МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теории функций и стохастического анализа

# РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ФИЛЬМОВ НА ОСНОВЕ ДАННЫХ КИНОПОИСКА

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 248 группы
направления 09.04.03 — Прикладная информатика

механико-математического факультета
Бахарева Сергея Николаевича

Научный руководитель
доцент, к. ф.-м. н. _____ Д. В. Мельничук

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2026

**Введение**. Одним из самых интригующих социальных явлений, вызванных достижениями в области информационных и коммуникационных технологий, является увеличение возможностей для передачи информации в письменной и устной формах. С помощью Интернета, беспроводных сетей и мобильной телефонии современные граждане формируют большой массив различных сообществ, в которых они обмениваются мнениями и опытом о компаниях, продуктах, услугах и даже мировых событиях. Хотя устная передача информации – явление старое, появление Интернета добавило два важных новых аспекта к этой концепции: беспрецедентная масштабируемость и скорость распространения, а также устойчивость и измеримость. Растёт объём данных, появляется потребность в их фильтрации.

С вышеназванной проблемой и тенденцией сталкиваются различные платформы, такие как Кинопоиск. Рекомендательные системы помогают удовлетворять потребность в фильтрации информации и отчасти прогнозируют дальнейшие действия человека

Целью настоящей работы является создание рекомендательной системы для фильмов на основе данных Кинопоиска. В связи с поставленной целью необходимо решить следующие задачи:

1. Изучение явления рекомендательных систем;
2. Сбор и обработка данных;
3. Выбор алгоритма и создание модели;
4. Создание информационной системы для взаимодействия с моделью.

**Структура работы.** Основная часть состоит из 3 разделов:

— Исследование «рекомендательных систем» и методов их построения;
— Построение источника данных и подбор модели;
— Проектирование и создание информационной системы.

**В первом разделе** было исследовано понятие «рекомендательной системы», описаны методы машинного обучения, на основе которых можно построить модель рекомендательной системы, и использованные инструменты разработки.

Рекомендательные системы достаточно широко применяются в некоторых видах задач анализа данных. такие системы призваны фильтровать большое количество информации и, тем самым, решать проблему избытка

контента, с которой может столкнуться пользователь.

В контексте сервиса задача рекомендательной системы звучит так: проинформировать пользователя о товаре, который может быть наиболее интересен ему в данный момент времени. Однако в более общем смысле такая задача сводится к прогнозированию некоего значения рейтинга для комбинации пользователь-объект. Предполагается наличие обучающих данных, отражающих предпочтения субъектов в отношении предметов рекомендации. Для $m$ пользователей и $n$ товаров это соответствует неполной матрице $m \times n$, где наблюдаемые значения используются для обучения. Значения ненаблюдаемые прогнозируются с использованием этой обучающей модели.

Рекомендательная система, разрабатываемая в этой работе, является коллаборативной. Такой вид не требует большого количества информации о фильмах – достаточно иметь множество записей вида «пользователь, фильм, оценка».

Рекомендательные системы применяются уже давно. На данный момент существует множество различных исследований на эту тему. Известно точно, что использование рекомендаций положительно влияет на продажи в интернет-магазинах. Такие крупные компании, как YouTube, Amazon и Netflix, уже достаточно давно используют коллаборативные рекомендательные системы в своих сервисах. При создании рекомендательных систем в качестве моделей часто использовались такие методы, как k-ближайших соседей (KNN) и сингулярное разложение (SVD). Для измерения качества прогнозов чаще использовались метрики F1, точность (precision), полнота (recall) и средняя абсолютная ошибка (MAE).

В настоящей работе были рассмотрены данные по отзывам на фильмы, снятые в Российской Федерации в период с марта 2015 по март 2025. В качестве источника данных была взята база данных Кинопоиска – русскоязычного интернет-сервиса с условно свободно редактируемой базой. Данный сервис предоставляет пользователям возможность размещать отзывы на своём сайте, однако каждая рецензия проходит процесс модерации. Также именно пользователь задаёт характер отзыва: положительный, нейтральный или отрицательный. Кинопоиск хранит информацию о фильмах: название, год выхода, даты проката, страны и многое другое. Пользователи имеют право

исправлять ошибки в данных, однако корректировки также проходят процесс модерации.

Для подключения к источнику данных был использован неофициальный API Кинопоиска ПоискКино. Этот сервис предоставляет пользователям бесплатный доступ к базе данных Кинопоиска, возможность скачивать данные об отзывах и фильмах с помощью множества фильтров.

В данной работе был использован сервис Google Colab, интегрированная среда разработки PyCharm 2021 Community edition, язык программирования Python версии 3.11, все данные были записаны в файлы формата json.

Python – это интерпретируемый высокоуровневый язык программирования с динамической строгой типизацией и автоматическим управлением памятью. Главными преимуществами Python можно назвать хорошую читаемость кода, широкую используемость и наличие множества библиотек для работы с данными, что делает этот язык одним из лучших решений для реализации поставленных задач.

Google Colaboratory – это облачная среда для выполнения Python-кода на базе Jupyter Notebook. Сервис позволяет писать и выполнять код в браузере и при этом не требует длительной настройки, предоставляет бесплатный доступ к графическим процессорам и документам, даёт возможность делиться своим блокнотом с другими пользователями.

PyCharm является бесплатной IDE с открытым исходным кодом. Платформа включает в себя множество возможностей: помощь в написании кода, быструю навигацию и поиск, работу с Git и GitHub, древовидную структуру отображения файлов проекта и многое другое. PyCharm предоставляет удобное взаимодействие с кодом, автоматическое создание виртуальной среды вокруг проекта, точечную настройку интерфейса, удобный механизм импорта и многое другое.

В работе были использованы следующие библиотеки и модули языка Python:

- для работы с данными: Pandas (2.2.2), json (2.0.9), NumPy (2.0.2);
- для отправки запросов и получения ответов: requests (2.32.4), time (встроенная);
- для визуализации данных: Matplotlib (3.10.0);

— для разработки моделей: Surprise (1.1.4), Catboost (1.2.8);
— для обработки данных и метрик: Scikit-learn (1.6.1);
— для упорядоченного вывода данных в консоль: pprint (встроенная);
— для связи Colaboratory с диском: модуль colab из библиотеки google (2.0.3);
— для создания информационной системы: Streamlit (1.51.0).

Surprise – это библиотека для Python, содержащая набор инструментов, предназначенных для создания и анализа рекомендательных систем, работающих с данными, содержащими явные оценки. Этот модуль умеет взаимодействовать только со старыми версиями некоторых библиотек, поэтому при работе с ним использовались NumPy 1.26.4 и Pandas 1.5.3.

В качестве основы для построения моделей были взяты алгоритмы k-ближайших соседей (KNN) и сингулярное разложение Фанка (Funk SVD) из библиотеки Surprise, а также более сложный CatBoostClassifier из библиотеки CatBoost.

Для оценивания системы использовались такие методы, как F1-мера, коэффициент Каппа, отчёт по классификации, матрица ошибок и количество правильно спрогнозированных величин. Все они были взяты из библиотеки Scikit-learn.

Для заключительной части работы был использован Streamlit – фреймворк с открытым исходным кодом для языка Python, предназначенный для специалистов по анализу данных, инженеров в области искусственного интеллекта и машинного обучения.

**Во втором разделе** была описана одна половина практической части работы, посвященная созданию исходных данных и подбору модели.

Сбор данных о фильмах и отзывах проводился отдельно, так как использовались разные разделы API.

Сбор данных о фильмах производился постранично. На каждой странице было по 249 записей (максимум, предоставляемый API). Всего было скачано 63 страницы за 4 попытки. Было получено 4 файла формата json с записями. Следующим этапом они объединялись и записывались в один файл. Всего было получено 15687 записей.

Набор данных содержит информацию о фильме, связанных с ним собы-

тиях и сопутствующих материалах. В наборе представлены следующие поля: id, name, alternativeName, type, typeNumber, year, description, shortDescription, status, rating, votes, movieLength, totalSeriesLength, seriesLength, ageRating, poster, genres, countries, releaseYears, isSeries, ticketsOnSale, backdrop, enName, logo, names. Получившийся набор данных являлся вполне приемлемым для анализа, однако требовал некоторых исправлений.

В самом начале была проведена проверка на уникальность столбцов «id» (уникальный идентификатор фильма) и «description» (описание фильма); дубликатов найдено не было. В ходе работы с данными были переформатированы значения в столбцах «genres» и «countries», удалена информация о не вышедших в прокат фильмах и о фильмах без оценок, а рейтинг Кинопоиска был вынесен в отдельный столбец. Количество записей (фильмов) сократилось до 5259.

Сбор данных об отзывах оказался чуть более сложным процессом. Были взяты идентификаторы фильмов из уже имеющихся обработанных ранее данных. Для скачивания пришлось снова прибегнуть к разделению. Было выяснено, что за один запрос можно скачать 400 фильмов. Значит, количество итераций $iterations = [\frac{5259}{400}] + 1 = 14$. Аналогично фильмам, получившиеся dataframe склеивались в один набор данных с помощью метода concat. После этого он был сохранён. Таким образом, данные об отзывах были собраны.

В наборе данных об отзывах были представлены следующие поля: id, movieId, title, type, review, date, author, userRating, authorId, reviewLikes, reviewDislikes, createdAt, updatedAt. Этот набор данных содержал достаточно много информации, пригодной для анализа.

В процессе обработки данных об отзывах была изменена оценка (тип отзыва) type на числовое значение. Положительным отзывам была присвоена «1», нейтральным – «0», отрицательным «$-1$» для мультиклассовой и «0» для бинарной классификаций.

Следующим этапом стало построение моделей на основе собранных данных и сравнение KNN, SVD (Funk SVD) и CatBoost.

В библиотеке Surprise содержалось несколько модификаций алгоритма KNN: KNNBasic, KNNWithZScore, KNNWithMeans и KNNBaseline. Каждая из них была протестирована с несколькими алгоритмами, вычисляющими ме-

ру схожести и задающимися в аргументах модели параметром sim_options: средняя квадратичная разность (msd, Mean Squared Difference), корреляция Пирсона обычная (pearson) и со смещением (pearson_baseline). Имеющиеся алгоритмы были опробованы по два раза: для вычисления сходства между пользователями и между объектами (параметр user_based в sim_options). Также рассматривались различные разбиения данных на тестовую и тренировочную части (от 0.1 до 0.35 для тестовой), при которых сохранялся тот же баланс классов, что и в исходных данных.

Подбор модели проходил в несколько этапов. На первом шаге вычислялись метрики для моделей с достаточно большим шагом для k. Среди всех получавшихся моделей отбирались такие, которые показывали первые и вторые наилучшие результаты Каппы и средней F1-меры по классам. На втором этапе шаг для k и диапазон этих значений становился меньше.

Далее был применён алгоритм SVD. Для него была написана программа с подбором лучшего результата, проходящая в 3 цикла и рассматривающая такие параметры, как количество скрытых признаков (n_factors), скорость обучения (lr_all), коэффициент регуляризации (reg_all) и различные разделения данных на тестовую и тренировочную части. Подбор проходил в один этап.

Наиболее результативная модель CatBoost также выбиралась методом перебора. Менялось количество деревьев (iterations), скорость обучения (learning_rate), максимальная глубина деревьев (depth) и различные разделения данных на тестовую и тренировочную части.

Результаты подбора и значения наиболее качественных моделей были выписаны в таблицу 1.

Таблица 1 – Результаты подбора

| Алгоритм | Доля тестовой | F1 | F1 для классов | Kappa |
|---|---|---|---|---|
| KNN (3 класса) | 0.11 | 0.364 | 0.1990172<br>0.32327167<br>0.56859362 | 0.152 |
| KNN (2 класса) | 0.1 | 0.662 | 0.58785249<br>0.73519164 | 0.331 |
| SVD (3 класса) | 0.18 | 0.38 | 0.21583851<br>0.32197562<br>0.60095071 | 0.164 |
| SVD (2 класса) | 0.21 | 0.655 | 0.59304905<br>0.71734292 | 0.311 |
| Catboost (3 класса) | 0.16 | 0.5 | 0.52181003<br>0.34132581<br>0.63688368 | 0.28 |
| Catboost (2 класса) | 0.12 | 0.668 | 0.65102421<br>0.68461797 | 0.34 |

Таким образом, для всех моделей разделение данных на 2 класса упростило задачу и дало возможность повысить качество прогнозирования. Возросли показатели F1 и Каппы, а точность прогнозирования между разными классами стала более сбалансированной. Среди всех моделей самые высокие результаты показал CatBoost для бинарной классификации, хотя разница с SVD и KNN оказалась небольшой.

**В третьем разделе** описывается разработка обзорной информационной системы для визуализации и взаимодействия с рекомендательной моделью.

В качестве инструмента был использован модуль Streamlit. Обзорная информационная система содержит статистику по фильмам и отзывам, описание данных, итоги построения моделей, возможность создания дополнительного пользователя и демонстрацию рекомендаций. Всё перечисленное было разбито и разнесено по разным вкладкам (разделам).

Приложение запускается и начинает выполняться из функции main(). После этого начинается построение самой страницы, в первую очередь – панели вкладок. В системе были определены следующие вкладки: «Обзор», «Данные», «Модели», «Демо рекомендаций», «Отзывы» и «Создание пользователя».

Первый раздел содержит название системы, обозначения используемых источников, статистические данные по фильмам, пользователям и отзывам, а также таблицу наиболее активных пользователей. Статистических переменных пять: количество отзывов, пользователей, фильмов, плотность матрицы взаимодействий «пользователь, фильм, рекомендация» и процент длинных рецензий (от 300 слов). Наиболее активные пользователи были внесены в таблицу. Внешний вид первого раздела представлен на рисунке 1.

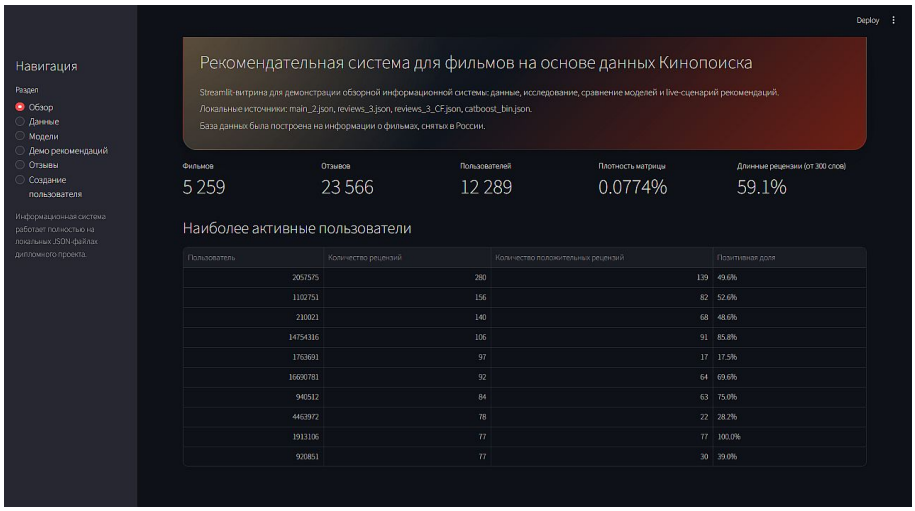


Рисунок 1 – Вкладка «Обзор»

Вкладка «Данные» содержит четыре графика, построенных с помощью виджетов bar_chart и line_chart, а также таблицу с самыми обсуждаемыми фильмами

На вкладке «Модели» представлены результаты подбора модели из второго раздела. Большая часть разделена на две половины: для бинарных и многоклассовых моделей. Приведены сводки по лучшим моделям в виде таблиц и графиков.

Раздел «Демо рекомендаций» является одной из важнейших частей системы. Именно на ней отображаются результаты рекомендации для пользователя. На вкладке располагается множество объектов: выбор пользователя,

выбор фильма, статистика по пользователю, информационная сводка по выбранному фильму, краткая история оценок пользователя в виде таблицы. Ниже указывается информация о связи пользователя с фильмом: есть ли фильм в истории пользователя, вероятность рекомендации и позитивная доля по фильму. В случае наличия отзыва на фильм от текущего автора система указывает, рекомендовал ли он это кино или нет. Наиболее важный элемент располагается в нижней части страницы – это таблица рекомендаций. В ней приводятся 20 самых рекомендуемых пользователю фильмов, то есть тех, которые имеют самый высокий процент рекомендации. Внешний вид раздела изображён на рисунке 2.

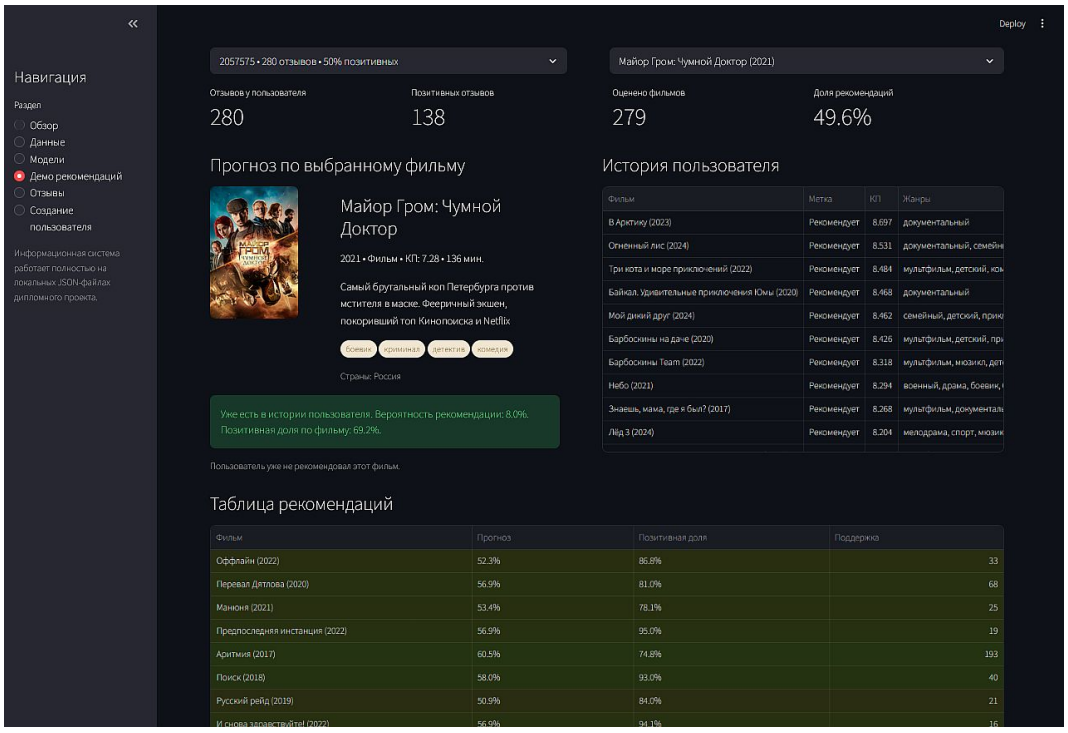


Рисунок 2 – Вкладка «Демо рекомендаций»

Раздел «Отзывы» даёт возможность изучить информацию о рецензиях на конкретный фильм. Начинается всё с выбора фильма и класса отзывов. Можно изучать информацию как о рецензиях в целом, так и о конкретных оценках: негативных, нейтральных и позитивных. Ниже приводится краткое описание фильма, его статистические показатели, график распределения типов рецензий для фильма и примеры отзывов в виде раскрывающегося списка с заголовками из их названий.

Раздел «Создание пользователя» содержит малое количество элемен-

тов интерфейса. Изначально на вкладке располагается только заголовок и кнопка «Добавить пользователя». При нажатии на кнопку появляется уникальный идентификатор нового пользователя, поле для ввода логина, кнопка сохранения и таблица для ввода. После заполнения всех обязательных полей данные сохраняются в системе. Внешний вид страницы представлен на рисунке 3.

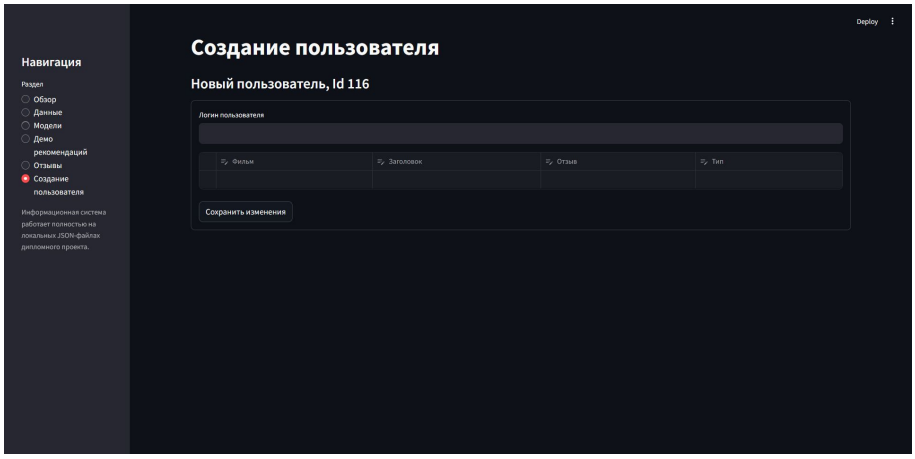


Рисунок 3 – Вкладка «Создание пользователя»

После сохранения данных можно будет увидеть рекомендации для нового пользователя в разделе «Демо рекомендаций» и его отзывы во вкладке «Отзывы».

**Заключение.** В результате проделанной работы было изучено, что собой представляют рекомендательные системы, то есть была выполнена задача 1. Для дальнейшего построения и работы системы посредством сбора и обработки данных был создан источник – выполнена задача 2. Были выбраны алгоритмы, подходящие для использования в рекомендательной системе, а также путем перебора и оценки была создана наилучшая модель для имеющихся данных – выполнена задача 3. В конце концов, при использовании библиотеки Streamlit была спроектирована и разработана обзорная информационная система для визуализации и взаимодействия с рекомендательной моделью. Система продемонстрировала свою работоспособность и корректность.

Таким образом, были решены все задачи, поставленные в начале работы, и достигнута конечная цель.