первичных ключей. Первичным ключом называют уникальный номер записи в таблице. Первичный ключ обозначают выражением: CREATE TABLE имя_таблицы (название поля с первичным ключом serial PRIMARY KEY).

Рассмотрим структуру датасета для исследования, предоставленного в таблице 3.

Таблица 3 – Структура датасета для исследования

Название поля	**Тип**	**Описание**
id	SERIAL	ID клиента
dateTime	TIMESTAMP	время события
utm	JSON	utm-метка
event	VARCHAR	Событие
campaignId	INT	ID компании
url	VARCHAR	
date	TIMESTAMP	Дата
pageTitle	VARCHAR	Заголовок
agentId	INT	ID агент
path	VARCHAR	Путь
device	VARCHAR	Устройство
details	JSON	
os	VARCHAR	
browser	VARCHAR	Браузер
adId	INT	ID рекламы
cost	VARCHAR	

Для будущего дата-пайплайна создадим таблицу, в которой можно хранить агрегированные данные:

```
CREATE TABLE agg_games_year (
    client_union_id serial PRIMARY KEY,
    campaign_union_id int,
    agency_union_id int,
    ad_id int,
    platform varchar,
    date_time timestamp,
    event varchar,
    ad_cost_type varchar
);
```

Рисунок 8 – Таблица для дата-пайплайна

2.3. Вертикальные и горизонтальные таблицы

При проектировании таблицы агрегированных данных важно помнить, что она должна быть как можно более вертикальной. Добиться этого можно, если следовать правилу: каждому признаку отводить отдельный столбец. Такой метод размещения данных намного лучше автоматизируется и масштабируется.

Таблица 4 – Вертикальная таблица

месяц	**id**	**сумма**
Январь 2022	1	100
Март 2022	1	200
Апрель 2022	1	30
Январь 2022	2	10
Апрель 2022	2	500
Январь 2022	3	14

Таблица 5 – Горизонтальная таблица

id	Январь 2022	Март 2022	Апрель 2022
1	100	200	30

Окончание таблицы 5

id	Январь 2022	Март 2022	Апрель 2022
2	10		500
3	14		

С «горизонтальной» таблицей возникнет проблема: при добавлении каждой новой колонки в пайплайне, а также в зависящих от него дашбордах и отчетах, придется заново определять тип столбца и правила обработки пропусков. Эту задачу, конечно, можно решить в коде. Однако гораздо проще строить первый вариант таблицы - в нем добавление новых данных не приведет к перераспределению их структуры.

2.4. Создание скрипта из пайплайна

Далее был написан скрипт, который читает данные из одной таблицы, модифицирует их и записывает в агрегирующую таблицу. Чтобы превратить его в полноценный пайплайн, нужно задать входные параметры. Для этого скрипту передаются два параметра: даты начала и конца временного интервала. Две даты нужны для случаев, когда понадобится запустить скрипт для нескольких лет разом, чтобы не вызывать его вручную несколько раз подряд. Также, чтобы не перегружать агрегированную таблицу будем удалять старые записи.

Листинг 1 - Автоматизированная обработка данных

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import sys
import getopt
from datetime import datetime
```

Продолжение листинга 1

```
import pandas as pd
from sqlalchemy import create_engine

if __name__ == '__main__':
    #Задаем входные параметры
    unixOptions = 'sdt:edt:'
    gnuOptions = ['start_dt=', 'end_dt=']
    fullCmdArguments = sys.argv
    argumentList = fullCmdArguments[1:] #excluding script
name

    try:
        arguments, values = getopt.getopt(argumentList,
unixOptions, gnuOptions)
    except
        getopt.error as err:
        print (str(err))
        sys.exit(2)

    start_dt = datetime.now()
    end_dt = datetime.timedelta(days=2*365)

    for currentArgument, currentValue in arguments:
        if currentArgument in ('-sdt', '--start_dt'):
            start_dt = currentValue
        elif currentArgument in ('-edt', '--end_dt'):
            end_dt = currentValue
```

Продолжение листинга 1

```
    db_config = {'user': 'my_user',
                 'pwd': 'my_user_password',
                 'host': 'localhost',
                 'port': 5432,
                 'db': 'db'}

    connection_string = 'postgresql://{}:{}@{}:{}/{}'.format(db_config['user'],
                                                             db_config['pwd'],
                                                             db_config['host'],
                                                             db_config['port'],
                                                             db_config['db'])
    engine = create_engine(connection_string)

    # Теперь выберем из таблицы только те строки,
    # которые были выпущены между start_dt и end_dt
    query = '''
    WITH report AS
    (
        SELECT
            id,
            dateTime,
            event,
            campaignId,
            agentId,
            device,
            os,
            adId,
```

Продолжение листинга 1

```
                cost
            FROM
                logs.event_log
            WHERE
                dateTime    BETWEEN    '{}'::TIMESTAMP    AND
'{}'::TIMESTAMP
                AND event in ('report', 'click', 'view')
        )

        SELECT
            id AS client_union_id,
            campaignId AS campaign_union_id,
            agentId AS agency_union_id,
            adId AS ad_id,
            os AS platform,
            dateTime AS date_time,
            event,
            CASE
                WHEN LEFT(cost, 3) = 'CPM'
                    THEN 'CPM'
                WHEN LEFT(cost, 3) = 'CPC'
                    THEN 'CPC'
                END AS ad_cost_type
        FROM
            report
            '''.format(start_dt, end_dt)
```

Продолжение листинга 1

```
    data_raw = pd.io.sql.read_sql(query, con = engine, index_col
= 'client_union_id')

        columns_numeric          =          ['client_union_id',
'campaign_union_id', 'agency_union_id', 'ad_id']
        columns_datetime = ['date_time']
        for  column  in  columns_numeric:  data_raw[column]  =
pd.to_numeric(data_raw[column], errors='coerce')
        for  column  in  columns_datetime:  data_raw[column]  =
pd.to_datetime(data_raw[column])

        #Удаляем старые записи
        query = '''DELETE FROM log_info
                        WHERE dat_time < '{}'::TIMESTAMP
                '''.format(start_dt)
        engine.execute(query)

        data_raw.to_sql(name  =  'log_info',  con  =  engine,
if_exists = 'append', index = False))
```

Окончание листинга 1

Основные результаты и выводы по второй главе

В данном разделе были описаны и продемонстрированы методы настройки сбора данных и автоматизации пайплайна.

На данном этапе была подготовлена вертикальная таблица, в которой будут храниться данные. Написан SQL - запрос для получения нужных данных из неструктурированной таблицы. Написан скрипт, который будет автоматически

выполнять подключение к базе данных, выполнять SQL - скрипт за нужный период времени, удалять старые записи в таблице и добавлять новые данные в таблицу. Данный этап позволит уменьшить нагрузку на сотрудников.

Данные метод является важным для дальнейшего мониторинга профайлов, так как он будет отвечать за постоянную выгрузку агрегированных данных.

3. СИСТЕМА МОНИТОРИНГА ПРОФАЙЛОВ

3.1. Выбор модели для оценки аномалий

Чтобы определять аномалии в бизнес-процессах, которые потребуется выбрать метрики, которые будут полностью отвечать поставленной задаче. одна или несколько метрик, которая наиболее подходящая для решения поставленной задачи

В данной выпускной квалификационной работе рассматривается три наиболее распространенные модели (набор математических уравнений), скользящих средних из них, будет выбрана оптимальная модель для поиска аномалий:

- простое скользящее среднее – SMA (simple moving average);

- взвешенное скользящее среднее – WMA (weighted moving average WMA) и экспоненциально взвешенное скользящее среднее;

- экспоненциальное скользящее среднее – EMA (exponentially weighted moving average —EWMA, exponential moving average).

Для модификации ряда, могут быть выбраны любые из 3-х моделей (SMA, WMA и EMA), в зависимости от типа расчетов и данных. Например, при расчете модели WMA в качестве весов может быть выбран номер очередности элемента ряда или показатель смежного ряда (например, объем продаж).

При сглаживании рядов и прогнозировании, применяются эти же формулы, с той разницей, что в первом случае расчетный период для SMAt является средним периодом, во втором он – последний, т. е. в случае прогнозирования, расчет основан на предшествующих периодах.

Скользящие средние обычно используются с данными временных рядов для сглаживания краткосрочных колебаний и выделения основных тенденций или циклов.

Экстраполяция тенденции как метод прогнозирования - основа большинства методов прогнозирования, в том числе – в адаптивных моделях на основе скользящих средних – с коротким прогнозным интервалом.

Адаптивные методы позволяют при изучении тенденции учитывать степень влияния предыдущих уровней на последующие значения динамического ряда. К адаптивным методам относятся методы скользящих и экспоненциальных средних, метод гармонических весов, методы авторегрессионных преобразований.

3.1.1. Простое скользящее среднее (SMA)

Простое скользящее среднее, или арифметическое скользящее среднее (simple moving average) численно равно среднему арифметическому значений исходной функции за установленный период и вычисляется по формуле:

$$SMA_t = \frac{1}{n}\sum_{i=0}^{n-1} p_{t-i} = \frac{p_{t-i} + p_{t-1} + \cdots + p_{t-i} + \cdots + p_{t-n+2} + p_{t-n+1}}{n} \qquad (1)$$

где:

- SMA_t — значение простого скользящего среднего в точке t;

- n — количество значений исходной функции для расчета, скользящего среднего (сглаживающий интервал), чем шире сглаживающий интервал, тем более плавным получается график функции;

- p_{t-i} — значение исходной функции в точке $t - i$.

Полученное значение простой скользящей средней относится к середине выбранного интервала, однако, традиционно его относят к последней точке интервала.

Из предыдущего своего значения простое скользящее среднее может быть получено по следующей рекуррентной формуле:

$$SMA_t = SMA_{t-1} - \frac{p_{t-n}}{n} + \frac{p_t}{n}, \qquad (2)$$

где:

- SMA_{t-1} — предыдущее значение простого скользящего среднего;

- p_{t-n} — значение исходной функции в точке $t-n$ (в случае временного ряда, самое «раннее» значение исходной функции, используемое для вычисления предыдущей скользящей средней);

- p_t — значение исследуемой функции в точке t (в случае временного ряда, текущее — последнее значение).

Данной формулой удобно пользоваться, чтобы избежать регулярного суммирования всех значений.

Выделяют следующие недостатки простого скользящего среднего:

1. Равенство весового коэффициента 1.

2. Двойная реакция на каждое значение: в момент входа в окно вычислений и в момент выхода из него.

3.1.2. Взвешенное скользящее среднее

Иногда при построении скользящей средней некоторые значения исходной функции целесообразно сделать более значимыми. Например, если предполагается, что внутри интервала сглаживания имеет место нелинейная тенденция, или, в случае временных рядов, последние — более актуальные — данные могут быть весомее предыдущих.

Бывает, что исходная функция многомерна, то есть представлена сразу несколькими связанными рядами. В этом случае может возникнуть необходимость объединить в итоговой функции, скользящей средней все полученные данные. Например, временные ряды биржевых цен обычно для каждого момента времени представлены как минимум двумя значениями — ценой сделки и ее объемом. Необходим инструмент для вычисления скользящей средней цены, взвешенной по объему.

В этих и подобных случаях применяются взвешенные скользящие средние.

Взвешенное скользящее среднее (weighted moving average), точнее линейно взвешенное скользящее среднее — скользящее среднее, при вычислении которого

вес каждого члена исходной функции, начиная с меньшего, равен соответствующему члену арифметической прогрессии. То есть, при вычислении WMA для временного ряда, мы считаем последние значения исходной функции более значимыми чем предыдущие, причем функция значимости линейно убывающая.

Формула вычисления скользящей средней примет вид:

$$WMA_t = \frac{n \cdot p_t + (n-1) \cdot p_{t-1} + \cdots + (n-i) \cdot p_{t-i} + \cdots + 2 \cdot p_{t-n+2} + 1 \cdot p_{t-n+1}}{n + (n-1) + \cdots + (n-i) + \cdots + 2 + 1} = \frac{2}{n \cdot (n+1)} \sum_{i=0}^{n-1} (n-i) \cdot p_{t-i}, \quad (3)$$

где:

- WMA_t — значение взвешенного скользящего среднего в точке t;

- n — количество значений исходной функции для расчета, скользящего среднего;

- p_{t-i} — значение исходной функции в момент времени, отдаленный от текущего на i интервалов.

При этом знаменатель функции, в этом случае, равен треугольному числу — сумме членов арифметической прогрессии с начальным членом и шагом равными 1:

$$\frac{n \cdot (n+1)}{2} \quad (4)$$

3.1.3. Экспоненциальное скользящее среднее

Экспоненциально взвешенное скользящее среднее, экспоненциальное скользящее среднее (exponentially weighted moving average) — разновидность взвешенной скользящей средней, веса которой убывают экспоненциально и никогда не равны нулю. Определяется следующей формулой:

$$EMA_t = \alpha \cdot p_t + (1-\alpha) \cdot EMA_{t-1} \quad (5)$$

где:

- EMA_t — значение экспоненциального скользящего среднего в точке t (последнее значение, в случае временного ряда);

- EMA_{t-1} — значение экспоненциального скользящего среднего в точке $t-1$ (предыдущее значение в случае временного ряда);

- p_t — значение исходной функции в момент времени (последнее значение, в случае временного ряда);

- α — (сглаживающая константа) коэффициент, характеризующий скорость уменьшения весов, принимает значение от 0 и до 1, чем меньше его значение, тем больше влияние предыдущих значений на текущую величину среднего.

Первое значение экспоненциального скользящего среднего, обычно принимается равным первому значению исходной функции:

$$EMA_t = p_0 \tag{6}$$

Коэффициент α, может быть выбран произвольным образом, в пределах от 0 до 1. Например, он может быть выражен через величину окна усреднения:

$$\alpha = \frac{2}{n + 1} \tag{7}$$

Изучив три модели скользящего среднего, было решено использовать экспоненциальное скользящее среднее, т.к. при использовании скользящего среднего почти всегда возникает лаг, и для его уменьшения необходима экспоненциальная скользящая средняя. Показатель Exponential Moving Average (EMA) придает последним данным больший вес по сравнению с предыдущими значениями. Этот факт позволяет реагировать на текущие изменения быстрее, чем при использовании простых скользящих средних. Вес последней зависит от периода, скользящего среднего, и чем этот период короче, тем больший вес имеет последняя цена. Проводя подробные вычисления, нетрудно заметить, что экспоненциальное скользящее среднее будет сложнее в расчете, чем простая средняя сумма.

3.2. Алгоритм для поиска аномалий

Во время учебы был написан алгоритм по поиску аномальных значений в рекламных объявлениях и успешно применен на практике. В данном пункте будет подробное описание работы функции.

3.2.1. Функция EMA и ее параметры

def exponential_moving_average(df, period ='H', window_size=2, plot=False, score_range=1.96, ax=None, bound_param='r--', current_value=True, trend_value=True, day=None, anomalies_value=False):

Рассмотрим каждый параметр функции:

1. df - Подставляем подготовленный Датафрейм.

Датафрейм – это двумерная структура данных со столбцами и строками. Это специальный аналог таблицы Excel или SQL – наборе Серий (Series) и наиболее часто используемый объект библиотеки Pandas.

2. Period - по какому временном промежутку будут агрегированны данные (дни/часы и т.п.). По Умолчанию алгоритм агрегирует данные по часу.

3. Window_size - выбираем нужный размер окна.

Окно — весовая функция, которая используется для управления эффектами, обусловленными наличием боковых лепестков в спектральных оценках (растеканием спектра). Имеющуюся конечную запись данных или имеющуюся конечную корреляционную последовательность удобно рассматривать как некоторую часть соответствующей бесконечной последовательности, видимую через применяемое окно. Например, последовательность наблюдаемых данных $x_0[n]$ из N отсчетов математически можно записать как произведение прямоугольной функции единичной амплитуды:

$$rect[n] = 1, 0 \leq n \leq N - 1 \quad (8)$$

и бесконечной последовательности $x[n]$:

$$x_0[n] = x[n] \cdot rect[n] \quad (9)$$

4. Plot - отрисовывать график или нет.

5. Score_range - Z-значение для доверительного интервала.

6. Bound_param - Цвет и тип линии.

7. Current_value - По умолчанию True, если на графике нужна линия наблюдаемых значений.

8. Trend_value - По умолчанию, True, если на графике нужна линия тренда.

9. Day - День, для которого смотрим статистику.

10. Anomalies_value - По умолчанию, True, если нужно вывести аномальные события.

Сначала функция группирует данные по временным промежуткам:

```
data_sample = pd.DataFrame(df.resample(rule='H').size(), columns=['cnt'])
```

date_time	cnt
2022-04-01 00:00:00	100
2022-04-01 01:00:00	46
2022-04-01 02:00:00	33
2022-04-01 03:00:00	46
2022-04-01 04:00:00	47
...	...
2022-04-16 19:00:00	465
2022-04-16 20:00:00	590
2022-04-16 21:00:00	530
2022-04-16 22:00:00	378
2022-04-16 23:00:00	314

Рисунок 9 – Группировка данных по временным промежуткам

Далее рассчитываем экспоненциально взвешенное скользящее среднее:

```
exp_ma = data_sample.ewm(span=2, adjust=False).mean()
```

date_time	cnt
2022-04-01 00:00:00	100.000000
2022-04-01 01:00:00	64.000000
2022-04-01 02:00:00	43.333333
2022-04-01 03:00:00	45.111111
2022-04-01 04:00:00	46.370370
...	...
2022-04-16 19:00:00	454.573367
2022-04-16 20:00:00	544.857789
2022-04-16 21:00:00	534.952596
2022-04-16 22:00:00	430.317532
2022-04-16 23:00:00	352.772511

Рисунок 10 – Расчёт экспоненциально взвешенного скользящего среднего

Следующий шаг — это получение верхних и нижних границ доверительного интервала:

```
std = np.std(data_sample[2:] - exp_ma[2:])
```

```
cnt    40.855504
dtype: float64
```

```
lower_bound = exp_ma - (1.96 * std)
lower_bound.head(3)
```

date_time	cnt
2022-04-01 00:00:00	19.923213
2022-04-01 01:00:00	-16.076787
2022-04-01 02:00:00	-36.743454

```
upper_bound = exp_ma + (1.96 * std)
upper_bound.head(3)
```

date_time	cnt
2022-04-01 00:00:00	180.076787
2022-04-01 01:00:00	144.076787
2022-04-01 02:00:00	123.410120

Рисунок 11 – Получение верхних и нижних границ доверительного интервала

Предпоследний шаг — это поиск аномальных значений:

```
anomalies = pd.DataFrame(index=data_sample.index, columns=data_sample.columns)
anomalies[data_res < lower_bound] = data_res[data_res < lower_bound]
anomalies[data_res > upper_bound] = data_res[data_res > upper_bound]
```

Рисунок 12 – Поиск аномальных значений

После получения всех значений приступим к финальному этапу - сравнений значений за разный период времени и отрисовке графика:

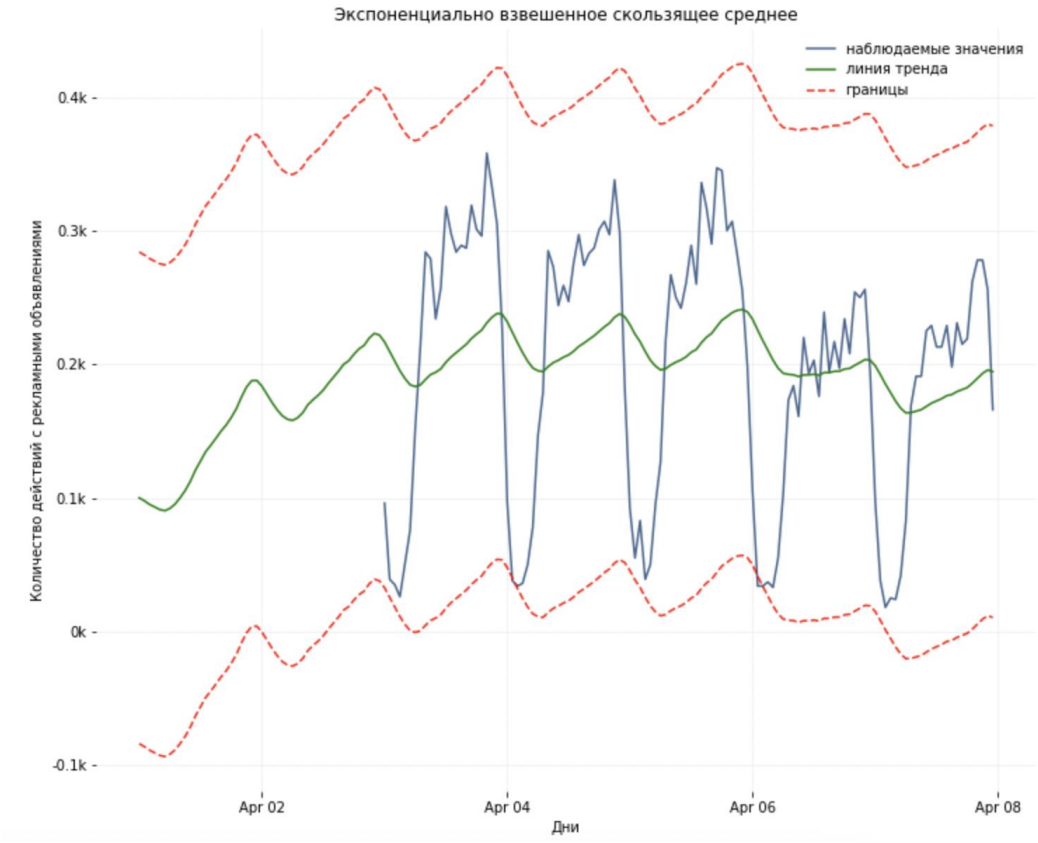

Рисунок 13 – График экспоненциально взвешенного скользящего среднего без аномалий (Python)

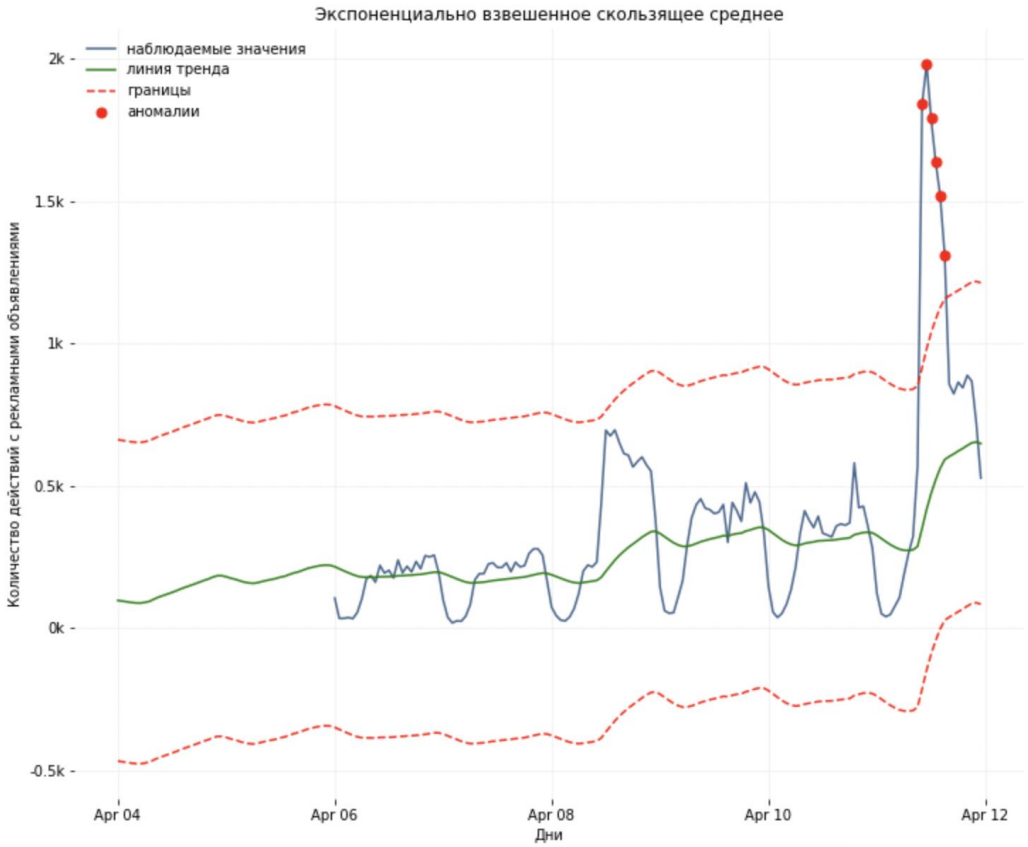

Рисунок 14 – График экспоненциально взвешенного скользящего среднего с аномалиями (Python)

После выполнения алгоритма, мы выявили 5 аномальных событий, данный мониторинг работает и будет сообщать о таких событиях ежедневно, чем безусловно поможет специалистам моментально детектировать различные отклонения.

3.3. Планировщик задач

Для организации задач по мониторингу был использован планировщик Airflow. Airflow — это набор библиотека для разработки, планирования и мониторинга рабочих процессов. Основная особенность Airflow: для описания (разработки) процессов используется код на языке Python. Отсюда вытекает масса преимуществ для организации проекта и разработки: ETL-проект — это просто

Python-проект, и его можно организовывать как нам удобно, учитывая особенности инфраструктуры, размер команды и другие требования.

Далее узнаем основные сущности Airflow. Поняв их суть и назначение, мы оптимально организуете архитектуру процессов.

3.3.1. Directed Acyclic Graph

Directed Acyclic Graph — это задачи, которые будут выполнены в строгой последовательности и по определенному расписанию записанные в один файл на Python. Задачи имеют некоторую смысловую связь, позволяющую считать их одним процессом. Файл, который создается для определения DAG по сути является не просто скриптом, обрабатывающим данные, а объектом. В процессе выполнения задачи DAG не могут пересекаться, так как они выполняются на разных объектах и в разное время.

Каждый DAG имеет собственное расписание - то, что определяет время и периодичность его запуска для планировщика задач. Типичный DAG может представлять собой последовательность следующих действий: забирает данные, обрабатывает их, выводит данные (записывает в базу данных, например). Этот процесс и является ETL (Extract-Transform-Load). Для выполнения отдельных задач в процессе разработчиком реализуются операторы. По сути, DAG — это набор операторов с заданной последовательностью их выполнения.

На рис. 15 представлен пример DAG, для запуска мониторинга каждый час.

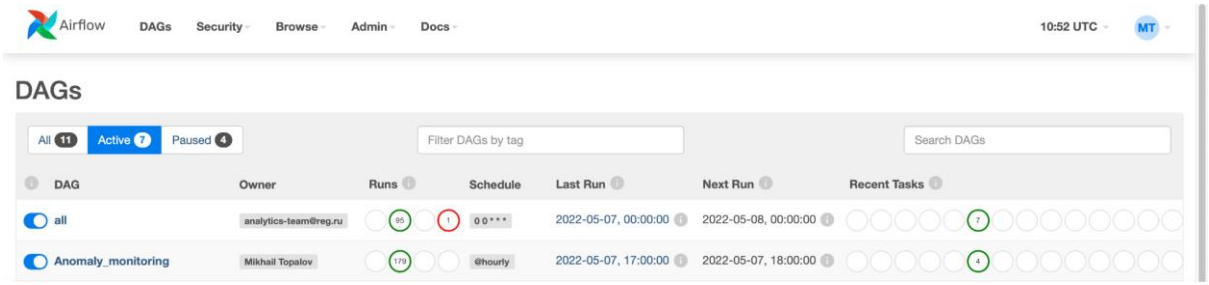

Рисунок 15 - DAG

Если рассмотреть DAG anomaly_monitoring, более подробно рис. 16 то можно увидеть, что сначала идет выполнение скрипта по сбору данных и их агрегации. Далее начинает выполняться сам скрипт по поиску аномалий. В самом конце в зависимости от результата проверки на аномалии, будет отправлено оповещение обнаружены ли какие-либо аномалии или нет.

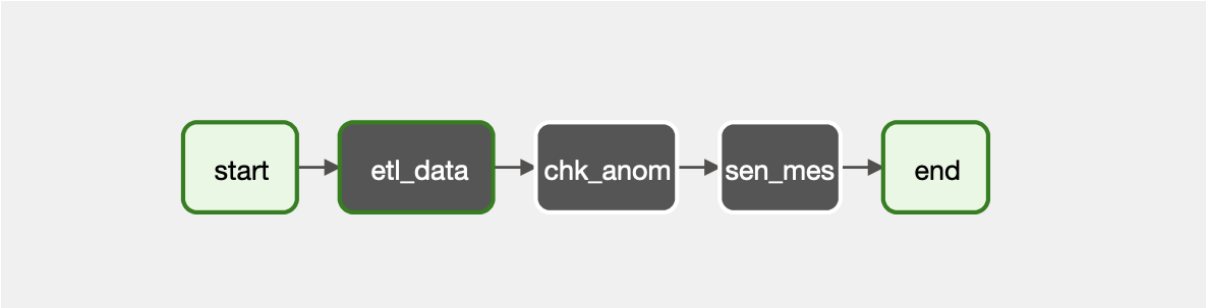

Рисунок 16 – Блок схема DAG

3.3.2. Операторы Airflow

Оператор — это сущность, на основании которой создаются экземпляры заданий, где описывается, что будет происходить во время исполнения экземпляра задания. Релизы Airflow с GitHub уже содержат набор операторов, готовых к использованию. Примеры:

- BashOperator — оператор для выполнения bash-команды;
- PythonOperator — оператор для вызова Python-кода;
- EmailOperator — оператор для отправки email;
- HTTPOperator — оператор для работы с http-запросами;
- SqlOperator — оператор для выполнения SQL-кода;
- Sensor — оператор ожидания события (наступления нужного времени, появления требуемого файла, строки в базе БД, ответа из API — и т. д., и т. п.).

Есть более специфические операторы: DockerOperator, HiveOperator, S3FileTransferOperator, PrestoToMysqlOperator, SlackOperator.

Также есть возможность разрабатывать операторы, ориентируясь на свои особенности, и использовать их в проекте. По сути, как только в проекте возникает часто используемый код, построенный на базовых операторах, можно задуматься

о том, чтобы собрать его в новый оператор. Это упростит дальнейшую разработку, и вы пополните свою библиотеку операторов в проекте.

3.3.3. Планировщик Airflow

Для планировщика в Apache Airflow используется Celery - Python-библиотека, которая позволяет организовать асинхронное и распределенное исполнение задач, а также очередь из них. Все задачи разделены на пулы, которые создаются вручную в интерфейсе Apache Airflow. Пулы используются в основном для ограничения нагрузки на один источник или для типизации задач. Для управления пулами используется так же web-интерфейс Apache Airflow. Для пулов задаются их размеры - количество доступных слотов для задач. При создании DAG приписывается одному из пулов.

Scheduler - собственно сам планировщик. Отвечает за планировку всех задач в Apache Airflow и занимается тем, что ставит эти задачи на выполнение. Алгоритм его работы следующий: если в DAG выполнены все предыдущие задачи, то происходит сортировка очереди по приоритетам задач и если в пуле есть свободное место под задачу, а так же есть свободный celery worker(процесс в Celery выполняющий работу) задача отправляется в него на выполнение.

Для того чтобы планировщик начал работать с DAG необходимо установить аргумент shedule_interval. Есть уже готовые свойства для задания времени. Например: @once, @hourly, @daily, @weekly, @monthly, @yearly.

3.3.4. Execution date Airflow

Ключевая особенность Apache Airflow состоит в том, что все запуски задач в DAG получают свою Execution date. То есть хранятся все запуски задач DAG за определенные промежутки времени. Это позволяет еще раз в точности воспроизводить результаты, полученные в той или иной Execution date. Так же получается, что различные задачи в DAG могут работать одновременно в разных

Execution date. При корректировке кода задачи, запуски задач в предыдущих Execution date будут уже с учетом этих корректировок. При этом, очевидно, теряется воспроизводимость результатов, но появляется возможность протестировать новый алгоритм на старых данных.

3.3.5. Хранилище Airflow

В Airflow есть свой бекенд-репозиторий. Поскольку Airflow был построен для взаимодействия с его метаданными с помощью большой библиотеки SqlAlchemy, должна быть возможность использовать любую базу данных, поддерживаемую в качестве серверной части SqlAlchemy. Рекомендуется использовать MySQL или Postgres. В базе хранятся состояния задач, DAG’ов, глобальные переменные и т. д.

Выводы по третьей главе

В данной главе был предложен и реализован алгоритм по поиску аномальных значений с использованием экспоненциально взвешенным скользящим средним. Данный метод помогает автоматизировано искать выбивающиеся значения.

Для автоматизации всего процесса поиска аномальных значений был использован планировщик airflow, который будет запускать скрипт по расписанию для отслеживания аномалий.

В настоящее время немногие компании используют мониторинг профайлов и из-за того, что вовремя не могут отследить проблемы, который происходят в будущем теряют деньги, что плохо сказывается на бизнесе.

Данные метод поможет бизнесу моментально отслеживать проблемы, если они возникают, принимать верные решения и развиваться в нужном направлении.

ЗАКЛЮЧЕНИЕ

При решении поставленных задач получены результаты, которые заключаются в следующем:

1. Выявлены специфические аспекты мониторинга профайлов, связанные с особенностями его использования и реализации.

2. Проведен анализ методов мониторинга профайлов. Выделены их преимущества и недостатки.

4. Предложены и реализованы модели скользящего среднего, позволяющие выявлять быстрее изменения в данных. Выделены основные преимущества и недостатки методов.

6. Разработана алгоритм, позволяющая автоматизировано проводить мониторинг профайлов, искать выбивающиеся значения. Что поможет бизнесу не терять деньги на не выявленных проблемах.

На основании полученных результатов были сделаны следующие выводы:

1. Предложенные методы мониторинга профайлов позволяют выявить основные аномалии в данных.

2. Разработанный алгоритм позволяют обрабатывать большой объем данных, а также моментально реагировать на возникающие проблемы в бизнесе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алискаров, С. Ж. Аналитика больших данных: скрапинг данных для мониторинга цен / С. Ж. Алискаров // "Вестник казахской академии транспорта и коммуникаций им М. Тынышпаева", 2018. - С. 210-216.

2. Антонова, В. М. Реализация технологии IoT для мониторинга данных через облачный сервис / В. М. Антонова // "T-comm", 2021. - С. 46-53.

3. Бекенева, Я. А. Преобразование данных от разнородных систем мониторинга / Я. А. Бекенева // Программные продукты и системы. 2019. - Т. 32. - № 2. - С. 197–206.

4. Бова, В. В. Оценка эффективности метода поиска ассоциативных правил для задач обработки больших данных / В. В. Бова, Э. В. Кулиев, С. Н. Щеглов // Известия ЮФУ. Технические науки. 2020. – № 2(212). – С. 66-78.

5. Гасанова, И. С. Разработка инструментов для мониторинга данных о региональной экономической безопасности / И. С. Гасанова // «Экономика: вчера, сегодня, завтра", 2020. - С. 123-133.

6. Свидетельство о государственной регистрации программы для ЭВМ № 2015613837 Российская Федерация. Автоматизированная система анализа и мониторинга данных единого портала государственных закупок: № 2015610608: заявл. 06.02.2015: опубл. 26.03.2015 / П. Н. Горбунов, С. Н. Лизин, М. А. Логинов [и др.]; заявитель Закрытое акционерное общество «Эволента». – EDN USTBFY.

7. Дубровин, М. Г. Модели и методы проактивного мониторинга ИТ-систем / М. Г. Дубровин, И. Н. Глухих // Моделирование, оптимизация и информационные технологии. – 2018. – Т. 6. – № 1(20). – С. 314-324.

8. Зунина, Н. В. HR аналитика: как превращать данные в бизнес-решения / Н. В. Зунина // Человек. Социум. Общество. – 2021. – № 1. – С. 51-56.

9. Колесников, М. В. Мониторинг субъектов и среды финансово-экономической деятельности / М. В. Колесников, Д. П. Мотренко // Вестник университета ГУУ. – 2016. – № 7-8. – С. 152-156.

10. Мартышов, М. И. Предварительная обработка данных системного мониторинга для анализа профиля загрузки высокопроизводительных вычислительных систем / М. И. Мартышов // "Numerical methods and programming", 2021. - С. 230-238.

11. Никандров, А. А. Анализ образовательных данных дисциплины "Основы математической обработки информации" / А. А. Никандров, К. Р. Пиотровская // Проблемы теории и практики обучения математике: сб. научных работ – СПБ: Российский государственный педагогический университет им. А. И. Герцена, 2020. – С. 91-97.

12. Павлов, А. Н. Автоматизация процессов анализа данных, полученных в результате групповой работы экспертов / А. Н. Павлов // Информатика: проблемы, методология, технологии: материалы XV международной научно-методической конференции, Воронеж, 12 - 13 февраля 2015 года. – Воронеж: Воронежский государственный университет, 2015. – Р. 121-125.

13. Павлов, А. Н. Видеокомплекс аппаратуры для экологического мониторинга окружающей среды и океанологических исследований / А. Н. Павлов, О. Г. Константинов, К. А. Шмирко // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2015. – № 2(36). – С. 29-32.

14. Полтавцева, М. А. Управление данными при мониторинге информационной безопасности КФС / М. А. Полтавцева // Защита информации. Инсайд. – 2022. – № 2(104). – С. 10-15.

15. Прохорова, М. М. Анализ больших данных как перспективный метод анализа данных в Отечественной статистике / М. М. Прохорова // Глобальная экономика в XXI веке: роль биотехнологий и цифровых технологий : Сборник научных статей по итогам работы четвертого круглого стола с международным участием, 15–16 июня 2020 года. – М: Общество с ограниченной ответственностью "КОНВЕРТ", 2020. – С. 116-120.

16. Сарьян В. К. Использование Data Envelopment Analysis (DEA) для расчета эффективности использования частотного спектра в

инфокоммуникационной среде (ИКС)/ Сарьян В. К., Сущенко Н. А. // Труды НИИР. — 2009. — № 1. — С. 75-79.

17. Фиофанова, О. А. Методология аналитики данных в проектном управлении государственными программами развития образования / О. А. Фиофанова // Большие данные в образовании: доказательное развитие образования: Сборник научных статей II Международной конференции, Москва, 15 октября 2021 года. – М: Издательский дом «Дело» РАНХиГС, 2021. – С. 7-18.

ПРИЛОЖЕНИЕ

Листинг поиска аномальных значений для мониторинга профайлов

```
# Загрузка библиотек
import re
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.ticker import FuncFormatter
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')

# Загрузка датасета
dateparse = lambda x: pd.datetime.strptime(x, '%Y-%m-%d')
df = pd.read_csv('/home/jovyan/work/ed/diplom.csv', parse_dates=['date'], date_parser=dateparse)
df['date_time'] = pd.to_datetime(df['time'], unit='s')

df = df.sort_values(by='time').reset_index(drop=True)
# время в индекс, чтобы можно было делать resample
df = df.set_index('date_time')

# Функция мониторинга
def exponential_moving_average(df, period ='H', window_size=2, plot=False, score_range=1.96, ax=None, bound_param='r--',
                               current_value=True, trend_value=True, day=None, anomalies_value=False):
```

```
        data_sample = pd.DataFrame(df.resample(rule=period).size(), columns=['cnt'])
        exp_ma = data_sample.ewm(span=window_size, adjust=False).mean()

        # доверительные интервалы
        std = np.std(data_sample[window_size:] - exp_ma[window_size:])
        lower_bound = exp_ma - (score_range * std)
        upper_bound = exp_ma + (score_range * std)

        # ищем аномалии
        anomalies = pd.DataFrame(index=data_sample.index, columns=data_sample.columns)
        anomalies[data_sample < lower_bound] = data_sample[data_sample < lower_bound]
        anomalies[data_sample > upper_bound] = data_sample[data_sample > upper_bound]

        if plot:
            if ax is None:
                fig, ax = plt.subplots()
            if current_value:
                ax.plot(data_sample[window_size:], label='наблюдаемые значения', color='#45678f')
            if trend_value:
                ax.plot(exp_ma, 'g', label='линия тренда')
            ax.plot(upper_bound, bound_param, label='границы')
            ax.plot(lower_bound, bound_param)
            ax.plot(anomalies, 'ro', markersize=7, label = 'аномалии')
            ax.set_frame_on(False)
```

```
            ax.yaxis.set_major_formatter(FuncFormatter( lambda y, p:
'{:,g}k'.format(y/1000)))
            ax.xaxis.set_major_locator(mdates.DayLocator(range(2, 32, 2)))
            ax.xaxis.set_major_formatter(mdates.DateFormatter('%b %d'))
            ax.xaxis.set_minor_locator(mdates.YearLocator(month=4, day=2))
            ax.xaxis.set_minor_formatter(mdates.DateFormatter('\n%Y'))
            ax.set(xlabel='Дни', ylabel='Количество действий с рекламными
объявлениями',
                   title='Экспоненциально взвешенное скользящее среднее')
            ax.grid(True, color='#e2e2e2', alpha=0.5)

        plt.show()
```