



# Urban**Umbra**

Умный Городской Гид

## Состав команды:



Петренко Валерия,  
Лилом Елизавета  
ML

Антонов Эдуард  
Анализ и  
подготовка данных

Лытов Дмитрий  
Фронтенд

Екатерина Чуйко  
Бэкенд



# Актуальность

Путешествия – это не просто перемещение из места в место, а увлекательное погружение в культуру и историю.

Создание рекомендательной системы на основе оценок туристических объектов поможет сделать посещение значимых мест в городах легкодоступным и увлекательным.



# Данные

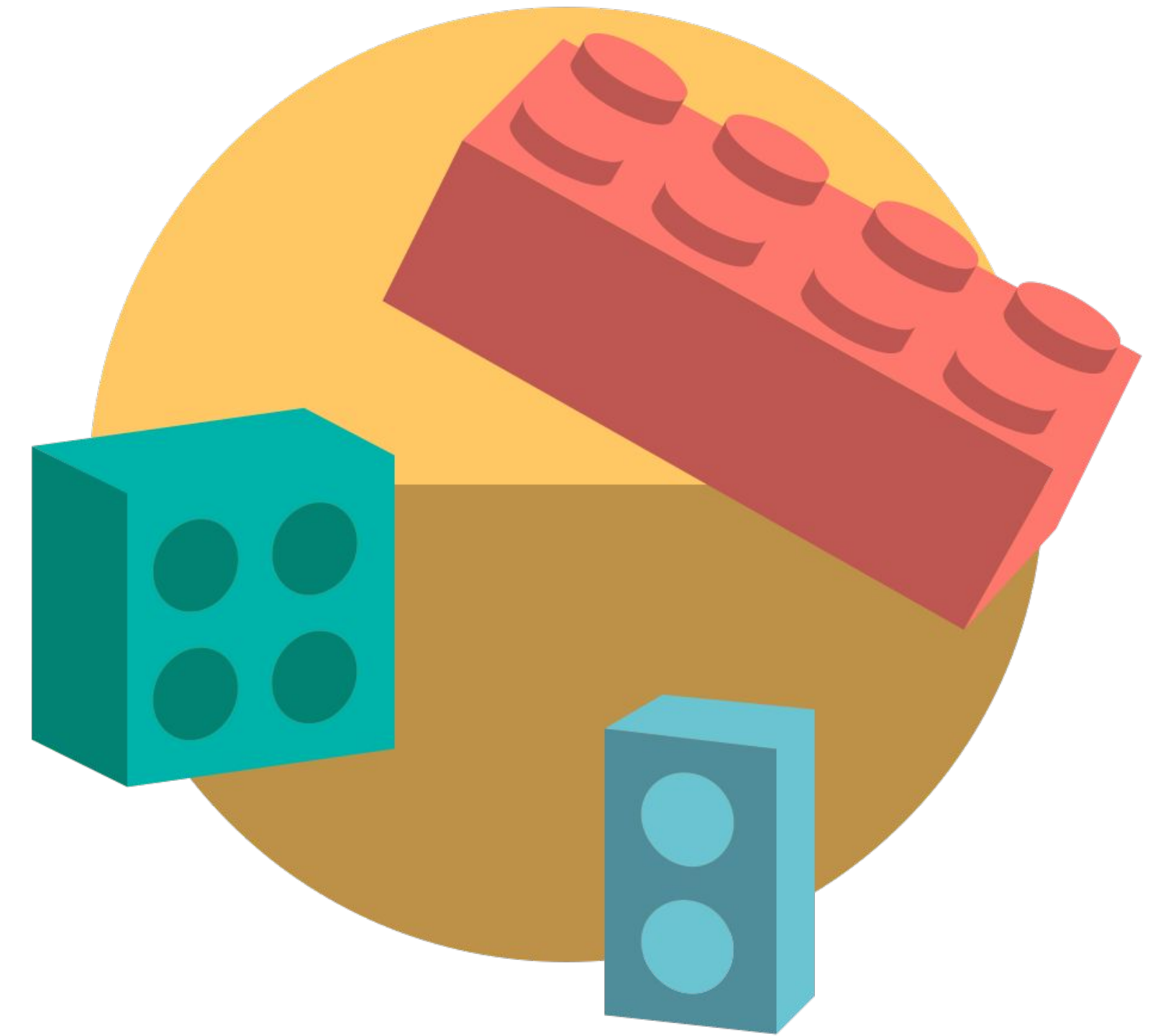
Выбранный нами датасет с платформы TripAdvisor содержит оценки и отзывы о местах в Южном Тироле, Италия. Эти данные помечены контекстными факторами: типом поездки, месяцем и годом.

Мы используем эти данные для персонализированных рекомендаций, учитывая предпочтения пользователей. Оценки и отзывы также позволяют нам анализировать места и предлагать информативные рекомендации.

 Link: [Dataset](#)

# Этапы работы

1. Анализ существующих продуктов и создание общей концепции.
2. Поиск данных для датасета.
3. Обработка исходных данных для получения необходимых признаков.
4. Разработка технического решения на каждой из сторон.
5. Реализация



# Анализ существующих решений

Мы изучили следующие сервисы, чтобы выявить тренд и потребность пользователей:

Имеют собственный сервис карт:

- **Google Maps:** предоставляет рекомендации по местам, ресторанам, магазинам и другим интересным местам на основе геолокации пользователя и истории его запросов.
- **Яндекс Карты/Путешествия:** предоставляют базовый набор рекомендаций, собранный заранее для некоторого набора городов.

Работают на основе других систем картографирования:

- **TripAdvisor:** использует ML для анализа и оценки отзывов и рейтингов, чтобы предлагать пользователям наиболее подходящие рекомендации на основе их предпочтений и истории поиска.
- **Островок.ру:** предоставляет сервис бронирования по поисковому запросу, на данный момент не имеет каких-либо систем для рекомендации на основе интересов пользователя

# Описание модели

Данная рекомендательная система использует коллаборативную фильтрацию на основе матрицы взаимодействий пользователей и объектов. Коллаборативная фильтрация - это технология, которая позволяет прогнозировать предпочтения пользователя, сравнивая его интересы с интересами других пользователей ресурса. На основании этой информации пользователю рекомендуются объекты, которыми интересовалась аудитория ресурса, а конкретный пользователь - ещё нет.

# Уже получилось:

ML	Бэкенд	Фронтенд
<ul style="list-style-type: none"><li>● Подготовка данных для построения рекомендаций</li><li>● Реализация коллаборативной фильтрации</li></ul>	<ul style="list-style-type: none"><li>● Запрос в Yandex Maps за координатами по названию места</li><li>● Запрос в ChatGPT за кратким описанием для каждого из мест</li><li>● Выполнить деплой бэка на сервер для внешнего доступа</li></ul>	<ul style="list-style-type: none"><li>● Предоставить пользователю выбор отеля</li><li>● Запрашивать рекомендуемые точки у сервера, опираясь на выбранный отель</li><li>● Отрисовывать полученные точки на яндекс карте</li><li>● При нажатии на точку выводить полученную от сервера информацию о ней</li><li>● Выполнить деплой фронта на github pages</li></ul>



# Какие боли возникли:

ML	Бэкенд	Фронтенд
<ul style="list-style-type: none"><li>● Недостаточно признаков в исходных данных для генерирования осмысленных рекомендаций по содержанию</li><li>● Ресурсные ограничения для сбора информации о взаимодействии пользователя с объектами</li><li>● Проблема “холодного старта” для коллаборативной фильтрации</li></ul>	<ul style="list-style-type: none"><li>●Yankees Maps не всегда отвечает точными координатами</li><li>●В бесплатной версии ChatGPT возможно делать не 3-х запросов* раз в 20 секунд</li></ul>	<ul style="list-style-type: none"><li>● Запрос очень долго выполняется</li><li>● Нет дизайна и компетенций в дизайне</li></ul>

# Что хотим выполнить дальше:

ML	Бэкенд	Фронтенд
<ul style="list-style-type: none"><li>● Рекомендации по контенту</li><li>● Индивидуальная настройка весов для рекомендаций</li><li>● Сбор и хранение истории взаимодействия пользователей и контента</li></ul>	<ul style="list-style-type: none"><li>● Настроить GitHub actions для автоматического деплоя</li><li>● Создать «секретницу» для ключей API, чтобы не подкладывать на сервер руками</li></ul>	<ul style="list-style-type: none"><li>● Декомпозировать API на несколько ручек, как минимум вынести запрос к GPT в другие, тк он занимает очень много времени, и подгружать их асинхронно</li></ul>

**Демонстрация:**

[https://kontrosha.github.io/urban\\_umbra/](https://kontrosha.github.io/urban_umbra/)

**Спасибо за внимание!**