**Документация, содержащая описание функциональных характеристик модуля программного обеспечения**

**«Персонализация контента»**

**г. Москва 2025**

**Содержание**

[1. Предназначение ПО 3](#_heading=h.s2xpnm9x5zoo)

[2. Описание функциональных характеристик 3](#_heading=h.7ep7zxhbxr7i)

[3. Архитектура модуля «Персонализация контента» 4](#_heading=h.fuxsqccpbuer)

[4. Схема работы 5](#_heading=h.spk4gfqhsfnh)

[5. Используемые языки программирования и фреймворкам 17](#_heading=h.pf377y6aljju)

[6. Описание технической инфраструктуры 17](#_heading=h.9nne77bsuqky)

[7. Эксплуатация системы 18](#_heading=h.s1q4gko3eb1g)

# Предназначение ПО

Программное обеспечение "Персонализация контента" предназначено для обработки поступающих запросов и формирования персонализированных рекомендаций видеоконтента для пользователей онлайн-кинотеатра Premier. Основные задачи модуля:

* Сбор и анализ данных каталога контента и пользовательских действий;
* Расчет персонализированных подборок контента с применением алгоритмов машинного обучения;
* Передача сформированных рекомендаций через API в клиентские системы;
* Обеспечение высокой точности рекомендаций на основе истории просмотров, предпочтений и статистики популярности контента.

# Описание функциональных характеристик

1. Персонализированные предложения по контенту. Генерация индивидуальных рекомендаций на основе предпочтений пользователей и их истории просмотров. Размещение персонализированных рекомендаций на полке "Вам понравится".
2. Управление профилем. Пользователи могут управлять своими профилями, корректировать предпочтения и получать соответствующие рекомендации.
3. Алгоритмы рекомендаций с использованием методов машинного обучения - коллаборативная фильтрация (выявление схожих пользователей и их предпочтений) и контентная фильтрация (рекомендации на основе характеристик контента). Гибридные подходы (комбинация различных методов, включая ранжирование с использованием градиентного бустинга).
4. Распределение видов запросов к системе:

а) Персонализированные (с учетом истории пользователя);

б) Групповые (на основе статистики просмотра);

с) "Холодные" (для пользователей без истории взаимодействий).

1. Сбор и анализ данных о взаимодействиях пользователей (просмотры, клики, лайки), анализ состояния системы и логирование ошибок, анализ эффективности рекомендаций.

# Архитектура модуля «Персонализация контента»

Модуль представляет из себя сервис, который собирает данные и предоставляет рекомендации для формирования рекомендательных полок.

Модуль "Персонализация контента" реализует микросервисную архитектуру, обеспечивающую гибкость, масштабируемость и отказоустойчивость системы. Основные компоненты модуля:

Обработка данных – агрегирует данные о пользователях, истории просмотров и контенте из внешних источников;

ML-рекомендации – включает алгоритмы коллаборативной и контентной фильтрации, а также гибридные модели на основе глубокого обучения;

API – предоставляет интерфейсы взаимодействия с внешними сервисами для получения и передачи рекомендаций;

Хранение данных – отвечает за хранение пользовательских профилей, метаданных контента и предварительно рассчитанных рекомендаций;

Логирование – контроль за работой системы, анализирует ошибки и собирает метрики производительности.

Пользователи получают доступ к персонализированному контенту через интернет в мобильном приложении и веб-приложении, а также на смарт-телевизорах и других устройствах. Сами интерфейсные приложения (например, мобильное приложение) не входят в рамки разработки Модуля.

Модель видеоконтента (тайтла) в рекомендательной системе представляет собой сущность, обладающую следующими атрибутами:

* Идентификатор – числовой код;
* Название – текстовое наименование. Для сериалов или шоу используется наименование всей франшизы, например «Реальные пацаны»;
* Тип видеоконтента – фильм, сериал, шоу;
* Возрастные ограничения (возрастной рейтинг) - 0+, 6+, 12+, 16+, 18+;
* Принадлежность к жанрам;
* Главный жанр;
* Географическая доступность;
* Категорийные признаки контента;
* Возможность просмотра без наличия платной подписки.

Запрос к рекомендательной системе определяет характер и ограничения для рекомендаций. Запрос может содержать следующую информацию:

* Тип рекомендации;
* Идентификатор пользователя;
* Возрастные ограничения;
* Географические ограничения;
* Категории контента;
* Жанровые ограничения.



Рис. 1 Архитектурная схема модуля «Персонализация контента»

# Схема работы

Ролевая модель подразумевает две роли:

***Зарегистрированный пользователь*** - пользователь, прошедший регистрацию и авторизацию. Имеет доступ к контенту в соответствии со своей подпиской. Получает персонализированные рекомендации при наличии достаточной истории просмотров в сервисе. Зарегистрированный пользователь идентифицируется по ID;

***Незарегистрированный пользователь*** - пользователь, не прошедший регистрацию и/или авторизацию. Таким пользователям доступна только часть контента (например, раздел «Бесплатное»). Незарегистрированные пользователи не получают персональные рекомендации.

Вне зависимости от типа запроса и выполняемого вида рекомендации все рекомендательные функции могут быть сведены к следующим основным этапам:

1. Анализ и обработка рекомендательного запроса;
2. Отбор кандидатов;
3. Фильтрация;
4. Ранжирование;
5. Формирование ответа

1. Анализ и обработка рекомендательного запроса

Сервис рекомендаций получает запросы от внешних систем, в частности от сервисов Uma.Tech. Каждый запрос проверен на корректность. В случае валидности запроса должна быть вызвана функция формирования рекомендаций соответствующего типа с применением ограничений, указанных в запросе. В случае некорректного запроса сервис должен возвращать сообщение о ошибке вызывающей стороне.



*Рис. 1 - Алгоритм анализа и обработка рекомендательного запроса*

1. Отбор кандидатов

При отборе кандидатов для формирования рекомендаций пользователя идентифицируется на основании ID пользователя из запроса рекомендаций.

Возможны три ситуации:

«холодный» пользователь - пользователь не указан в запросе. В этом случае некоторые типы рекомендательных пайплайнов могут формировать так называемые «холодные рекомендации», если они заложены в рекомендательный сервис. Так как информация о пользователе полностью отсутствует, то фильтрация задействуется только на основе ограничений запроса.

«теплый» пользователь - пользователь указан, но для него в системе присутствует только часть необходимой информации и персональные рекомендации еще не были предрасчитаны. В этой ситуации, при наличии необходимой информации, часть пайплайнов могут выполнить подбор кандидатов либо использовать холодные рекомендации, но накладывать на этапе фильтрации и ранжирования все возможные типы ограничений или предпочтений, известные на текущий момент.

«горячий» пользователь – пользователь, о котором сервис уже имеет достаточно информации для формирования полноценных персональных рекомендаций.

1. Фильтрация

В каждом типе рекомендаций может формироваться свой набор фильтров. В общем случае для работы сервиса требуются следующие виды фильтров:

* фильтр «включенных» тайтлов фильтр бесплатных тайтлов фильтр возрастного ограничения фильтр жанра
* фильтр географической доступности
* фильтр категорийных признаков
* фильтр пересечения с другим типом рекомендаций фильтр просмотренных тайтлов



*Рис. 2 - Алгоритм фильтрации кандидатов*

1. Ранжирование

В зависимости от типа рекомендательного пайплайна отфильтрованные кандидаты могут быть переупорядочены по различным критериям. Итоговый порядок кандидатов должен отражать максимальный интерес для пользователя с точки зрения Сервиса.

Некоторые наборы кандидатов могут быть уже заранее отсортированы по предпочтениям пользователей, в этом случае при ранжирование используются только те признаки, которые не учтены в начальном наборе.

Ранжирование может выполняться по следующим критериям:

* Сортировка на основе векторной близости векторов признаков (embeddings) для пользователя и кандидата. Чем ближе оказываются вектора, тем более подходящим для пользователя должен оказаться контент;
* Сортировка на основе новизны контента. Новые тайтлы более вероятно окажутся интересны пользователю, чем те которые он уже просматривал или видел в рекомендациях;
* Сортировки на основе специальных бизнес-правил. Например, целевое поднятие тайтлов в рейтинге.

Финальным этапом в блоке ранжирования может выступать блок ограничения – лимиттер. В этом блоке отсекаются кандидаты, если существует ограничение на максимальное количество кандидатов, либо формируется пустой список кандидатов, если не удалось подобрать минимально необходимое их число.



*Рис. 3 - Алгоритм ранжирования кандидатов*

1. Формирование ответа

Генерация рекомендаций для разных контентных полок осуществляется различными пайплайнами согласно следующим принципам:

1. Полка главной страницы «Популярное» формируется двухуровневой моделью рекомендаций, в которой кандидатными моделями первого уровня выступают:

ALS, LightFM, BPR, строящиеся на матрице взаимодействия, ячейки которой заполняются на основе данных об успешности просмотров фильмов и сериалов (для фильмов такая оценка выводится из доли просмотра, для сериала – на основе работы отдельной модели, предсказывающей вероятность начала просмотра следующего сезона). При заполнении ячеек матрицы применяется дисконтирование по времени;

аналогичная ALS, ограниченная бесплатными тайтлами;

контентная модель, определяющая близость текстовых embeddings описаний тайтлов к описаниям успешно просмотренных пользователем;

DSSM-модель с пользовательскими и контентными признаками, обученная на успешности просмотров.

Помимо этих моделей, отдельно обучаются:

ALS на жанрах и возрастных ограничениях для получения векторных представлений этих сущностей и пользователей;

ALS на матрице, ячейки которой заполняются данными о соотношении кликов по постерам и их показам (impressions), для учета персонального интереса к постеру и обеспечения динамичности рекомендаций.

Выборка кандидатов от перечисленных моделей ранжируется моделью градиентного бустинга, обучаемой на гибридный просмотрово-кликовый регрессионный таргет с фичами, полученными из всех полученных векторных представлений (как меры близости), фичами популярности, новизны, контентными фичами тайтлов.

При обучении и валидации кандидатных моделей и итогового ранжирования используется хронологическое разбиение сформированного датасета во избежание data leaks.

2. Полка главной страницы «Вам понравится» формируется секвенциональной трансформерной моделью типа XLNet, обученной на предсказании следующего тайтла в последовательности по событиям просмотров и кликов на постеры. В итоговом ранжировании используются рейтинги (IMDB, Кинопоиск) и предсказание индивидуальной вероятности предпочтения от ранжируюшей модели полки «Популярное».

3. Холодные бейзлайны, показываемые на упомянутых выше полках пользователям без достаточной истории или без авторизации, формируются на основе популярных в последнем периоде и трендовых (резко набирающих популярность тайтлов) таким образом.

4. Из полок отфильтровываются пересечения между собой и находящимися выше главными редакторскими. Фильтруются также ранее просмотренные тайтлы.

5. Жанровые полки формируются засчет отбора тайтлов нужного жанра из следующих после отображаемых тайтлов «Вам понравится».

6. Полка «Похожее на X» формируется на основе выбора якорного тайтла из последних наиболее успешно просмотренных пользователем, и последующего поиска ближайших соседей в векторной БД Qdrant. Аналогичным образом формируются и рекомендации после просмотра в плеере (с учетом естественной логики порядка серий сериалов).

7. Переранжирование редакторских полок ведется сервисом, работающим в онлайн-режиме на основе модели градиентного бустинга из полки «Популярное», фичи для которой предрасчитываются, в частности, на основе векторных представлений кандидатных моделей и записываются в БД Postgre.

Потоки данных включая протоколы обмена и сценарии, включая альтернативные ветвления, с указанием акторов взаимодействий в виде диаграмм DFD, sequence.



*Рис. 4 Диаграмма DFD организации потоков данных от источников до сервиса API*

Формат хранения данных в витрине с информацией о просмотрах, непосредственно используемой для генерации рекомендаций, приведен в следующей таблице 1:

Таблица 1

| **Параметр** | **Пример** | **Описание** |
| --- | --- | --- |
| passport\_id | --5DvmO6RhSsUnD-uyIXa | Passmedia ID пользователя |
| tv\_id | 20834 | tv\_id тайтла |
| season\_num | 1 | Номер сезона сериала, для фильмов 1 |
| episode\_num  | 5 | Номер эпизода сериала, для фильмов 1 |
| user\_ep\_total\_view\_duaration\_s | 2629 | Время просмотра эпизода в секундах |
| user\_ep\_first\_view\_dt | 2023-12-11 19:22:33  | Старт просмотра эпизода |
| pdp\_created\_at  | 2023-12-12 09:04:36.381263 | Время создания записи |
| pdp\_updated\_at | 2023-12-13 09:04:28.11316 | Время актуализации записи |
| actual\_title\_ru | Последователи | Название тайтла |
| season\_duration\_s  | 16769 | Общая продолжительность эпизода в секундах |
| user\_seas\_total\_view\_duarat | 14557 | Суммарное время просмотра сезона в секундах |
| ion\_s  | 0.8680899278430437  | Доля просмотра сезона |
| user\_ep\_seconds\_from\_origin\_median | 10705105 | Медианное время в секундах с выхода эпизода до просмотра |
| user\_ep\_seconds\_from\_origi | 2642 | Медианное время в секундах между просмотрами эпизодов в сезоне |
| has\_next\_season  | false | Начал ли пользователь просмотр следующего сезона |
| user\_nextseas\_percent\_view | 0.8680899278430437 | Процент просмотра пользователем следующего сезона |
| ed actual\_tvtype\_name | series | Тип тайтла (фильм, сериал, шоу, трансляция) |

Таблица 2 - Поля витрины с информацией о метаданных каталога

| **Параметр** | **Описание** |
| --- | --- |
| absolute\_url |  |
| accessibility |  |
| age\_restriction |  |
| can\_subscribe |  |
| content |  |
| contenttvstype |  |
| countries |  |
| description |  |
| external\_ids |  |
| genres |  |
| global\_release |  |
| has\_allow\_download |  |
| id |  |
| is\_active |  |
| keywords |  |
| labels |  |
| last\_updated\_ts |  |
| last\_video\_add\_ts |  |
| local\_release |  |
| name |  |
| original\_title |  |
| picture |  |
| poster\_url |  |
| provider |  |
| related\_showcase |  |
| restriction\_notices |  |
| sensitive\_content |  |
| slogan |  |
| slug |  |
| smoking\_restriction |  |
| studios |  |
| type |  |
| uniform\_url |  |
| year |  |
| year\_end |  |
| year\_start |  |
| persons |  |
| studio |  |
| rightholder |  |
| is\_original |  |
| geo\_tag |  |
| available\_to\_geo\_RU |  |
| available\_to\_geo\_WW |  |
| available\_to\_age\_0 |  |
| available\_to\_age\_6 |  |
| available\_to\_age\_12 |  |
| available\_to\_age\_16 |  |
| available\_to\_age\_18 |  |
| is\_erotics |  |
| is\_test |  |
| item\_id |  |
| genre\_ids  |  |
| country\_ids |  |

Таблица 3 - Поля таблицы с информацией о кликах из Яндекс.Метрики

| **Параметр** | **Описание** |
| --- | --- |
| watchID |  |
| deviceCategory |  |
| counterID |  |
| date |  |
| dateTime |  |
| URL |  |
| clientID |  |
| lastTrafficSource |  |
| regionCountry |  |
| cid |  |
| session\_id |  |
| user\_id |  |
| user\_auth |  |
| user\_profile\_id |  |
| screen\_name |  |
| event\_group |  |
| event\_name |  |
| event\_category |  |
| event\_action |  |
| event\_label |  |
| event\_context |  |
| event\_json |  |
| user\_age\_category |  |
| channel\_id |  |
| content\_id |  |
| content\_category\_id |  |
| content\_subcategory\_id |  |
| content\_list\_position |  |
| content\_list\_id |  |
| content\_type |  |
| ab\_name |  |
| ab\_variant |  |
| user\_subscription |  |
| user\_subscription\_id |  |
| touch\_point |  |

Таблица 4 - Поля таблицы с информацией о показах постеров (impressions) из Яндекс.Метрики

| **Параметр** | **Описание** |
| --- | --- |
| date |  |
| clientID |  |
| impression\_id |  |
| url |  |
| sess\_num |  |
| json\_pass\_id |  |
| authorized |  |
| country |  |
| json\_object\_id |  |
| json\_response\_model\_tag |  |
| json\_items\_id |  |
| ua\_device\_type |  |

# Используемые языки программирования и фреймворки

| **Часть Сервиса** | **Языки программирования и фреймворки** |
| --- | --- |
| Data Engineering | Python PySpark Clickhouse Postgresql Dagster |
| Machine Learning | Python (implicit, RecTools, PyTorch, CatBoost) MlFlow |
| Backend  | Java, Postgresql |
| DevOps | Terragrunt, Helm, GitOps, ArgoCD, Yandex Cloud |

# Описание технической инфраструктуры

Основные компоненты инфраструктуры:

Облачная платформа: Yandex Cloud

Система виртуализации: Yandex Compute Cloud

Контейнеризация: Docker и Kubernetes для управления развертыванием сервисов

Балансировка нагрузки: Yandex Application Load Balancer и Yandex Network Load Balancer

Облачное хранилище: Yandex Object Storage для хранения данных и моделей

База данных: Yandex Managed Service for PostgreSQL, Redis (в контейнерах) и Qdrant для векторного поиска

Платформа машинного обучения: Yandex Data Sphere для обработки ML-моделей

Логирование:

Prometheus – сбор метрик производительности

Grafana – визуализация и анализ данных

Loki – сбор и анализ логов

Tempo – распределенный трейсинг

CI/CD: ArgoCD, Terragrunt и Helm для автоматического развертывания

Средства безопасности: Аутентификация и авторизация пользователей, защита API от SQL-инъекций и XSS-атак, шифрование данных при передаче и хранении

Механизмы отказоустойчивости: Автоматический перезапуск сервисов при сбоях, горизонтальное масштабирование, ежедневное резервное копирование баз данных

# 7. Эксплуатация системы

7.1 Запуск системы

Развертывание инфраструктуры через Terraform и Helm. Запуск контейнеров с использованием Kubernetes. Проверка доступности API и БД.

7.2 Управление

Управление параметрами и конфигурацией осуществляется через DevOps-инструменты (ArgoCD, GitOps). Настройки системы хранятся в конфигурационных файлах в Yandex Cloud.

7.3 Резервное копирование и восстановление

Автоматическое резервное копирование БД раз в сутки.

Возможность восстановления данных в течение 1 часа.

5.4 Обновление

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