Deloitte.

Movie Recommendation Engine Based on Movie Features

By: Santiago Alvarez, Santiago Cataño, Alex Cordova, Sarah Stallman, Shalini Vijayaraghavan

OCTOBER 2024

Agenda

Topic	Presenter
Introduction (Business Foundations)	Alex Cordova
Our Solution and Dataset Insights	Santiago Cataño
Dataset Insights and Introduction to AI Model	Santiago Álvarez
AI Model: Data Pre-processing, item-item Recommendations with kNN, Cosine vs Euclidean Similarity Scores	Shalini Vijayaraghavan
AI Model and Data Challenges: Cold-Start Problem, Building a Movie Finder, Recommendations for Improvement	Sarah Stallman

Introduction

Movie Recommendation Background

Recommendation System Foundations

- Decision Fatigue
- Maximizing Engagement
- Recommendations Based on Preference of Similar Users
- Business Scalability
- Previous Challenges – Cold Start Problem

Project Approach

Client Objective:
- Stand out in competitive streaming market.

Content Strategy:
- Select popular movies, genre, based on ratings.

Market Gap:
- Address frustration with inconsistent content

Solution Approach:
- Collaborative and Content-Based Filtering

Project Motivation:
- Create stable, appealing movie library.

Outcome Goals:
- Drive growth, exceed user expectations.

Success Metrics

Popularity: Average User Rating, Rating Distribution, Top-Rated Movies

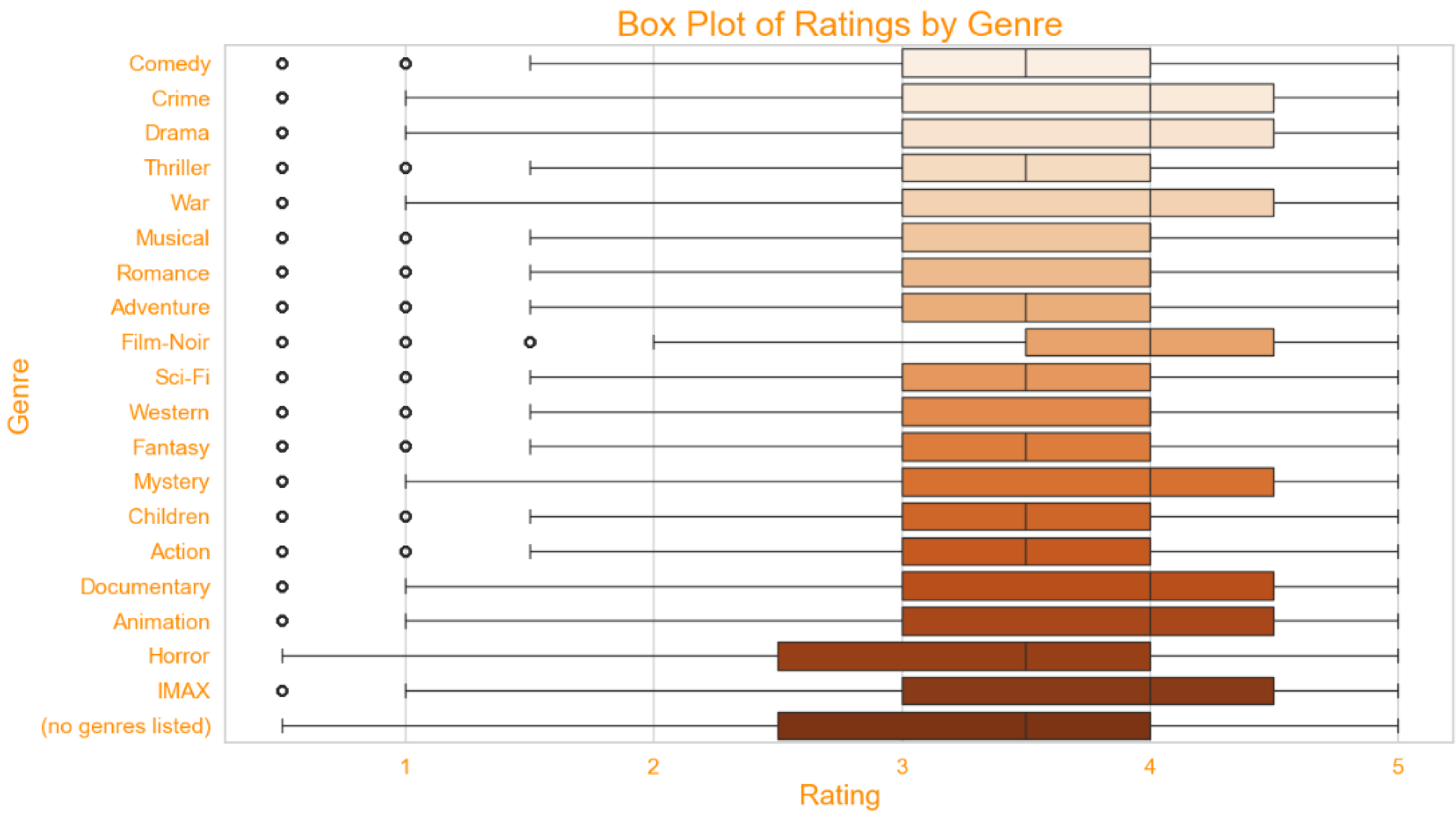
Engagement: Ratings per Genre, Generating 10 most relevant movies

Genre-Specific: Average Rating per Genre

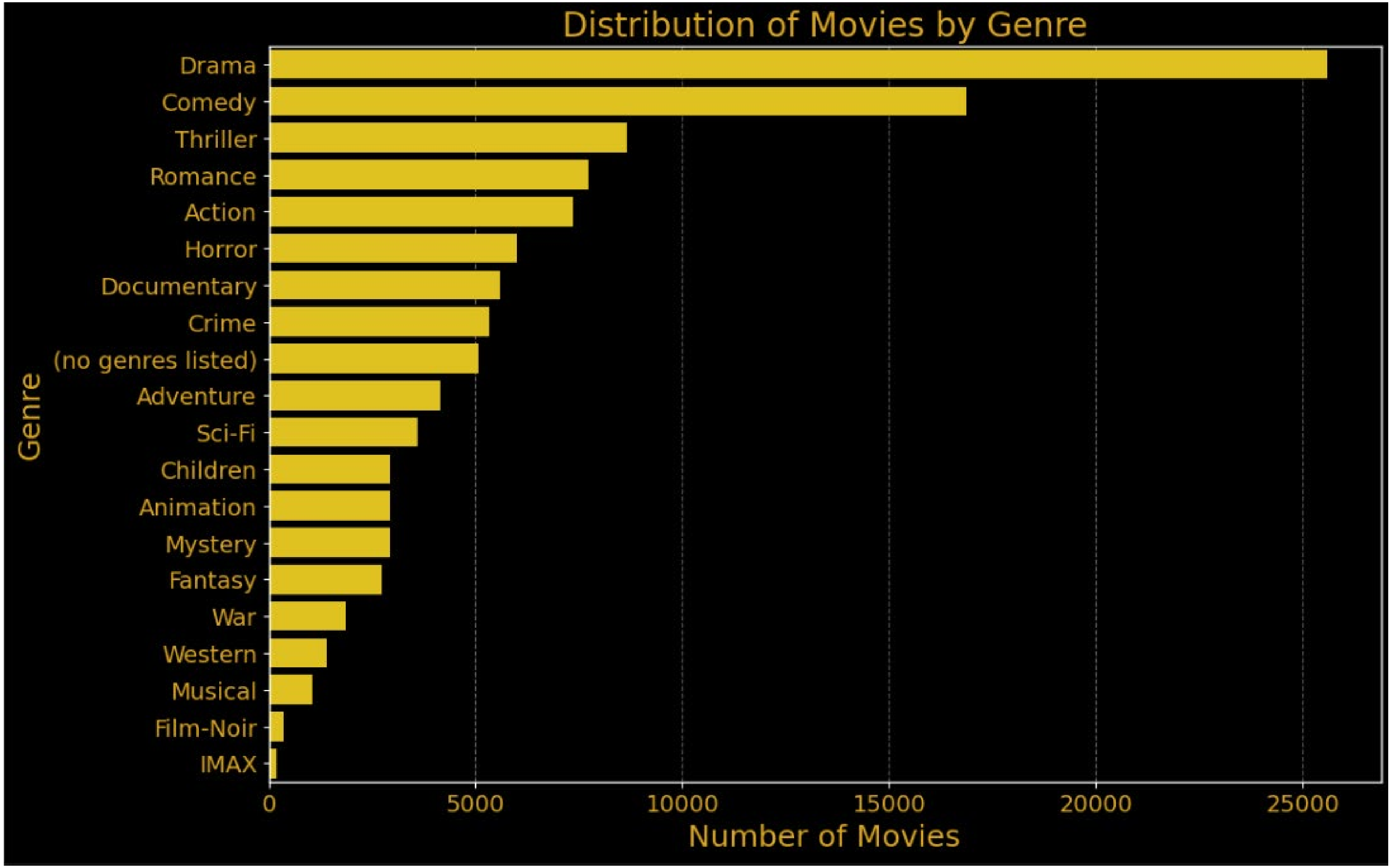
Dataset Insights

Some aspects worth noting

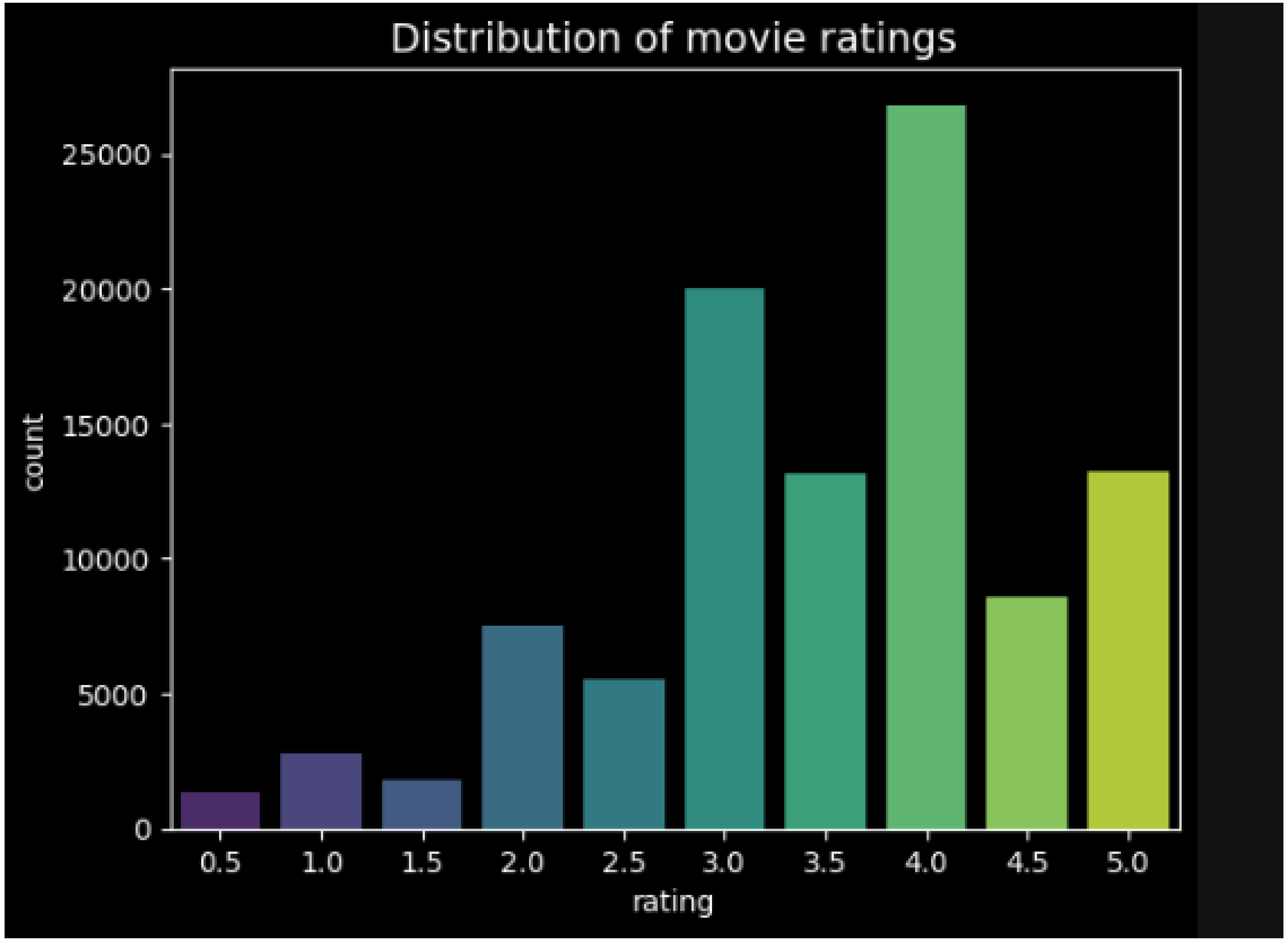
Dataset Insights



Dataset Insights



Distribution of Movie Ratings



AI Model

Data Preparation and Modelling

Intro to AI Model

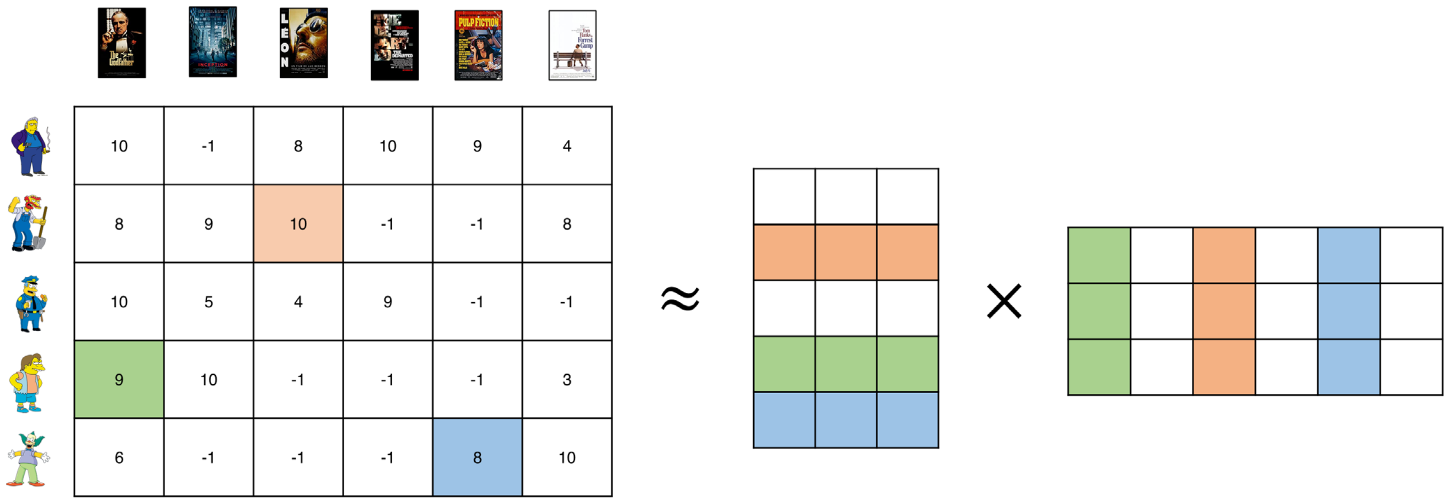
This model is centered around generating movie recommendations, taking a movie as an input, and generating a list of related movies as an output. We developed two approaches for accomplishing this

Collaborative Filtering	Content-based Filtering
• Uses **k-Nearest Neighbors** • Identifies movies that a user may like, based on the review history of other users • If a specific user has rated two movies highly, it uses one as a recommendation for the other	• Uses **Cosine & Euclidian** similarity , based on the movie genres. • Is able to recommend content with few reviews (handling the Cold-Start Problem).

Data Pre-processing

Purpose:

- Data pre-processing is a critical step in preparing the dataset for collaborative filtering.
- Specifically, we created a user-item matrix - where rows represent users and columns represent movies.
- This matrix will be used to identify patterns and generate recommendations.



10	-1	8	10	9	4
8	9	10	-1	-1	8
10	5	4	9	-1	-1
9	10	-1	-1	-1	3
6	-1	-1	-1	8	10

$\approx$ $\times$

User-item Interaction Matrix (R)

User Matrix (Q)

Item Matrix (P)

Benefits:

Efficient Storage

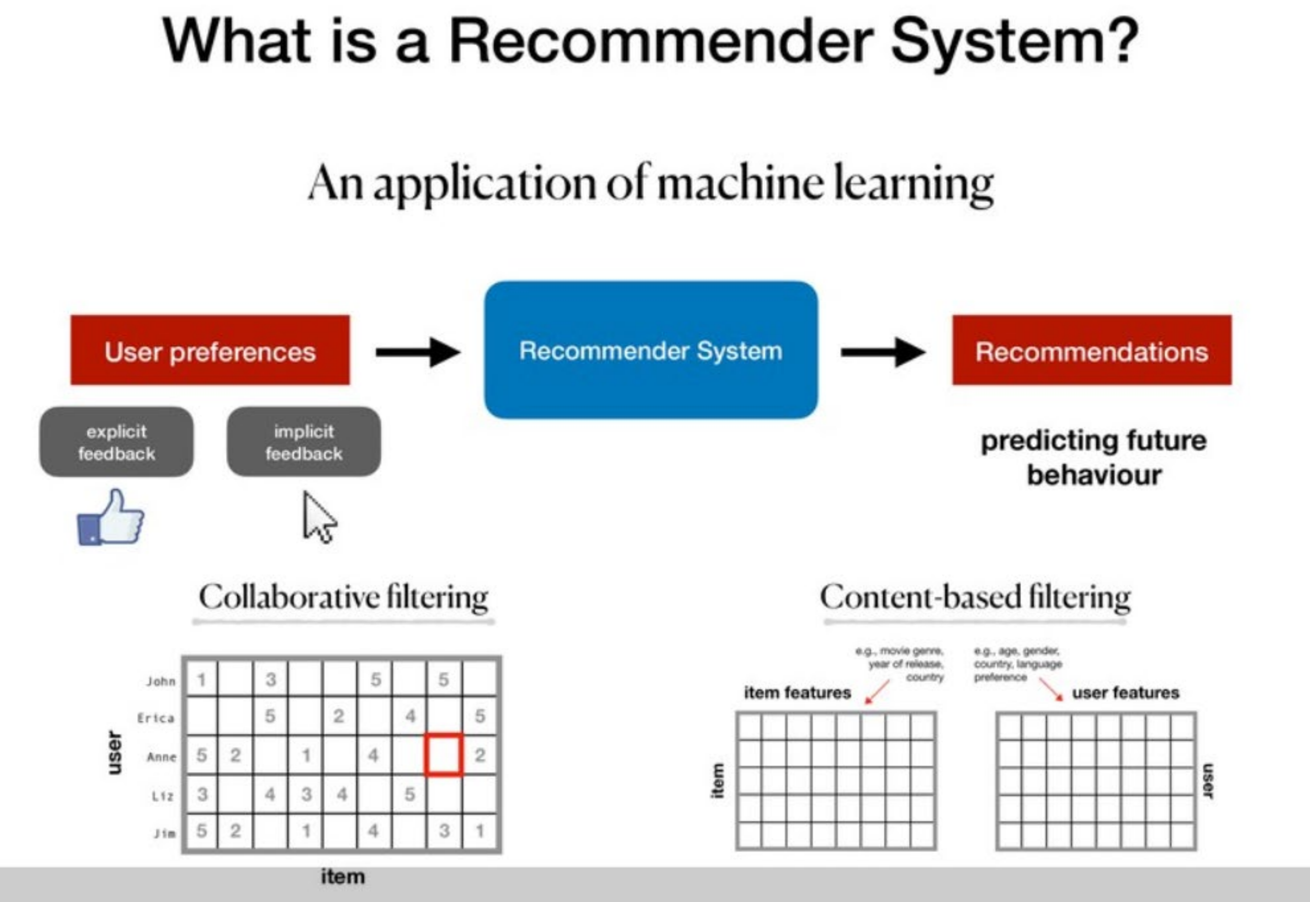
Fast Lookups

Scalability

Item-Item Recommendations with k-nearest Neighbors

Purpose:

The primary purpose of this step is to implement item-item collaborative filtering using the k-nearest neighbors (k-NN) algorithm to recommend movies that are similar to a given movie based on user engagement vectors.



Benefits:

- Improved User Experience
- Efficient Conputation
- Scalability
- Personalization

Cosine Vs Euclidean Similarity

Cosine Similarity

```
Because you watched Toy Story (1995):
Toy Story 2 (1999)
Jurassic Park (1993)
Independence Day (a.k.a. ID4) (1996)
Star Wars: Episode IV - A New Hope (1977)
Forrest Gump (1994)
Lion King, The (1994)
Star Wars: Episode VI - Return of the Jedi (1983)
Mission: Impossible (1996)
Groundhog Day (1993)
```

Euclidean Similarity

```
Because you watched Toy Story (1995):
Toy Story 2 (1999)
Mission: Impossible (1996)
Independence Day (a.k.a. ID4) (1996)
Bug's Life, A (1998)
Nutty Professor, The (1996)
Willy Wonka & the Chocolate Factory (1971)
Babe (1995)
Groundhog Day (1993)
Mask, The (1994)
```

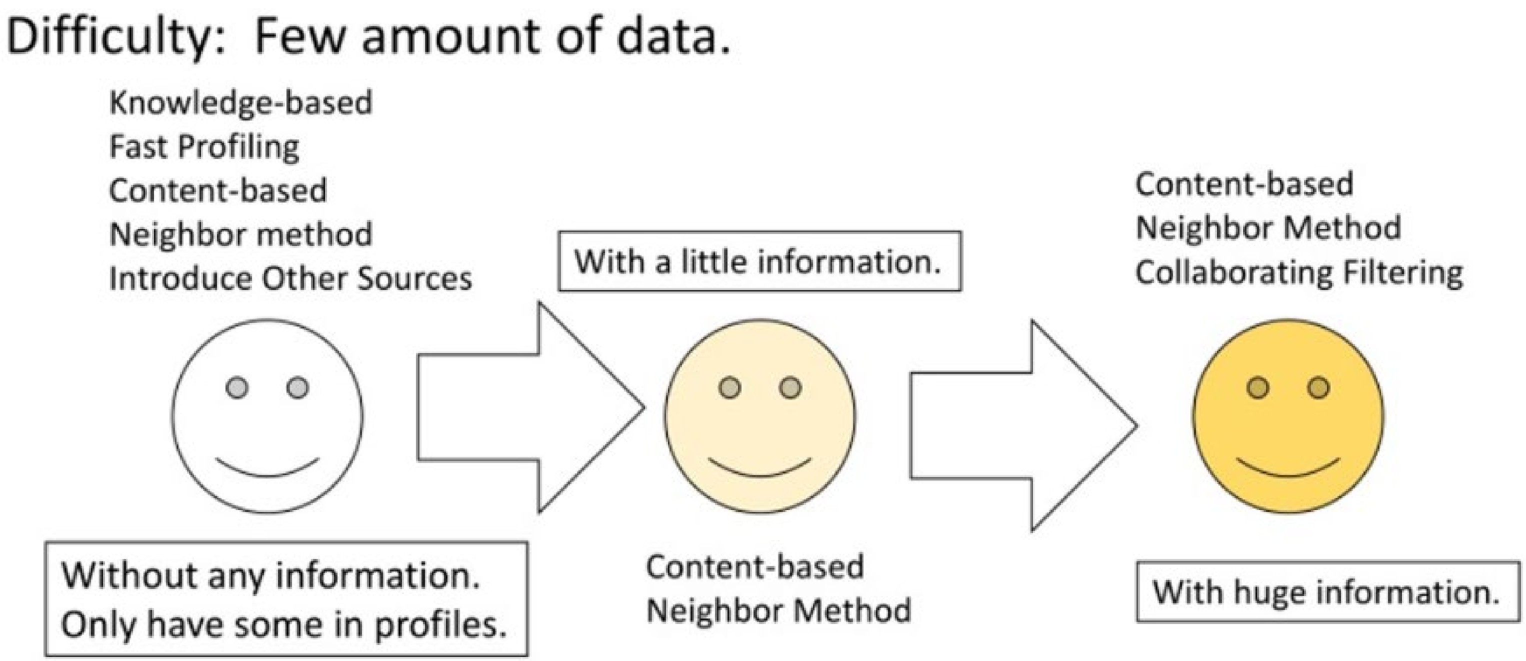
Handling the cold-start problem

Problem

- Collaborative filtering relies solely on user-item interactions within the utility matrix.
- Brand new users or items with no interactions get excluded from the recommendation system.

Solution

- Content-based filtering to generate recommendations based on user and item features.
- Convert the genres column into binary/boolean features.



Creating a Movie Finder

Problem	Solution
• To get results from our recommender, we need to know the exact title of a movie in our dataset. • Recommender can't identify misspelled movie titles or if year of release is not included.	• Python package fuzzywuzzy: Finds the most similar title to a passed in string.

Data Challenges

Data Sparsity
- User-item matrices are often sparse.
- Difficulty finding sufficient data to generate accurate recommendations.

Cold Start Problem
- New users and movies lack sufficient interaction data.
- User and item cold start.

Diversity vs. Accuracy
- More accuracy may lead to less diversity.
- Users may receive similar types of movies repeatedly.

Evaluation & Metrics
- Subjective nature of user satisfaction.

Recommendations for Improvement

Increase Coverage

- Coverage should be a proportion between 0 and 1
- Reviewing the calculation ensures that it accurately reflects the recommendation diversity.

Enhance Content-based Filtering

- Hit rate
- Improve Ranking Quality
- Incorporation of additional features improves the relevance of recommendations.

Regression Metrics

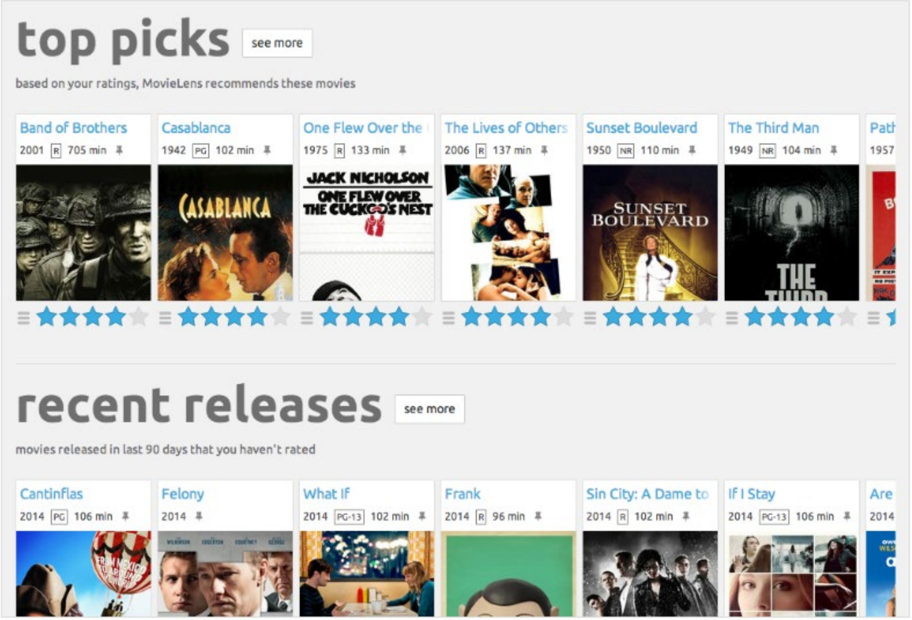
- **(MAE and RMSE):** The very low values of MAE (0.14) and RMSE (0.15) suggest that the predicted ratings are highly accurate and close to the actual ratings.

Classification Metrics

- **(Precision, Recall, F1-Score, AUC):**
- The perfect scores (1.0) for precision, recall, F1-Score, and AUC indicate that the recommendation system is performing exceptionally well in distinguishing and recommending relevant items without any errors.

Improving Ranking Quality

- Both MAP@K and NDCG@K are 0.0
- Consider tuning recommendation algorithms by adjust # of k-neighbors or different distance metrics.



Deloitte.

Questions?