**🔥 Цель проекта**

1. **EDA (разведочный анализ)**
   * Красивые графики: динамика выхода фильмов по годам, топ-10 жанров, самых активных актёров и режиссёров.
   * Тепловая карта «жанр × десятилетие».
   * Это показывает, что ты умеешь работать с реальными данными.
2. **Несколько методов рекомендаций (для сравнения)**
   * **Content-based filtering**: рекомендации по жанрам, актёрам, описанию (TF-IDF + cosine similarity).
   * **Collaborative filtering**: матричная факторизация (SVD) или библиотека Surprise.
   * **Hybrid model**: комбинация двух подходов (например, LightFM).
   * Покажи таблицу/график с метриками: Precision@k, Recall@k, NDCG.
3. **NLP-бонус**
   * Использовать описания фильмов и применить BERT/Doc2Vec для «смыслового» сходства.
   * Это будет выглядеть современно и продвинутo.
4. **Графовый подход**
   * Построить граф (NetworkX): «фильм → актёры → режиссёры → жанры».
   * Сделать визуализацию «связей фильмов».
   * Это редко встречается в портфолио и сразу выделяет проект.
5. **Интерактивность**
   * Выложить простой **Streamlit/Dash веб-прототип**, где пользователь вводит фильм → получает рекомендации.
   * Работодателю понравится, что это «живое приложение», а не только Jupyter Notebook.
6. **Документация и оформление**
   * Чистый Jupyter Notebook с комментариями.
   * README.md: цели, методы, результаты, визуализации.
   * Несколько слайдов-презентации (или PDF) с краткими итогами.

**📌 Итог**

Если ты сделаешь **3–4 подхода (Content-based + Collaborative + NLP + Graph)** и выложишь результаты с интерактивным прототипом → это будет выглядеть как **сильный, продвинутый проект**, который реально можно показать на собеседовании.

**📂 Архитектура проекта (рекомендационная система на IMDb)**

movie-recommender/

│

├── data/ # Данные

│ ├── raw/ # Исходные данные (как скачали)

│ ├── processed/ # Обработанные данные (после cleaning)

│ └── external/ # Вспомогательные данные (жанры, словари и т.д.)

│

├── notebooks/ # Jupyter Notebooks

│ ├── 01\_EDA.ipynb # Разведочный анализ данных

│ ├── 02\_Content\_Based.ipynb # Content-based filtering

│ ├── 03\_Collaborative.ipynb # Collaborative filtering

│ ├── 04\_Hybrid.ipynb # Hybrid модели

│ └── 05\_Final\_Dashboard.ipynb # Подготовка к Streamlit/FastAPI

│

├── src/ # Код проекта

│ ├── data\_preprocessing.py # Функции для очистки и подготовки данных

│ ├── recommender.py # Алгоритмы рекомендаций

│ ├── evaluation.py # Метрики и сравнение моделей

│ └── utils.py # Вспомогательные функции

│

├── app/ # Приложение

│ ├── streamlit\_app.py # Streamlit дашборд

│ └── fastapi\_app.py # FastAPI API

│

├── reports/ # Итоговые материалы

│ ├── figures/ # Графики, визуализации

│ └── presentation.pdf # Слайды с итогами проекта

│

├── requirements.txt # Зависимости (pip install -r requirements.txt)

├── README.md # Описание проекта

└── .gitignore # Игнорируем лишние файлы

**📌 Как это будет работать**

1. **data/** → все исходные данные (IMDb датасет).
2. **notebooks/** → основная исследовательская работа, код с графиками и пояснениями.
3. **src/** → аккуратный код с функциями (чтобы не дублировать всё в ноутбуках).
4. **app/** → твой финальный интерактивный прототип:
   * streamlit\_app.py → для дашборда;
   * fastapi\_app.py → API (по желанию).
5. **reports/** → красивые графики и PDF-презентация.
6. **README.md** → витрина проекта (описание, как запустить, примеры рекомендаций).

**🎯 Следующие шаги**

1. Скачиваем датасет IMDb.
2. Делаем 01\_EDA.ipynb → быстрый анализ данных (распределения, графики).
3. Реализуем первую модель (content-based).