Система подбора фильмов

Работу выполнили:

Гущин Дмитрий, Гордеев Андрей, Высокос Вадим

Руководитель:

Чумаченко Никита Сергеевич

2. Содержание

[3. Паспорт проекта 3](#_Toc193969968)

[4. Введение 4](#_Toc193969969)

[5. Теоретическое обоснование проекта 5](#_Toc193969970)

[6. Практическое описание проекта 6](#_Toc193969971)

[7. Перспективы развития проекта 8](#_Toc193969972)

[8. Список литературы и интернет-источников 8](#_Toc193969973)

[9. Скриншоты 9](#_Toc193969974)

# 3. Паспорт проекта

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название | Проектная группа | Аннотация | Проблема, на решение которой направлен проект | Актуальность | Цель проекта | Задачи | Сроки реализации проекта | Ожидаемые результаты |
| Система подбора фильмов и показ рекомендаций на основе предпочтений | Гущин Дмитрий, Гордеев Андрей, Высокос Вадим | Система рекомендаций фильмов на Python предлагает персонализированные фильмы на основе предпочтений пользователя. Она анализирует вкусы с помощью машинного обучения, создаёт профили и позволяет добавлять фильмы в избранное. Интуитивный интерфейс и постоянный анализ данных улучшают опыт просмотра. | Перегруженность информацией: Пользователи могут почувствовать себя подавленными из-за слишком большого количества доступных вариантов и неудовлетворенность выбором: часто выбор оказывается неудачным, что может привести к разочарованию и снижению интереса к просмотру | Рост объема контента: с каждым годом количество доступных фильмов и сериалов растет, что усложняет процесс выбора и увеличивает потребность в персонализированных рекомендациях и изменение привычек потребления: С переходом на стриминговые платформы пользователи ожидают, что им будет легко находить фильмы, которые соответствуют их интересам и настроению, что делает системы подбора особенно важными | Упростить поиск фильмов и сделать его доступней для всех | Сбор данных: необходимо собрать данные о фильмах, включая жанры, актеров, рейтинги и отзывы. Это может быть реализовано через API киноиндустрии или веб-скрейпинг.Также необходимо разработать механизм создания и хранения профилей пользователей, включающих их предпочтения и интересы, а также историю взаимодействий с системой | 01.03.2025 - 26.03.2025 | Положительные |

# 4. Введение

Создание системы подбора фильмов на Python — это увлекательный проект, который предоставляет пользователям возможность находить подходящие фильмы на основе их индивидуальных интересов. Система использует алгоритмы машинного обучения для анализа предпочтений, что позволяет формировать рекомендации, соответствующие личному вкусу каждого пользователя.

Основные функции включают возможность добавления фильмов в избранное, что делает процесс выбора еще удобнее. Интуитивно понятный интерфейс упрощает взаимодействие, а регулярное обновление данных обеспечивает актуальность рекомендаций. Такой подход помогает пользователям открывать для себя новые фильмы и наслаждаться просмотром

# 5. Теоретическое обоснование проекта

Теоретическое обоснование системы подбора фильмов включает следующие ключевые аспекты:

1. Сбор информации

- Пользовательские данные: оценки, история просмотров, жанры. Реализация через регистрацию или анонимный сбор.

- Данные о фильмах: название, жанр, описание, год, актерский состав. Использование API (TMDb, IMDb).

- Метаданные: отзывы критиков и пользователей для улучшения рекомендаций.

2. Анализ информации

- Предобработка данных: очистка, заполнение пропусков, нормализация.

- Классификация: определение жанров для точных рекомендаций.

- Анализ предпочтений: выявление паттернов с помощью кластеризации и статистики.

- Изучение взаимосвязей: ассоциативный анализ для нахождения похожих фильмов.

3. Построение рекомендательной модели

- Коллаборативная фильтрация: использование пользовательских данных для рекомендаций.

- Контентная фильтрация: опора на характеристики фильмов.

- Гибридные методы: совмещение первых двух подходов для повышения качества.

4. Функционал избранного

- Позволить пользователям добавлять фильмы в избранное с хранением данных в базе.

5. Тестирование и оценка

- Важность регулярной оценки системы по метрикам, таким как точность и полнота, и получение обратной связи.

# 6. Практическое описание проекта

Для создания системы подбора фильмов и показа рекомендаций на основе предпочтений пользователя на языке программирования Python надо в первую очередь написать код для получения API.py. Наш код не показан по соображениям безопасности

После выполненных вышеуказанных действий нужно закодировать основной код main.py (см. 9. main). Наш код:

**from api import get\_films**

**def display\_film\_card(film, index=None):**

**st.image(film['posterUrl'], width=200)**

**st.write(f"Фильм: {film['nameRu']}")**

**st.write(f"Рейтинг: {film['ratingKinopoisk']}")**

**if isinstance(film['genres'], list):**

**genres = film['genres']**

**genres\_str = ', '.join([g['genre'] for g in genres])**

**st.write(f"Жанры: {genres\_str}")**

**else:**

**st.write(film['genres'])**

**if 'kinopoiskId' in film:**

**button\_key = f"add\_{film['kinopoiskId']}"**

**else:**

**button\_key = f"add\_{film['nameRu']}\_{film['year']}"**

**if st.button(f"Добавить в избранное: {film['nameRu']}", key=button\_key):**

**if 'favorites' not in st.session\_state:**

**st.session\_state['favorites'] = []**

**else:**

**st.session\_state['favorites'].append(film)**

**st.success(f"Фильм {film['nameRu']} успешно добавлен в избранное!")**

**st.title('Фильмотека')**

**films = get\_films()**

**for index, film in enumerate(films):**

**if index > 0:**

**display\_film\_card(film, index=index)**

**st.sidebar.title("Избранное")**

После выполненных выше действий и написания кода остается только тестировать проект

# 7. Перспективы развития проекта

- Фикс багов сайта, если такие присутствуют, и выход публичной версии в интернет

- Добавление расширенного функционала

- Более точный подбор рекомендаций, основывающийся не только по жанру фильма, но и по актерскому/режиссерскому составу, кинокомпании и т.д.

# 8. Список литературы и интернет-источников

(АйТиФай, б.д.)

# 9. Скриншоты

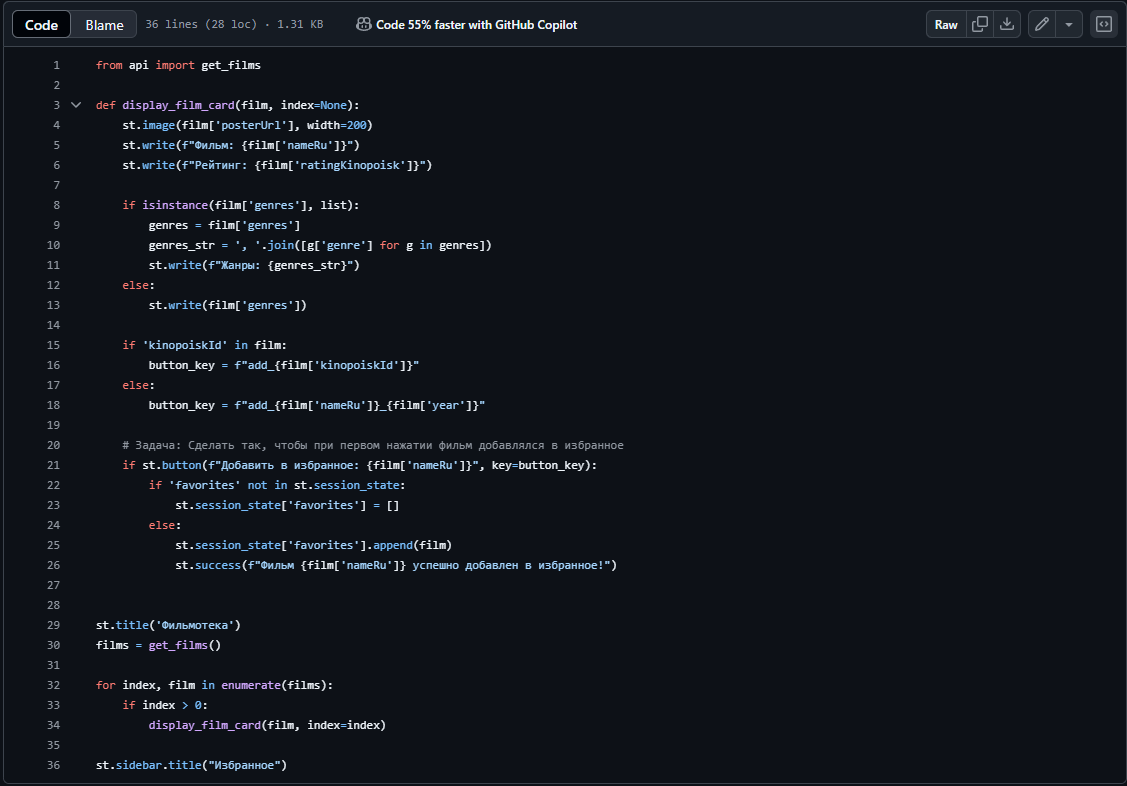


Рисунок 2. Основной код (main)