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

**Модуль «Тегизация»**

**г. Москва 2025**

[1. Введение 3](#_heading=h.g7u99exsnl6z)

[2. Запуск модуля 3](#_heading=h.c4w7lwpe93g2)

[3. Основные функции модуля 3](#_heading=h.yvuq6v81w2kf)

[4. Работа с системой 4](#_heading=h.nkkr1we1zvui)

[5. Алгоритмы формирования тегов 5](#_heading=h.iotojvbp9pjs)

[6. Алгоритм формирования ответа на запрос 5](#_heading=h.q17yo57zlpja)

[6. Масштабируемость и производительность 5](#_heading=h.7lp38vwt6r2u)

[7. Завершение работы 6](#_heading=h.qdgy3awzeuj2)

##

## **1. Введение**

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.

# 2. 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.

# 3. 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.

# 4. Работа с системой

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/](https://uma.media/api/metainfo/tv/%24tv_id/)

# 5. 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.

# 6. 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.

# 6. 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.

# 7. 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.