**Документация, содержащая информацию, необходимую для эксплуатации экземпляра программного обеспечения**

**Комплекс модулей «Платформа Предикто»**

[1. Программный комплекс «Платформа Предикто» 3](#_s2fbl5o0xrr7)

[2. Как начать работу 4](#_67nh06ptzrbm)

[- Получите учётную запись и доступ к платформе (через PassMedia ID или корпоративную авторизацию); 4](#_d6okniru2z98)

[- Войдите в веб-интерфейс; 4](#_f6apnx2zypmw)

[- Ознакомьтесь с доступными модулями и интерфейсами; 4](#_wietmysg5kj3)

[- Используйте встроенные дашборды или настройте свои (если есть права); 4](#_3pnxbpbf0sfk)

[- Подключите Telegram-бота для оперативного получения метрик; 4](#_nkeobmh32f1h)

[- Работайте с сегментами и настройками профиля для повышения персонализации. 4](#_o68dzjwyjqxb)

[3. Модуль «Персонализация контента» 5](#_un7wrqx0ex4k)

[4. Модуль «Предсказание оттока» 10](#_o45hdps7plk6)

[5. Модуль «Сквозная аналитика» 13](#_scu6s4ilmkni)

[6. Модуль «Инфраструктура для сервиса» 24](#_gyl0i73mk41i)

[7. Модуль «BI и adhoc» 26](#_p8b57rqzzzfa)

[6. Модуль «Telegram бот» 34](#_ntlko78osnt4)

[7. Модуль «Модерация контента» 37](#_brweuk89qlun)

[8. Модуль «Сегментатор данных» 39](#_r8syevabfggb)

[9. Модуль «Тегизация» 43](#_g7u99exsnl6z)

[10. Модуль «Рекомендательная система для стриминговых сервисов» 47](#_h9vqitmibr9s)

[11. «Поисковый модуль для стриминговых сервисов» 50](#_nsx2269v8x8z)

[12. Модуль «BI аналитика для стриминговых сервисов» 59](#_wfisjt295qah)

Данный документ предназначен для специалистов технических подразделений: DevOps-инженеров, администраторов платформ, системных архитекторов, специалистов по ML/DS-инфраструктурам, BI-аналитиков и разработчиков, осуществляющих проектирование, установку, эксплуатацию и сопровождение Платформы «Предикто».

# 

# **Комплекс модулей «Платформа Предикто»**

представляет собой интегрированную модульную систему, предназначенную для поддержки задач аналитики, предиктивного моделирования, рекомендательных алгоритмов и автоматизации взаимодействия с пользователями медиасервисов. Архитектура комплекса построена на принципах микросервисной организации, что обеспечивает масштабируемость, отказоустойчивость и гибкость интеграции в существующую ИТ-инфраструктуру заказчика.

Платформа реализует сквозную обработку данных — от первичного сбора и хранения до их последующей обработки, агрегации и визуализации результатов. Компоненты платформы развертываются в облачной среде (предпочтительно Yandex Cloud) с использованием инструментов контейнеризации и оркестрации (Docker, Kubernetes, Helm), обеспечивая автоматическое масштабирование и высокую доступность.

Каждый модуль платформы разрабатывается как самостоятельный сервис с собственным API, хранилищем данных и конфигурацией, и может быть задействован как в составе общей системы, так и в изолированном режиме. Все модули взаимодействуют через защищённые REST/gRPC-интерфейсы и единый слой авторизации (например, PassMedia ID).

В состав программного комплекса входят следующие модули (см. структуру документации по разделам):

1. **Модуль персонализации контента**;
2. **Модуль предсказания оттока пользователей**;
3. **Модуль сквозной аналитики**;
4. **Модуль BI и Adhoc аналитики**;
5. **Модуль Telegram-бота мониторинга бизнес-показателей**;
6. **Модуль инфраструктурного обеспечения и хранения данных**;
7. **Модуль сегментации пользовательской базы**;
8. **Модуль автоматической тегизации контента**;
9. **Модуль модерации контента**;
10. **Поисковый модуль для стриминговых сервисов**;
11. **Рекомендательная система для стриминговых платформ**;
12. **BI-аналитика и A/B тестирование для медиаплатформ**.

# 

# **Как начать работу**

# 

# Получите учётную запись и доступ к платформе (через PassMedia ID или корпоративную авторизацию);

# Войдите в веб-интерфейс;

# Ознакомьтесь с доступными модулями и интерфейсами;

# Используйте встроенные дашборды или настройте свои (если есть права);

# Подключите Telegram-бота для оперативного получения метрик;

# Работайте с сегментами и настройками профиля для повышения персонализации.

# 

# 

# **Модуль «Персонализация контента»**

Программное обеспечение модуль "Персонализация контента" предназначено для обработки пользовательских данных и формирования рекомендаций видеоконтента на основе предпочтений и истории просмотров. Основная цель системы – предоставление пользователям персонализированного опыта потребления медиаконтента в онлайн платформе.

Данный документ содержит информацию, необходимую для установки, настройки и эксплуатации системы. Он предназначен для технических специалистов, администраторов системы, инженеров DevOps и других экспертов, осуществляющих проверку и развертывание системы в рабочем окружении. Также в документе приведены инструкции по управлению системой, работе с API, мониторингу состояния и обеспечению безопасности.

Система основана на технологиях машинного обучения и использует алгоритмы коллаборативной и контентной фильтрации. Основная цель - формировать рекомендации, учитывая как предпочтения отдельных пользователей, так и глобальные тренды среди всех зрителей платформы.

Документ включает описание архитектуры системы, требований к программному и аппаратному обеспечению, инструкцию по установке и эксплуатации, а также руководство пользователя, описывающее основные функции системы.

2.2. Установка программы

Требования к окружению

Перед установкой ПО необходимо убедиться, что система соответствует следующим требованиям:

Операционная система: Ubuntu (рекомендуется 20.04 и выше)

Виртуализация: Yandex Cloud, Docker, Kubernetes

База данных: PostgreSQL, Redis, Qdrant

Облачное хранилище: Yandex Object Storage

Программные компоненты: FastAPI, gRPC, PyTorch, CatBoost, scikit-learn

***Минимальные аппаратные требования:***

* 4 vCPU
* 8GB RAM
* 100GB SSD

***Рекомендуемые аппаратные требования:***

* 8 vCPU
* 16GB RAM
* 500GB SSD

2.3. Процесс установки

**Подготовка инфраструктуры:**

* + Развертывание облачного окружения в Yandex Cloud.
  + Настройка контейнеризации с использованием Docker и Kubernetes.
  + Конфигурация базы данных PostgreSQL и кеширования Redis.

**Развертывание приложения:**

* + Запуск контейнеров с использованием Helm:  
    helm install personal-recommendation ./charts
  + Подключение к облачному хранилищу для обработки данных.

**Настройка API и взаимодействия с внешними сервисами:**

* + Обеспечение связи с каталогами контента онлайн-кинотеатра Premier.
  + Настройка взаимодействия через API с внешними системами.

**Начало работы**

Запуск системы

Подготовка окружения

Обновите пакеты и установите зависимости:

sudo apt update && sudo apt upgrade -y

sudo apt install -y curl wget git unzip

**Установите Docker и Docker Compose:**sudo apt install -y docker.io

sudo systemctl enable --now docker

sudo usermod -aG docker $USER  
Перезагрузите систему или выполните newgrp docker для активации прав.

**Установите Kubernetes (kubectl) и Helm:**curl -fsSL https://get.k3s.io | sh -

curl -fsSL -o /usr/local/bin/helm https://get.helm.sh/helm-v3.9.0-linux-amd64.tar.gz

chmod +x /usr/local/bin/helm

**Убедитесь, что все компоненты установлены:**docker --version

kubectl version --client

helm version

Развертывание инфраструктуры в Yandex Cloud

**Авторизуйтесь в Yandex Cloud:**yc init  
Следуйте инструкциям на экране для настройки аккаунта.

**Создайте кластер Kubernetes в Yandex Cloud:**yc managed-kubernetes cluster create --name my-cluster --zone ru-central1-a --network-id default --node-group memory-optimized

**Настройте контекст Kubernetes:**yc managed-kubernetes cluster get-credentials my-cluster --external

Развертывание базы данных

**Создайте PostgreSQL-базу данных в Yandex Cloud:**yc managed-postgresql cluster create --name my-db --network-id default --host zone=ru-central1-a

**Настройте подключение к базе:**export DATABASE\_URL=postgresql://user:password@host:port/dbname

Развертывание приложения

**Клонируйте репозиторий приложения:**git clone https://github.com/your-repo/personalization-service.git

cd personalization-service

**Настройте переменные окружения:**cp .env.example .env

nano .env  
Укажите API-ключи, параметры базы данных и другие конфигурации.

**Запустите приложение с помощью Helm:**helm install personal-recommendation ./charts

**Проверьте статус развертывания:**kubectl get pods -n recommendation-system

**Настройте балансировщик нагрузки:**kubectl expose deployment personal-recommendation --type=LoadBalancer --port=80

**Проверьте доступность сервиса:**curl -X GET "http://your-service-ip/recommendations?user\_id=1234"

**После успешного развертывания проверьте, что API сервиса работает корректно, а также выполните тестирование всех функциональных возможностей. Система готова к эксплуатации**

**Руководство пользователя**

Вход в систему и работа с интерфейсом

Взаимодействие с системой происходит через API. Пользовательский интерфейс предоставляет доступ к персонализированным рекомендациям и настройке предпочтений.

Получение персонализированных рекомендаций

Для получения рекомендаций пользователь должен войти в систему с использованием PassMedia ID.

Рекомендации доступны через API-запрос:

curl -X GET "https://api.premier-recsys.com/recommendations?user\_id=1234"

В ответ система вернет список подходящего контента.

Управление профилем

Пользователь может изменить свои предпочтения для улучшения рекомендаций:

curl -X POST "https://api.premier-recsys.com/profile" -d '{"user\_id": "1234", "preferences": {"genre": "comedy"}}'

Возможные настройки профиля включают жанры, актеров, языки контента.

Работа с типами рекомендаций

**Персонализированные рекомендации** – генерируются на основе истории просмотров пользователя.

**Трендовые рекомендации** – основываются на общем поведении пользователей.

**Жанровые подборки** – учитывают предпочтения пользователя по жанрам.

Фильтрация контента

Система поддерживает фильтры, позволяющие исключать нежелательный контент по возрастным рейтингам, регионам или жанрам.

Для применения фильтрации необходимо передать параметры в API-запросе.

Функциональные возможности

Генерация персонализированных рекомендаций

Система формирует индивидуальные подборки контента на основе истории просмотров и предпочтений пользователей.

Поддержка различных типов рекомендаций

Персонализированные рекомендации – на основе истории пользователя.

Общие трендовые рекомендации – на основе популярности контента среди всех пользователей.

Модели машинного обучения

Коллаборативная фильтрация – анализ предпочтений схожих пользователей.

Контентная фильтрация – анализ характеристик контента для подбора релевантных рекомендаций.

Обслуживание

Резервное копирование

Автоматическое резервное копирование базы данных раз в сутки. Возможность восстановления данных в течение 1 часа.

Мониторинг системы

**Метрики производительности:**

Среднее время ответа API – не более 200 мс. Поддержка нагрузки до 1500 rps.

**Инструменты мониторинга:**

Prometheus – сбор метрик производительности.

Grafana – визуализация данных.

Loki – логирование и анализ ошибок.

**Безопасность**

Аутентификация и авторизация пользователей через PassMedia ID. Защита API от SQL-инъекций и XSS-атак. Шифрование данных при передаче и хранении.

Обновление системы

Обновления ПО разворачиваются через CI/CD. Обновления моделей машинного обучения выполняются по заранее определенному графику.

**Завершение работы**

**Остановка обработки запросов API:**kubectl delete -f deployment.yaml

**Остановка контейнеров:**docker-compose down

**Выключение облачной инфраструктуры (при необходимости).**

**Заключение**

Программное обеспечение "Персонализация контента" обеспечивает обработку запросов, расчет персонализированных рекомендаций и их передачу через API. Использование облачной инфраструктуры и машинного обучения позволяет достигать высокой точности рекомендаций, масштабируемости и надежности работы системы.

# **Модуль «Предсказание оттока»**

Модуль «Предсказание оттока» предназначен для прогнозирования вероятности оттока подписчиков онлайн-кинотеатра Premier. Основная цель системы – заблаговременное выявление пользователей с высоким риском оттока для последующей работы с ними через маркетинговые инструменты и повышения бизнес-метрик Заказчика.

Функционал модуля

Предсказание вероятности оттока на 30 дней (+2 дня на возврат) для всех типов подписок;

Поддержку отдельных моделей для триалистов и пользователей с платной подпиской;

Качество классификации пользователей с ROC-AUC не ниже 0.7;

Автоматическое обновление витрины данных с результатами скоринга ежедневно;

Интерпретируемость предсказаний с возможностью определения ключевых факторов, влияющих на вероятность оттока.

Начало работы и запуск модуля

Окружение

Модуль «Предсказание оттока» работает в среде Яндекс.Облака и использует следующие сервисы:

Yandex Managed Service for PostgreSQL – для хранения результатов предсказаний;

Проект Data Sphere (Yandex Cloud) – для обучения и предсказаний модели;

S3 bucket (Yandex Cloud) – для хранения промежуточных данных.

**Запуск модуля**

Для работы с модулем администратор должен:

* Войти в Data Sphere (Yandex Cloud)
* Запустить соответствующий Jupyter Notebook

( Воркбенч обновления оттока, Воркбенч с MVP uplift для оттока)

* Проверить соединение с базами данных PostgreSQL и хранилищем S3.
* Запустить пайплайн предсказания.

Описание алгоритмов и математического обеспечения

Пайплайн предсказания оттока выполняется в три фазы:

Предварительный расчет фичей на Apache Spark, включающий анализ:

Истории просмотров, данных биллинга, показов и кликов (Яндекс.Метрика + Trex) и активности пользователя, сумм его трат и действий на сайте.

Обучение и предсказание с помощью временной кросс-валидации моделей градиентного бустинга с бинарным классификатором (отток/не-отток), используя данные за последние 12 месяцев.

Обновление и интеграция данных, выгрузка предсказаний в базу PostgreSQL и подготовка отчетов по качеству модели.

**Потоки данных**

Данные поступают от различных сервисов, включая:

Pass.Media – идентификация пользователей;

Бэкенд-сервисы Premier – информация о подписках и биллинге;

Trex (плеер) – сбор логов событий;

AppMetrica и Яндекс.Метрика – пользовательские события в мобильных и веб-приложениях.

**Модель данных**

Основные параметры витрины данных о просмотрах:

| **Поле** | **Описание** |
| --- | --- |
| passport\_id | Идентификатор пользователя |
| tv\_id | Идентификатор контента |
| season\_num | Номер сезона |
| episode\_num | Номер эпизода |
| user\_ep\_total\_view\_duaration\_s | Время просмотра эпизода (сек) |
| pdp\_created\_at | Время создания записи |
| actual\_title\_ru | Название контента |
| season\_duration\_s | Длительность сезона (сек) |

**Интерфейсы взаимодействия**

Модуль передает предсказания через внутренние системы компании в платформу маркетинговых коммуникаций Altcraft, откуда осуществляется взаимодействие с пользователями (email, push, SMS и т.д.).

**Ролевая модель**

Система предусматривает три категории пользователей:

Администраторы – сотрудники Предикто, управляющие системой;

Сотрудники Premier – маркетологи, использующие предсказания для кампаний;

Пользователи Premier – получатели персонализированных маркетинговых предложений.

**Администраторы.** Контролируют работу модуля и корректность данных. Запускают пайплайны предсказаний и обновления витрины данных. Настраивают параметры модели и следят за качеством предсказаний. Мониторят работу базы данных и серверов.

**Сотрудники**. Используют дашборды для анализа предсказаний оттока. Планируют маркетинговые кампании на основе полученных данных. Настраивают сегменты пользователей для удержания через Altcraft.

**

*Рис. 1 пример одного из операционных дашбордов для монитроинга состояния базы данных*

**Пользователи**. Получают персонализированные предложения на основе предсказаний оттока. Взаимодействуют с маркетинговыми коммуникациями (push, email, SMS). Реагируют на предложения, что может повлиять на их статус в системе.

# **Модуль «Сквозная аналитика»**

Модуль «Сквозная аналитика» предназначен для интеграции и анализа данных из различных источников, включая веб-аналитику, биллинговые системы и плееры. Он позволяет анализировать поведение пользователей, оптимизировать контент и проводить финансовый анализ.

Основные задачи модуля

Предоставить сотрудникам платформы доступ к детализированной информации, необходимой для маркетинговой, продуктовой и контентной аналитики, а также обеспечить возможность пользовательской аналитики в различных сегментах и когортах пользователей:

* Анализ поведения пользователя: Отслеживание действий пользователя на

платформе для понимания предпочтений и поведенческих паттернов, а также построение карты путей клиента (CJM, Customer Journey Map).

* Продуктовая и маркетинговая отчетность: Интерактивные дашборды,

позволяющие пользователям быстро и удобно отслеживать ключевые метрики и показатели, такие как продажи, конверсии, расходы на маркетинг и т.д.

* Сегментация пользовательской базы: Интеграция с модулем «Сегментатор

данных» для выделения пользовательских сегментов на основе и предоставления аналитики на основе создаваемых сегментов.

* Оптимизация контента: Анализ популярности различных видов контента для оптимизации закупок и создания собственного контента.
* Управление подписками: Мониторинг активности подписок для улучшения стратегий удержания и привлечения клиентов.

Пользователи модуля

| **Категория пользователей** | **Основные задачи** | **Цель использования модуля** |
| --- | --- | --- |
| 1. Продуктовые аналитики | - Анализ поведения пользователей  - Анализ пользовательских сегментов  - Формирование гипотез | Повышение вовлечённости и удержания пользователей |
| 2. Продуктовые менеджеры | - Мониторинг ключевых метрик  - Принятие решений на основе данных  - Постановка задач командам | Оптимизация продукта и достижение бизнес-целей |
| 3. Маркетологи | - Анализ аудитории и её поведения  - Оценка эффективности рекламных кампаний  - Выявление трендов | Привлечение и удержание целевой аудитории |
| 4. Руководители | - Отслеживание стратегических показателей (DAU, MAU, время просмотра)  - Анализ общей динамики использования платформы | Поддержка управленческих решений и стратегического планирования |
| 5. Отдел контента | - Анализ потребления контента  - Оценка предпочтений по сегментам  - Мониторинг популярности категорий | Оптимизация контентной политики и стратегия закупок |

Процесс установки модуля

Подготовка окружения

Перед установкой убедитесь в наличии Kubernetes-кластера (Yandex Managed Service).

Проверьте доступ к Yandex Object Storage, Redis, ClickHouse, PostgreSQL.

Подключите внешние источники данных: AppMetrica, Яндекс.Метрика, логи плеера и др.

Установка

Выполните деплой через скрипт установки или CI/CD пайплайн.

Проверьте доступность сервиса по домену.

Настройка

Укажите параметры доступа к внешним API.

Настройте роли и права в интерфейсе.

Задайте конфигурации для подключения к "Платформе данных"

В файле values.yaml укажите параметры подключения:

dataPlatform:

url: https://data-platform.internal

apiKey: <ваш-ключ-доступа>

segmentator:

url: https://segmentator.internal

token: <токен-доступа>

Убедитесь, что сервисы доступны по указанным адресам внутри кластера.

Проверьте сетевые политики и DNS-резолвинг между сервисами.

Выполните перезапуск подов для применения изменений:  
kubectl rollout restart deployment skvoz-analytics -n analytics

Основные аналитические дашборды

Примеры продуктовых дашбордов.

Описание метрики "Активность пользователей"

Этот показатель является ключевым для оценки эффективности продукта и помогает в принятии стратегических решений.



*рис.1 метрика "Активность пользователей"*

Параметры графика:

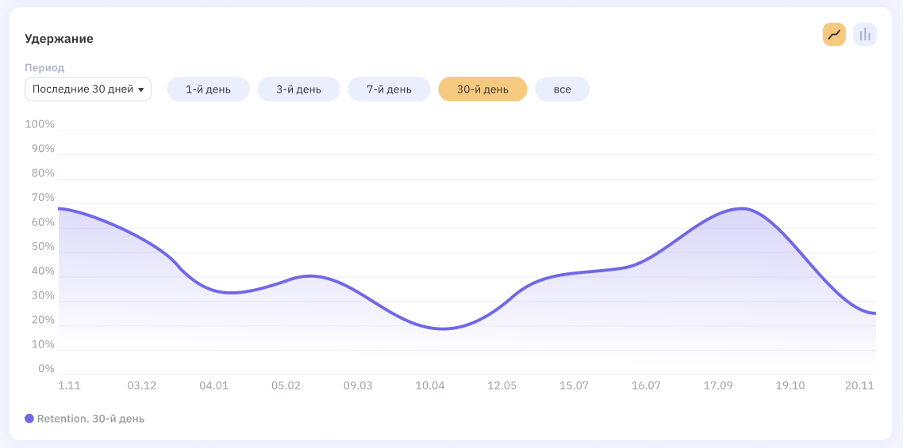
* Периодичность анализа: График позволяет выбрать период анализа данных (например, год), что дает возможность оценивать долгосрочные тренды в активности пользователей.
* Метрики активности: Представлены различные измерения, такие как MAU (Monthly Active Users), DAU (Daily Active Users), и время, проведённое пользователями (timespent).

Анализ MAU (Monthly Active Users):

* MAU: Показатель, отражающий количество уникальных пользователей, активно пользующихся платформой в течение месяца.
* Визуализация: Линейный график показывает изменения в количестве активных пользователей за год, что помогает выявить тренды, сезонные колебания или периоды роста и снижения.

Описание метрики "Удержание"

Эта метрика ключевая для оценки успешности взаимодействия пользователей с платформой и позволяет принимать решения по улучшению продуктовых стратегий.



*рис.2 метрика "Удержание"*

Параметры графика:

Периодичность анализа: Пользователи могут выбрать период (например, последние 30 дней), что позволяет отслеживать краткосрочные тенденции в удержании.

Выбор дней удержания: Возможность анализа удержания на 1-й, 3-й, 7-й и 30-й дни, позволяя оценивать, насколько пользователи возвращаются через разные временные промежутки.

Анализ 30-дневного удержания:

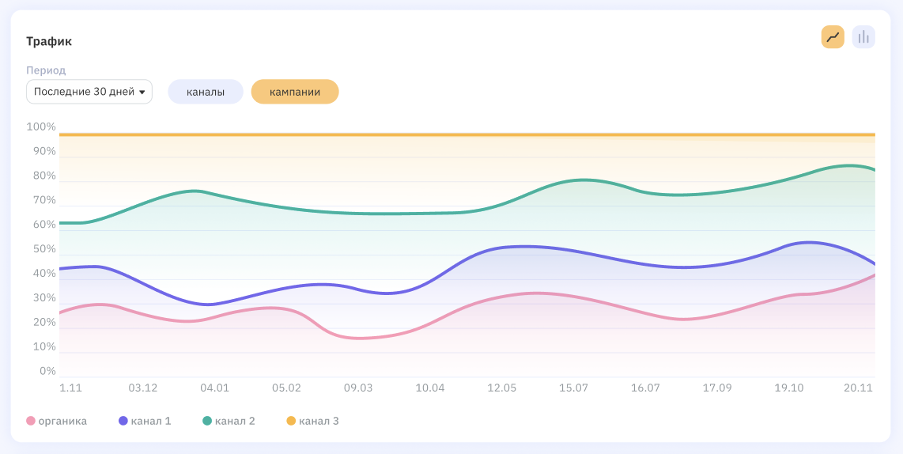
Retention Rate: Показатель отображает процент пользователей, которые продолжают пользоваться платформой через 30 дней после первого визита.

Визуализация: Линейный график демонстрирует изменения в уровне удержания пользователей, что помогает выявить периоды, когда пользователи наиболее активно возвращаются.

Примеры маркетинговых дашбордов

Описание метрики "Трафик"

Эта метрика критически важна для понимания источников трафика и помогает в оптимизации стратегий привлечения пользователей на платформу.



*рис.3 метрика "Трафик"*

Параметры графика:

* Периодичность анализа: Возможность выбора периода (например, последние 30 дней), что позволяет отслеживать изменения в источниках трафика.
* Категории анализа: Доступны два анализа — по каналам и кампаниям, что дает возможность детализировать источник трафика.

Анализ источников трафика:

* Органический трафик и каналы: График показывает распределение трафика между органическим источником и различными каналами (канал 1, канал 2, канал 3).
* Визуализация: Стековая диаграмма иллюстрирует колебания в процентах для каждого канала, что помогает оценивать эффективность и изменение доли каждого источника трафика.

Описание метрики "Эффективность маркетинговых инвестиций (ROMI)"

Этот показатель является важным инструментом для принятия обоснованных решений о стратегиях маркетинга и распределении бюджетов на рекламу.



*рис.4 метрика "Эффективность маркетинговых инвестиций (ROMI)"*

Параметры графика:

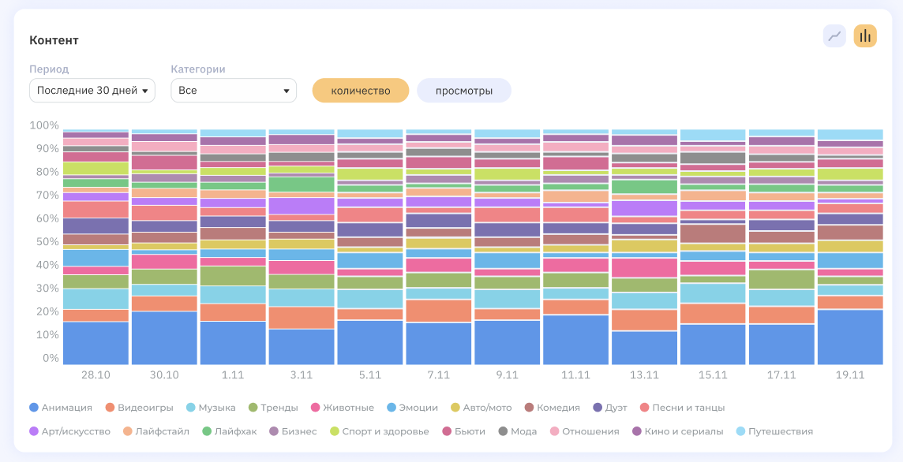
* Периодичность анализа: Возможность выбора периода (например, последние 30 дней), что позволяет оценить краткосрочную эффективность инвестиций.
* Визуализация по объему и эффективности: График представляет данные в виде пузырьковых диаграмм, где каждая точка (пузырек) рассчитана по двум ключевым параметрам: объем и эффективность.

Анализ кампаний:

* Эффективность: По вертикали отложена эффективность кампании с учетом возврата на инвестиции.
* Объем инвестиций: По горизонтали отложен объем вложенных средств.
* Пузырьки: Размер пузырька может указывать на общий размер кампании или другой важный фактор.

Примеры контентных дашбордов

Этот показатель ключевой для управления и улучшения контентных предложений, чтобы максимально соответствовать интересам аудитории.



*рис.5 Пример дашборда за 30 дней по различным категориям*

Параметры графика:

* Периодичность анализа: Возможность выбора периода (например, последние 30 дней), что позволяет отслеживать долгосрочные тенденции в потреблении контента.
* Категории: Разделение данных по различным категориям контента, таким как анимация, видеоигры, музыка и другие.

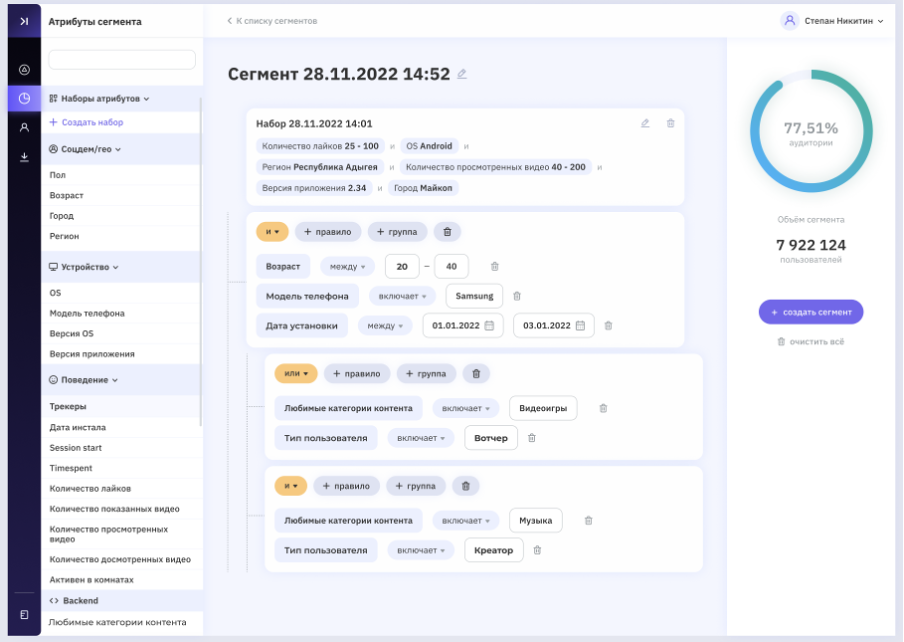
Анализ контента:

* Количество и просмотры: Возможность переключаться между анализом по количеству контента и числу его просмотров, что помогает оценивать как объем, так и вовлеченность.
* Цветовая кодировка: Каждая категория имеет свой цвет, что облегчает визуальную идентификацию и анализ тенденций в популярности различных типов контента.

Инструменты для работы с сегментами

Создание и управление сегментами

Модуль «Сквозная аналитика» предоставляет интерфейс для создания и управления сегментами пользователей. Вычисление и обновление сегментов происходит через интеграцию с модулем «Сегментатор данных».



*рис.6 интерфейс для создания и управления сегментами пользователей*

Предоставляются следующие функциональные возможности по работе с сегментами:

Атрибуты и правила сегментации:

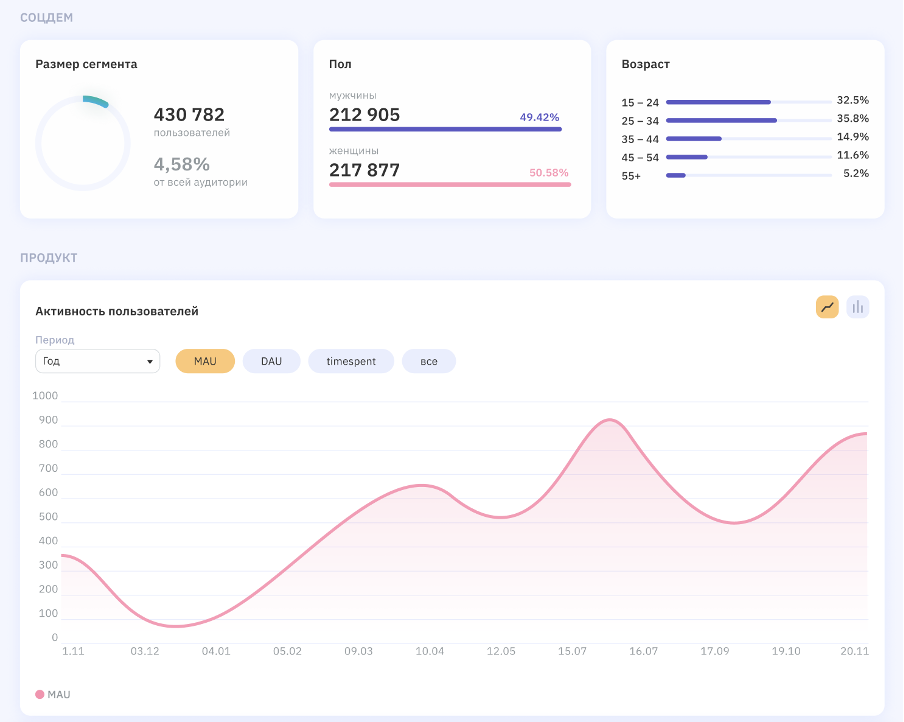
* Определение сегментов пользователей по различным критериям, включая возраст, количество лайков, модель телефона, и другие параметры.
* Использование логических операторов «и» и «или» для создания сложных условий сегментации, которые позволяют уточнять аудиторию на основе комбинации атрибутов.

Фильтрация по демографическим и поведенческим факторам:

* Возможность фильтрации по региону, версии ОС, городам, и предпочтениям в контенте (например, видеоигры или музыка).
* Учет пользователей с различными уровнями активности и типов, такими как «Вочтер» или «Креатор».

Аналитика на уровне сегмента

Для созданного сегмента доступны определенные аналитические возможности – включая базовую аналитику по сегменту и визуализацию дополнительных метрик.



*рис.7 визуализация аналитики на уровне сегмента*

Основные характеристики сегмента:

* Размер сегмента: Показывает количество пользователей в сегменте и процент от всей аудитории.
* Пол: Отображение распределения пользователей по полу с процентным соотношением.
* Возраст: Графическое представление возрастных групп и их процент в общем сегменте, что помогает понять демографический состав пользователей.

Дополнительные метрики (пример):

График отображает динамику изменения активности пользователей на протяжении года, что помогает выявить пики и спад активности. Предоставляются следущие возможности:

* Периодичность: Выбор периода анализа для отслеживания активности.
* Метрики активности:
* MAU (Monthly Active Users): Анализирует месячное количество активных пользователей.
* DAU (Daily Active Users): Возможность переключения для анализа ежедневной активности.
* Временной показатель (timespent): Анализ времени, проведённого пользователями в медиаплатформе.

Профиль пользователя медиаплатформы

Возможности анализа пользователей

Модуль «Сквозная аналитика» позволяет получить глубокий анализ поведения пользователей и их взаимодействия с платформой, с возможностью получения сведений на уровне конкретного пользователя медиаплатформы.

Основные метрики пользователей

***Активность***: Количество опубликованных, показанных, просмотренных и досмотренных видео.

***Подписки***: Информация о количестве подписок.

***История посещений***: Даты первого и последнего визитов.

Общая информация:

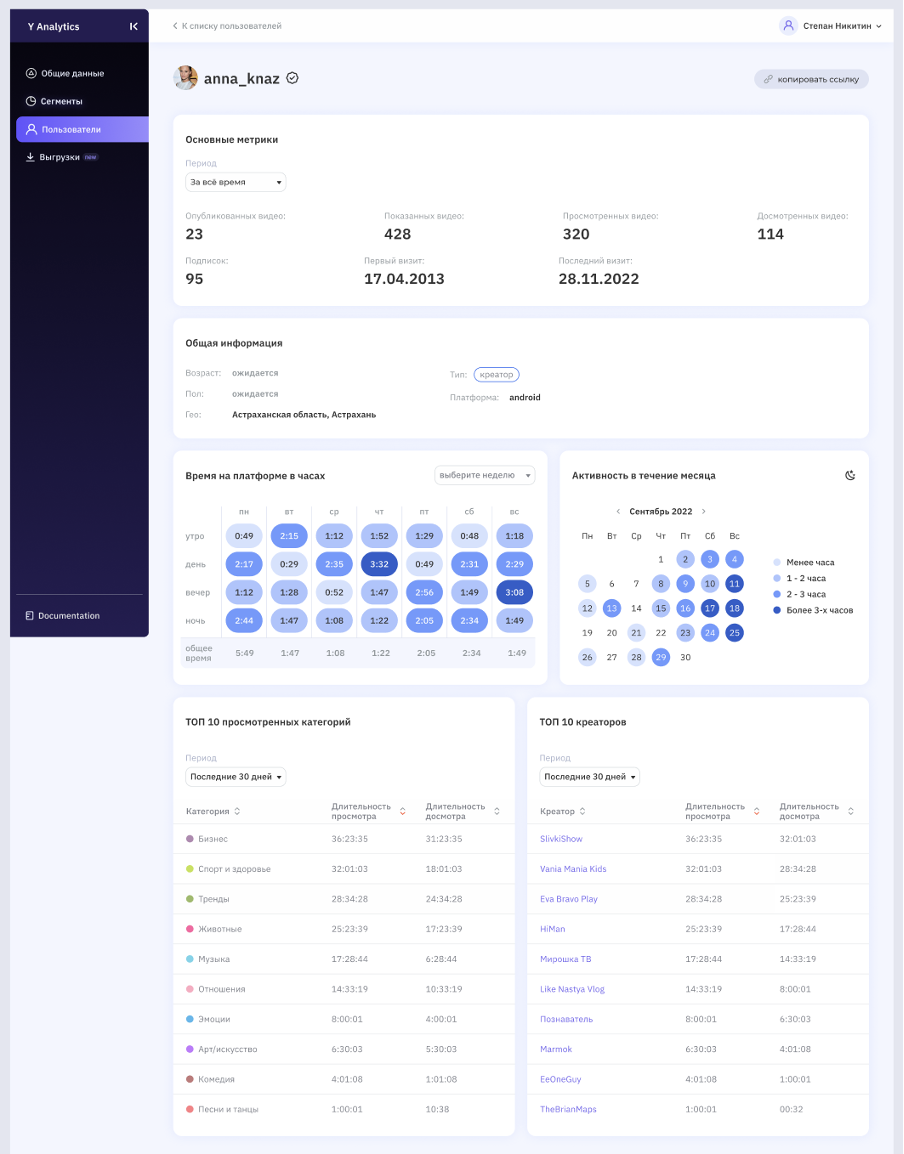
* Тип пользователя: Информация, например, как "Креатор".
* Географические данные: Регион и платформа пользователя.

Время на платформе:

* Еженедельный анализ: Детализированные данные по времени, проведённому на платформе, разбитые по дням и времени суток.
* Месячный календарь активности: Визуализация активности по дням в течение месяца.

Потребление контента:

* ТОП 10 просмотренных категорий: Перечень популярных категорий, включая продолжительность просмотра и досмотра.
* ТОП 10 креаторов: Популярные креаторы, чьё видео чаще всего просматривают пользователи, с аналогичной продолжительностью просмотра.

*рис.8 Пример отчета по пользователю*

## **Модуль «Инфраструктура для сервиса»**

Данный документ предназначен для пользователей, ответственных за эксплуатацию программного обеспечения (ПО), которое предоставляет сервисы для хранения исторических данных, интеграции с источниками данных и их обработки. Основное внимание уделено процессу начала работы в системе, запуску модулей, а также использованию функционала для обработки данных.

Описание программного обеспечения

ПО предназначено для обработки и хранения исторических данных, а также интеграции с аналитическими системами и источниками данных. Среди ключевых возможностей — автоматическая и периодическая загрузка данных, хранение и обработка больших объемов информации, создание аналитических витрин данных, а также интеграция с инструментами для бизнес-анализа и отчетности.

Основные функциональные требования к ПО:

* Автоматическая загрузка данных из аналитических систем (Яндекс.Метрика, АппМетрика).
* Периодическая загрузка данных о пользовательском поведении.
* Долгосрочное хранение данных в объектном хранилище на базе Яндекс.Облака.
* Обработка данных с использованием SQL и Apache Spark.
* Интеграция с BI-инструментами для визуализации данных.
* Обработка видео и использование мультимодальных нейросетей.

Начало работы с программным обеспечением

Для эффективной работы с программным обеспечением необходимо выполнить настройку инфраструктуры, интеграцию с источниками данных, а также запустить соответствующие модули для обработки и хранения данных. В случае возникновения вопросов или проблем, обратитесь к документации по конкретным модулям или к технической поддержке.

2.1 Установка и настройка

Проверьте соответствие системных требований (среда выполнения Java, версии Python, доступ к облачным ресурсам).

Настройте доступ к облачной инфраструктуре (например, Яндекс.Облако), указав ключи API для подключения к сервисам хранения и обработки данных.

Настройте коннекторы для источников данных (например, Яндекс.Метрика) с использованием соответствующих API. Для настройки автоматической загрузки данных следуйте руководству по интеграции с каждой системой.

Запуск модуля

Запуск основной службы:

Для запуска программного обеспечения, откройте командную строку и выполните команду для старта основного сервиса:

shell  
$ ./start\_service.sh

Убедитесь, что все службы запустились без ошибок.

Запуск модулей загрузки данных:

Для запуска модуля автоматической загрузки данных выполните следующую команду:

shell  
$ ./start\_data\_loader.sh

Этот модуль будет автоматически загружать данные из аналитических систем (Яндекс.Метрика, АппМетрика) и внутренних сервисов Яппи.

Запуск модуля обработки данных:

Для обработки больших объемов данных используйте команду для старта инфраструктуры Apache Spark:  
shell  
$ ./start\_spark\_processing.sh

Этот процесс позволит использовать распределённую обработку данных с использованием SQL.

Запуск видеопроцессинга:

Для параллельной обработки видео и использования мультимодальных нейросетей используйте команду:  
shell  
$ ./start\_video\_processing.sh

Рабочие процессы

Загрузка данных

**Автоматическая загрузка:** Каждый день осуществляется автоматическая загрузка данных о поведении пользователей и активности из интегрированных систем (например, Яндекс.Метрика).

**Периодическая загрузка:** Раз в неделю выполняется загрузка данных о просмотрах, лайках, подписках и других событиях внутри приложения Yappy.

**Модели загрузки данных:** Все данные, загруженные в систему, проходят через предварительную обработку и нормализацию.

Обработка данных

**Обработка через Apache Spark:**

Использование SQL-запросов для обработки больших данных. Все запросы к данным обрабатываются с использованием распределённой системы для повышения производительности.

**Мультимодальные нейросети:**

Для обработки видео данных используется интеграция с нейросетями, такими как XClip и ViCLIP.

Хранение данных

Все данные, включая видео и пользовательские данные, сохраняются в объектном хранилище на базе Яндекс.Облака. Реализовано георезервирование, что обеспечивает надежность и доступность данных в разных регионах.

Визуализация и бизнес-аналитика

Для визуализации данных используется Tableau, что позволяет создавать интерактивные отчеты и графики, помогая в принятии бизнес-решений.

Обслуживание системы

**Мониторинг работы системы.** Используйте встроенные инструменты для мониторинга состояния серверов и загрузки данных.

Для отслеживания ошибок и производительности системы доступны логи, которые можно просматривать через административную панель.

**Обновления и поддержка.** Программное обеспечение регулярно обновляется. Следите за выпусками и установите последние версии для улучшения функционала и безопасности.

**Резервное копирование данных.** Все данные регулярно резервируются в облаке для обеспечения их сохранности.

## **Модуль «BI и adhoc»**

Модуль «BI и adhoc» представляет собой систему бизнес-аналитики, предназначенную для сбора, обработки и визуализации данных, связанных с бизнес-показателями онлайн-кинотеатра «Премьер». Данный документ содержит подробное описание эксплуатации экземпляра программного обеспечения, включая установку, настройку, запуск и обновление системы.

Архитектура системы

Основные компоненты BI и adhoc модуля:

* BI-сервер (виртуальная машина в Yandex Cloud) – обеспечивает доступ к пользовательскому интерфейсу и визуализации;
* ClickHouse – хранилище агрегированных данных;
* Платформа данных – источник формирования витрин и агрегатов;
* Встроенная СУБД (PostgreSQL) – для хранения метаданных и конфигураций;
* OpenJDK 11 – среда выполнения сервера.

Потоки данных:

* Данные формируются в Платформе данных;
* Сохраняются в ClickHouse в витрины;
* BI-сервер отображает дашборды, подключённые к ClickHouse в реальном времени или по расписанию.

Процесс установки модуля

Системные требования:

ОС: Ubuntu 18.04.6 LTS

CPU: 16 vCPU

RAM: 64 GB

Disk: 500 GB SSD

**Процесс установки:**

1. Загрузка пакета:

wget https://downloads.example.com/bi-server-2022-1-4\_amd64.deb

1. Установка:

sudo dpkg -i bi-server-2022-1-4\_amd64.deb

1. Инициализация менеджера управления:

sudo /opt/bi/bi\_server/packages/scripts/initialize-adminctl

1. Активация лицензии:

adminctl licenses activate -k <ключ\_лицензии>

1. Запуск сервиса:

adminctl start

**Подключение к ClickHouse:**

Установить ClickHouse JDBC Driver v0.3.22

Настроить подключения в конфигурационном файле BI-сервера

**PostgreSQL:**

sudo systemctl start postgresql

psql -U admin -d metastore -c "SELECT \* FROM system\_users;"

Доступ и интерфейс

Пользователь переходит по ссылке на сервер,развернутый в Yandex Cloud. Через веб-интерфейс с логином/паролем. После успешной идентификации система проверяет, к каким директориям, дашбордам, инструментам управления имеет доступ данный пользователь. Пользователю отображаются: «Главный» экран интерфейса; соответствующие дашборды: «Бизнес», «Контентный», «Продуктовый» и т.д., в зависимости от уровня доступа (например, аналитик, руководитель, редактор и т.д.).

Вход в систему через браузер по адресу: https://analytics.predicto.ru/#/

Управление пользователями и ролями производится через административный веб-интерфейс

Интерфейсы и их функциональность

Бизнес дашборд

Функциональность: Этот дашборд позволяет отслеживать ключевые метрики бизнеса в реальном времени и анализировать финансовые показатели по подпискам и пользователям.

Разделы:

Фактические показатели   
Стоимость притока/оттока

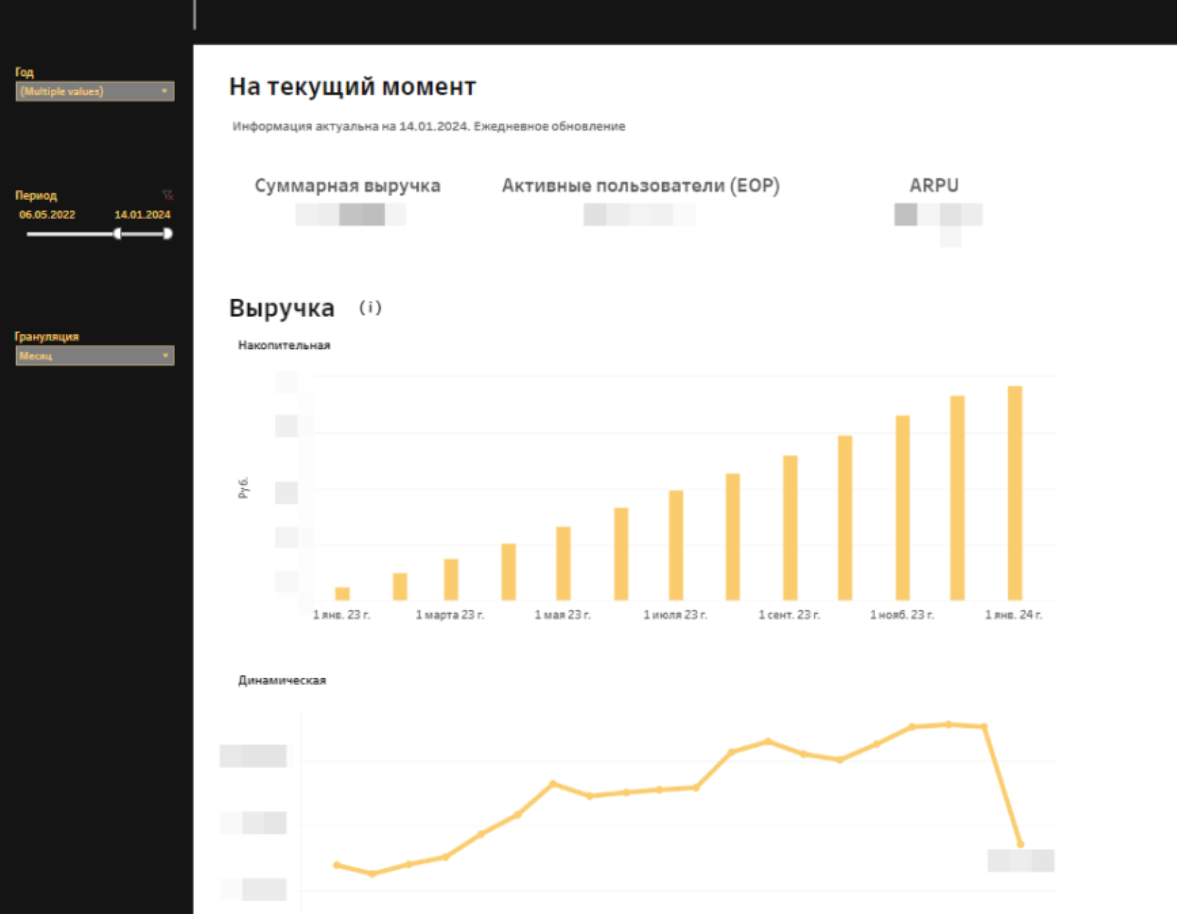
Активная база пользователей

Приток / Отток пользователей

Доля оттока

Время жизни в продукте (LT)

ARPU (Средний доход на пользователя)



*Рис.1 Бизнес дашборд. Фактические показатели*

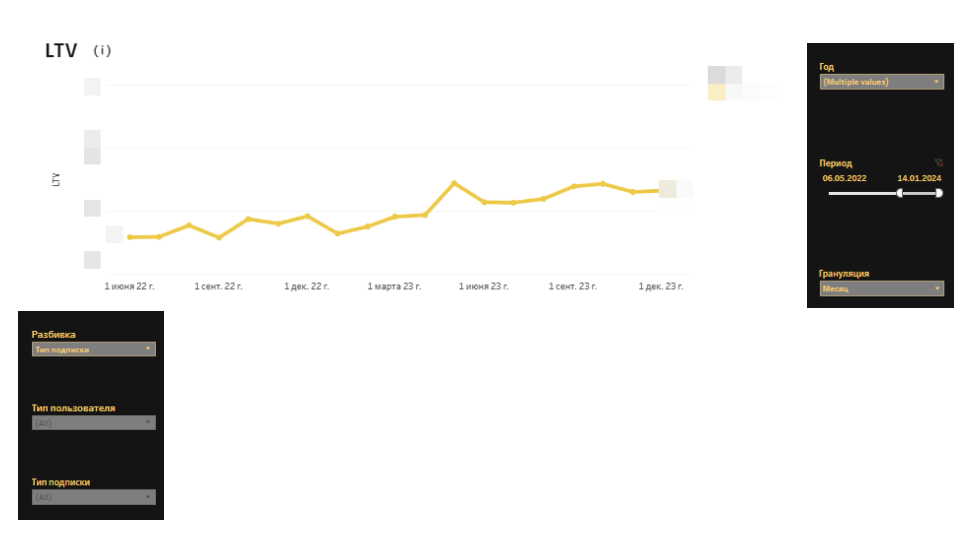
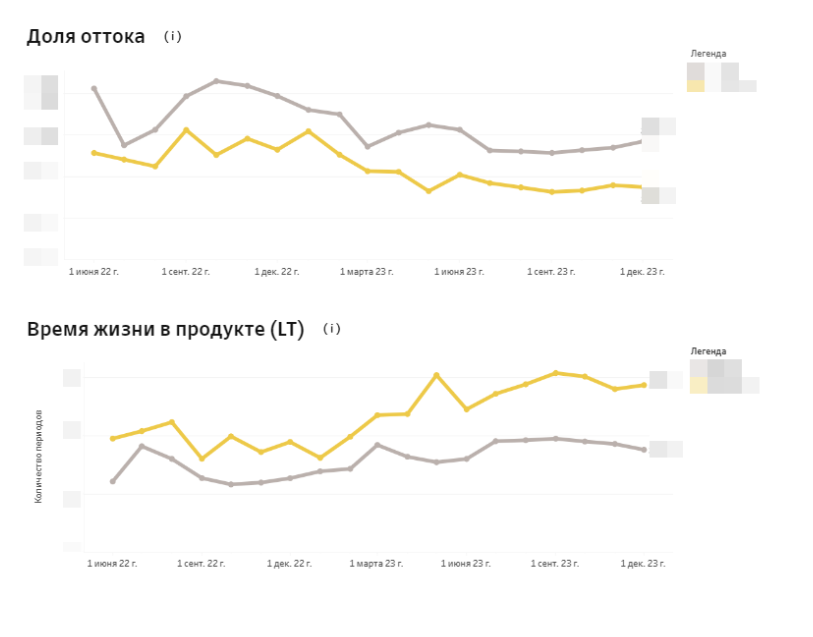
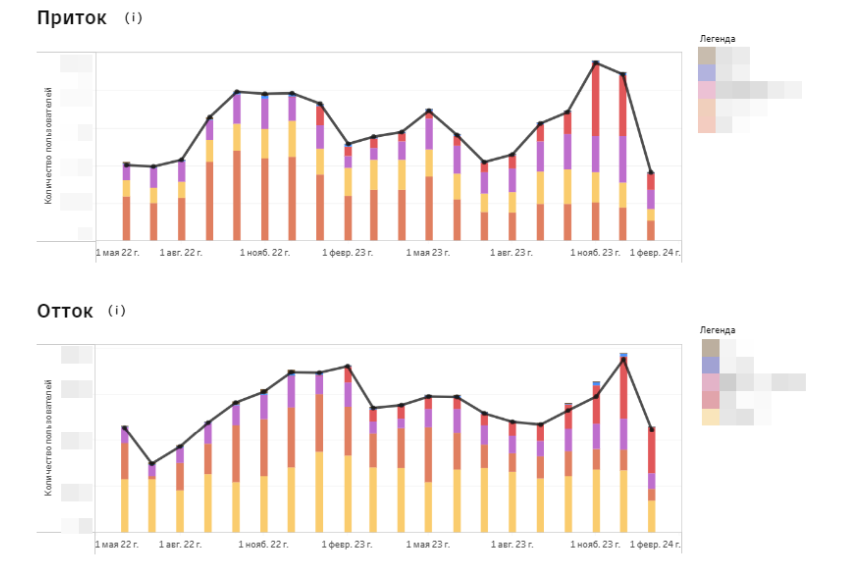


Рис.2 Бизнес дашборд

План-факт

Функциональность: Сравнение запланированных и фактических показателей финансовых результатов. Оценка отклонений от запланированных значений и выявление факторов, влияющих на изменения. Контроль эффективности финансовых стратегий.

Разделы:

Расширенный план по выручке, руб. без НДС

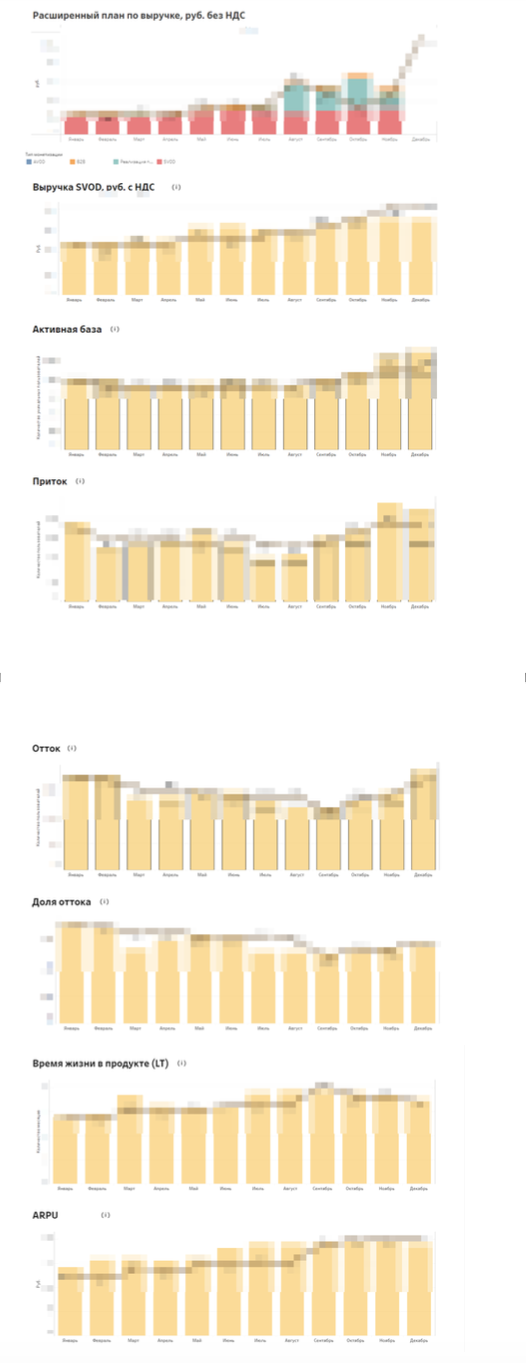
Выручка SVOD, руб. с НДС

Активная база

Приток (+) / Отток (-)

Доля оттока

Время жизни в продукте (LT)



*Рис.3 План-факт дашборд экран интерфейса*

Контентный дашборд

Функциональность: Анализ вовлеченности пользователей в контент и оценка его популярности. Определение наиболее популярных тайтлов. Анализ времени просмотра и удержания аудитории.

Разделы:

ТОП тайтлов по количеству зрителей

Выручка по смотрению

Динамика потребления контента

Доля тайтла относительно других



*Рис.4 Контентный дашборд экран интерфейса*

**Дашборд по притоку**

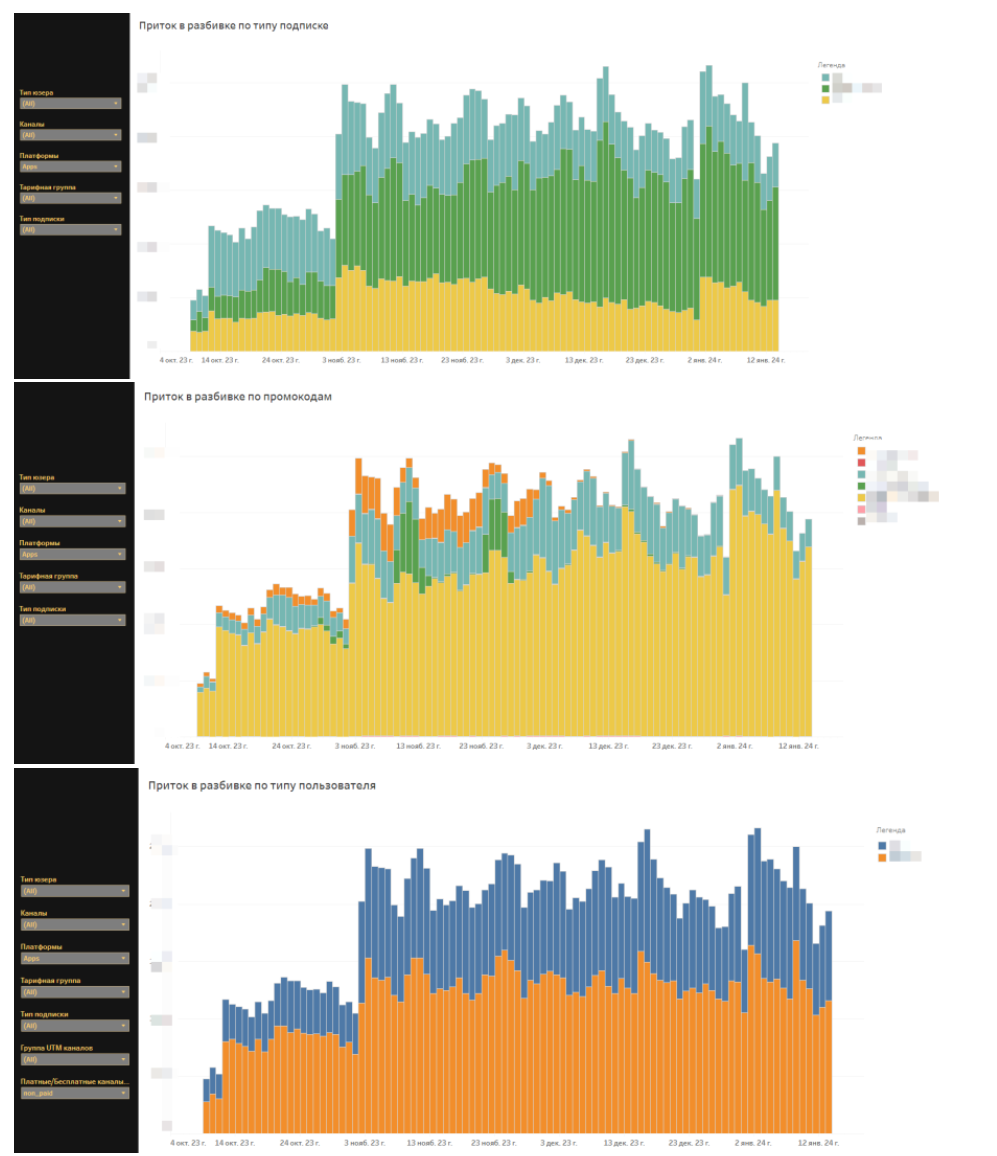
Функциональность: Мониторинг каналов привлечения пользователей и их конверсии. Оценка эффективности рекламных кампаний. Анализ того, какие каналы приводят пользователей с наибольшей вероятностью оформления подписки.

Разделы:

Приток по каналам привлечения

Приток в разбивке по подпискам

Приток в разбивке по промокодам



*Рис.5 Дашборд по притоку экран интерфейса*

Продуктовый дашборд

Функциональность: Анализ продуктовых метрик и динамики подписок. Анализ динамики роста подписчиков и их взаимодействия с контентом. Оценка уровня удержания и средней стоимости привлеченного пользователя.

Разделы:

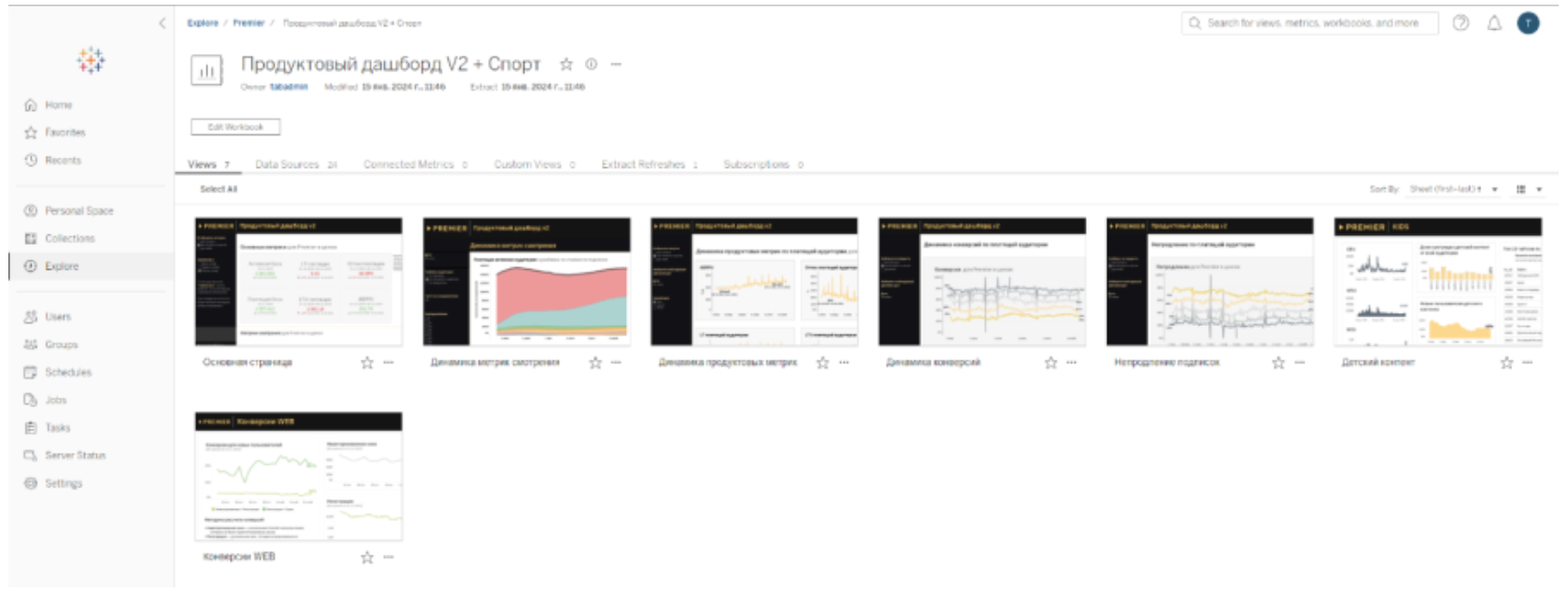
Активная база

LT платящих пользователей

Отток платящих пользователей

LTV платящей аудитории

Среднее время потребления контента



*Рис.6 Продуктовый дашборд экран интерфейса*

Возможности и действия пользователя в системе

Пользователь модуля BI и adhoc может выполнять следующие действия:

* Получать доступ к дашбордам, соответствующим его роли и правам доступа;
* Применять фильтры (по датам, сегментам, платформам, типам подписки и другим параметрам);
* Изучать ключевые бизнес-метрики: выручка, отток, приток, ARPU, LTV, LT, вовлечённость и др.;
* Строить гипотезы на основе визуализации данных;
* Оценивать эффективность каналов привлечения и действия промокодов;
* Сравнивать плановые и фактические значения по подписке и выручке;
* Осуществлять анализ по контенту: какие тайтлы популярны, каков средний просмотр и удержание;
* Сохранять пользовательские представления (если включена соответствующая настройка);

Резервное копирование и восстановление

Создание бэкапа:

adminctl maintenance backup -f backup.bak

Восстановление из бэкапа:

adminctl maintenance restore -f backup.bak

## **Модуль «Telegram бот»**

Модуль «Telegram бот» предназначен для предоставления актуальной и структурированной информации о состоянии бизнеса онлайн-кинотеатра «Премьер» через интерфейс мессенджера Telegram. Данный документ содержит руководство по установке, настройке, запуску и эксплуатации бота.

**Процесс установки модуля**

Системные требования

Операционная система: Ubuntu 18.04 или выше

Процессор: 4 vCPU

Оперативная память: 16 GB

Дисковое пространство: 50 GB SSD

Сеть: Доступ к интернету, разрешенные соединения с API Telegram и Yandex Cloud

Установка необходимых зависимостей

Обновление пакетов:  
sudo apt update && sudo apt upgrade -y

Установка Python 3.8+ и pip:  
sudo apt install python3 python3-pip -y

Установка необходимых Python-библиотек:  
pip3 install aiogram clickhouse-driver pandas schedule

Развертывание сервиса в контейнере Docker (если используется контейнеризация):  
docker build -t telegram-bot .

docker run -d --restart always --name telegram-bot -p 8080:8080 telegram-bot

Настройка подключения к базе данных

Конфигурация подключения к ClickHouse:  
from clickhouse\_driver import Client

client = Client(host='clickhouse\_host', user='default', password='password', database='premier\_bi')

Настройка токена Telegram API:  
import os

1. TOKEN = os.getenv('TELEGRAM\_BOT\_TOKEN')

**Запуск системы**

Запуск бота вручную

python3 bot.py

**Автоматический запуск при загрузке системы**

Создать systemd сервис:  
sudo nano /etc/systemd/system/telegram-bot.service

Добавить следующий конфигурационный файл:  
[Unit]

Description=Telegram Bot Service

After=network.target

[Service]

ExecStart=/usr/bin/python3 /path/to/bot.py

WorkingDirectory=/path/to/bot

Restart=always

User=bot\_user

[Install]

WantedBy=multi-user.target

Активировать сервис:  
sudo systemctl enable telegram-bot

sudo systemctl start telegram-bot

**Функциональность**

Команды бота

/start – Запуск бота и получение краткой справки по возможностям.

/help – Описание доступных команд и работы с ботом.

/reports – Доступ к основным отчетам (выручка, просмотры, подписки).

/subscriptions – Информация о статусе подписок пользователей.

/trends – Анализ изменения ключевых метрик по времени.

**Примеры взаимодействий**

Запрос отчета по выручке

Пользователь: /reports revenue

Бот: Выручка за последние 30 дней: 12 345 678 ₽

Текущий прирост: +3.2%

Запрос информации о подписках

Пользователь: /subscriptions

Бот: Активные подписки: 120 000

Отмена подписок за день: 1 500

**Роли пользователей**

Пользователь

Пользователь может получать отчеты, анализировать данные и взаимодействовать с ботом через Telegram. Доступ зависит от уровня разрешений, установленного администратором.

Администратор

Администратор управляет списком пользователей, назначает роли и права доступа, а также настраивает параметры работы бота, включая частоту обновления отчетов.

Сервер

Сервер выполняет обработку запросов, взаимодействует с базами данных и API Telegram, а также обеспечивает бесперебойную работу бота.

Платформа данных

Платформа данных отвечает за сбор, подготовку и агрегирование информации, которая затем передается пользователям через Telegram бот.

**Начало работы**

Подключиться к боту

* Откройте Telegram.
* В строке поиска введите название бота или используйте ссылку, предоставленную администрацией.
* Нажмите «Начать» или введите команду /start.

Основные команды

Бот поддерживает несколько команд для получения информации:

/start – Начало работы с ботом.

/help – Информация о доступных командах.

/reports – Получение отчетов по ключевым метрикам.

/subscriptions – Проверка статистики по подпискам.

/trends – Динамика изменений ключевых показателей.

**Работа с отчетами**

Запросить отчет

Введите команду /reports, затем выберите или напишите нужный тип отчета. Например:

Пользователь: /reports revenue

Бот:

Выручка за последние 30 дней: 12 345 678 ₽

Текущий прирост: +3.2%

Запрос подписок

Введите /subscriptions, чтобы получить актуальные данные:

Пользователь: /subscriptions

Бот:

Активные подписки: 120 000

Отмена подписок за день: 1 500

Динамика изменений (тренды)

Введите /trends, чтобы получить сводку по ключевым показателям:

Пользователь: /trends

Бот:

Средний рост аудитории: +2.5%

Динамика выручки: -1.3%

Дополнительные настройки

Если бот поддерживает несколько языков, используйте /settings для выбора нужного языка.

* Введите /settings.
* Выберите «Настройки уведомлений».
* Укажите частоту уведомлений (ежедневно, еженедельно, отключить).

**Часто задаваемые вопросы**

Что делать, если бот не отвечает?

* Проверьте подключение к интернету.
* Перезапустите Telegram и попробуйте снова.
* Убедитесь, что бот активен (обратитесь к администратору).

Как получить доступ к дополнительным данным?

Некоторые отчеты доступны только для определенных пользователей. Обратитесь к администратору для расширения доступа.

Как выйти из бота?

Просто закройте чат или используйте команду /stop.

## **Модуль «Модерация контента»**

Данный документ содержит инструкции по эксплуатации, настройке и обновлению модуля «Модерация контента». Модуль предназначен для автоматизированной разметки видеоконтента (фильмы, сериалы) в онлайн-кинотеатре для модерации и обеспечения безопасной и здоровой онлайн-среды для пользователей.

Архитектура системы

Модуль «Модерация контента» включает в себя следующие компоненты:

* Серверная часть: Обрабатывает запросы, анализирует контент и формирует отчеты.
* База данных ClickHouse: Хранит информацию о модерации и метаданные контента.
* Интерфейс командной строки: Позволяет запускать модули анализа и формировать отчеты.
* Подключение к внешним сервисам (Кинопоиск, IMDB): Получает данные о видеоконтенте.
* Система логирования: Записывает процесс работы и фиксирует возможные ошибки.

Процесс установки модуля

Требования к окружению

Перед установкой необходимо убедиться, что в системе доступны следующие компоненты:

* Операционная система: Ubuntu (рекомендуемая версия 20.04 и выше).
* Python: Версия 3.10 или выше.
* Библиотеки Python: Pandas, Selenium, Requests, OpenPyXL, Loguru и другие (полный список приведен в разделе 5).
* ClickHouse: Должен быть развернут и доступен для записи данных.
* Доступ к интернету: Необходим для взаимодействия с API Кинопоиск и IMDB.

Установка модуля

1. Клонировать репозиторий с исходным кодом:  
git clone <https://mlgit.cyrm.ru/ml-team/mediasurf.git>

2. Перейти в папку с проектом:  
cd mediasurf

3. Установить зависимости:  
pip install -r requirements.txt

4. Проверить корректность установки:  
python app.py --help  
Если команда вывела справочную информацию, установка прошла успешно.

Запуск системы

Запуск модуля осуществляется через командную строку.

* Подготовка входных данных

Перед запуском необходимо подготовить входной файл XLSX с перечнем контента.

* Запуск модуля

1. Указать путь к входному файлу в параметрах запуска:  
python app.py --input data.xlsx --output results.xlsx

2. Дождаться завершения обработки.

3. Проверить выходной файл results.xlsx с разметкой контента.

Функциональность

Модуль «Модерация контента» выполняет следующие задачи:

* Автоматическая классификация видеоконтента на основе предустановленных критериев (наличие сцен курения, алкоголя, наркотиков, ЛГБТ).
* Получение данных о контенте из внешних сервисов (Кинопоиск, IMDB).
* Формирование отчетов с цветовой разметкой (зеленый – безопасно, желтый – возможно, красный – нарушение).
* Ведение логов работы модуля.
* Возможность интеграции с BI-системами для визуализации данных.

Обновление и поддержка

Обновление модуля выполняется стандартными командами:

git pull origin main

pip install -r requirements.txt --upgrade

В случае обнаружения ошибок или необходимости доработки функционала, следует обратиться в службу технической поддержки.

# **Модуль «Сегментатор данных»**

Модуль «Сегментатор данных» предназначен для автоматизированного сбора, обработки и анализа данных пользователей, с возможностью их сегментации по различным критериям. Он предоставляет интерфейс для настройки сегментов и их выгрузки в различных форматах.

Назначение модуля

Модуль разработан для использования в компаниях, работающих с большими объемами данных, включая:

* Электронную коммерцию (анализ эффективности маркетинговых кампаний);
* Рекламные агентства (оптимизация рекламных кампаний);
* Разработчиков мобильных приложений (анализ поведения пользователей);
* Компании-разработчики ПО (оптимизация работы сервисов);
* Онлайн-сервисы (повышение удовлетворенности клиентов).

Принцип работы

1. Пользователь загружает данные из различных источников.

2. Происходит обработка данных и разбиение их на сегменты по заданным критериям.

3. Пользователь получает сформированные сегменты и может выгрузить их в различных форматах.

4. Сегменты доступны для анализа и интеграции с другими сервисами.

Структура данных

Модель данных PostgreSQL:

| Поле | Тип | Описание |
| --- | --- | --- |
| id | integer | Уникальный идентификатор |
| name | varchar(50) | Имя сегмента |
| rule | jsonb | Правило выгрузки |
| updated\_at | timestamp(6) | Дата обновления |

Модель данных ClickHouse:

| Поле | Тип | Описание |
| --- | --- | --- |
| user\_id | String | Идентификатор пользователя |
| date | Date | Дата события |
| platform\_main | String | Платформа пользователя |
| cohort | Date | Когорта |

Роли и полномочия

Администратор

* Управление пользователями (создание, удаление, назначение ролей);
* Контроль доступа к данным и функциям системы;
* Настройка параметров выгрузки и отчетности.

Пользователь

* Просмотр и управление сегментами данных;
* Выгрузка данных;
* Анализ сегментов через веб-интерфейс.

Использование модуля

Запуск системы

* Авторизация в системе через Яндекс ID.
* Выбор необходимых данных для анализа.
* Формирование сегментов с использованием гибких фильтров.
* Выгрузка сегментов в удобных форматах (CSV, XLSX).

Работа с сегментами

Создание сегмента: Пользователь задает название сегмента, выбирает атрибуты для фильтрации, группирует их и может копировать ранее созданные атрибуты.

Просмотр сегментов: Все ранее созданные сегменты отображаются в списке с возможностью поиска и фильтрации.

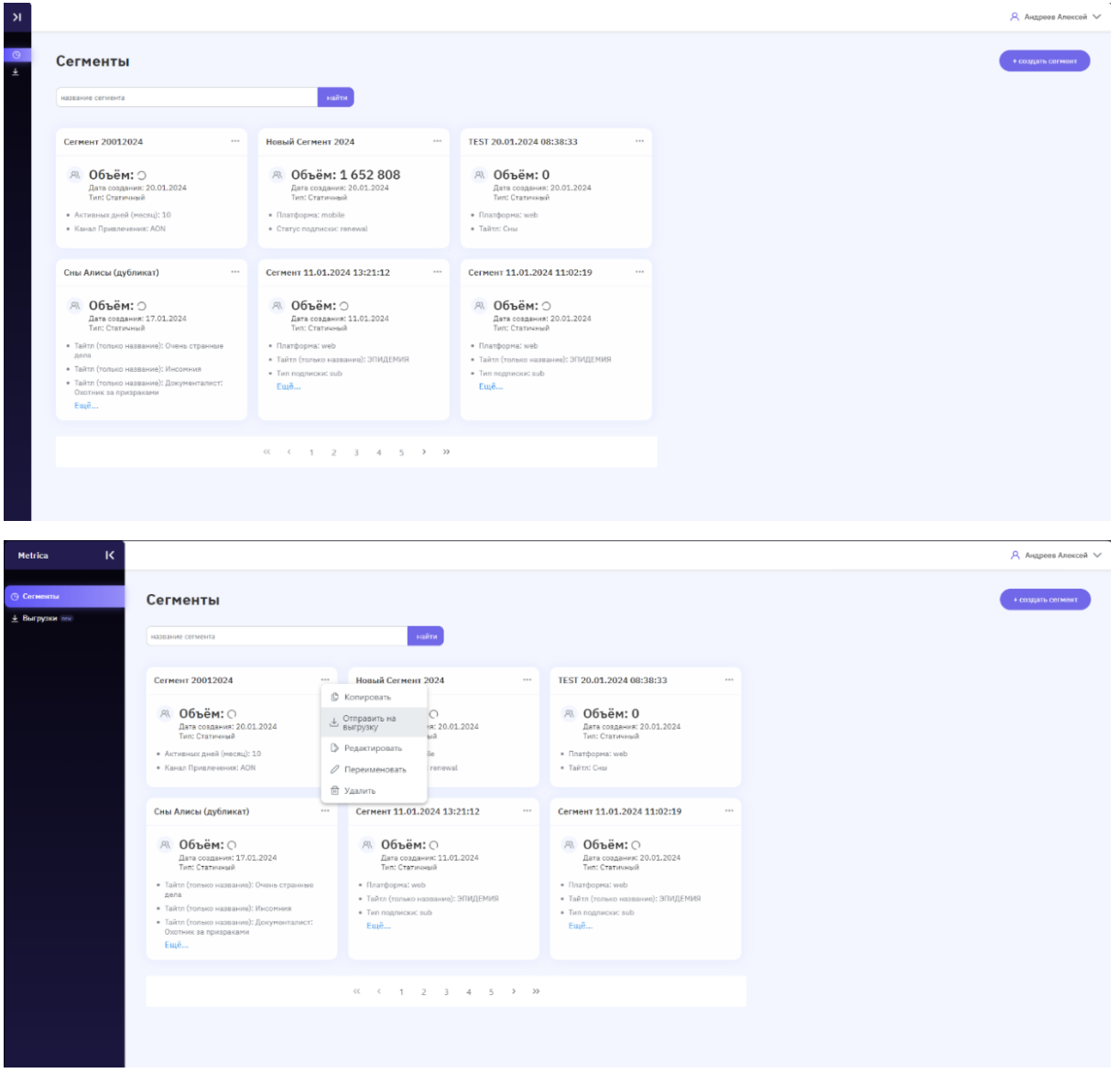
Редактирование сегмента: Пользователь может изменить параметры сегмента, обновить его название и настройки фильтрации.

Удаление сегмента: Сегмент может быть удален по запросу пользователя.

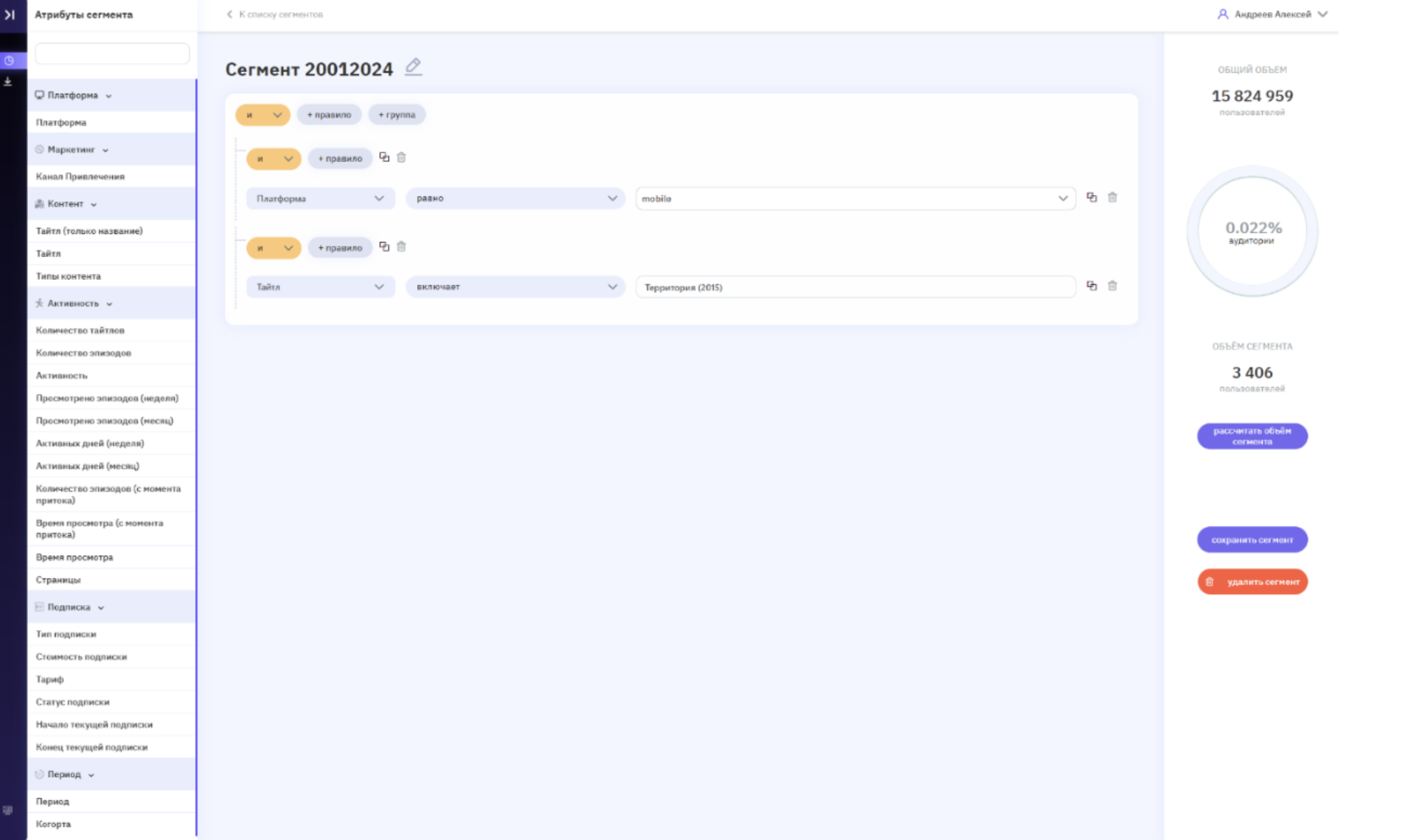
Выгрузка сегмента: Сегменты могут быть экспортированы в формате CSV или XLSX.

Пагинация: Если список сегментов велик, доступна пагинация для удобства навигации.

Создание групп атрибутов: Пользователи могут объединять атрибуты в группы для удобного фильтрования.



*Рис. 1 - Отправка сегмента на выгрузку*

**

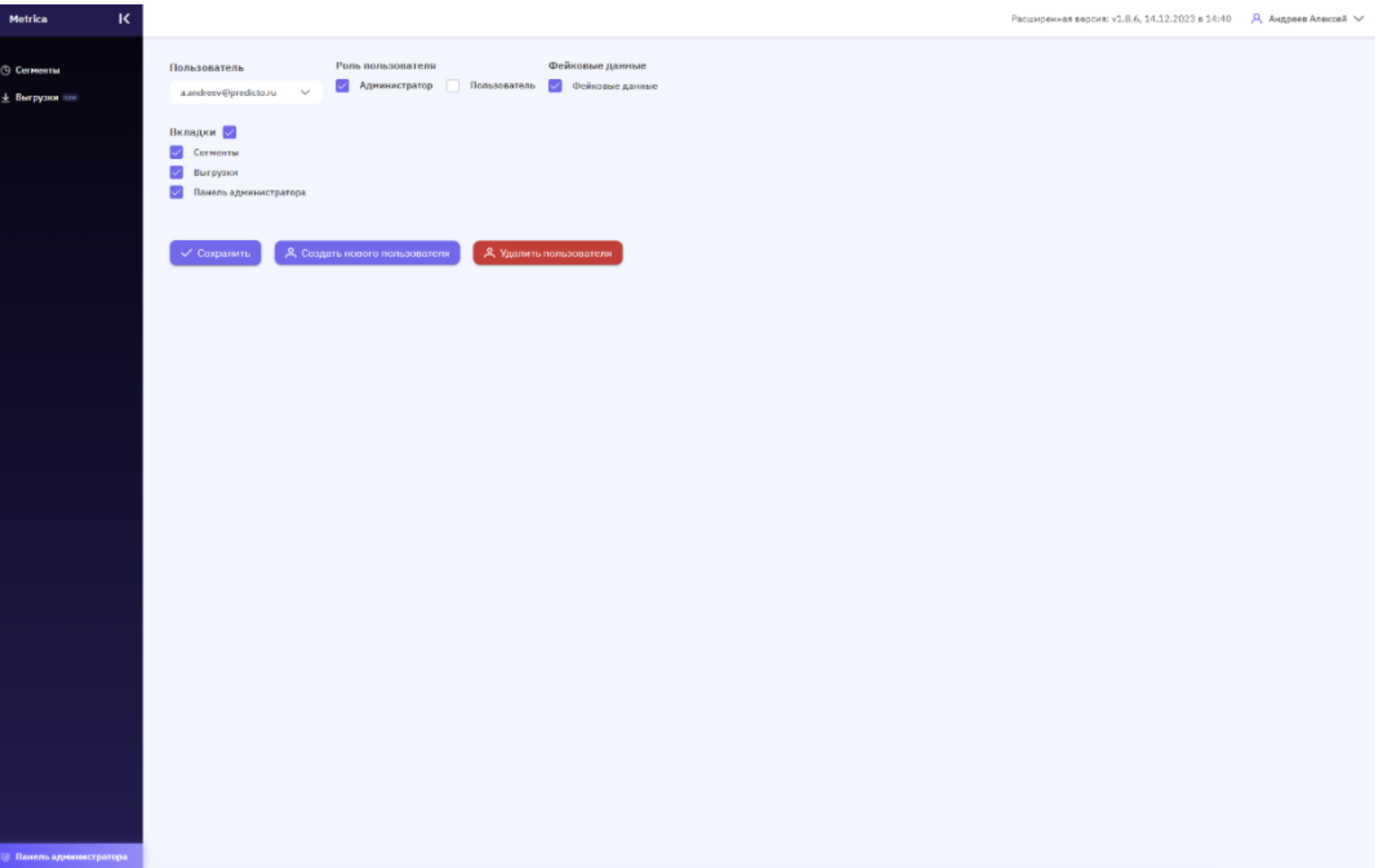
*Рис. 2 - Создание группы для объединения атрибутов фильтрации.*

Элементы страницы создания сегмента:

* + Общий объем пользователей,
  + Объем сегмента пользователей,
  + Кнопки "Рассчитать объем сегмента", "Сохранить сегмент", "Удалить сегмент".
  + Меню с группами атрибутов отображается слева.

Дополнительные возможности для Администратора:

* + Управление пользователями (создание, удаление, изменение ролей),
  + Назначение вкладок и изменение доступов,
  + Поиск пользователей по email.



*Рис. 3 - Функционал администратора по работе с пользователями.*

Завершение работы

Убедитесь, что все необходимые сегменты сохранены.

Закройте активные сессии работы с данными.

Выйдите из системы через интерфейс выхода.

## **Модуль «Тегизация»**

Модуль «Тегизация» предназначен для автоматического формирования тематик контента на основе интересов пользователей. Система анализирует историю просмотров, действия пользователей и другие внешние источники, создавая персонализированные тематические рекомендации. Основная цель модуля — автоматизация процесса тегирования, упрощение работы редакторов контента и повышение удовлетворенности пользователей.

Запуск модуля

Установка и настройка

1. Убедитесь, что серверная среда соответствует требованиям:

* + **ОС**: Ubuntu 20.04 и выше
  + **База данных**: PostgreSQL
  + **Облачная инфраструктура**: Yandex Cloud
  + **Контейнеризация**: Docker

2. Разверните микросервисы, используя Docker Compose:  
docker-compose up -d

3. Выполните миграцию базы данных:  
python manage.py migrate

4. Запустите сервер API:  
python manage.py runserver

5. Убедитесь, что сервис доступен по API.

Основные функции модуля

**Получение данных**

Система собирает актуальные и исторические данные о пользователях и контенте из внешних источников.

Данные хранятся в базе и регулярно обновляются для обеспечения актуальности рекомендаций.

**Анализ данных и формирование тегов**

Использует алгоритмы машинного обучения для анализа данных.

Создает облако тегов, релевантное каждому элементу каталога и истории просмотров пользователя.

Определяет тематики контента, наиболее подходящие пользователю.

**Генерация персонализированных рекомендаций**

Пользователи получают подборки контента, сформированные на основе тегов.

Система фильтрует контент по возрастным ограничениям и личным предпочтениям.

В случае недостатка данных используются "холодные" кластеры, формируемые на основе анализа поведения всех пользователей.

**Работа с модулем**

Запрос и обработка теговых рекомендаций

Как получить рекомендации?

1. Система получает запрос, содержащий **идентификатор пользователя** и **номер полки (tag1, tag2, tag3…)**.

2. Определяется, какие кластеры наиболее подходят пользователю.

3. Из кластера выбираются релевантные тайтлы (видеоконтент).

4. Выбранные тайтлы сортируются по степени релевантности и формируются в персонализированную полку.

Как фильтруется контент?

Используются векторные модели (эмбеды) и алгоритмы машинного обучения для сравнения интересов пользователя с контентом.

Применяются дополнительные фильтры, такие как возрастные ограничения и статус просмотра.

Использование API для работы с тегами

Запрос списка просмотренных пользователем тайтлов

GET https://premier-recsys.dev.predicto-tech.ru/api/dev/watchedfilter/users/$passmediaId/watchedfilter?watched=$boolean

**Пример ответа:**

{

"passmedia\_id": "EwNZX96fUmK8V5Lr8KF9f5",

"tv\_ids": [

{ "tv\_id": 1078, "title": "Stand Up", "watched": true },

{ "tv\_id": 2638, "title": "Сладкая жизнь", "watched": true }

]

}

Запрос списка пользователей, которые посмотрели указанные тайтлы

GET https://premier-recsys.dev.predicto-tech.ru/api/dev/watchedfilter/tv\_ids?tv\_ids=$tv\_id&limit=10

**Пример ответа:**

{

"passmedia\_ids": [

"123zNMtjVorGms4vdkqY4K",

"124j8yTrJuC47iw4uMxUVU",

"124PMUb9jV3HuTyncum57r",

"125nhatvd5dx2A4MEiyUrM"

]

}

Получение информации о тайтле по его идентификатору

GET <https://uma.media/api/metainfo/tv/$tv_id/>

Алгоритмы формирования тегов

**Кластеризация контента**

1. Используется модель BERT для обработки корпуса данных.

2. Выделяются потенциально наиболее кликабельные кластеры.

3. Применяется тематическая разметка (создание справочника тегов).

4. Определяется близость пользователя к конкретным кластерам на основе его действий и истории просмотров.

Меры схожести

ALS-расстояние между пользователем и тайтлом.

Сравнение тегов (по принципу tf-idf векторов).

Главная тематическая компонента полки пользователя.

Косинусное сходство между эмбеддингами пользователя и контента.

Алгоритм формирования ответа на запрос

Приходит запрос, содержащий идентификатор пользователя и номер полки - passmediaID + “tag1”...”tag5” (tagN - определяет порядковый номер полки/кластера для пользователя, сейчас запрашивается максимум 5 полок).

По PassmediaID и номеру полки определяется соответствующий кластер из таблицы TAGS\_USER\_CLUSTER.

Выбираются все кластеры пользователя с весом > 0, сортируются по убыванию веса и выбираются нужные кластеры.

По найденному кластеру выбирается список входящих в него tv\_id из таблицы TAGS\_CLUSTER\_ITEMS.

Выбранные тайтлы проходят фильтрацию и ранжирование по косинусной близости эмбеддингов пользователя и тайтла.

Если рекомендаций достаточно, формируется ответ. В противном случае, происходит переключение на другой кластер.

Масштабируемость и производительность

Обеспечена горизонтальная масштабируемость для работы с большими объемами данных.

Используются кластеризация серверов и балансировка нагрузки.

Скорость обработки API-запросов не более 200 мс для 95% запросов.

Система выдерживает нагрузку до 500 rps.

Завершение работы

При завершении работы с модулем рекомендуется:

1. Убедиться, что все изменения в тегах и кластерах сохранены.

2. Проверить, что все API-запросы были обработаны корректно.

3. Закрыть активные сессии, если работа производилась в администраторском режиме.

## 

## **Модуль «Рекомендательная система для стриминговых сервисов»**

Модуль "Рекомендательная система для стриминговых сервисов" представляет собой программное решение, основанное на алгоритмах машинного обучения и анализа данных, предназначенное для персонализированного подбора видеоконтента. Система учитывает предпочтения пользователей, историю просмотров, реакции на контент и общие тренды для формирования релевантных рекомендаций.

Запуск модуля

Предварительные требования

Перед запуском модуля необходимо:

* Убедиться, что серверная инфраструктура развернута (Kubernetes, Docker, Nginx).
* Проверить доступность баз данных (PostgreSQL, Redis, ClickHouse, OpenSearch).
* Настроить интеграцию с платформами, использующими рекомендации (например, Rutube).

Установка и запуск

Модуль устанавливается и настраивается как любое API-приложение. Установка производится с использованием контейнеризации (Docker/Kubernetes) и включает развертывание всех зависимых сервисов.

Запуск осуществляется с помощью контейнеризации Docker/Kubernetes:

kubectl apply -f recommendation-system-deployment.yaml

Или в случае локального развертывания:

docker-compose up -d

После запуска необходимо проверить доступность API:

curl -X GET http://localhost:8000/api/v2/rec/status

Если система запущена корректно, в ответ будет получен статус 200 OK.

Основные функции модуля

Персонализированные рекомендации

Анализ предпочтений пользователей.

Подбор видео на основе истории просмотров.

Категорийные рекомендации

Учет жанров и тематики контента.

Коллаборативные рекомендации

Подбор контента на основе схожих пользователей.

Контекстные рекомендации

Формирование рекомендаций, основанных на текущем видео.

Трендовые рекомендации

Выбор популярного контента по актуальным темам.

Работа с системой

Запрос персонализированных рекомендаций

Пример запроса к API:

curl -X POST http://localhost:8000/api/v2/rec/ \

-H "Content-Type: application/json" \

-d '{"user\_id": "12345", "limit": 10}'

Ответ сервера:

{

"version": "xclip\_1000\_2023\_08\_14",

"total": 10,

"payload": [

{"content\_id": "abcd1234", "score": 0.89},

{"content\_id": "xyz7890", "score": 0.75}

]

}

Логирование и мониторинг

Для анализа работы модуля используются:

* **Prometheus** – сбор метрик.
* **Grafana** – визуализация данных.
* **Grafana Loki** – логирование событий.

Алгоритмы формирования тегов

Система анализирует метаданные видео (название, описание, категории) и поведенческие данные пользователей (запросы, лайки) для присвоения тегов. Используемые методы:

* NLP (Natural Language Processing) – обработка текста и извлечение ключевых слов.
* Кластеризация – объединение схожих тегов.
* Семантический анализ – анализ взаимосвязей между терминами.

Алгоритм формирования ответа на запрос

Сбор данных

* Извлечение истории просмотров пользователя.
* Определение предпочтений на основе взаимодействия с контентом.

Применение алгоритмов

* Коллаборативная фильтрация – анализ схожих пользователей.
* Контентная фильтрация – подбор схожего контента.
* Машинное обучение – предсказание интересов.

Генерация списка рекомендаций

* Агрегация оценок по алгоритмам.
* Сортировка видео по релевантности.
* Формирование итогового списка рекомендаций.

Масштабируемость и производительность

Автоматическое масштабирование

* Использование Kubernetes для горизонтального масштабирования.
* Балансировка нагрузки через Yandex Application Load Balancer.

Оптимизация запросов

* Кэширование данных в Redis.
* Обработка запросов в потоковом режиме через Kafka.
* Хранение больших объемов данных в ClickHouse.

A/B тестирование

* Оценка эффективности рекомендаций через экспериментальные группы.
* Анализ метрик вовлеченности (время просмотра, количество кликов).

Завершение работы

Для корректного завершения работы модуля необходимо:

kubectl delete -f recommendation-system-deployment.yaml

Или для локального окружения:

docker-compose down

После остановки сервисов проверить состояние процессов:

ps aux | grep recommendation-system

При необходимости завершить зависшие процессы:

kill -9 <PID>

# 

# **«Поисковый модуль для стриминговых сервисов»**

Модуль используется для предоставления пользователям эффективного поиска контента на видео-платформах. Система ориентирована на обработку больших объемов данных и учета предпочтений пользователей для формирования персонализированных результатов поиска.

Установка программы

Требования к окружению

Для корректной работы поискового модуля необходимо следующее окружение:

Операционная система: Ubuntu или Debian.

Система виртуализации: Yandex Cloud (или аналогичные облачные сервисы).

Контейнеризация: Docker и Kubernetes.

Базы данных: PostgreSQL, OpenSearch, Redis, ClickHouse.

Языки программирования: Python.

Модели машинного обучения: PyTorch, scikit-learn, CatBoost, XGBoost и другие.

Программные платформы: Yandex Cloud, Kubernetes, Yandex Managed Services.. Использование поискового модуля в Rutube

Процесс установки

1. Установите Docker и настройте контейнеризацию.

2. Разверните Kubernetes для оркестрации контейнеров.

3. Настройте PostgreSQL, OpenSearch, Redis и ClickHouse для работы с данными.

4. Установите необходимые Python-библиотеки: scikit-learn, PyTorch, Optuna, CatBoost, XGBoost, и другие.

5. Настройте и подключите облачные сервисы, такие как Yandex Object Storage и Yandex Compute Cloud.

Начало работы

Для запуска системы необходимо выполнить несколько подготовительных шагов:

Подготовка окружения:

1. Подготовьте контейнеры с базовыми операционными системами (например, Ubuntu или Debian).

2. Настройте кластер Kubernetes для развертывания контейнеров и распределения нагрузки.

3. Разверните облачные сервисы для хранения и обработки данных (например, Yandex Cloud).

4. Убедитесь, что все сервисы, включая базы данных, работают корректно.

Для запуска системы используйте команду в командной строке:

bash

Копировать

docker-compose up

Подключите API для интеграции с другими системами, включая внешний контент и авторизацию. Настройте параметры кеширования запросов для оптимизации работы поисковой системы.

Эксплуатация модуля

Поисковый модуль предоставляет функциональность для быстрого и точного поиска контента на платформе. Пользователи могут искать видео, каналы, плейлисты, а также получать персонализированные рекомендации на основе предыдущих поисковых запросов и предпочтений.

Основные функциональные возможности

* Поиск по разным типам сущностей. Поиск по видео, каналам, персонам и другим сущностям.
* Поиск по контенту. Поддержка различных типов контента: UGC, фильмы, сериалы, музыкальные клипы и другие.
* Автодополнение и исправление ошибок. Поиск с предложением вариантов запросов в реальном времени и автоматическим исправлением ошибок.
* Персонализация поиска. Модуль учитывает историю поиска и предпочтения пользователя для улучшения релевантности результатов.
* Ранжирование результатов. Результаты поиска ранжируются на основе популярности контента и предпочтений пользователя.

Основные сценарии поиска

* Поиск по видео: Пользователи могут найти видео по ключевым словам, тегам, названиям и описаниям.
* Поиск по каналам: Возможность искать каналы по тематике, названию и описанию.
* Поисковые подсказки: Автодополнение запросов и исправление опечаток в реальном времени.
* Поиск внутри канала: Поиск видео или плейлистов в пределах конкретного канала.

Использование API для интеграции

Поисковый модуль предоставляет API для внешних интеграций, что позволяет интегрировать его с другими платформами и сервисами.

Пример запроса:

json

Копировать

POST /api/v2/search/

{

"scenario": "search.test.online.random",

"query": "Сериал Друзья",

"user\_id": "675ceb78-3d03-41b4-af5e-907499dfb3c0",

"limit": 10

}

Пример ответа:

json

Копировать

{

"version": "xclip\_1000\_2023\_08\_14",

"timestamp": "2024-04-11T16:27:07.468613",

"total": 10,

"payload": [

{

"content\_id": "a534f645-ce7b-4d34-b3a3-e7888ca88926",

"score": 0.494,

"type": "unknown"

}

]

}

Обработка ошибок:

Если запрос не может быть выполнен, система вернет ошибку с кодом и описанием. Возможные коды ошибок включают:

* **400**: Неправильный запрос.
* **404**: Ресурс не найден.
* **500**: Внутренняя ошибка сервера.

Интеграция модуля в Rutube

Поисковая система для стриминговых сервисов интегрирована в видео платформу Rutube, где используется в качестве основного инструмента для поиска и навигации по контенту на нескольких типах устройств.

Данный модуль интегрируется через API и используется для обработки поисковых запросов, формирования и ранжирования релевантных результатов, которые отображаются на видео платформе Rutube.

Поисковая система интегрирована в веб-версию rutube.ru для мобильных и desktop браузеров, а также в мобильные приложения на iOS и Android и в Smart TV (умные телевизоры).

Сценарии поиска

Поисковый модуль для стриминговых сервисов используется в нескольких основных сценариях поиска в видео платформе Rutube.

* Поиск по видео

Пользователи могут находить конкретные видео по ключевым словам, названиям, тегам или описаниям. Система учитывает релевантность, популярность и персонализированные предпочтения для выдачи наиболее подходящих результатов.

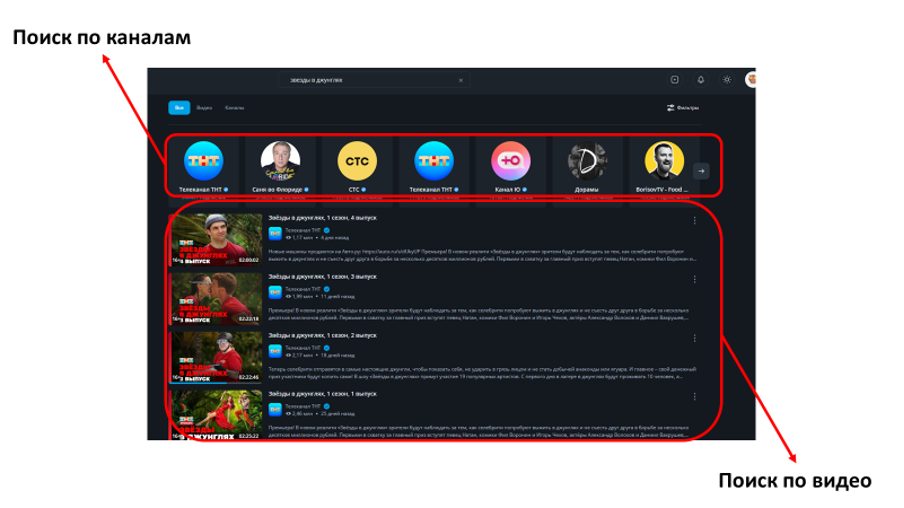


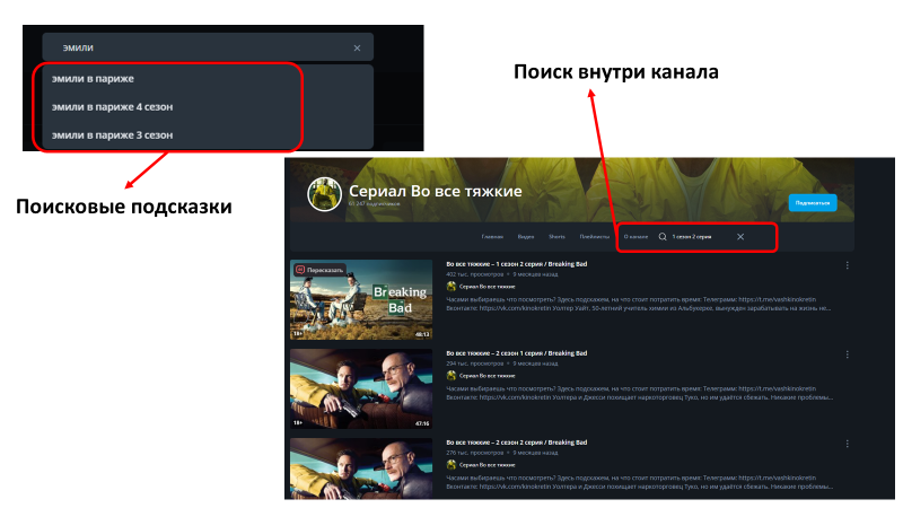
Рис.1 Поиск по каналам и поиск по видео

* Поиск по каналам

Возможность находить каналы по их названиям, тематике или описанию. Пользователи могут быстро перейти к интересующим их авторам или тематическим подборкам контента.

* Поисковые подсказки

Автодополнение запросов в реальном времени, исправление опечаток и предложение популярных или релевантных вариантов поиска, чтобы упростить и ускорить процесс ввода запросов.



*Рис.2 Поисковые подсказки и поиск внутри канала*

* Поиск внутри канала

Пользователи могут искать видео или плейлисты в рамках конкретного канала, что позволяет быстро находить нужный контент у любимых авторов или по определенной тематике.

Пользователи поискового модуля в Rutube

* Горячие пользователи

Это активные пользователи, которые регулярно используют поиск на платформе. Они формируют подробную историю поисковых запросов, просматривают результаты, взаимодействуют с найденным контентом (лайки, комментарии, подписки) и уточняют свои запросы. Их предпочтения и поведение хорошо известны системе, что позволяет предлагать им персонализированные и релевантные результаты поиска.

* Холодные пользователи

Это новые пользователи, которые только начали использовать платформу, или те, у кого еще нет истории поисковых запросов. Поскольку система не имеет данных об их предпочтениях, результаты поиска для них строятся на основе общих трендов, популярного контента, региональных предпочтений или контекста текущего запроса.

* Зарегистрировавшиеся пользователи

Пользователи, которые прошли регистрацию на платформе, предоставив свои данные (номер телефона или электронную почту). Они могут быть как горячими (активно используют поиск), так и холодными (новые пользователи). Регистрация позволяет системе лучше анализировать их поисковое поведение, учитывать историю запросов и предлагать более точные и персонализированные результаты.

* Незарегистрировавшиеся пользователи

Пользователи, которые еще не зарегистрировались на платформе. Они идентифицируются с помощью cookie или других технических методов. Незарегистрировавшиеся пользователи также могут быть как горячими (активно используют поиск), так и холодными (только начали взаимодействовать с платформой). Для них результаты поиска формируются на основе текущего запроса, популярного контента и общих трендов.

Задачи поискового модуля в Rutube

* Удержание пользователя за счет быстрого и точного поиска контента, повышение его интереса и вовлеченности благодаря релевантным результатам, соответствующим запросам и предпочтениям.
* Улучшение пользовательского опыта в части навигации и поиска контента за счет предоставления персонализированных и удобных инструментов поиска, включая автодополнение, исправление ошибок и фильтрацию результатов.
* Повышение пользовательских метрик на сервисе — например, увеличение времени просмотра (watchtime) за счет быстрого нахождения интересующего контента и повышение возвращаемости пользователя (retention rate) благодаря удобству и эффективности поиска.

Спецификация интеграционного API

Описание принципов интеграции

«Поисковый модуль для стриминговых сервисов» предоставляет API для внешних интеграций с целью формирования персонализированной поисковой выдачи релевантного контента в ответ на запрос пользователя на крупных медиа платформах (Rutube или подобные платформы). Этот API осуществлять поиск на основе различных сценариев, а также персонализировать поисковую выдачу, учитывая основную мета информацию о контенте, его популярность и трендовость, предпочтения пользователей, историю просмотров и другие параметры.

Основные возможности API

**Получение релевантных результатов:**

Возможность искать разные типы контента на основе разных сценариев - например, поисковые каналы, поиск по видео или поисковые подсказки.

Поддержка пагинации через параметры limit и offset.

Возможность использования кеширования запросов для оптимизации производительности.

**Персонализация поиска.** Учет идентификатора пользователя (user\_id) для формирования персонализированных результатов поиска.

**Кеширование:**

Настройка режима кеширования через заголовок x-cache-mode.

Возможность использования реальных данных в режиме реального времени (off), кеширования на основе параметров запроса (use\_request\_fields).

**Гибкость сценариев:**

Возможность указывать различные сценарии поиска через параметр scenario.

Поддержка запасных сценариев (fallback), если основной сценарий не может быть выполнен.

**Обработка ошибок:**

Поддержка различных HTTP-статусов для обработки ошибок (400, 401, 403, 404, 408, 422, 429, 495, 500, 503).

Возврат детализированных сообщений об ошибках в формате JSON.

7. Спецификация API запросов

7.1 Поиск по контенту

**Запрос:** POST /api/v2/search/

**Параметры запроса:**

● x-request-id (опционально): Уникальный идентификатор запроса для отслеживания. Если не предоставлен, генерируется автоматически.

● x-cache-mode (опционально): Режим кеширования. Возможные значения:

● off — игнорировать кеш, использовать только real-time данные.

● use\_request\_fields — формировать ключ кеширования на основе параметров запроса.

**Тело запроса (JSON):**

● scenario (обязательно): Название сценария поиска.  
Формат: [service].[source].[type].[algorithm].  
Пример: search.test.online.random.

● query: Текстовый запрос пользователя.

● limit (опционально): Максимальное количество результатов в ответе. По умолчанию: 20.

● user\_id (опционально): Идентификатор пользователя для персонализации поиска.

● offset (опционально): Смещение для пагинации. По умолчанию: 0.

**Ответ (JSON):**

● version: Версия контента.

● cache: Режим кеширования, использованный для запроса.

● timestamp: Временная метка ответа.

● fallback: Список запасных сценариев, если они были использованы.

● total: Общее количество результатов в ответе.

● payload: Список найденного контента. Каждый элемент содержит:

● content\_id: Идентификатор контента.

● score: Оценка релевантности контента.

● type: Тип контента.

*Пример ответа:*

{

"version": "xclip\_1000\_2023\_08\_14",

"cache": "use\_request\_fields",

"timestamp": "2024-04-11T16:27:07.468613",

"fallback": [],

"total": 10,

"payload": [

{

"content\_id": "a534f645-ce7b-4d34-b3a3-e7888ca88926",

"score": 0.494,

"type": "unknown"

},

{

"content\_id": "d0943291-8dc5-431d-b4bf-0be83b30a437",

"score": 0.161,

"type": "unknown"

}

]

}

Обработка ошибок

API возвращает различные HTTP-статусы для индикации ошибок:

● 400: Неправильный запрос.

● 401: Отсутствие токена авторизации.

● 403: Запрет доступа.

● 404: Ресурс не найден.

● 408: Таймаут запроса.

● 422: Ошибка валидации данных.

● 429: Слишком много запросов.

● 495: Ошибка сертификата.

● 500: Внутренняя ошибка сервера.

● 503: Сервис временно недоступен.

*Каждая ошибка сопровождается JSON-объектом с деталями:*

{

"details": "Упс, что то пошло не так...",

"code": "P012345E"

}

Безопасность

API использует авторизацию через заголовок x-authorization-token. Токен должен быть передан в каждом запросе для доступа к защищенным endpoint'ам.

*Пример использования:*

POST /api/v2/search/ HTTP/1.1

Content-Type: application/json

x-authorization-token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

x-cache-mode: use\_request\_fields

{

"scenario": "search.test.online.random",

“query”: "Сериал Друзья",

"user\_id": "675ceb78-3d03-41b4-af5e-907499dfb3c0",

"limit": 10

}

*Ответ:*

{

"version": "xclip\_1000\_2023\_08\_14",

"cache": "use\_request\_fields",

"timestamp": "2024-04-11T16:27:07.468613",

"fallback": [],

"total": 10,

"payload": [

{

"content\_id": "a534f645-ce7b-4d34-b3a3-e7888ca88926",

"score": 0.494,

"type": "unknown"

}

]

}

## **Модуль «BI аналитика для стриминговых сервисов»**

Модуль «BI аналитика для стриминговых сервисов» — это инструмент для продуктовой аналитики на видео платформах, таких как Rutube. Он позволяет анализировать данные о пользователях, контенте и их взаимодействиях, обеспечивая глубокий анализ поведения аудитории, эффективности контента и работы алгоритмов.

Модуль интегрируется с модулем «Инфраструктура для сервиса», поисковыми и рекомендательными системами для стриминговых сервисов, получая данные напрямую. Это позволяет быстро адаптироваться к новым источникам информации и приступать к полноценной аналитике с минимальными задержками.

Сервис поддерживает анализ ключевых метрик, таких как время просмотра, CTR, retention rate, а также помогает выявлять тренды, оптимизировать контент и тестировать гипотезы для повышения качества рекомендаций и поиска.ий.

Запуск модуля

Предварительные требования

Доступ к Yandex Cloud и соответствующим ресурсам (ClickHouse, DataSphere, Apache Superset).

Настроенная интеграция с модудем «Платформой данных»

Доступ к данным стримингового сервиса (Rutube и др.).

Административные права для настройки BI-инструментов.

Установка и запуск

Развертывание инфраструктуры с помощью Kubernetes (Yandex Managed Service for Kubernetes®).

Запуск контейнеров с ClickHouse, DataSphere и Apache Superset.

Настройка интеграции с источниками данных.

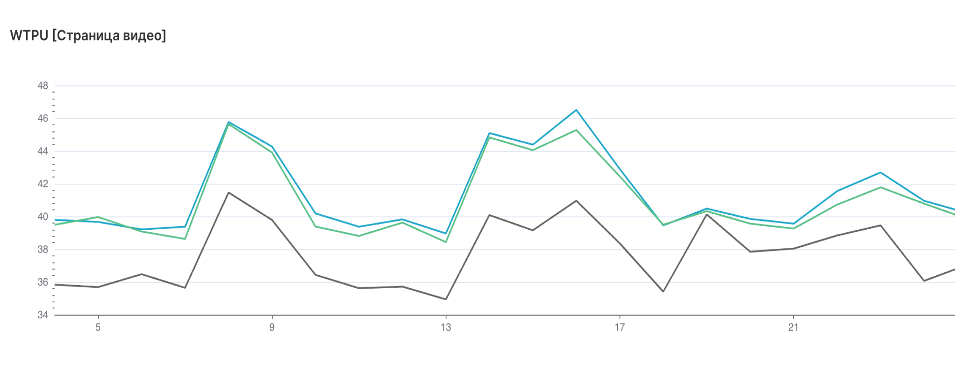
Авторизация в системе и предоставление прав доступа пользователям.

Аналитические инструменты

Примеры доступных дашбордов

Среднее время просмотра на пользователя

Дашборд показывает среднее время просмотра на пользователя (WTPU, Watch Time Per User). Это метрика, которая измеряет среднее время, проведенное пользователем на платформе за определенный период (например, за день, неделю или месяц). Она рассчитывается как общее время просмотра контента, деленное на количество активных пользователей.



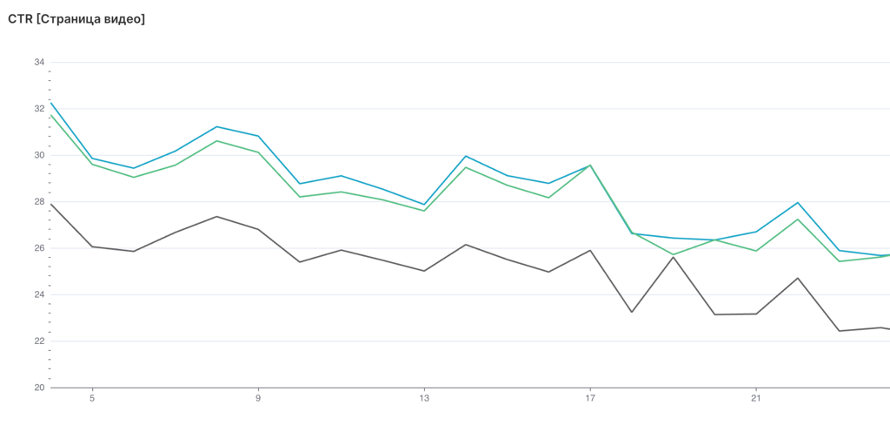
*Рис. 1 Дашборд WTPU (Watch Time Per User)*

Дашборд позволяет рассматривать данные в следующих разрезах:

* Период: Фильтрация данных по дате.
* Категория: Выбор одной или нескольких категорий видео.
* Группа: Группа в A/B тестировании.
* Флаг регистрации пользователя: Анализ в разрезе зарегистрированных / незарегистрированных пользователей.
* Свежесть видео: Время после публикации видео на платформе.
* Длина видео: Длина опубликованного видео.
* Устройство: Устройство, с которого происходит просмотр (web браузер, мобильный телефон, планшет или Smart TV).

Click-Through Rate (CTR) в рекомендациях

Дашборд показывает долю кликов (CTR, Click-Through Rate) на определенной рекомендательной поверхности. Это метрика, которая показывает процент пользователей, которые кликнули на определенный элемент рекомендательной поверхности относительно общего числа пользователей, которым этот элемент был показан.



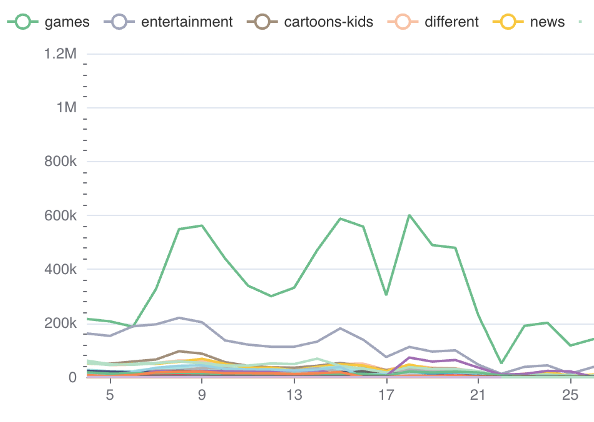
*Рис. 2 Дашборд CTR (Click-Through Rate)*

Дашборд позволяет рассматривать данные в следующих разрезах:

* Период: Фильтрация данных по дате.
* Категория: Выбор одной или нескольких категорий видео.
* Группа: Группа в A/B тестировании.
* Флаг регистрации пользователя: Анализ в разрезе зарегистрированных / незарегистрированных пользователей.
* Свежесть видео: Время после публикации видео на платформе.
* Длина видео: Длина опубликованного видео.
* Устройство: Устройство, с которого происходит просмотр (web браузер, мобильный телефон, планшет или Smart TV).

Число показов видео по категориям

Метрика "Число показов видео в рекомендациях по категориям" измеряет, сколько раз видео из определенной категории (например, "Сериалы", "Музыка", "Игры") было показано пользователям в рекомендательной ленте за определенный период.



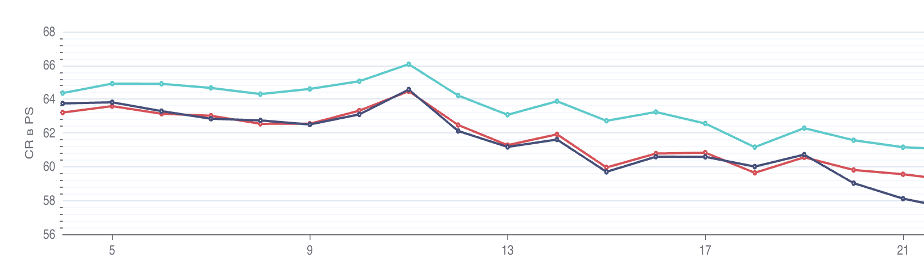
*Рис. 3 Дашборд “Число показов видео в рекомендациях по категориям”*

Дашборд позволяет рассматривать данные в следующих разрезах:

* Период: Фильтрация данных по дате.
* Категория: Выбор одной или нескольких категорий видео.
* Группа: Группа в A/B тестировании.
* Флаг регистрации пользователя: Анализ в разрезе зарегистрированных / незарегистрированных пользователей.
* Свежесть видео: Время после публикации видео на платформе.
* Длина видео: Длина опубликованного видео.

Click-Through Rate (CTR) в поиске

Дашборд показывает долю кликов по поисковым результатам (CTR, Click-Through Rate). Это метрика, которая показывает, какой процент пользователей кликнул на результаты поиска после ввода запроса.



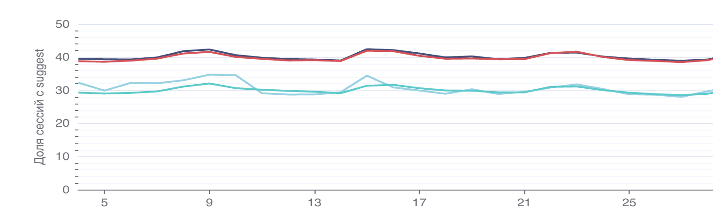
*Рис. 4 Дашборд Click-Through Rate (CTR) в поиске*

Дашборд позволяет рассматривать данные в следующих разрезах:

* Период: Фильтрация данных по дате.
* Группа: Группа в A/B тестировании.
* Флаг регистрации пользователя: Анализ в разрезе зарегистрированных / незарегистрированных пользователей.
* Популярность запрос: Анализ в разрезе по частотности запроса (популярный, середина или хвост).

Доля сессий с поисковыми подсказками

Метрика "Доля сессий с поисковыми подсказками" измеряет, какой процент пользовательских сессий на платформе включает использование поисковых подсказок (автодополнения). Она рассчитывается как отношение числа сессий, в которых пользователи взаимодействовали с подсказками, к общему числу сессий.



*Рис. 5 Дашборд "Доля сессий с поисковыми подсказками"*

Дашборд позволяет рассматривать данные в следующих разрезах:

* Период: Фильтрация данных по дате.
* Группа: Группа в A/B тестировании.

Средства Adhoc аналитики

Аналитика в ClickHouse

Агрегированные витрины данных, формируемые модулем «Платформы данных» (также фигурирует в документации как «Инфраструктура для сервиса»), доступным для adhoc аналитики при использовании модуля «BI аналитика для стриминговых сервисов».

ClickHouse — это высокопроизводительная колоночная СУБД, которая идеально подходит для adhoc аналитики по агрегированным витринам данных. Благодаря своей скорости обработки запросов и эффективной работе с большими объемами данных, ClickHouse позволяет аналитикам быстро выполнять сложные запросы к агрегированным данным без необходимости предварительной подготовки. Это особенно полезно для анализа исторических данных, построения отчетов и исследования трендов, где требуется гибкость и оперативность.

ClickHouse поддерживает стандартный SQL, что делает его удобным для аналитиков, а также предоставляет возможности для работы с материализованными представлениями и предварительно агрегированными данными. Это позволяет ускорить выполнение запросов к часто используемым витринам, сохраняя при этом возможность глубокого анализа.

Аналитика в Apache Superset

Apache Superset — это современный инструмент для визуализации и анализа данных, который отлично подходит для adhoc аналитики. Он предоставляет интуитивный интерфейс для создания интерактивных дашбордов и отчетов, что позволяет аналитикам и бизнес-пользователям быстро исследовать данные без необходимости написания сложных запросов. Superset поддерживает множество источников данных, включая ClickHouse, что делает его универсальным решением для работы с агрегированными витринами.

С помощью Apache Superset пользователи могут легко создавать визуализации, выполнять фильтрацию данных "на лету" и проводить глубокий анализ с помощью встроенного SQL-редактора. Это особенно полезно для adhoc аналитики, где требуется гибкость и оперативность.

Аналитика в Apache Spark

При необходимости пользователи имеют возможность работать с сырыми даными, используя Apache Spark – мощный фреймворк для распределенной обработки данных. Благодаря архитектуре, ориентированной на распределенные вычисления, Apache Spark может обрабатывать данные на сотнях узлов, что делает его идеальным решением для задач, требующих высокой производительности и масштабируемости.

Apache Spark поддерживает работу с сырыми данными в различных форматах (например, JSON, CSV, Parquet) и предоставляет инструменты для их очистки, трансформации и анализа. Это позволяет аналитикам и инженерам данных работать с необработанными данными напрямую, не требуя предварительной агрегации.

Работа с системой

Авторизация в системе через корпоративные учетные записи.

Доступ к дашбордам с основными метриками и отчетами.

Работа с WTPU Dashboard:

Выберите период анализа (день, неделя, месяц).

Используйте фильтры (категория видео, возраст контента, длина видео, устройство просмотра).

Интерпретируйте результаты для оценки вовлеченности пользователей и улучшения стратегии контента.

Фильтрация данных по различным параметрам (категория контента, временные диапазоны, пользовательские сегменты).

Формирование пользовательских запросов через SQL-редактор для Adhoc-анализа.

Сохранение и экспорт данных в виде отчетов (CSV, JSON, Excel, PDF).

Доступ к дашбордам с основными метриками и отчетами.

Фильтрация данных по различным параметрам (категория контента, временные диапазоны, пользовательские сегменты).

Формирование пользовательских запросов через SQL-редактор для Adhoc-анализа.

Сохранение и экспорт данных в виде отчетов (CSV, JSON, Excel, PDF).

Аналитические инструменты

WTPU (Watch Time Per User) Dashboard:

Позволяет фильтровать данные по дате, категории контента, длине видео и другим параметрам.

Используется для анализа эффективности видео и привлечения пользователей.

Интегрирован в Apache Superset и доступен для Adhoc-анализа.

ClickHouse – высокопроизводительная колоночная СУБД для Adhoc-аналитики.

Apache Superset – инструмент BI для визуализации и построения дашбордов.

DataSphere – среда для интерактивного анализа данных.

Apache Spark – распределенная обработка больших объемов данных.

Настройка и администрирование

Настройка источников данных (интеграция с платформой данных).

Конфигурация Apache Superset (создание дашбордов, управление отчетами).

Оптимизация ClickHouse (индексация данных, настройка запросов).

Мониторинг работы системы (Prometheus, Grafana)

Обновление и масштабирование (управление контейнерами, авто-масштабирование в Kubernetes).

**Завершение работы**

Завершение сеанса работы пользователя (выход из системы).

Остановка BI-системы при необходимости (выключение контейнеров с ClickHouse, Apache Superset и др.).

Архивирование данных и отчетов.

Мониторинг логов для выявления возможных ошибок.