

RFC для рекомендательной системы

Автор: Михаил Самбуев

Используемые понятия и сокращения

Рекомендательная система (RS) — алгоритм подбора наиболее подходящих вариантов общежитий и комнат

Контентная фильтрация — метод рекомендаций на основе характеристик студентов

Коллаборативная фильтрация — метод рекомендаций на основе истории взаимодействий студентов

Directus — headless CMS для управления данными

FastAPI — backend для API

Nuxt.js — фреймворк для создания универсальных Vue.js-приложений.

Аннотация

Данный документ описывает архитектуру, функциональные требования и спецификацию API для рекомендательной системы, предназначенной для подбора общежитий и комнат для студентов на основе их предпочтений, академических данных и других факторов.

Цели

- Разработать гибридную рекомендательную систему (контентная + коллаборативная фильтрация)
- Улучшить удобство и точность подбора мест проживания

Область применения

Студенты, желающие подать заявку на место в общежитии

Архитектура системы

Компоненты

База данных (PostgreSQL)

Микросервис рекомендаций (FastAPI)

Контентная и (коллаборативная) фильтрация

// Интеграция с Directus

Web-приложение (Nuxt.js)

UI для студентов

// Интеграция с Directus

Отображение рекомендаций

Функциональные требования

1. Хранение данных о студентах и об общежитиях
 - 1.1. Данные о студентах: персональные данные (ФИО, возраст, пол, курс обучения, факультет), предпочтения, хобби, языки, особые условия.
 - 1.2. Данные об общежитиях: адрес, тип общежития, количество блоков, комнат, мест, удобства.
2. Получение и обновление данных
3. Получение рекомендаций на основе профиля и предпочтений
 - 3.1. Формирование списка подходящих студентов
 - 3.1.1. Система должна анализировать схожесть профилей студентов
 - 3.1.2. Студенты одного пола должны попадать в один список рекомендаций
 - 3.2. Оценка совместимости
 - 3.2.1. Расчет схожести между студентами
4. Взаимодействие с Directus (с базой данных)
 - 4.1. Получение данных через Directus SDK
 - 4.2. Внесение рекомендаций в Directus
5. Интерфейс пользователя
 - 5.1. Ввод данных
 - 5.1.1. элемент управления — кнопка "Получить рекомендации"
 - 5.2. Отображение рекомендаций

5.3. Отображение ошибок (например, если данные для получения рекомендаций отсутствуют)

Алгоритм рекомендаций

Контентная фильтрация

Контентная фильтрация (Content-Based Filtering) основана на анализе характеристик студентов. Система сравнивает профиль студента с характеристиками студентов и рекомендует наиболее подходящие варианты.

Принцип работы:

- Векторизация профилей студентов
- Поиск ближайших студентов
- Рекомендация проживающих в свободных комнатах студентов, которые максимально соответствуют профилю студента.

Для оценки качества рекомендаций можно использовать:

Precision@K — доля правильных рекомендаций среди K лучших.

Mean Average Precision (MAP) — средняя точность по всем запросам

Коллаборативная фильтрация

Коллаборативная фильтрация (Collaborative Filtering) используется для поиска похожих студентов и рекомендаций наиболее подходящих общежитий и соседей на основе предпочтений других студентов. Она основывается на предположении, что студенты с похожими характеристиками и интересами предпочитают схожие условия проживания.

Методы для реализации данного алгоритма:

1. **User-Based Collaborative Filtering** - метод основан на поиске похожих студентов

Шаги:

Найти студентов с похожими профилями (интересы, возраст, образование, язык, предпочтения).

Вычислить метрику схожести между студентами (например, косинусное сходство или коэффициент Пирсона).

Использовать информацию о заселении схожих студентов для рекомендации общежитий и соседей.

2. **Item-Based Collaborative Filtering** – метод основан на поиске похожих общежитий

Шаги:

Анализировать, какие общежития выбирали похожие студенты.

Вычислить метрику схожести между общежитиями на основе их характеристик (комнаты, удобства, рейтинг и т. д.).

Рекомендовать общежития, которые выбирали студенты с похожими профилями.

Для оценки качества рекомендаций можно использовать:

Precision@K — доля правильных рекомендаций среди K лучших.

RMSE (Root Mean Squared Error) — Среднеквадратичная ошибка. Но для этого необходимы данные с рейтингами или оценками.

Для коллаборативной фильтрации необходимы исторические данные (история заселений)

Пока ещё думаю со своим преподавателем как можно реализовать и проверить алгоритм без реальных данных.

Развёртывание микросервиса рекомендательной системы

Создание Dockerfile

Создание docker-compose.yml

Запуск и тестирование