Подбор фильма по интересам

Ансамблевый подход с использованием алгоритмов Machine Learning

Автор: [Ваше Имя]

От проблемы выбора к умному поиску

Проблема

Фильмов тысячи, и найти «тот самый» сложно. Стандартные фильтры по жанрам часто неточны, предлагая либо слишком много вариантов, либо фильмы низкого качества.

Цель проекта

Разработать ML-алгоритм, который учитывает не только жанровую принадлежность, но и рейтинг фильма. Создать систему, предлагающую качественные рекомендации, похожие на любимые фильмы пользователя.

Технологический стек и данные

Основа проекта: датасет IMDb (5000+ фильмов) и современные библиотеки Python.

Pytho
n

Основной язык разработки и логики.

Panda
s

Обработка, очистка и анализ табличных данных.

Scikit-learn

Реализация алгоритмов KNN и препроцессинга.

Подготовка данных: Ключ к точности

Очистка данных

Заполнение пропусков в жанрах значением "Unknown" и в рейтингах — средним значением по датасету.

One-Hot Encoding

Преобразование текстовых жанров в бинарный формат (0 и 1), чтобы алгоритм мог математически сравнивать фильмы.

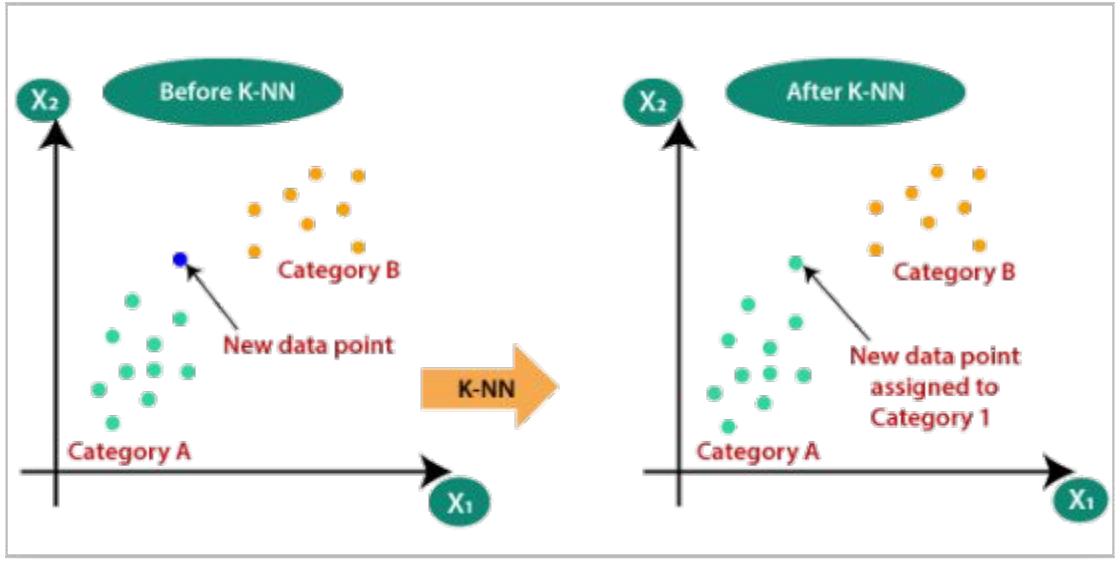
StandardScaler

Масштабирование рейтинга. Приведение оценки (1-10) к единому диапазону с жанрами для равноценного учета.

Основной алгоритм: K-Nearest Neighbors

Мы используем алгоритм **KNN** (K-ближайших соседей) не для классификации, а для поиска сходства.

- Каждый фильм — это точка в многомерном пространстве признаков.
- Алгоритм вычисляет расстояние между выбранным фильмом и всеми остальными.
- Результат: список из 5 "соседей", находящихся геометрически ближе всего.



Визуализация поиска ближайших соседей

Ансамбль метрик: Три взгляда на фильм

Для повышения качества рекомендаций мы используем три математических подхода:

Cosine Similarity

Измеряет угол между векторами.

Фокус: Состав Жанров

Euclidean Distance

Измеряет прямую дистанцию.

Фокус: Строгое соответствие рейтинга

Manhattan Distance

Сумма модулей разностей

Фокус: Баланс для сложных данных

Теория: Как метрика меняет результат?

Направление (Cosine)

Косинусное расстояние "смотрит" на профиль фильма. Ему важно, чтобы набор жанров совпадал. Разница в рейтинге (например, 8.0 против 5.0) влияет на результат слабо.
Результат: Находит тематически похожие фильмы, даже если их качество (рейтинг) сильно отличается.

Дистанция (Euclidean & Manhattan)

Эти метрики работают как строгая линейка. Благодаря масштабированию, разница в рейтинге становится такой же важной, как и разница в жанре.
Результат: Ищет фильмы, которые не только похожи по жанру, но и имеют практически идентичный рейтинг.

Сравнение: The Matrix (8.7)

Cosine (Фокус на жанр)

- Kantara (9.3)
- LOTR: Return of the King (9.0)
- The Dark Knight (9.0)

Нашел культовые фильмы с макс. рейтингом.

Euclidean (Фокус на рейтинг)

- Empire Strikes Back (8.7)
- Inception (8.8)
- LOTR: Two Towers (8.8)

Подобрал фильмы с рейтингом ~8.7.

Manhattan

- Matrix Reloaded (7.2)

Нашел сиквел (идеальное совпадение жанров).

Итоги проекта

- **Полноценная система рекомендаций:** Разработан алгоритм, решающий задачу "холодного старта" поиска похожих фильмов.
- **Комплексный анализ:** Внедрена логика учета рейтинга через масштабирование данных, что повысило качество выдачи.
- **Ансамблевый подход:** Использование трех метрик дает пользователю выбор: искать "шедевры" (Cosine) или фильмы "того же уровня" (Euclidean).
- **Масштабируемость:** Код подготовлен для работы с любыми новыми данными без ошибок (автоматическая очистка).

Спасибо за внимание!

Проект доступен на GitHub.

Готов ответить на ваши вопросы.