



ЛИДЕРЫ ЦИФРОВОЙ
ТРАНСФОРМАЦИИ



Pegasus

Рекомендательная система для пользователей
московских библиотек



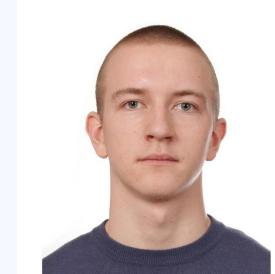
Захаров Сергей
@sergei3000
Backend



Исаев Кисмат
@kismat02
ML Engineer



Крашенинников Егор
@egorkrash
ML Engineer

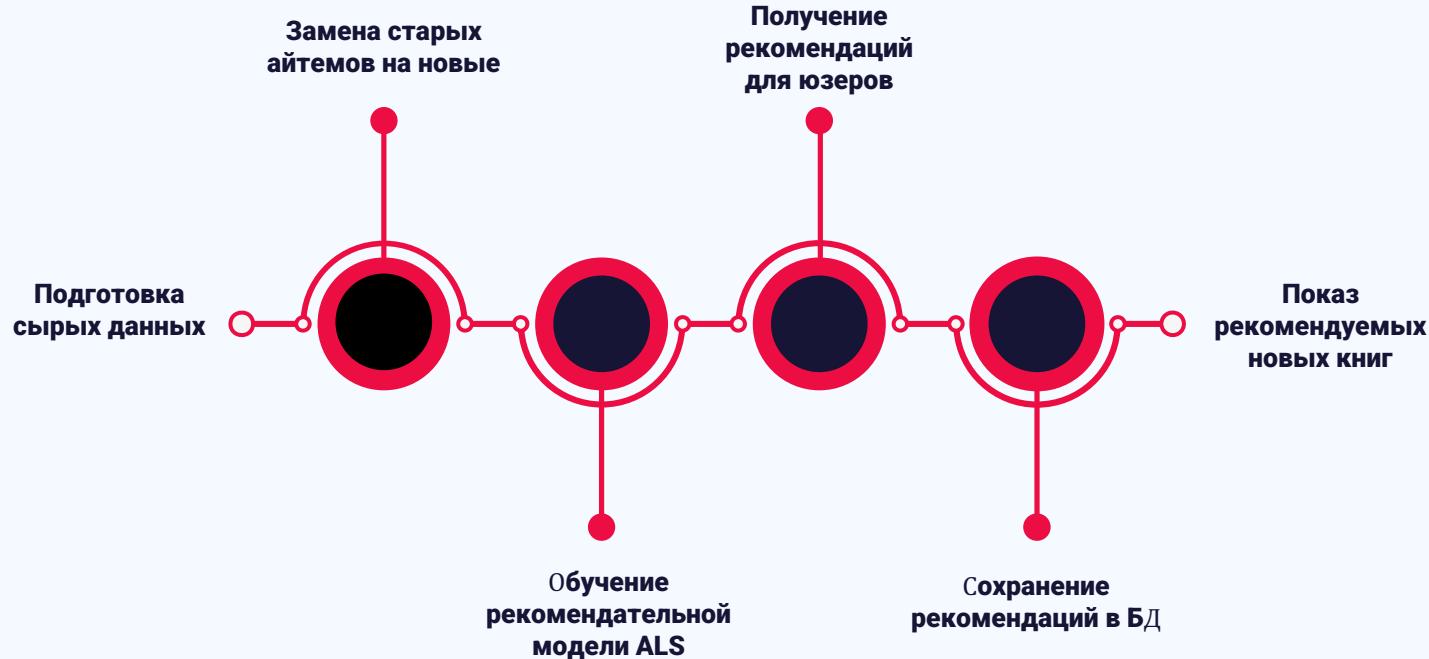


Назаров Николай
@nazarov_ni
Frontend

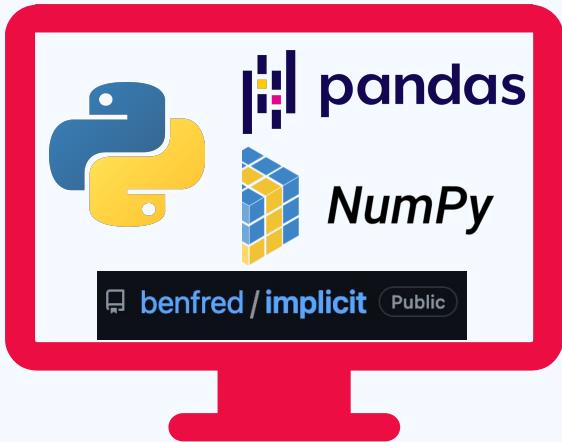




Схема работы алгоритма



Описание технологий



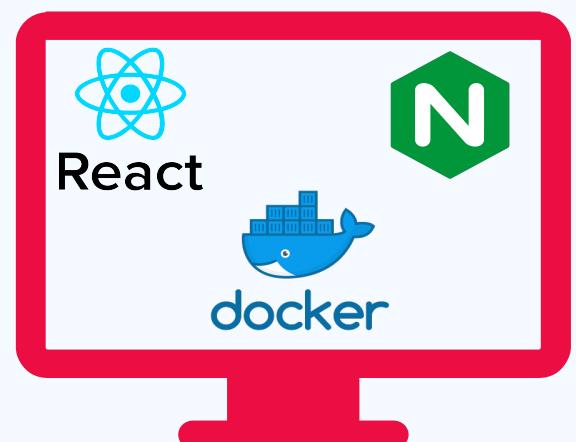
Recommender system

ALS модель как рекомендательный движок, написанный на Python



Backend

HTTP API фреймворк FastAPI + gunicorn как надежный HTTP сервер



Frontend

Написан на React с использованием Nginx в качестве веб-сервера. Обернут в Docker



Подготовка данных



Исходные данные

Данные представляют собой книгооборот в московских библиотеках за 2 года. Для построения рекомендательной системы нужно ввести три понятия: юзеры (пользователи), айтемы (книги) и рейтинг. В нашем случае явного рейтинга в данных нет, поэтому будем исходить из неявного взаимодействия – взял пользователь книгу или нет.



Подготовка айтемов

В исходных данных айтем представлял собой экземпляр книги. То есть одна и та же книга от одного автора могла иметь много разных id, что увеличивало разреженность данных и в рекомендации могли попадать одинаковые книги, но разные экземпляры. Для решения этих проблем исходные экземпляры книг были схлопнуты по автору и названию книги в единый новый айтем.

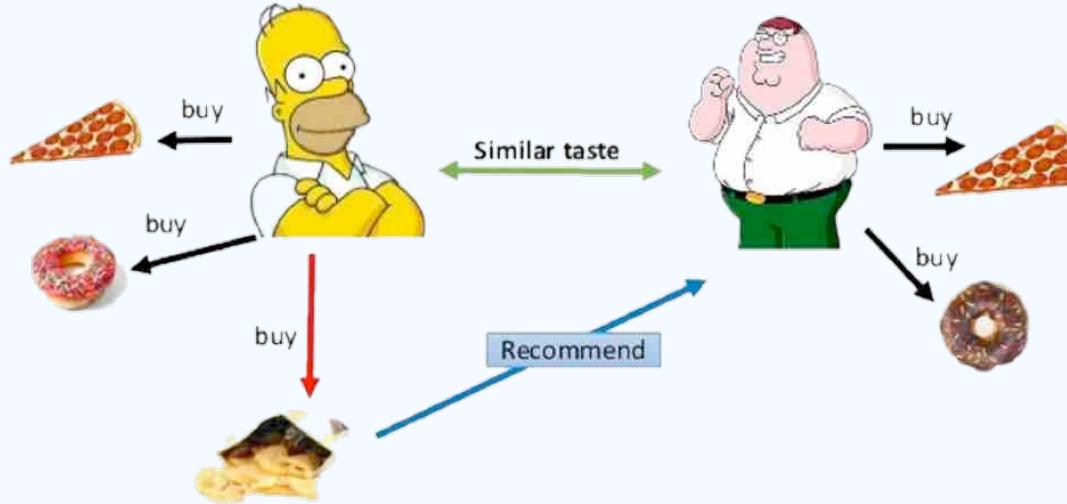


Авторы и названия

В справочниках о книгах и их авторах были пропуски. Они были заполнены значением “неизвестно”. Схлопыванию по автору и названию книги позволило сократить число айтемов вдвое, что улучшило качество итоговых рекомендаций.



Рекомендательный движок



Основной моделью рекомендаций является ALS – разложение матрицы взаимодействий пользователей и книг на 2 меньшие матрицы, произведение которых приблизит исходную матрицу.

Каждая строка в матрице юзеров – это представление интересов юзера в пространстве айтемов, столбец в матрице айтемов – тематическое представление по интересам юзеров.

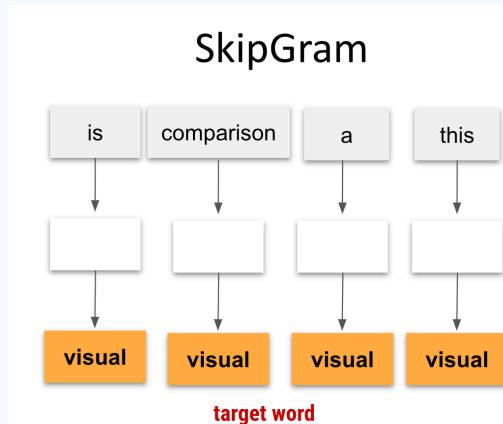
При помощи этих матриц можно искать похожих юзеров, близкие айтемы. Можно быстро рассчитывать представление нового юзера или айтема по небольшой истории в режиме онлайн.



Другой подход к рекомендациям

Item2Vec

Еще одна идея, которую мы реализовали – **Item2Vec** – нейросетевая модель, основанная на методе **skip-gram**, которая учится для слова (в данном случае слова – книги) предсказывать его контекст. Основная идея – выучить векторы книг (айтемов) аналогично тому, как **Word2Vec** выучивает векторы слов, имея множество примеров каждого слова в разных контекстах (под контекстом подразумеваются слова, стоящие рядом с данным в предложении). В данном случае контекст – книги, которые брал этот же пользователь. Получается, мы выучиваем векторы книг в зависимости от того, с какими книгами они берутся вместе. Далее считаем, что среднее векторов книг по истории пользователя – вектор пользователя, к которому можно найти ближайшие k книг по косинусной близости.





Backend



Мы разработали рассчитанный на высокую нагрузку асинхронный backend, построенный в соответствии с T3 хакатона



MySQL – одна из популярнейших opensource RDBMS (подключение - через aiomysql)



Gunicorn – надёжный HTTP сервер (запускает ASGI воркеры uvicorn для FastAPI app)

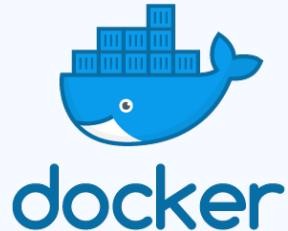


FastAPI – высокопроизводительный HTTP API фреймворк со встроенной валидацией, сериализацией и асинхронностью. Более того, это **автоматическая документация API**, доступная по ip:port/docs

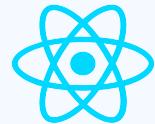


Frontend

Frontend - технологии



Для запуска фронтенда используется Docker. Docker позволяет быстро развертывать и масштабировать приложения в любой среде и сохранять уверенность в том, что код будет работать.



React

Приложение написано на React. React - JavaScript-библиотека для создания пользовательских интерфейсов. React предоставляет высокую скорость, простоту и масштабируемость.



В качестве веб-сервера используется Nginx. Nginx позволяет обрабатывать сотни тысяч единовременных подключений на одном физическом сервере.



Frontend интерфейс

Форма поиска пользователя

id пользователя

История пользователя

Введите id пользователя Search

user_id=72346

Рекомендации

rank	id	author	title
1	1282	Астафьев Виктор Петрович	Конь с розовой гривой
2	1350790	Пришвин Михаил Михайлович	Кладовая солнца
3	708268	Паустовский Константин Георгиевич	Заячий лапы
4	845023	Свифт Джонатан	Путешествия Гулливера
5	332326	Сент-Экзюпери Антуан де	Маленький принц

История

id	author	title
167349	Дойл Артур Конан	Затерянный мир
170815	Зощенко Михаил Михайлович	Рассказы для детей
1025036	Куприн Александр Иванович	Белый пудель
1954877	Житков Борис Степанович	Рассказы для детей
293648	Казаков Юрий Павлович	Арктур - гончий пес
957244	Успенский Эдуард Николаевич	Вниз по Волшебной реке
83273	Рыбаков Анатолий Наумович	Приключения Кроша ; Каникулы Кроша ; Неизвестный солдат
569430	Янссон Туве	Сказки про Муми-Тролля
473602	Паустовский Константин Георгиевич	Мещерская сторона
1225659	Хаггард Генри Райдер	Прекрасная Маргарет

Итоговые рекомендации были сделаны для всех **555 тысяч** пользователей.



Результат

Работа с данными

Мы очистили данные, схлопнули экземпляры одинаковых книг в единый айтем “книга + автор”

Рекомендательная система

Построили две системы: ALS (коллаборативная фильтрация) и Item2Vec (на основе word2vec)

Backend

Разработали рассчитанный на высокую нагрузку асинхронный backend

01



02



03



04



05



Frontend

Написали пользовательский интерфейс на React + Nginx, что предоставляет высокую скорость, простоту, масштабируемость позволяет обрабатывать сотни тысяч единовременных подключений на одном физическом сервере

Микросервис

Весь сервис работает на Yandex.Cloud с использованием OS Linux, версионирование кода поддерживается GitHub