МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

# Разработка рекомендательного сервиса для музыкального стримингового сервиса

**Рецензент:**
Начальник отдела суперкомпьютерного моделирования НИУ ВШЭ,
к.ф.-м.н., доцент
П.С. Костенецкий

**Научный руководитель:**
доцент кафедры СП,
к.ф.-м.н., доцент
Г.И. Радченко

**Автор:**
студент группы КЭ-229
А.Е. Колмаков

Челябинск, 2024 г.

# Актуальность

- Рынок музыкального стриминга демонстрирует заметный рост: аудитория увеличилась на 25% в течение 2023 года
- Рост пользовательского интереса к рекомендательным системам
- Рынок характеризуется высокой степенью конкуренции, с участием таких значимых игроков как «Яндекс Музыка», «VK Музыка», и «Звук» применяющие рекомендательные системы

# Цель и задачи исследования

**Цель:**

Разработка рекомендательного сервиса для музыкального стримингового сервиса.

**Задачи:**

1. Провести обзор существующих решений и методов
2. Описать требования и смоделировать систему
3. Подготовить и проанализировать исходные данные
4. Реализовать сервис для взаимодействия с обученной моделью
5. Провести тестирование реализованного сервиса

# Обзор существующих решений

**Проприетарные решения (в рамках сервисов):**

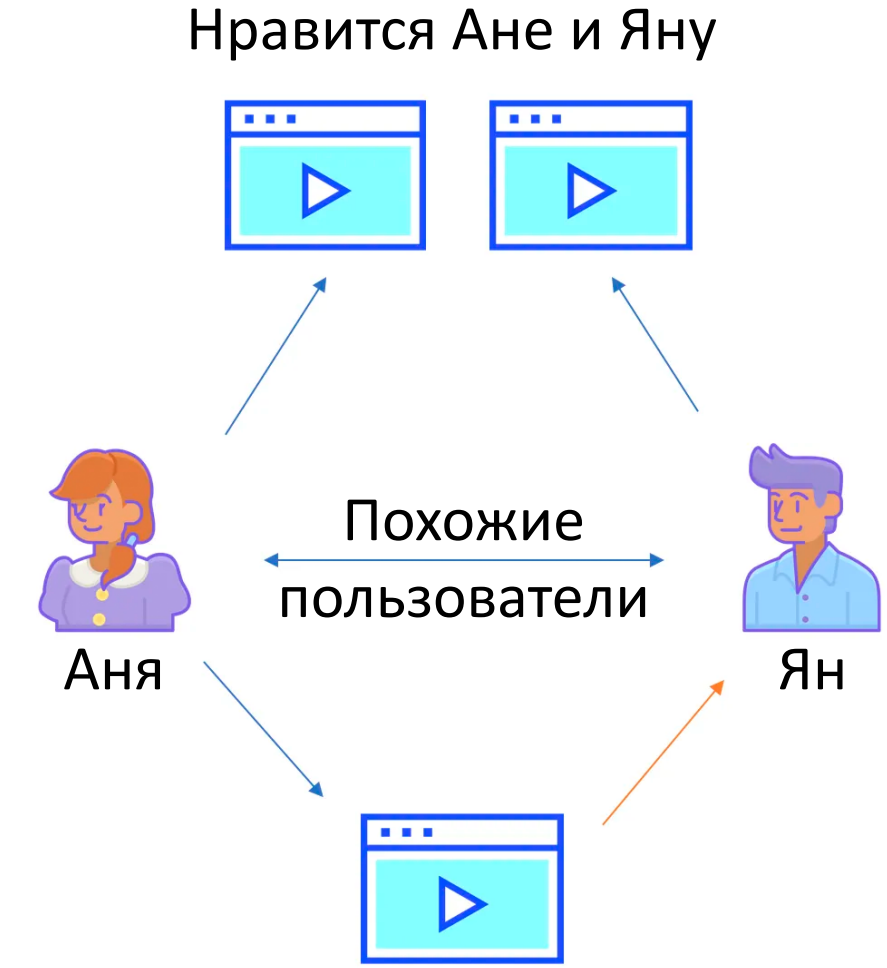
- Spotify
- Apple Music
- Яндекс Музыка
- Звук
- VK Музыка
- …

**Открытое программное обеспечение:**

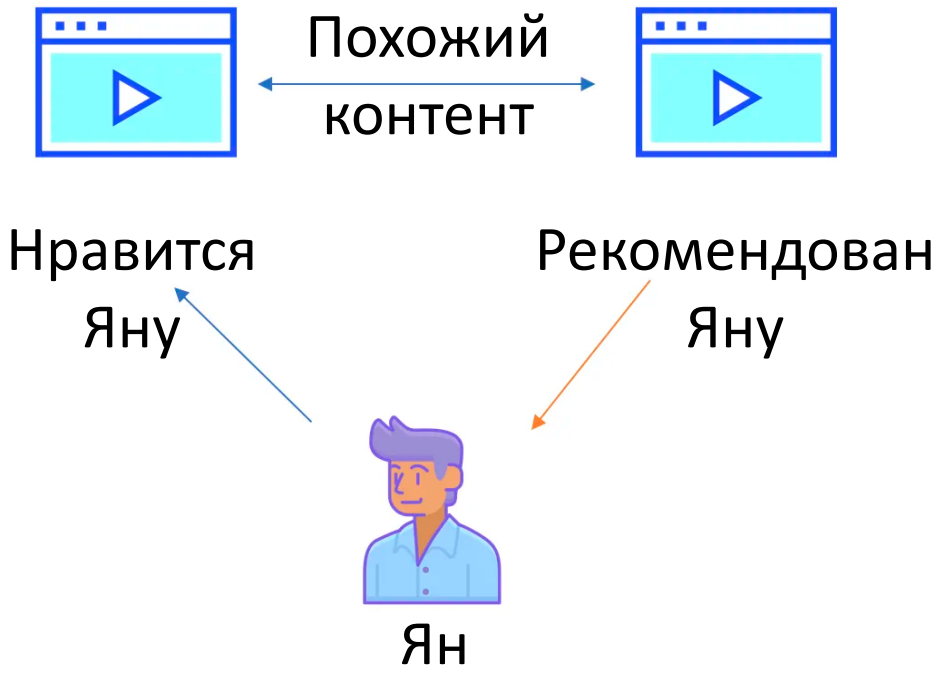
- Surprise
- LensKit

# Обзор существующих методов

*Коллаборативная фильтрация*




*Контентные методы*

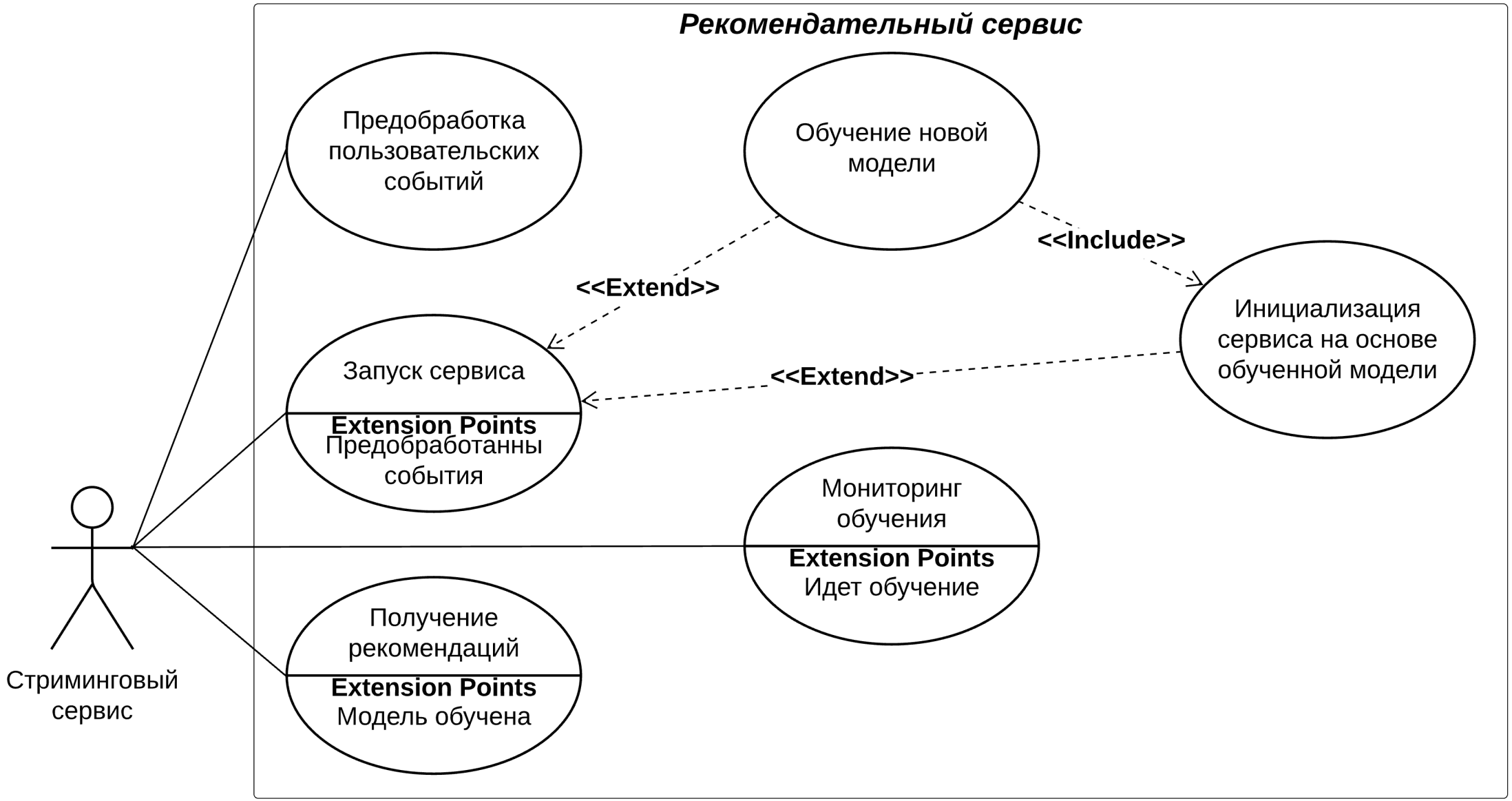


# Нефункциональные требования

1. Время отклика на запрос 60 рекомендаций: < 50 [мс]
2. Использование GPU для обучения модели
3. Время обучения модели: < 24 [часов] на NVIDIA RTX 3070
4. Микросервисная архитектура с взаимодействием через REST API
5. Применение контейнеризации: Docker

Требования основываются на входных данных, состоящих из 1,7 миллионов строк.

# Функциональные требования



# Анализ исходных данных

## Общая информация

| Файл | Описание | Количество записей |
|---|---|---|
| artists.csv | Таблица с артистами | 75 839 |
| authors.csv | Таблица с авторами | 57 125 |
| events.csv | Таблица с событиями | 147 499 237 |
| genres.csv | Таблица с жанрами | 388 |
| tracks.csv | Таблица с треками | 417 494 |

# Анализ исходных данных

Таблица с артистами

| Признак | Описание |
|---|---|
| artist_id | ID артиста |
| artist_name | Имя артиста |

# Анализ исходных данных

Таблица с авторами

| Признак | Описание |
|---|---|
| id | ID автора |
| name | Имя автора |
| union_id | Перечисление ID всех авторов (в случае нескольких авторов) |

# Анализ исходных данных

## Таблица с событиями

| Признак | Описание |
|---|---|
| trackId | Уникальный идентификатор трека |
| type | Тип события |
| personId | Уникальный идентификатор авторизированного пользователя |
| actionTypeContext | Причина возникновения события |
| initiatorContext | Место возникновения события |
| deviceTimestampUtc | Время на устройстве пользователя |
| artistId | Уникальный идентификатор пользователя |
| trackTimeMilliseconds | Время трека, на котором возникло событие (в миллесекундах) |
| durationMilliseconds | Длительность прослушивания трека (в миллисекундах) |
| countryIsoCode | Код страны в формате ISO 3166-1 |
| regionIsoCode | Код региона в формате ISO 3166-2 |

# Анализ исходных данных

Таблица с жанрами

| **Признак** | **Описание** |
|---|---|
| id | ID жанра |
| name | Название жанра |

# Анализ исходных данных

Таблица с треками

| Признак | Описание |
|---|---|
| TRACK_ID | ID трека |
| TRACK | Название трека |
| DURATION | Продолжительность трека в секундах |
| ARTIST | ID артиста |
| AUTHORS | ID авторов |
| GENRES | ID жанров |

# Анализ исходных данных

**Вывод:**

- Используем методы коллаборативной фильтрации

| | 🍈 | 🥫 | 🍍 | 🥝 | 🍅 | 🥕 | 🍔 |
|---|---|---|---|---|---|---|---|
| 👴 | 5 | | 4 | | 5 | | |
| 👵🏻 | 4 | 5 | | 5 | | 3 | |
| 👨🏾 | | 4 | | | | 5 | 4 |
| 👵🏿 | 5 | 5 | | | | | |

Необходимо составить матрицу взаимодействий

# Обработка исходных данных

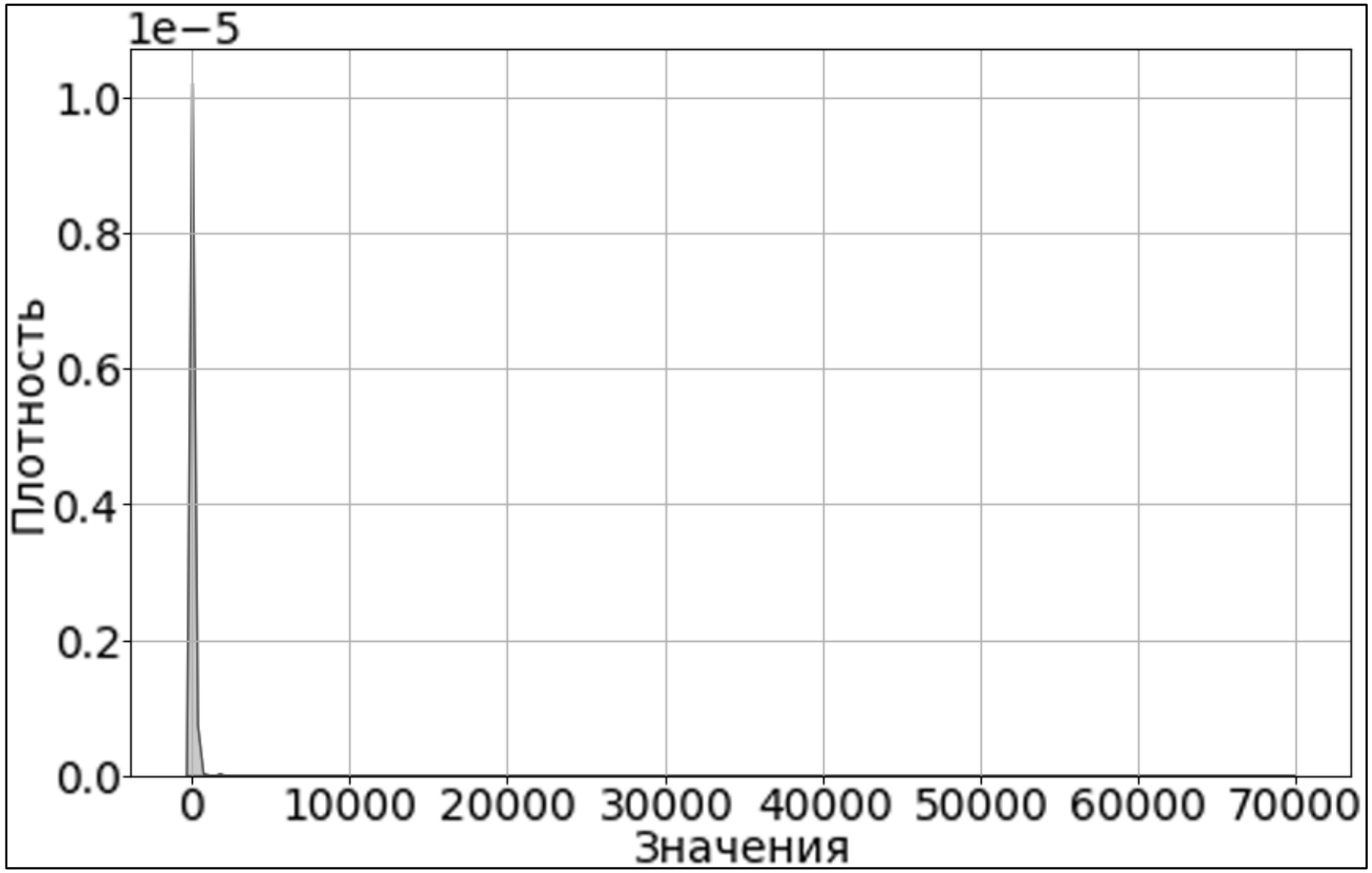
| Событие | Вес |
|---|---|
| STREAMING_END | От -0,3 до +0,7 |
| STREAMING_FULL_WITHOUT_REWIND | +1 |
| FAVORITE_ADD | +5 |
| NOT_SUGGEST | -5 |
| FAVORITE_REMOVE | -5 |
| UN_NOT_SUGGEST | +5 |

Вес события STREAMING_END рассчитывается по формуле:

$$weight_{STREAMING\_END} = \frac{eventDuration}{trackDuration} - 0{,}3$$

# Обработка исходных данных

## График плотности вероятности рейтингов до обработки



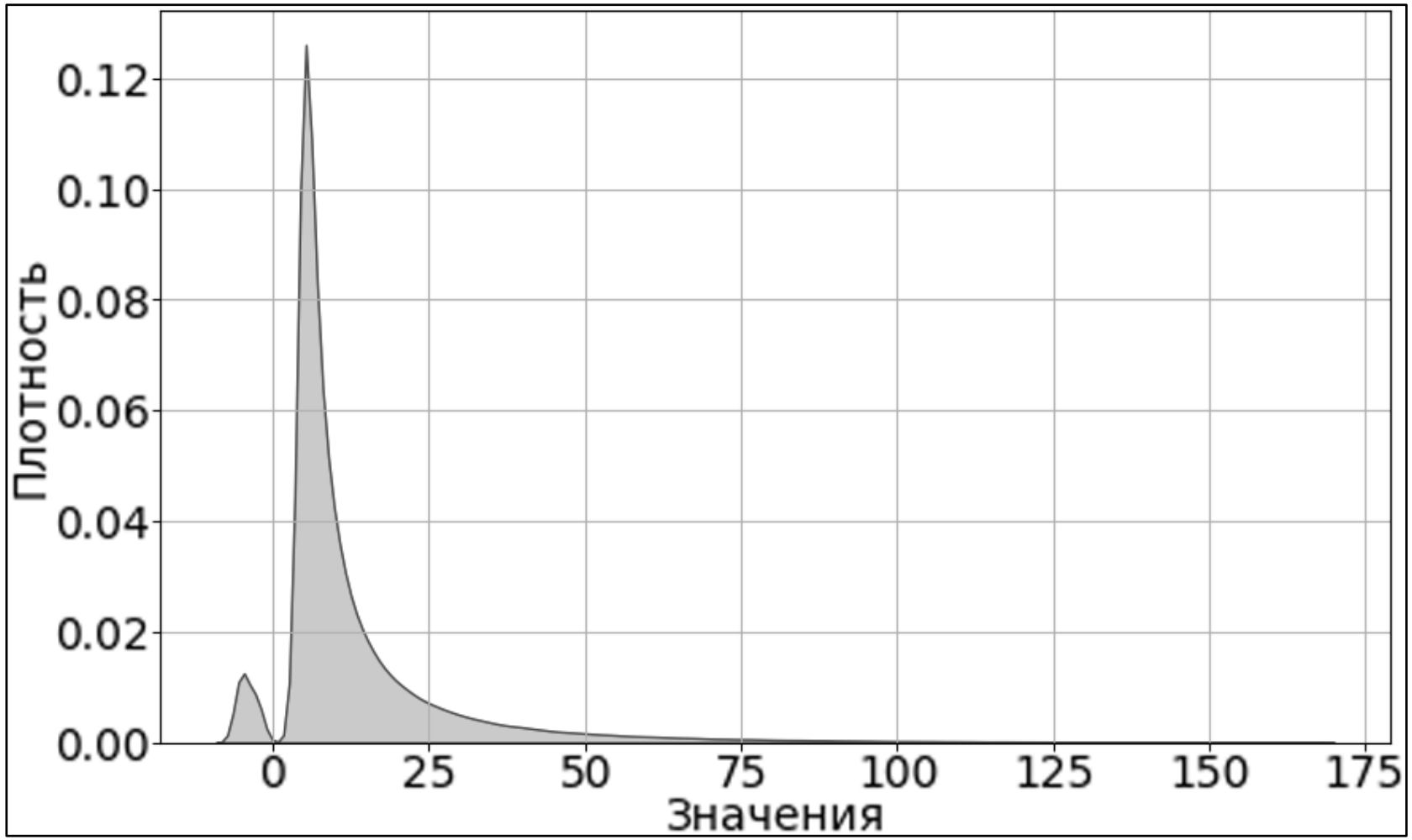

| Параметр | Рейтинг |
|---|---|
| count | 1,565386e+07 |
| mean | 2,032512e+00 |
| std | 2,328187e+01 |
| min | -3,651977e+02 |
| 25% | -2,743103e-01 |
| 50% | -3,636306e-02 |
| 75% | 1,700000e+00 |
| max | 6,999969e+04 |

# Обработка исходных данных

- Удалили рейтинги от -2 до 4
- Удалили крайние 0,5% самых низких и самых высоких оценок (квантили уровня 0,005 и 0,995)

# Обработка исходных данных

## График плотности вероятности рейтингов после обработки




| Параметр | Рейтинг |
|---|---|
| count | 1,709440e+06 |
| mean | 1,383649e+01 |
| std | 1,821466e+01 |
| min | -6,016488e+00 |
| 25% | 5,100000e+00 |
| 50% | 7,781010e+00 |
| 75% | 1,502485e+01 |
| max | 1,669990e+02 |

# Выбор модели коллаборативной фильтрации

Обученные модели:

- Alternating Least Squares (ALS)
- Bilateral Variational Autoencoder (BiVAE)
- Bayesian Personalized Ranking (BPR)
- FastAI Embedding Dot Bias (FastAI)
- Neural Collaborative Filtering (NCF)
- Simple Algorithm for Recommendation (SAR)

Andreas A., Miguel G., Le Z. Microsoft Recommenders: Best Practices for Production-Ready Recommendation Systems. // Companion Proceedings of the Web Conference, 2020. – 50–51 pp. DOI: 10.1145/3366424.3382692

# Выбор модели коллаборативной фильтрации

**Параметры датета:**

- Размер датасета: 3,5% от исходного
- Соотношение обучающей и тестовой выборки: 4/1
- Размер обучающей выборки: 136 725
- Размер тестовой выборки: 45 576

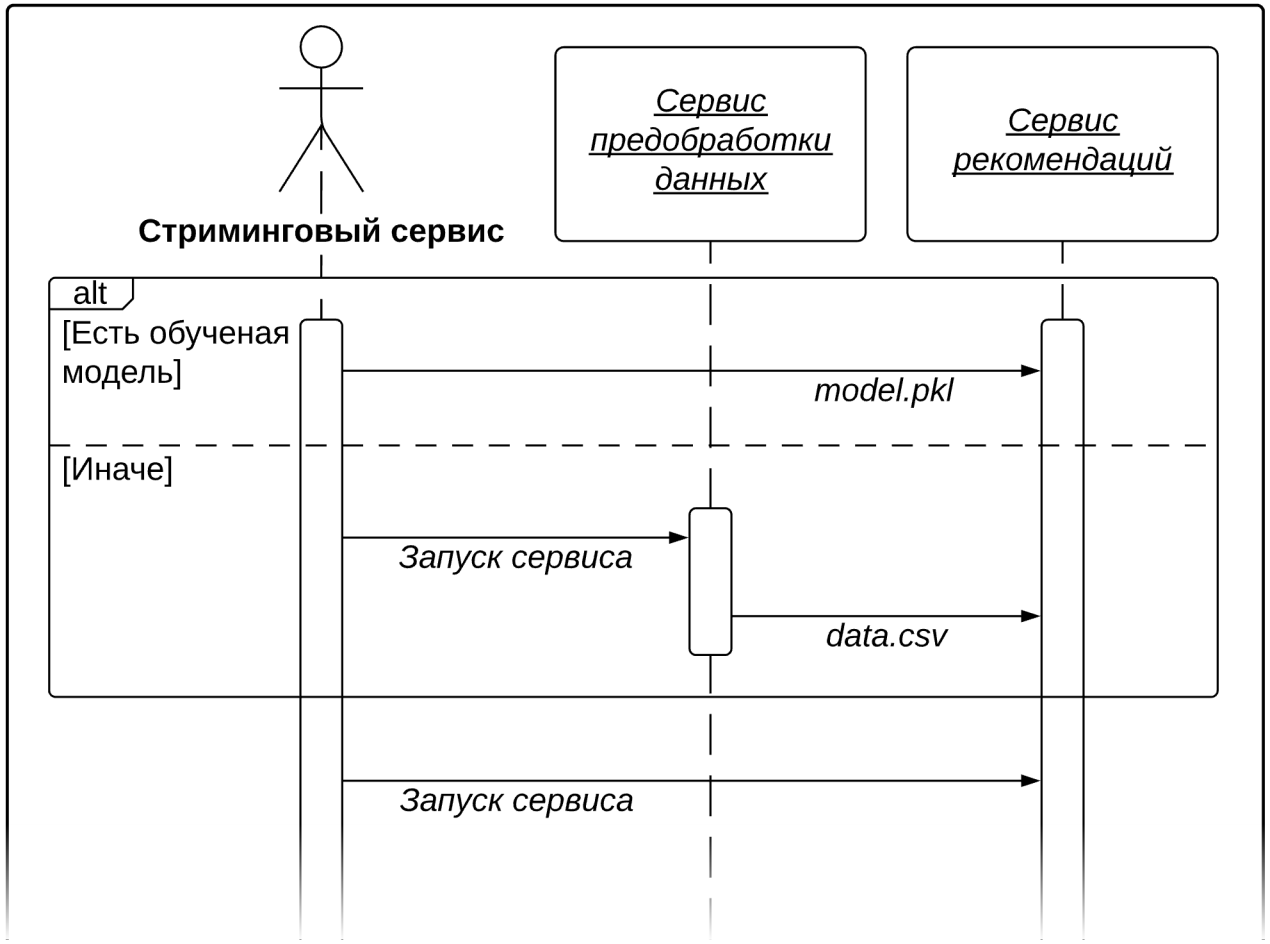
**Метрики (k=10):**

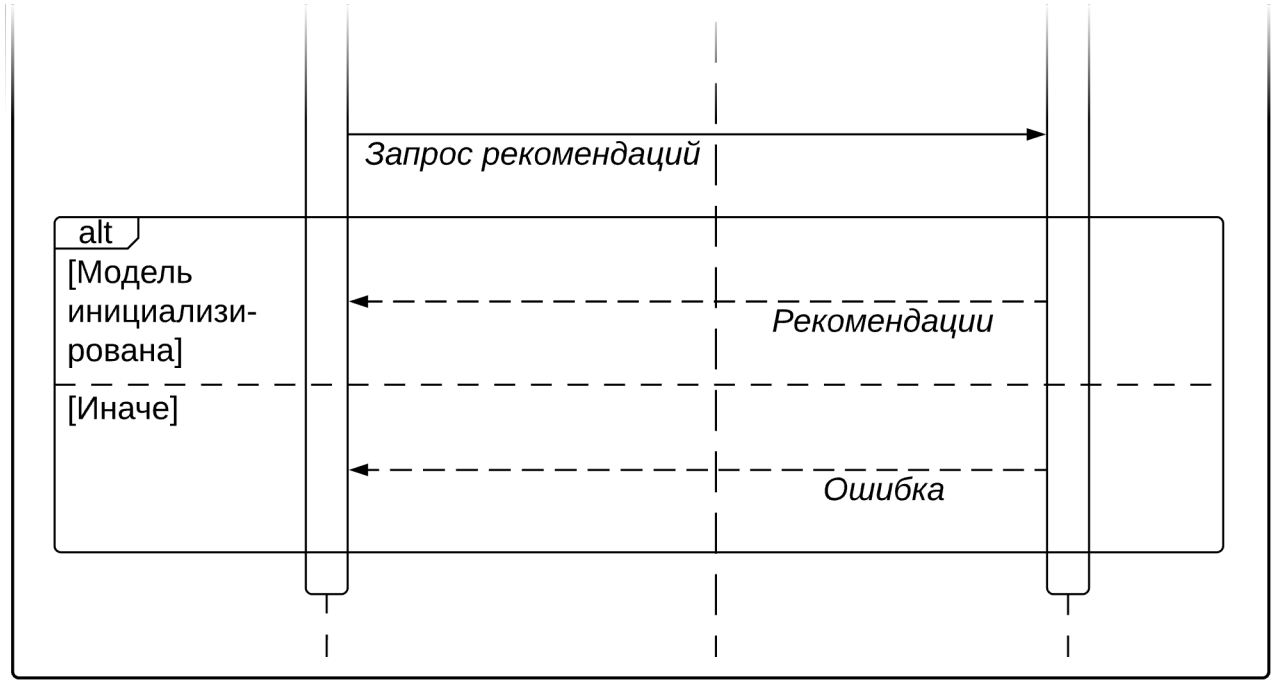
- MAP (Mean Average Precision): Средняя точность по всем запросам
- nDCG@k (normalized Discounted Cumulative Gain at k): Эффективность ранжирования с учетом полезности и позиции до k-го места
- Precision@k: Доля релевантных элементов среди первых k
- Recall@k: Доля релевантных элементов в первых k, относительно всех релевантных элементов

# Выбор модели коллаборативной фильтрации

| Модель | MAP | nDCG @k | Precision @k | Recall @k | Время обучения, [с] | Время предсказания, [с] |
|---|---|---|---|---|---|---|
| BiVAE | 0.10554 | 0.17291 | 0.10902 | 0.19227 | 368.32 | 36.12 |
| SAR | 0.08201 | 0.14547 | 0.08227 | 0.15915 | 3.13 | 1.34 |
| BPR | 0.05159 | 0.09739 | 0.06684 | 0.11617 | 1.18 | 38.72 |
| FastAI | 0.04638 | 0.08345 | 0.05432 | 0.10162 | 18.52 | 33.45 |
| NCF | 0.04118 | 0.08520 | 0.05221 | 0.10073 | 7,968.79 | 177.27 |
| ALS | 0.00013 | 0.00028 | 0.00016 | 0.00018 | 6.38 | 55.69 |

# Моделирование сервиса



# Реализация сервиса

**Сервис предобработки данных:**

- Реализуется индивидуально для каждого стримингового сервиса. Должен возвращать предобработанный CSV файл

**Сервис рекомендаций:**

- Отвечает за обучение модели (cornac.models.bivaecf)
- Отвечает отдачу рекомендаций (FastAPI GET Endpoint)
- Запускается в Docker контейнере с настроенным CUDA окружением и необходимыми библиотеками

# Работа сервиса

```
2024-05-07 03:21:24 initializing a model from a file...
2024-05-07 03:21:24 initializing a new model [GPU: True]...
2024-05-07 03:21:24 saving model...
100%|          | 500/500 [9:43:35<00:00, 70.03s/it, loss_i=0.521, loss_u=0.994]
2024-05-07 13:05:50 INFO:     Started server process [1]
2024-05-07 13:05:50 INFO:     Waiting for application startup.
2024-05-07 13:05:50 INFO:     Application startup complete.
2024-05-07 13:05:50 INFO:     Uvicorn running on http://0.0.0.0:80 (Press CTRL+C to quit)
2024-05-15 03:21:13 INFO:     Shutting down
2024-05-15 03:21:13 INFO:     Waiting for application shutdown.
2024-05-15 03:21:13 INFO:     Application shutdown complete.
2024-05-15 03:21:13 INFO:     Finished server process [1]
```

# Работа сервиса

192.168.0.2

GET /recommendations/ Get Recommendations

Parameters Cancel

| Name | Description |
|---|---|
| user_id * required<br>integer<br>(query) | 52168275 |
| track_count<br>integer<br>(query) | 60 |
| remove_known<br>boolean<br>(query) | false |

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://192.168.0.2/recommendations/?user_id=52168275&track_count=60&remove_known=false' \
  -H 'accept: application/json'
```

Request URL

```
http://192.168.0.2/recommendations/?user_id=52168275&track_count=60&remove_known=false
```

Server response

Code Details

200

Response body

```
[
  24768474,
  11803242,
```

# Функциональное тестирование

| Дейвствие | Ожидаемое поведение | Пройден |
|---|---|---|
| Предобработка данных | Получение предобработанного датасета в формате CSV | Да |
| Сборка Docker образа | Docker образ собирается | Да |
| Запуск Docker контейнера с предобработанными данными | Docker контейнер запускается. Начинается процесс обучения модели. По завершению обучения становится доступен GET метод REST API для получения рекомендаций | Да |
| Запуск Docker контейнера с обученной моделью | Docker контейнер запускается. Модель инициализируется обученной моделью. По завершению инициализации модели становится доступен GET метод REST API для получения рекомендаций | Да |

# Функциональное тестирование

| Дейвствие | Ожидаемое поведение | Пройден |
|---|---|---|
| Мониторинг процесса обучения | В консоль Docker контейнера выводится прогресс обучения | Да |
| Получение рекомендаций для существующего пользователя | Ответ с кодом 200 со списком рекомендованных треков размером 60 | Да |
| Получение рекомендаций для не существующего пользователя | Ответ с кодом 500 с описанием ошибки | Да |
| Получение ограниченного числа рекомендаций (параметр track_count = 10) | Ответ с кодом 200 со списком рекомендованных треков размером 10 | Да |
| Получение рекомендаций без известных пользователю треков (параметр remove_known = True) | Ответ с кодом 200 со списком рекомендованных треков. Список может содержать ранее прослушанные треки | Да |

# Нефункциональное тестирование

| Параметр | Ожидаемый результат | Итоговый результат |
|---|---|---|
| Время отклика (получения рекомендаций) | 50 миллисекунд | 30 миллисекунд |
| Время обучения модели | Менее 24 часов | 9 часов 45 минут |
| Использование графических ускорителей для обучения модели | Используется | Используется |
| Микросервисная архитектура с использованием REST API | Используется | Используется |
| Применение Docker контейнеризации | Используется | Используется |

# Основные результаты

1. Проведен анализ предметной области
2. Проанализированы требования и исходные данные, на основании которых было произведено проектирование архитектуры рекомендательного сервиса
3. Исследованы и обучены рекомендательные модели. На базе наиболее результативной модели была реализована рекомендательная система
4. Произведено тестирование разработанного рекомендательного сервиса

Разработанный сервис предложен к внедрению в рамках стримингового сервиса «Zaycev.net»