



Факультет компьютерных наук

Науки о данных

Москва 2024

# Построение системы музыкальных рекомендаций при помощи методов машинного обучения

Выполнено студентом группы МНОД231  
Шкляром Михаилом Игоревичем

Научный руководитель:  
Доцент департамента Анализа данных и  
искусственного интеллекта,  
Николенко Сергей Игоревич

# Описание предметной области

**Определение:** Системы музыкальных рекомендаций — это технологии, которые анализируют предпочтения и поведение пользователей для предложения музыкального контента, который может им понравиться. Эти системы используют различные алгоритмы для анализа музыкальных треков, жанров и поведенческих данных.

**Применение:** Эти системы широко применяются в музыкальных стриминговых сервисах, таких как Spotify, Apple Music и Яндекс.Музыка, где они помогают пользователям находить новую музыку в соответствии с их вкусами и предыдущей активностью прослушивания.

**Значение:** Системы рекомендаций улучшают пользовательский опыт, увеличивают продолжительность прослушивания и способствуют увеличению пользовательской базы и доходов сервисов за счет предоставления персонализированного контента.





# Цели и задачи проекта

**Цель работы:** Разработка эффективной системы музыкальных рекомендаций, способной предсказывать музыкальные жанры с высокой точностью и предоставлять пользователям персонализированные музыкальные рекомендации в реальном времени.

## Задачи:

- **Анализ существующих решений** в области музыкальных рекомендаций для определения недостатков и возможностей улучшения.
- **Определение необходимых функций системы**, включая методы и модели машинного и глубокого обучения, которые могли бы быть использованы для анализа и классификации музыкальных треков.
- **Создание и обучение различных моделей машинного и глубокого обучения**, включая CatBoost, XGBoost, LightGBM, SVM и нейронные сети, для точного предсказания музыкальных жанров на основе аудио характеристик.
- **Разработка и интеграция веб-приложения и телеграм-бота**, которые позволяют пользователям вводить названия треков или ссылки на Spotify и получать музыкальные рекомендации.

## Актуальность работы

- Музыкальная индустрия активно развивается, требуя новых подходов к персонализации контента.
- Повышение интереса к музыкальным рекомендациям среди пользователей.
- Необходимость интеграции современных технологий машинного и глубокого обучения для улучшения качества рекомендаций.
- Глобализация музыкального контента и потребность в универсальных решениях, способных адаптироваться под разные культурные контексты.





## Анализ конкурентов

Функция/Характеристика	Яндекс Музыка	Spotify	Разработанная система
Интеграция с видеоплатформами	Нет	Нет	Да
Доступ ко всему мировому репертуару	Нет	Да	Да
Интеграция с Телеграм-ботом	Нет	Нет	Да
Географические и лицензионные ограничения	Нет	Нет	Нет
Начало рекомендаций с выбранной песни	Нет	Нет	Да
Возможность сохранять музыку в свои плейлисты	Да	Да	Нет



## Анализ публикаций

Публикация	Основные идеи	Ограничения	Выводы
"Deep Content-based Music Recommendation"	Использование сверточных нейронных сетей (CNN) для анализа спектрограмм музыкальных треков. Высокая точность предсказаний на основе содержания треков.	Не учитывает социальные и контекстные аспекты музыкального восприятия, такие как популярность треков или сезонные предпочтения слушателей.	Комбинирование содержательного анализа и других подходов улучшит точность рекомендаций.
"Collaborative Filtering for Music Recommendation"	Применение методов коллаборативной фильтрации для выявления скрытых паттернов в музыкальных предпочтениях на основе пользовательских оценок.	Проблемы масштабирования, холодного старта и разреженности данных.	Гибридные системы, объединяющие коллаборативную фильтрацию с контент-ориентированными подходами, улучшают качество рекомендаций.





## Функциональные требования

- Минимальная точность 0,66 при предсказании жанра
- Разнообразие моделей (машинное обучение и глубокое обучение)
- Продвинутая инженерия признаков
- Тщательная проверка и тестирование
- Оптимизация гиперпараметров
- Использование различных метрик оценки
- Интеграция с Telegram, Spotify и YouTube
- Разработка API





## Стек технологий

- **Язык программирования - Python:**  
(pandas, matplotlib, seaborn, sklearn, xgboost, catboost, tensorflow)
- **Веб-фреймворк:** Flask
- **Обработка данных:** Pandas
- **Сериализация моделей:** Joblib
- **Тестирование и взаимодействие с API:** Postman, cURL
- **Разработка и тестирование кода:**  
Jupyter Notebook, Visual Studio Code







# Сбор данных

## Парсинг данных со Spotify

- Выбранные жанры: Hip-hop, K-pop, Metal, Indian, Classical, Blues
- Использование Python и Spotify API для сбора данных
- Функция `collect_tracks` для поиска треков по жанрам
- Лимит: 5000 треков для каждого жанра
- Извлечение подробной информации и аудио-характеристик треков
- Объединение данных в единый датасет

		Unnamed: 0	track_id	genre_track	duration_ms	explicit	danceability	energy	key
13220	13219	13236	JrMHTP3FfqsqnIRg0FM	classical	246840.0	False	0.623	0.861	0.0



# Преппроессинг

## Применение StandardScaler:

- Стандартизация данных для улучшения работы моделей
- Среднее значение = 0, стандартное отклонение = 1

## Попытка логарифмирования:

- Снижение асимметрии распределения признаков
- В данном случае не привело к улучшению

## Удаление выбросов:

- Использование метода межквартильного размаха (IQR)
- Удаление значений, выходящих за пределы 1.5 межквартильного диапазона
- Повышение качества данных для анализа



## Анализ данных

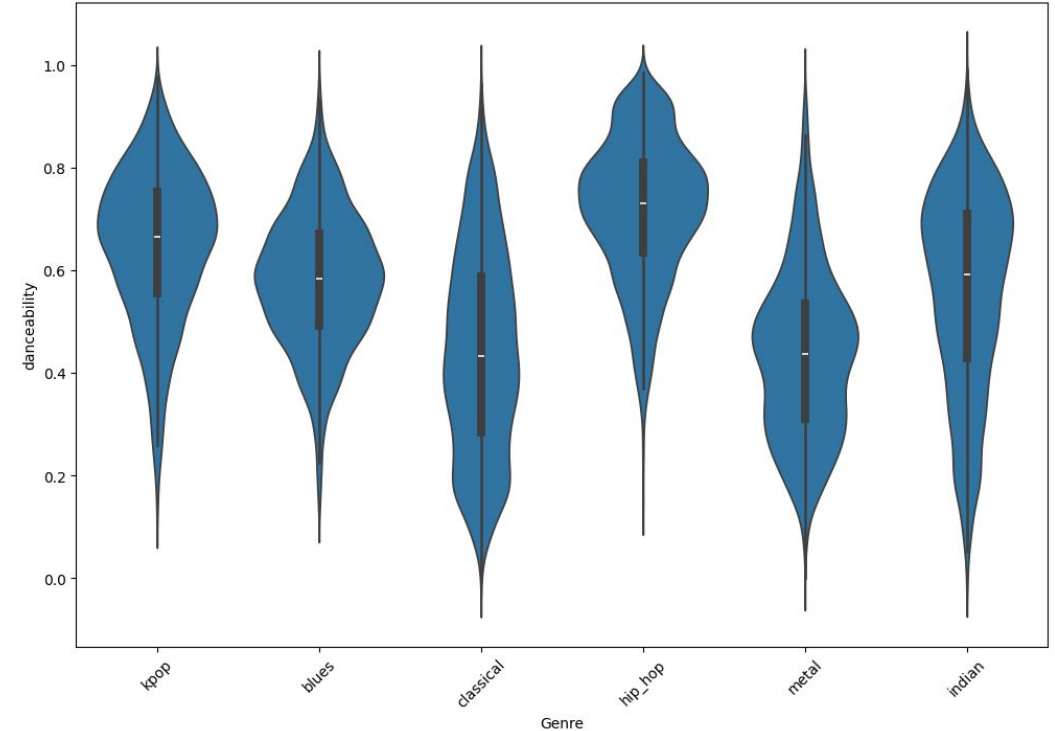
genre_track	1	-0.067	0.033	-0.1	-0.32	-0.0053	-0.26	0.079	-0.077	0.31	0.081	-0.027	-0.035	-0.064	-0.1
duration_ms	-0.067	1	0.083	-0.17	-0.0051	0.006	-0.039	0.021	-0.073	0.052	-0.029	0.067	-0.12	-0.024	-0.0095
explicit	0.033	0.083	1	-0.24	-0.16	0.013	-0.23	0.01	-0.4	0.24	0.25	-0.048	-0.016	-0.021	-0.083
danceability	-0.1	-0.17	-0.24	1	0.16	0.029	0.34	-0.039	0.19	-0.24	-0.29	-0.12	0.52	0.0037	0.19
energy	-0.32	-0.0051	-0.16	0.16	1	0.046	0.82	-0.064	0.19	-0.77	-0.38	0.23	0.29	0.23	0.18
key	-0.0053	0.006	0.013	0.029	0.046	1	0.051	-0.17	0.029	-0.029	-0.04	0.0048	0.062	0.016	-0.00031
loudness	-0.26	-0.039	-0.23	0.34	0.82	0.051	1	-0.029	0.15	-0.66	-0.52	0.15	0.34	0.21	0.17
mode	0.079	0.021	0.01	-0.039	-0.064	-0.17	-0.029	1	-0.051	0.069	0.001	-0.0021	-0.0014	-0.0098	-0.023
speechiness	-0.077	-0.073	-0.4	0.19	0.19	0.029	0.15	-0.051	1	-0.19	-0.22	0.11	0.059	0.073	0.072
acousticness	0.31	0.052	0.24	-0.24	-0.77	-0.029	-0.66	0.069	-0.19	1	0.36	-0.14	-0.17	-0.22	-0.17
instrumentalness	0.081	-0.029	0.25	-0.29	-0.38	-0.04	-0.52	0.001	-0.22	0.36	1	-0.11	-0.27	-0.092	-0.091
liveness	-0.027	0.067	-0.048	-0.12	0.23	0.0048	0.15	-0.0021	0.11	-0.14	-0.11	1	0.019	0.033	0.027
valence	-0.035	-0.12	-0.016	0.52	0.29	0.062	0.34	-0.0014	0.059	-0.17	-0.27	0.019	1	0.12	0.11
tempo	-0.064	-0.024	-0.021	0.0037	0.23	0.016	0.21	-0.0098	0.073	-0.22	-0.092	0.033	0.12	1	0.021
time_signature	-0.1	-0.0095	-0.083	0.19	0.18	-0.00031	0.17	-0.023	0.072	-0.17	-0.091	0.027	0.11	0.021	1
	genre_track	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature



### Heatmap:

- Положительная корреляция: громкость и энергия
- Отрицательная корреляция: акустичность и энергия
- Танцевальность и валентность

Distribution of danceability Across Different Genres



### Violin Plot:

- Высокая танцевальность: K-pop, Hip-Hop
- Низкая танцевальность: Classical
- Средняя танцевальность: Blues, Metal, Indian

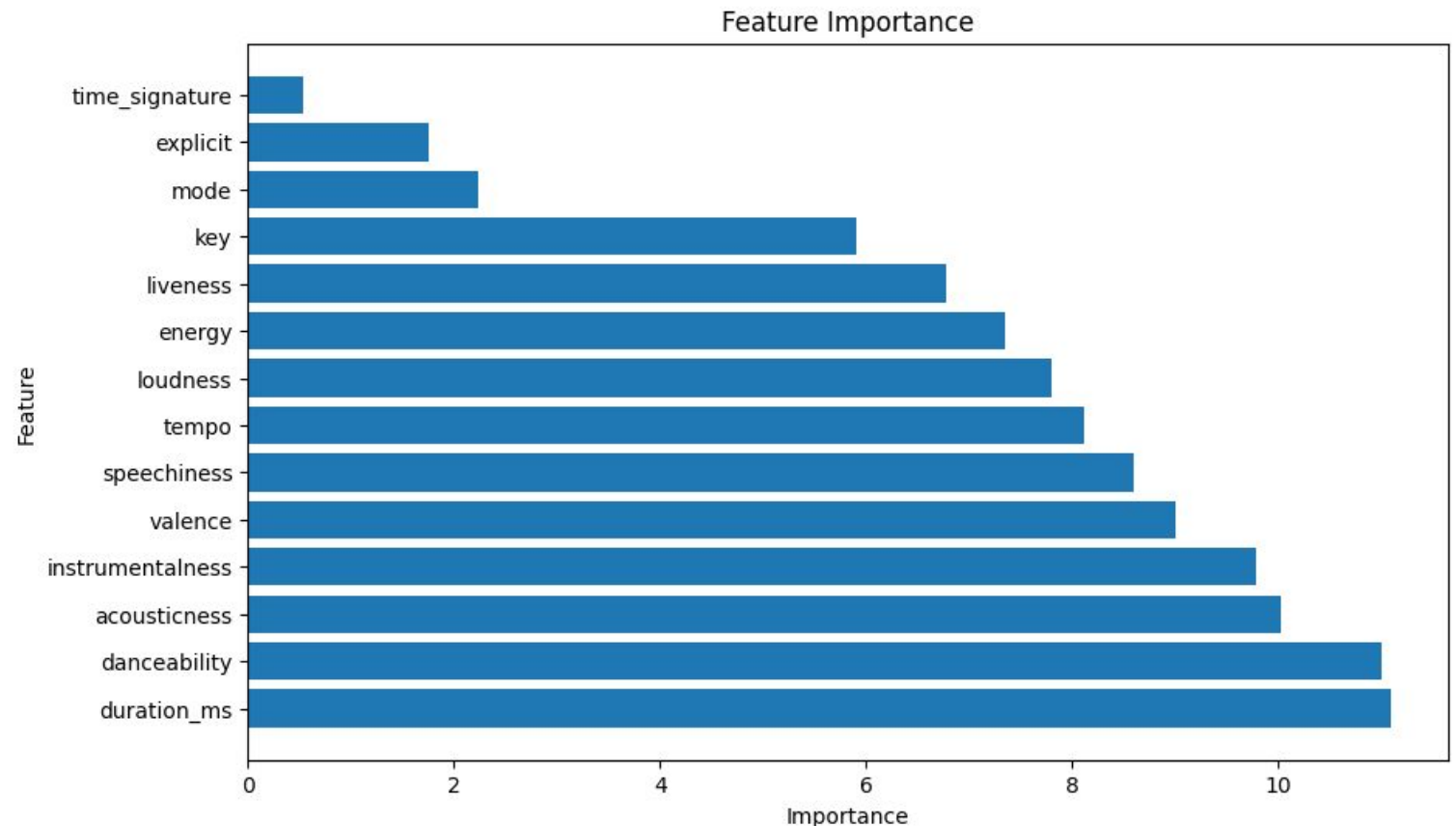
# Машинное обучение. XGBoost + feature importance

## Настройка и оценка модели:

- Гиперпараметры: `n_estimators`, `learning_rate`, `max_depth`
- Использование `GridSearchCV` для выбора лучших параметров
- Точность модели на тестовой выборке: 68%

## Важность признаков:

- `Duration_ms` (длительность трека)
- `Danceability` (танцевальность)
- Наименьшее влияние:  
`time_signature` (музыкальный размер) и `explicit` (наличие нецензурной лексики)



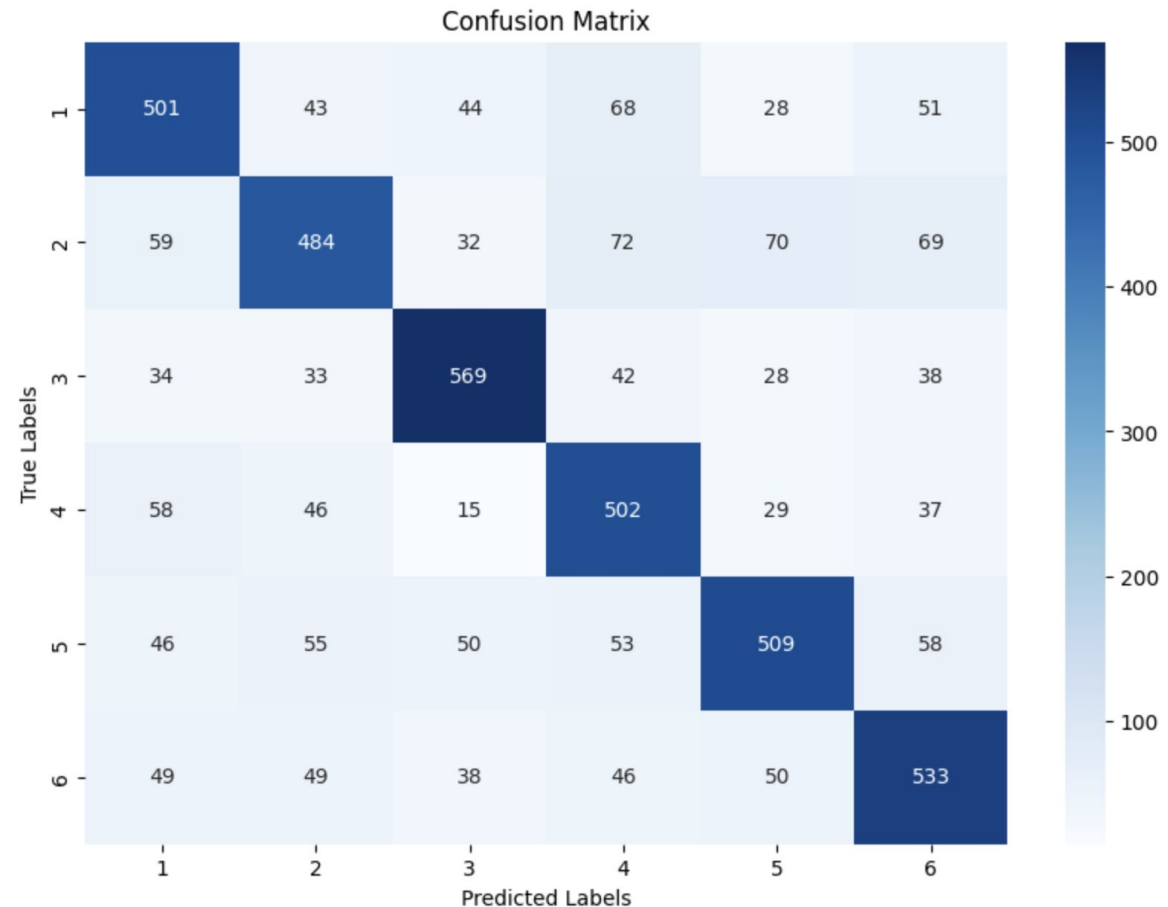
# Машинное обучение. CatBoost + confusion matrix

## CatBoost:

- Градиентный бустинг на деревьях решений
- Эффективно работает с категориальными данными
- GridSearchCV для настройки параметров
- Точность на тестовой выборке: 70%

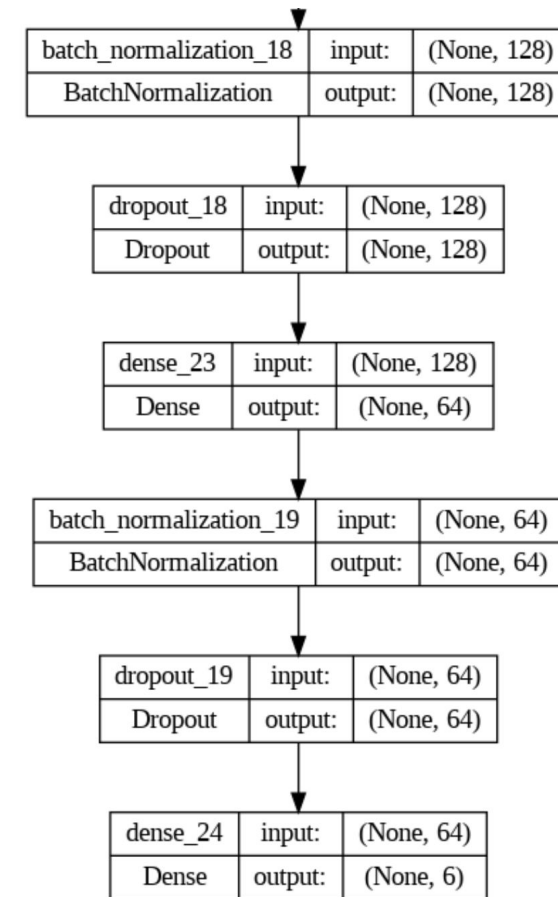
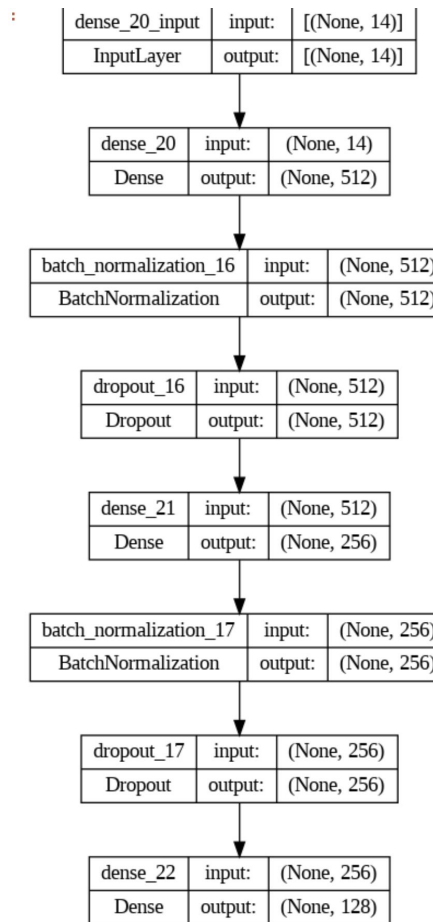
## Матрица ошибок:

- Хорошие результаты для жанров k-pop (1) и blues (6)
- Сложности с классификацией жанров metal (3) и classical (5)



# Построение нейронной сети

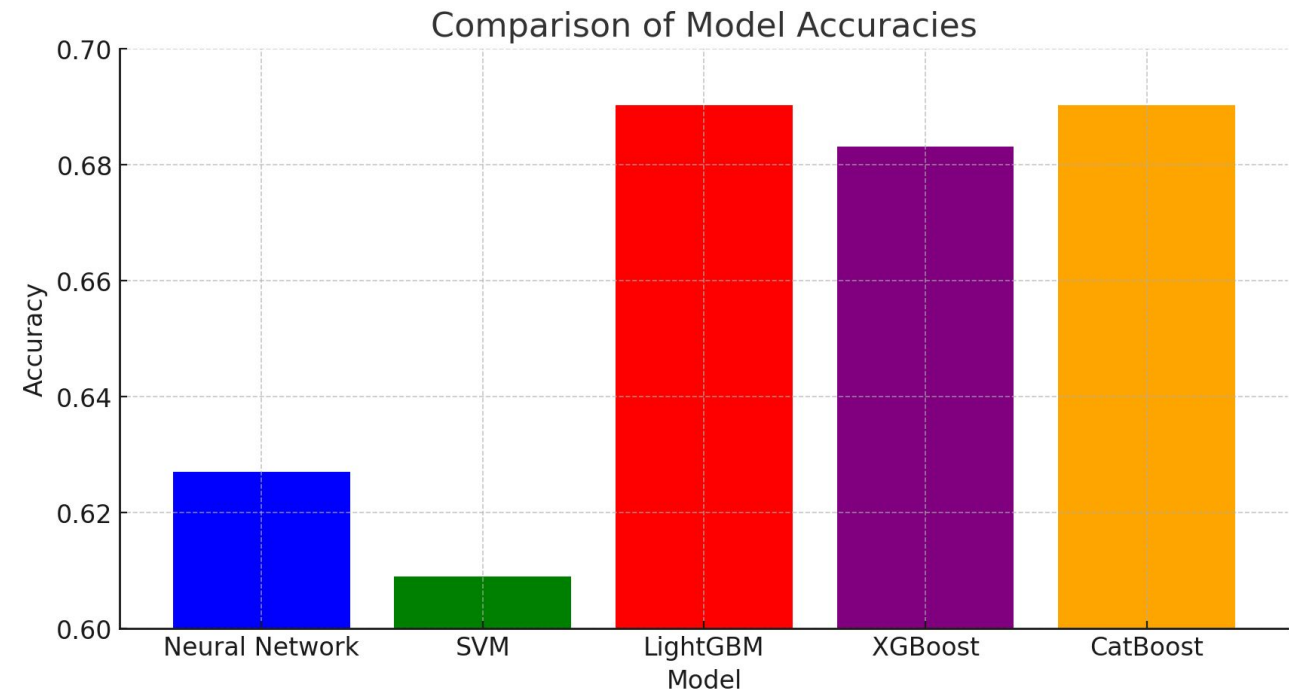
- Использованы CNN, MLP, RNN, LSTM
- Лучшая производительность у MLP
- Архитектура с Keras:
  - Плотный слой на 512 нейронов
  - Dropout и BatchNormalization
  - Завершающий слой с функцией softmax
- Оптимизация и обучение:
  - Оптимизатор Adam
  - Функция потерь categorical\_crossentropy
  - Метрика accuracy
  - Ранняя остановка и ModelCheckpoint
- Точность на тестовом наборе: 62.7%
- Архитектура MLP и предотвращение переобучения





## Выводы по обученным моделям

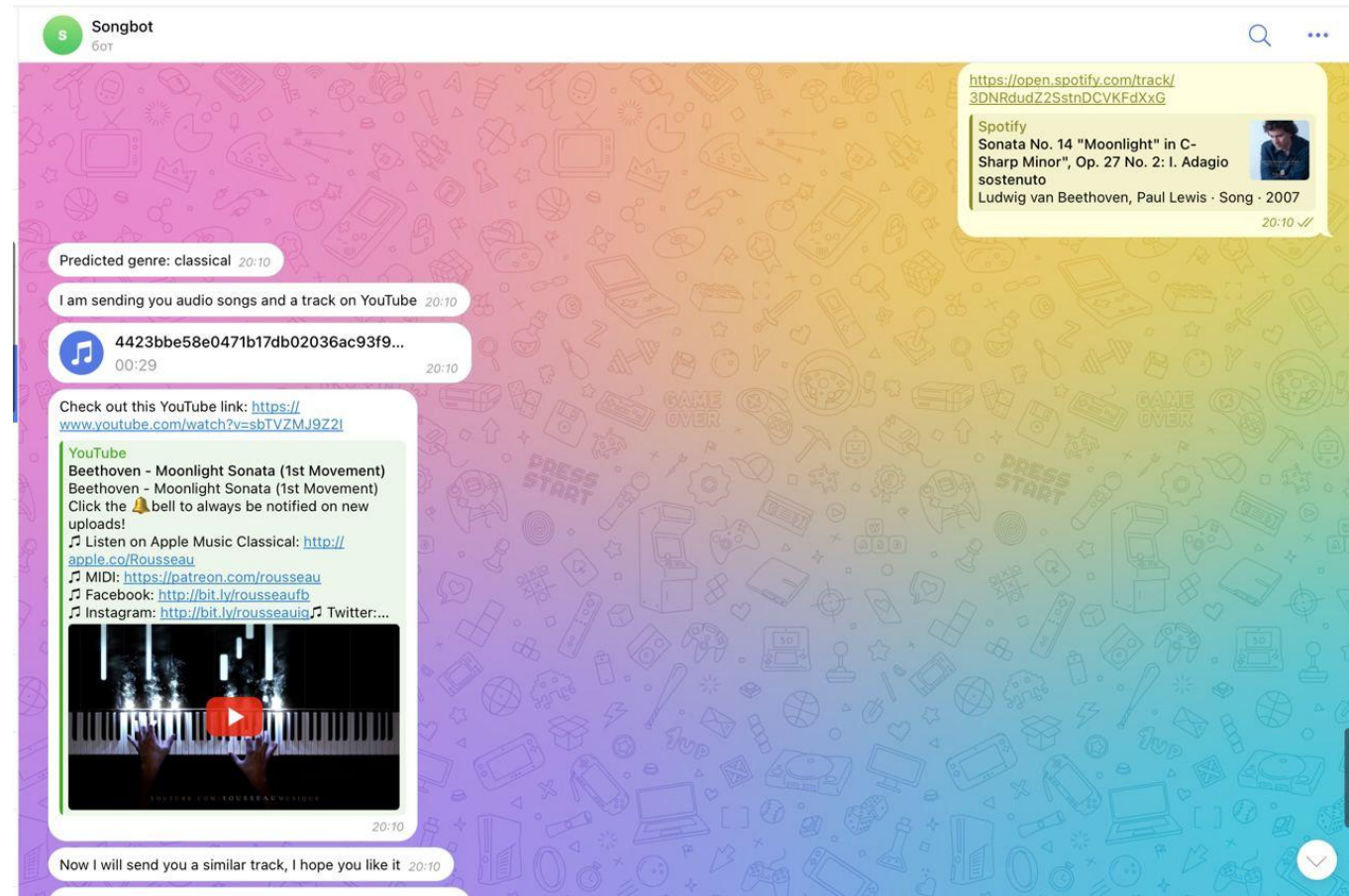
- Использовано несколько методов машинного обучения для классификации музыкальных жанров.
- Методы: CatBoost, XGBoost, LightGBM, SVM, нейронные сети.
- Проведена предварительная обработка данных: стандартизация, удаление нерелевантных признаков и выбросов.
- Лучшие результаты показали модели градиентного бустинга (LightGBM и CatBoost) с точностью 69.03%.
- CatBoost показал высокую способность к выявлению нелинейных зависимостей.
- График сравнения моделей подтверждает превосходство бустинговых моделей над SVM и нейронной сетью



# Разработка Телеграм-бота

## Основные функции бота:

- Получение ID трека от пользователя.
- Извлечение информации о треке через Spotify API.
- Предсказание жанра.
- Поиск видео на YouTube.
- Рекомендация похожего трека.



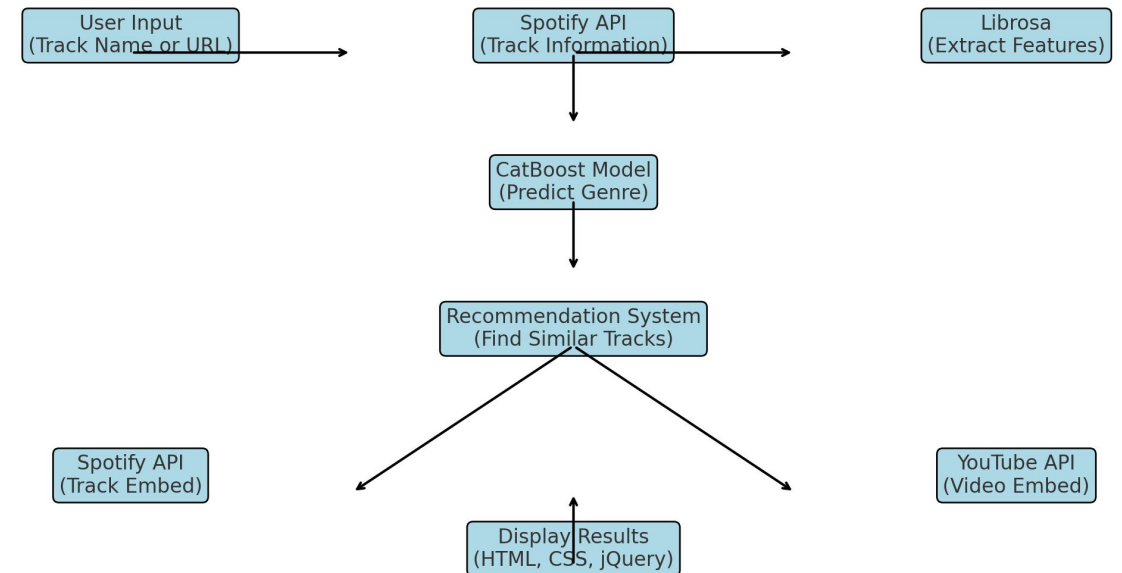
# Разработка веб-приложения

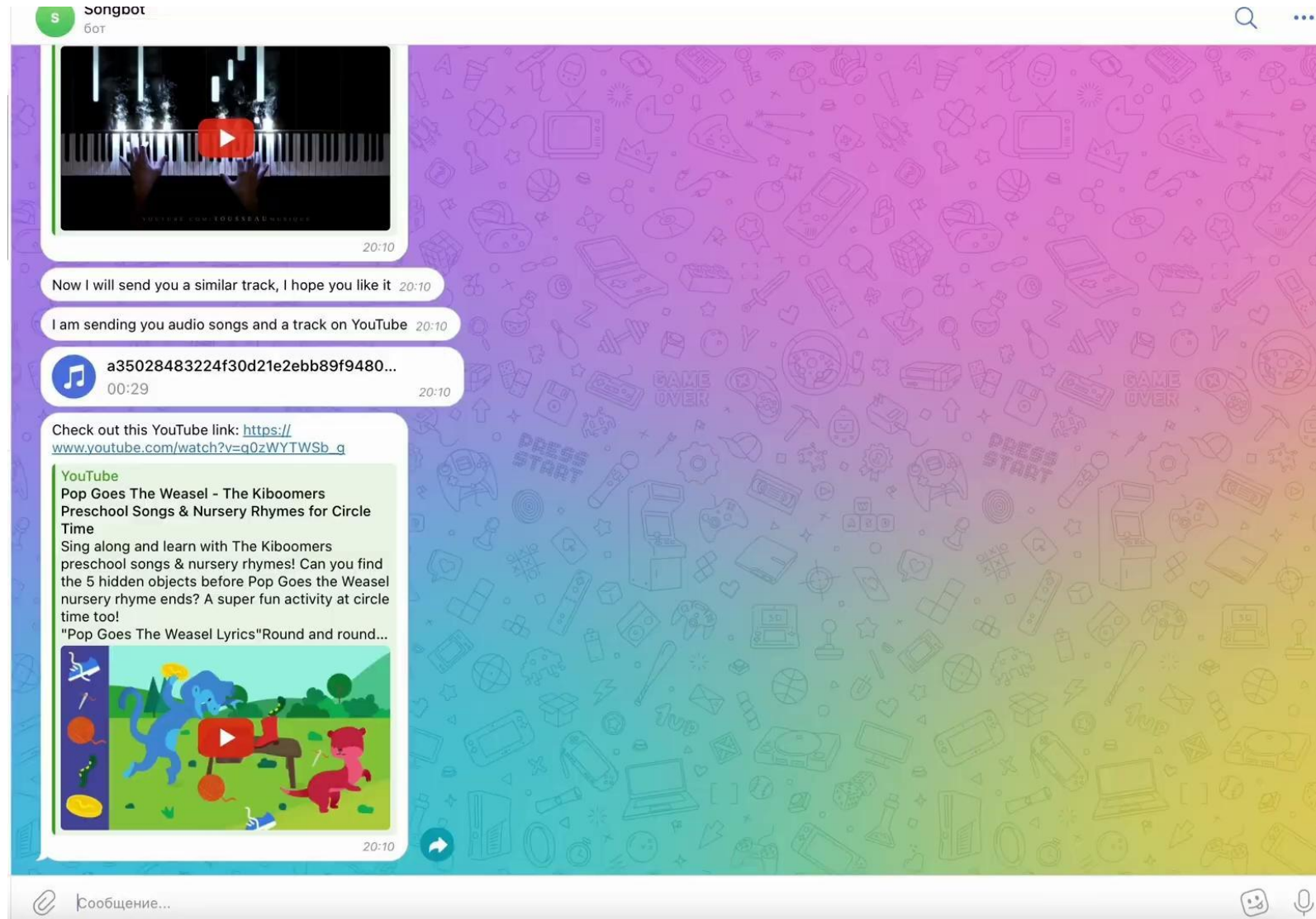
## Web-приложение

- Система музыкальных рекомендаций с использованием методов машинного и глубокого обучения.
- Анализ треков и предоставление рекомендаций на основе музыкальных предпочтений.
- Интерактивный интерфейс для ввода названий треков или ссылок на Spotify.
- Использование современных технологий веб-разработки и интеграция с внешними музыкальными сервисами через API.

## Архитектура приложения

- Пользовательский интерфейс: HTML, CSS, jQuery.
- Серверная логика: Flask.
- Интеграция с API: Spotify API, YouTube API, librosa.
- Модель машинного обучения: CatBoost.
- Выбор рекомендаций: на основе предсказанного жанра.
- Интеграция с YouTube и Spotify: получение ссылок на клипы и аудиозаписи.

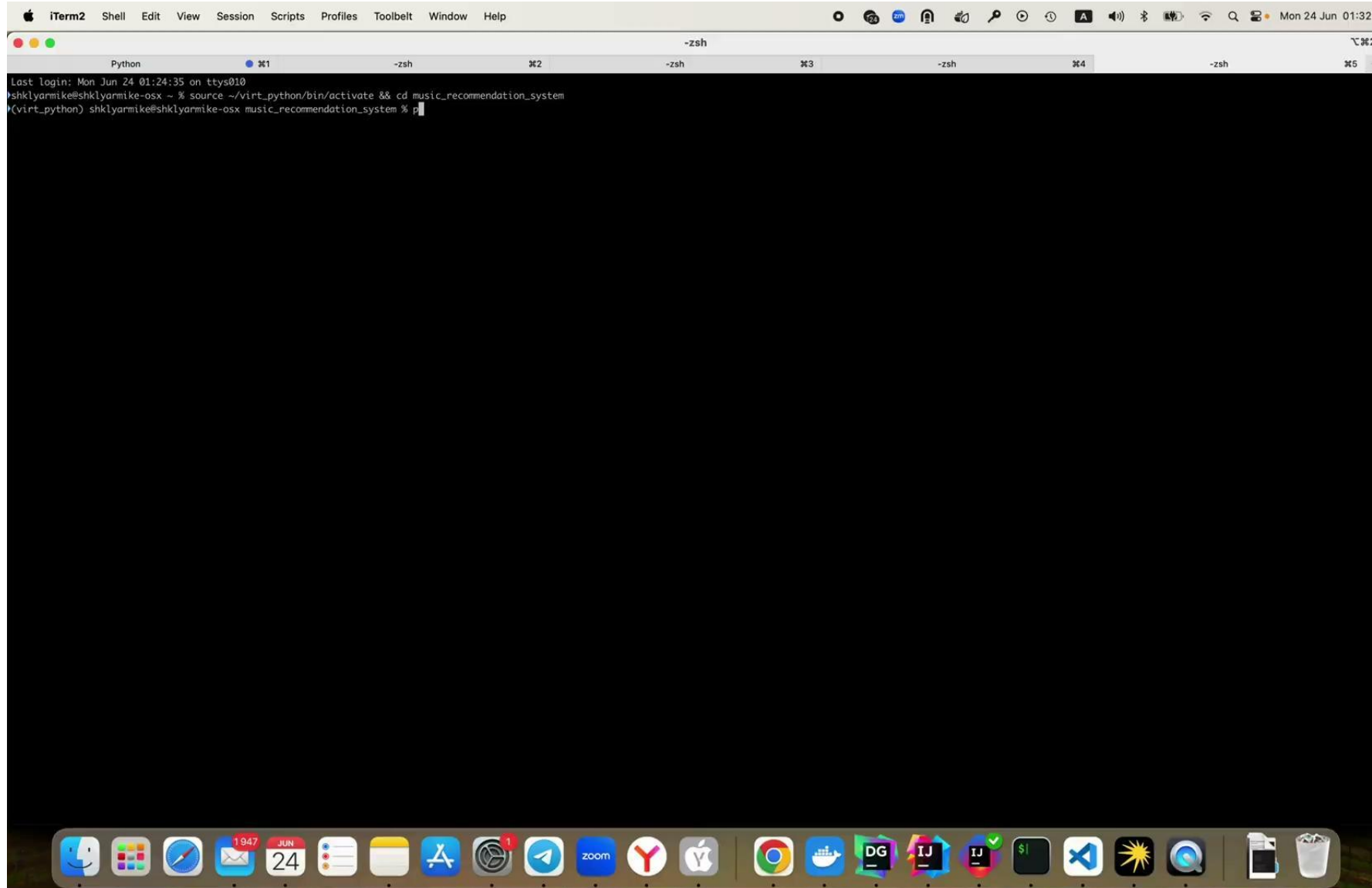








# Demonstration web-app





## Будущие улучшения

**Расширение поддержки жанров:** Добавление большего количества жанров для более разнообразных рекомендаций.

**Улучшение обучения моделей:** Использование более крупного и разнообразного набора данных для повышения точности моделей.

**Реальные рекомендации:** Внедрение обработки данных в реальном времени для актуальных рекомендаций.

**Персонализация:** Учет истории прослушиваний и предпочтений пользователей для более точных рекомендаций.

**Интеграция с другими платформами:** Расширение интеграции на другие музыкальные платформы, такие как Apple Music и Amazon Music.

**Расширенные функции анализа аудио:** Использование продвинутых методов анализа аудио для более точного извлечения характеристик.

**Обратная связь от пользователей:** Внедрение механизма обратной связи для улучшения алгоритма рекомендаций на основе оценок пользователей.





## Список источников

1. Tzanetakis, George, et al.: "Music Genre Classification with Machine Learning." Journal of Machine Learning Research - <http://jmlr.org/papers/volume13/tzanetakis12a/tzanetakis12a.pdf> (Дата обращения: 25.08.2023)
2. Choi, Keunwoo, et al.: "Automatic Tagging using Deep Convolutional Neural Networks." ISMIR - [https://ismir.net/archives/2016/Choi\\_Automatic\\_Tagging\\_using.pdf](https://ismir.net/archives/2016/Choi_Automatic_Tagging_using.pdf) (Дата обращения: 01.09.2023)
3. Spotify Technology S.A.: "Developing Spotify's Music Recommendation Engine." - <https://www.spotify.com/us/about-us/contact/> (Дата обращения: 09.09.2023)
4. Baccigalupo, Claudio, and Juan Manuel Pacheco: "Music Recommendation: A Multi-level Perceptual Approach." Artificial Intelligence Review - <https://link.springer.com/article/10.1007/s10462-008-9101-4> (Дата обращения: 21.10.2023)
5. Statista: "Global Music Streaming Market Trends and Forecasts (2024–2029)." - <https://www.statista.com/statistics/652140/global-music-streaming-revenue/> (Дата обращения: 04.01.2024)
6. Dieleman, Sander, et al.: "End-to-end Learning for Music Audio." IEEE International Conference on Acoustics, Speech, and Signal Processing - <https://ieeexplore.ieee.org/document/7952134> (Дата обращения: 12.02.2024)
7. McFee, Brian, et al.: "LibROSA: A Python Package for Music and Audio Analysis." Journal of Open Source Software - <https://joss.theoj.org/papers/10.21105/joss.00534> (Дата обращения: 29.03.2024)
8. Music Machinery: "How Music Recommendation Works — Challenges and Solutions." - <https://musicmachinery.com> (Дата обращения: 15.04.2024)
9. Pedregosa, Fabian, et al.: "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research - <http://jmlr.org/papers/v12/pedregosa11a.html> (Дата обращения: 05.05.2024)
10. Ronacher, Armin: "Flask Documentation." - <https://flask.palletsprojects.com/en/2.0.x/> (Дата обращения: 22.05.2024)
11. Van den Oord, A., Dieleman, S., & Schrauwen, B.: "Deep content-based music recommendation." Advances in Neural Information Processing Systems, 26 - <https://proceedings.neurips.cc/paper/2013/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf> (Дата обращения: 27.05.2024)
12. Chen, S., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. <https://www.semanticscholar.org/reader/26bc9195c6343e4d7f434dd65b4ad67efe2be27a> (Дата обращения: 01.06.2024)



**Спасибо за внимание!**

Шкляр Михаил Игоревич

Построение системы музыкальных рекомендаций при  
помощи методов машинного обучения

[mishklyar@edu.hse.ru](mailto:mishklyar@edu.hse.ru)