

# Building the AI Movie Recommender

SEMPALA DANIEL

2023/BCS/118/PS

ALOYSIUS OWEN JUUUKO

2023/BCS/029/PS

TIBASAGA JAIZAH

2023/BCS/125/PS

OWOMUGISHA MARIA TRACY

2023/BCS/157/PS

LUKYAMUZI ABUBAKAR HUSSEI

2023/BCS/071/PS

NANKYA SOPHIA

2023/BCS/093/PS

“Imagine Netflix knowing exactly what you want to watch tonight — that's what we built.”

# Too Many Choices!

Have you ever...?

- Spent 30 minutes scrolling and just gave up?
- @ Watched something random and hated it?
- @ Wished a friend could just pick for you?

The Overwhelming Choice



**15,000+**

Netflix: movies

**24,000+**

Amazon Prime: movies

**100,000+**

All Streaming: movies

Our Quest

To build a system that learns **YOUR** taste and suggests movies **YOU'LL** actually enjoy.

# Our Dataset: The MovieLens Dataset

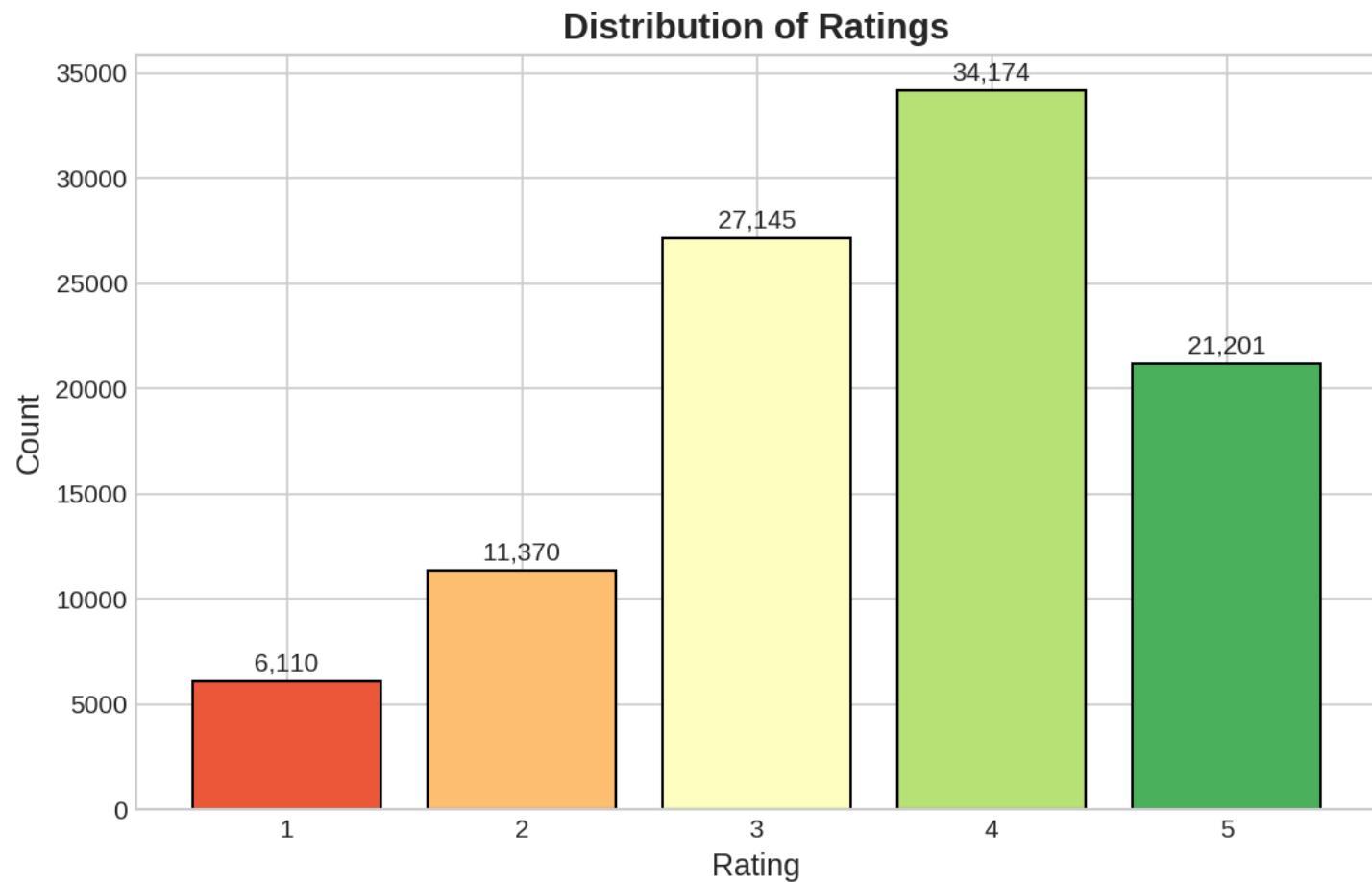
A research project by University of Minnesota where real people rated real movies

What	How Much
Users	162,000
Movies	62,000
Ratings	25M
Time Span	1900-2000s

*"Quality data is our foundation. 25 million real ratings are far more valuable than 100 million fake ones."*

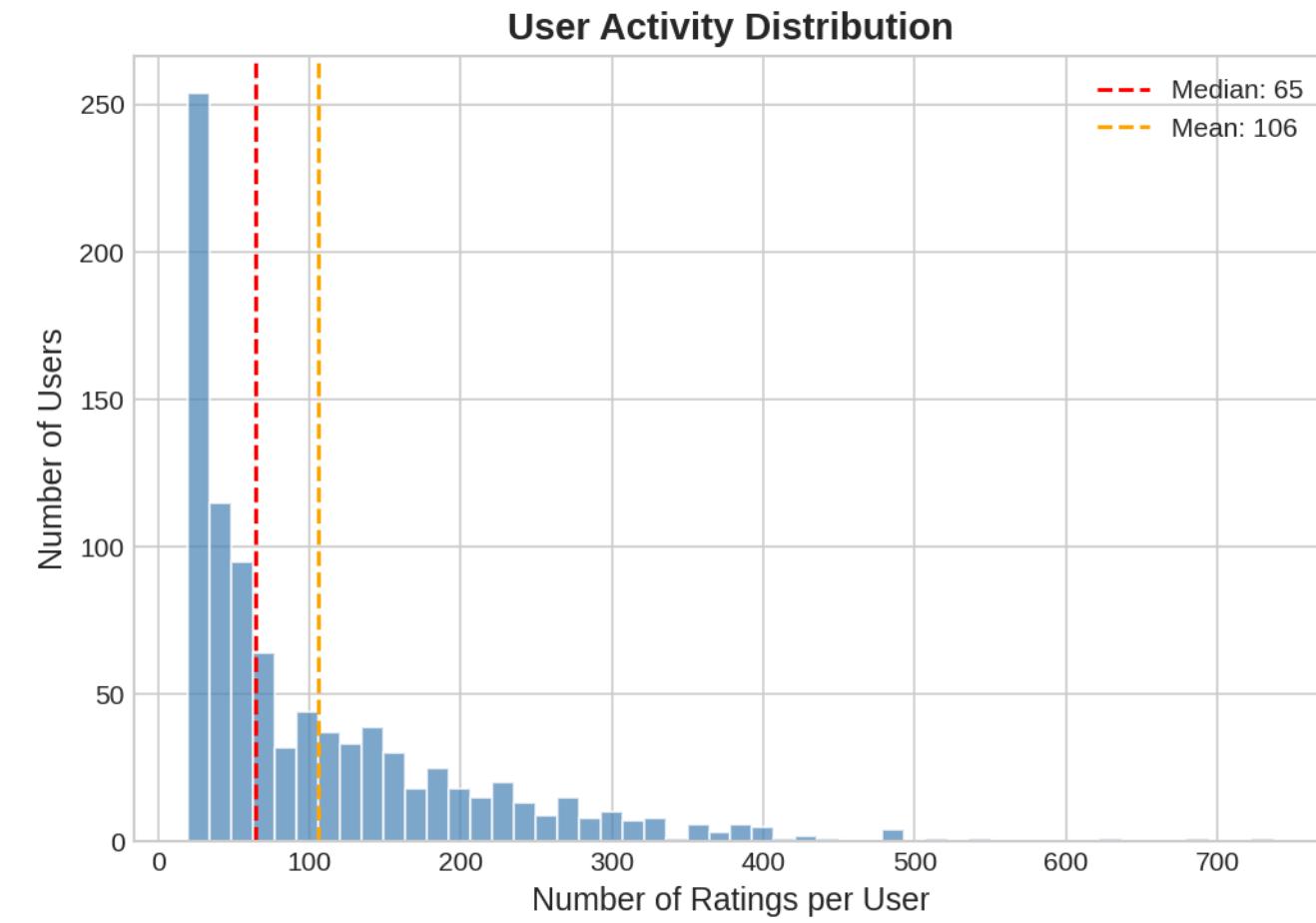
# Exploring Our Data, Part I

Insight: People are generous raters, with 4 stars being the most common choice.



Key Stat: The average rating is 3.5 stars. This gives us a simple but important baseline.

Insight: Activity is highly skewed. A few “super users” provide tons of data, while many provide very little.

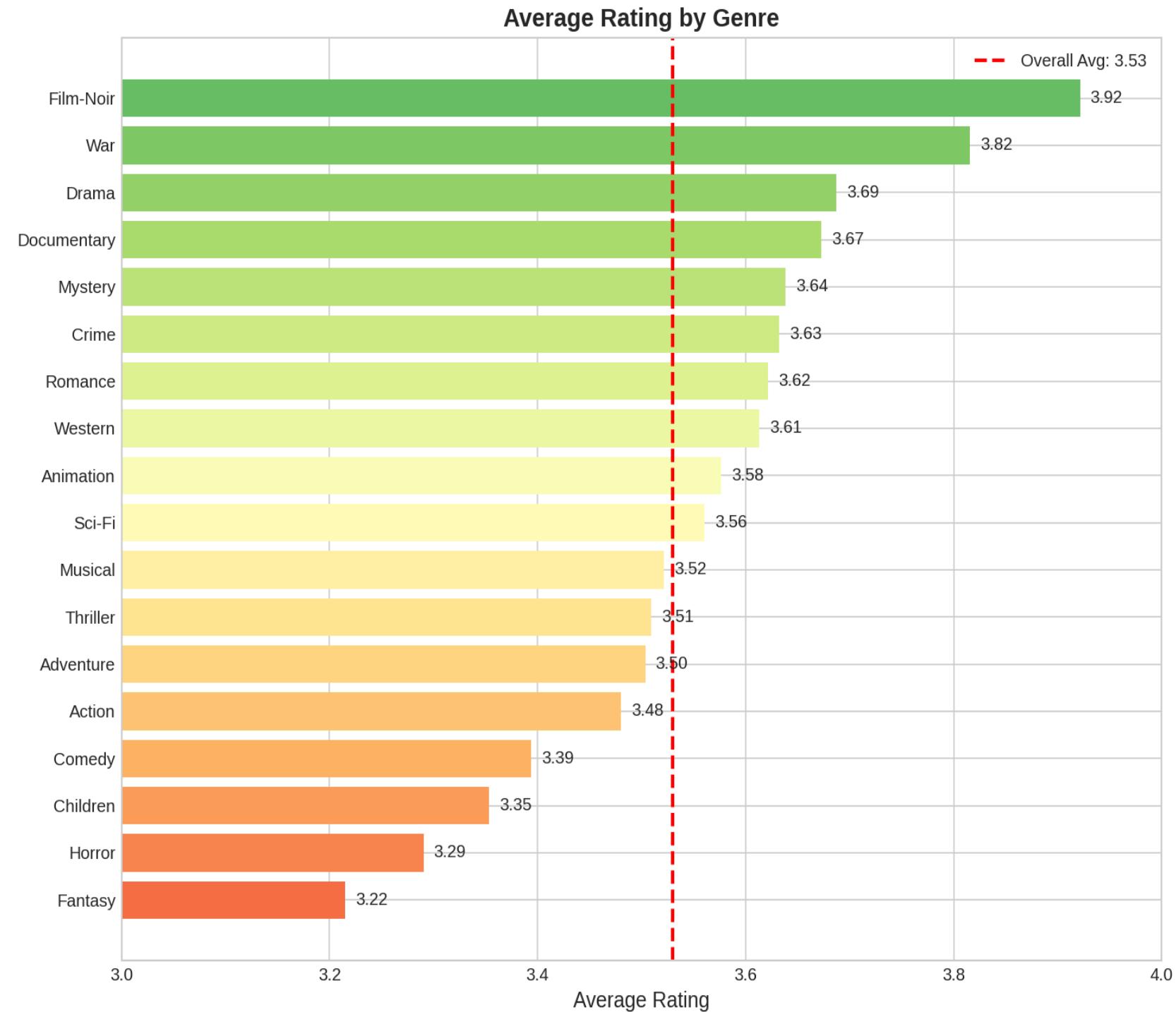
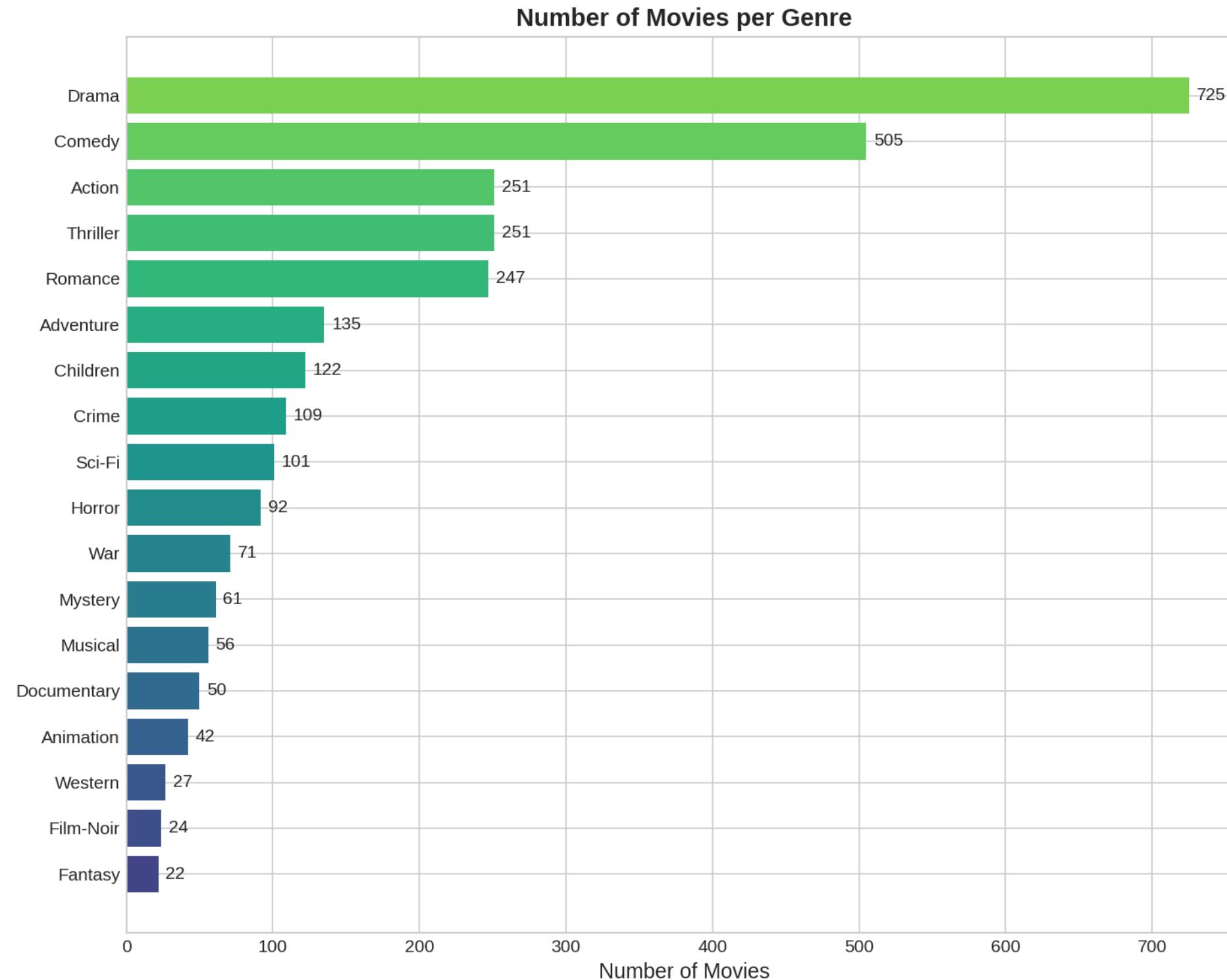


Key Challenge Introduced: This is where we first encounter the Cold Start Problem: How can we recommend anything to a new user with zero ratings?

# Exploring Our Data, Part II

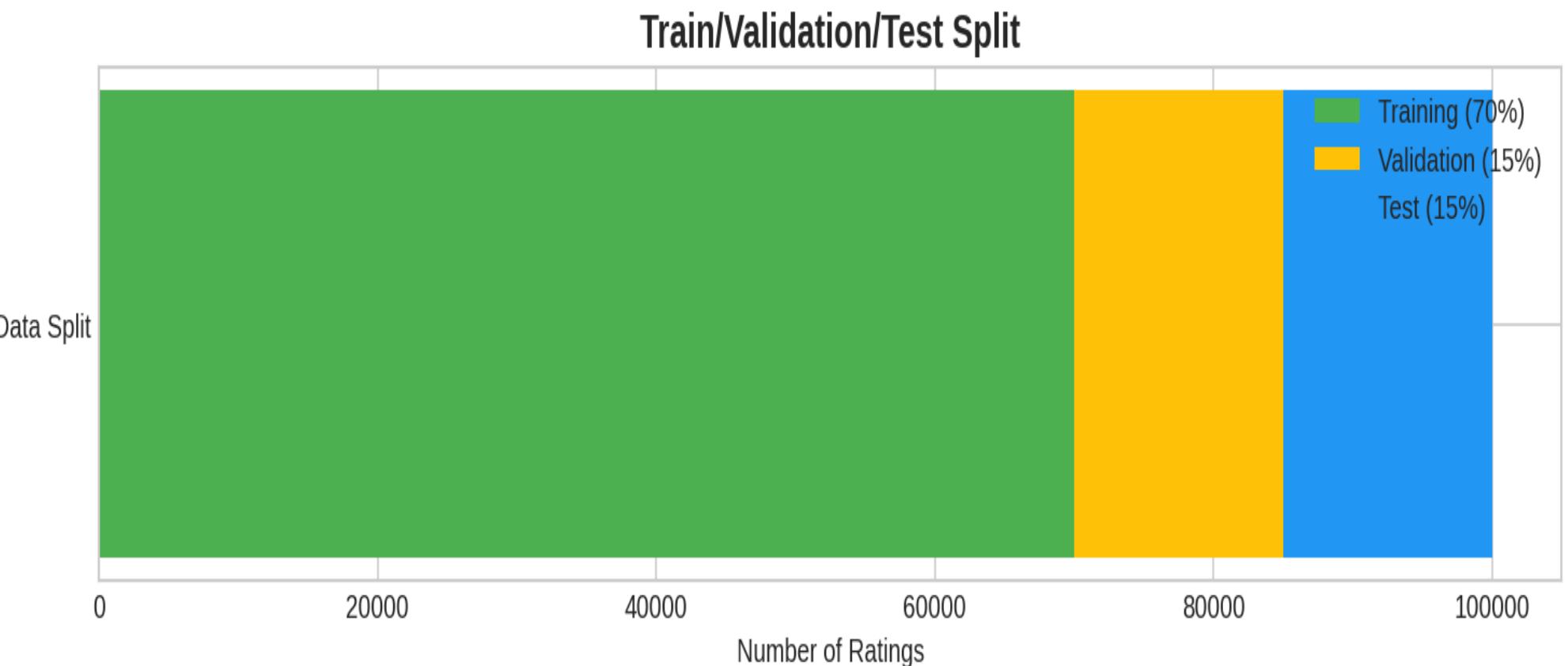
Insight: The catalog is dominated by Drama and Comedy, which means our model will see far more

Insight: A genre's average rating doesn't define its quality for a fan. A 4-star rating for a horror fan is a major success.



# Data Preprocessing

Step	What
1	Removed movies with < 5 ratings
2	Removed users with < 5 ratings
3	Extracted year from title
4	Split genres into list
5	Normalized ratings



# CineMind AI Architecture

Supervised Learning → Hybrid Recommendation Engine



# Approach #1: Learning from the Crowd with Collaborative Filtering(CF)

The guiding principle: People who agreed in the past are likely to agree in the future.



## The Technology: Singular Value Decomposition (SVD)

SVD is a powerful mathematical technique that distills a massive, sparse matrix of user ratings into dense, meaningful patterns about user tastes and movie characteristics. It's how we find "similar users" at scale.

Big Sparse Matrix → Two Smaller Dense Matrices  
(mostly empty) (filled with learned patterns)

# The SVD Algorithm

	<b>Movie1</b>	<b>Movie2</b>	<b>Movie3</b>	<b>Movie4</b>	.....	<b>Movie62000</b>
<b>User1</b>	4	?	?	2	.....	?
<b>User2</b>	?	5	?	?	.....	3
<b>User3</b>	3	?	4	?	.....	?
.....	.....	.....	.....	.....	.....	.....
<b>User162000</b>	?	?	?	?	.....	?

## Prediction:

$$\begin{aligned}
 \text{Rating(User1, Movie5)} &= \text{User1\_vector} \cdot \text{Movie5\_vector} \\
 &= [0.2, 0.8, \dots] \cdot [0.4, 0.6, \dots] \\
 &= 3.7 \text{ stars (predicted!)}
 \end{aligned}$$

Original Matrix (Users × Movies):

4 ? ? 2 ? ...	162,000 users
? 5 ? ? 3 ...	x
3 ? 4 ? ? ...	62,000 movies

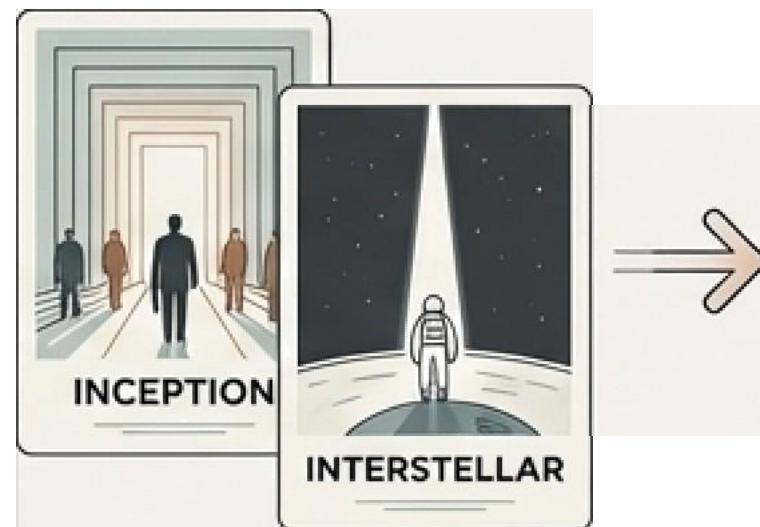
Becomes:

$$\begin{array}{ccc}
 \text{User Matrix} & \times & \text{Movie Matrix} \\
 \left[ \begin{array}{c} u_1: [0.2, 0.8, \dots] \\ u_2: [0.9, 0.1, \dots] \\ \dots \end{array} \right] & \times & \left[ \begin{array}{c} m_1: [0.3, 0.5, \dots] \\ m_2: [0.7, 0.2, \dots] \\ \dots \end{array} \right] \\
 (162K \times 100) & & (100 \times 62K)
 \end{array}$$

# Approach #2: Learning from the Movie's DNA with Content-Based Filtering

The guiding principle: If you liked a specific movie, you'll probably enjoy other movies with similar features (genre, actors, keywords).

This approach works perfectly for new movies with zero ratings and is crucial for solving the cold-start problem.



dream  
**dream** ••  
heist Space time  
reality



## 1. Turning Movies into Numbers (TF-IDF)

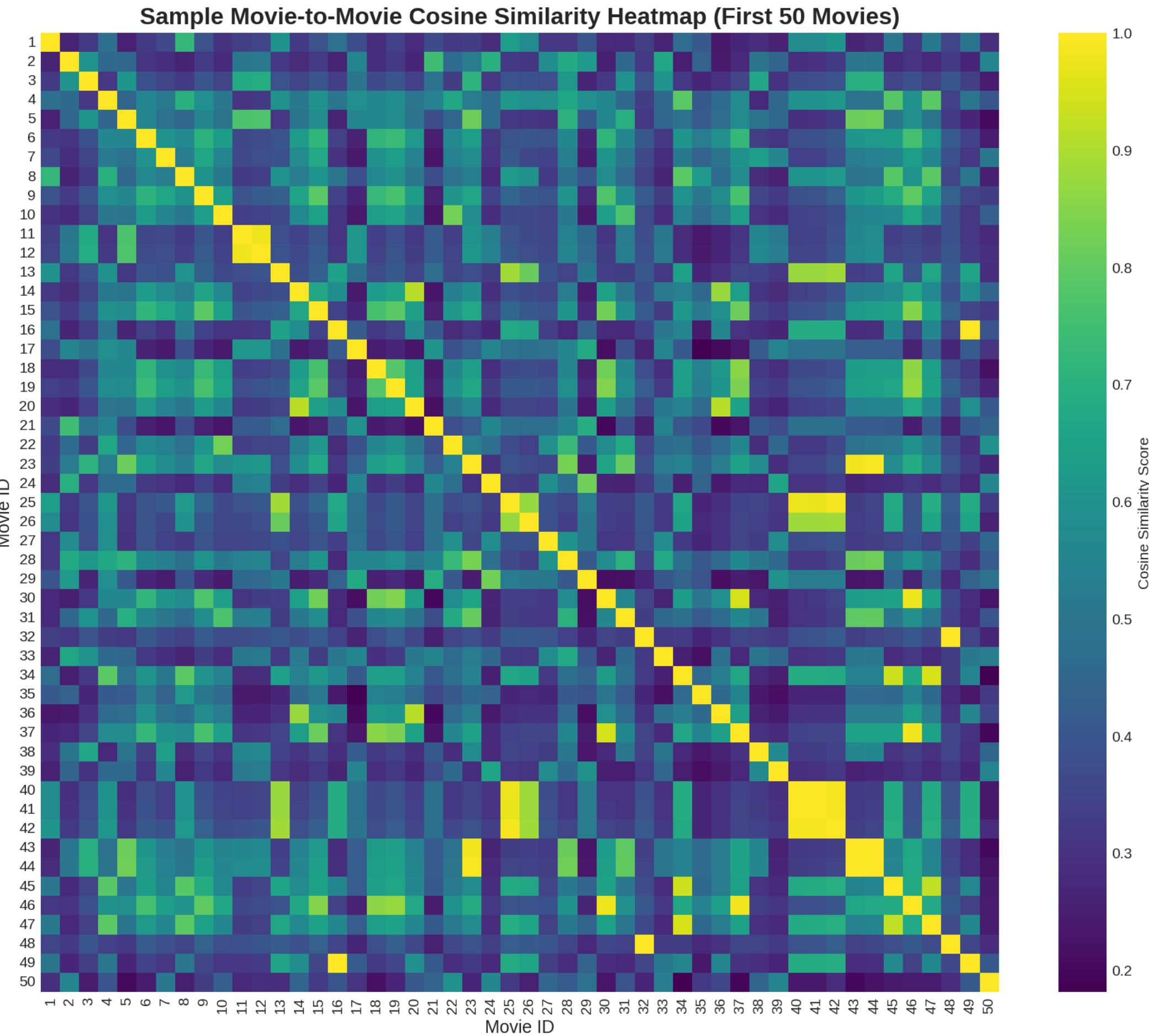
We use TF-IDF to convert movie features like 'dream' and 'heist' into a meaningful numerical vector. Important words get a higher score.

## 2. Measuring Similarity (Cosine Similarity)

We then calculate the 'angle' between two movie vectors. A smaller angle means they are more similar.

Similarity(*Inception*,  
*Interstellar*) = 0.85  
(Very Similar!)

# Cosine Similarity



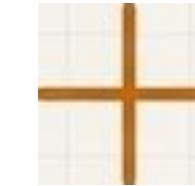
# The Architectural Innovation: A Hybrid System For the Best of Both Worlds

The Dilemma.

## Collaborative Filtering

Strength: Uncovers surprising, serendipitous recommendations.

Weakness: Fails on new users and new movies (the “cold start” problem).



## Content-Based Filtering

Strength: Solves the cold start problem and works with item metadata. O

Weakness: Can get stuck in a “similarity bubble,” only recommending very similar items.

## The Solution

A weighted blend that leverages the strengths of both models.

$$\text{**Final Score} = (0.7 \cdot \text{Collaborative Score}) + (0.3 \times \text{Content Score})^{**}$$

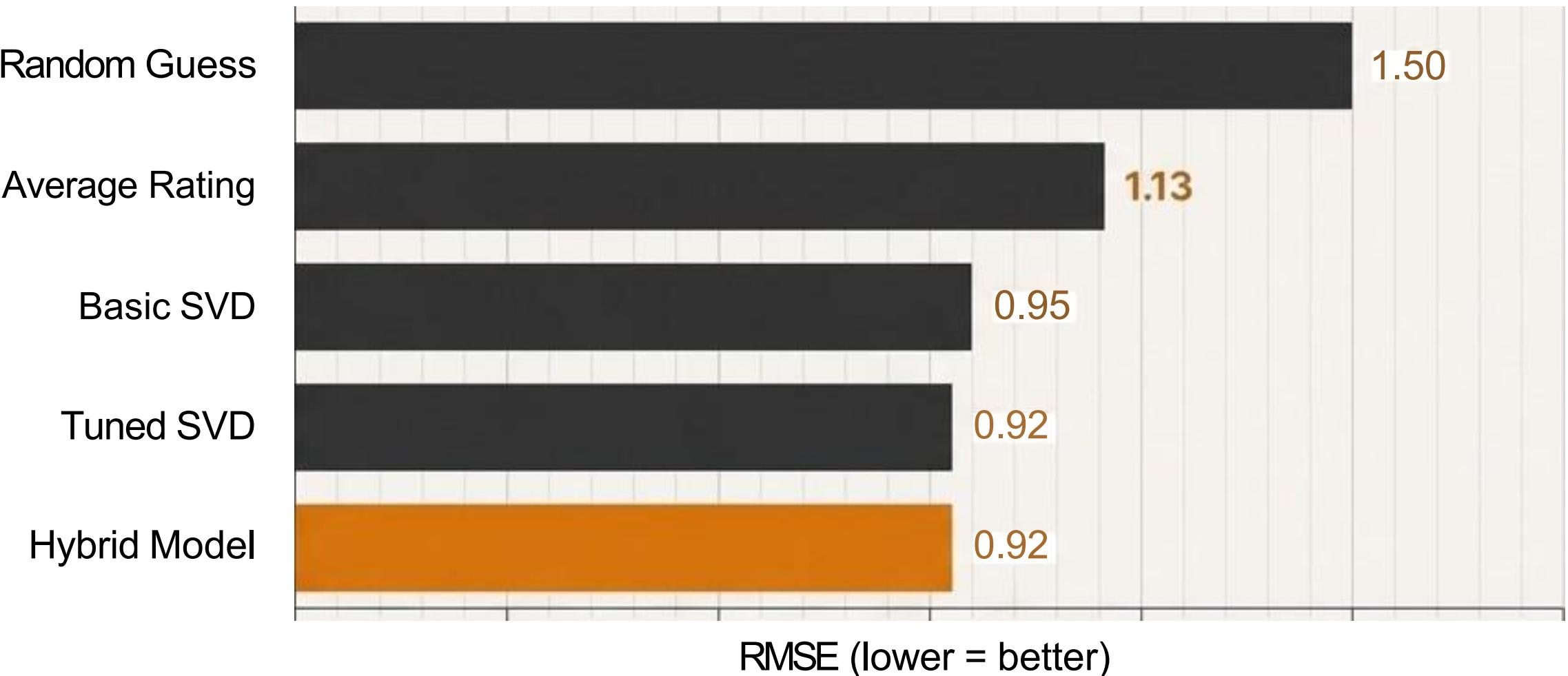
The Justification: We tested multiple weightings, and a 70/30 split in favor of the collaborative model gave us the lowest prediction error.

# The Verdict, Part I: Our System Predicts Ratings with High Accuracy

A performance comparison of recommendation models using Root Mean Square Error (RMSE).

## Metric Explained

Root Mean Square Error (RMSE) answers: “On average, how many stars off is our prediction?” (Lower is better).



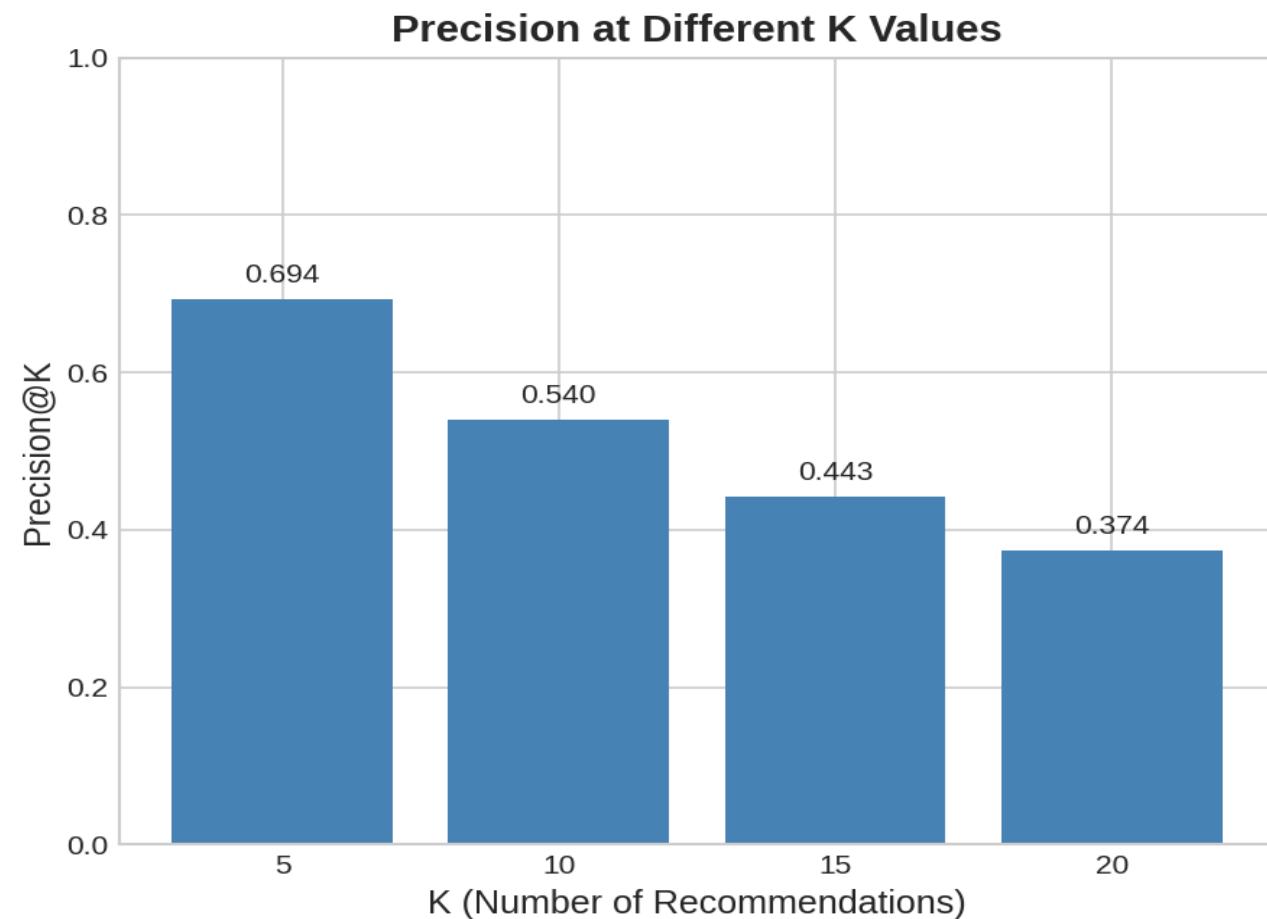
## (So What?

Our final RMSE of 0.92 means we're off by less than one star on average. For context, the winning Netflix Prize model achieved around 0.85 RMSE, placing our model in a highly competitive performance bracket.

# The Verdict, Part II: Recommendations are Relevant and Discoverable

## The Questions

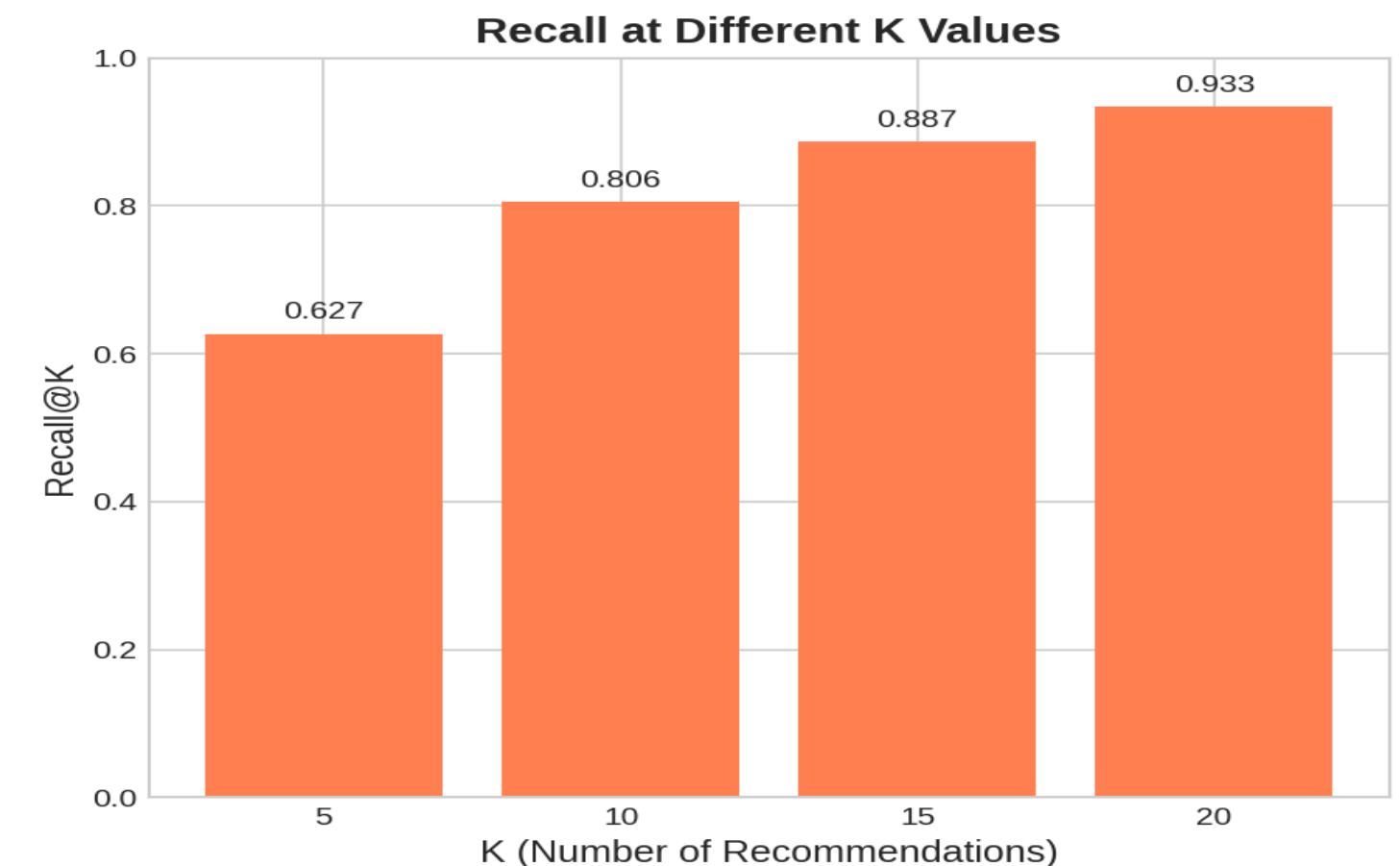
1. Precision: Of the 10 movies we recommend, how many are actually good?



Precision@10: 54%

*Interpretation: Over half of the movies in our top 10 list are ones the user will actually like.*

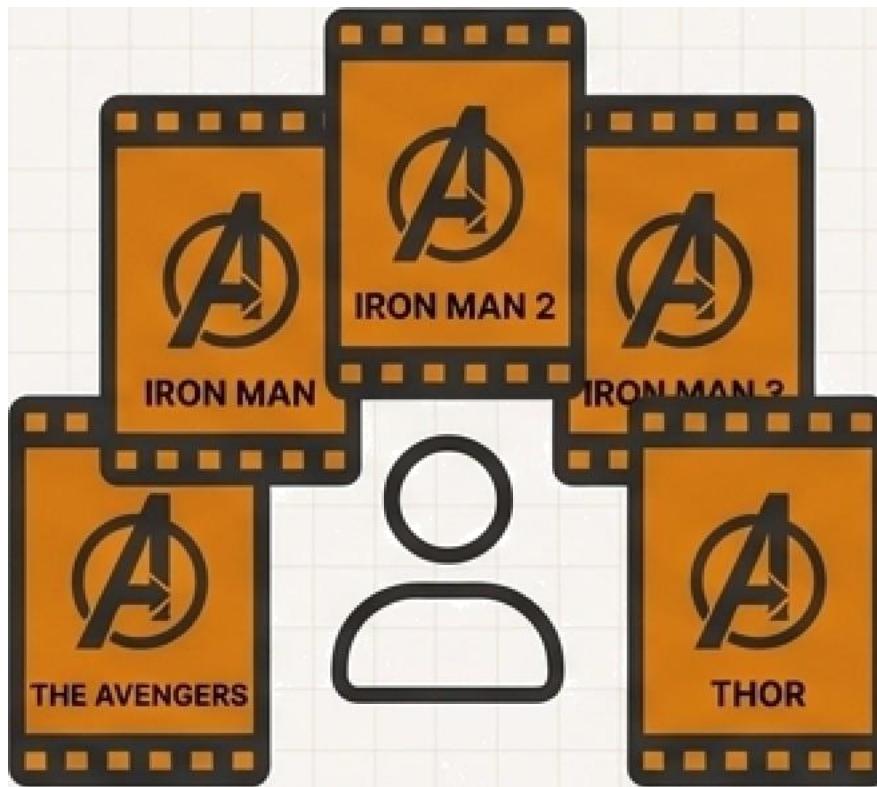
2. Recall: Of all the movies the user would like, how many did we find in our top 10?



Recall@10: 80.6%

*interpretation. Our top 10 list successfully finds over 80% of the movies a user would have enjoyed.*

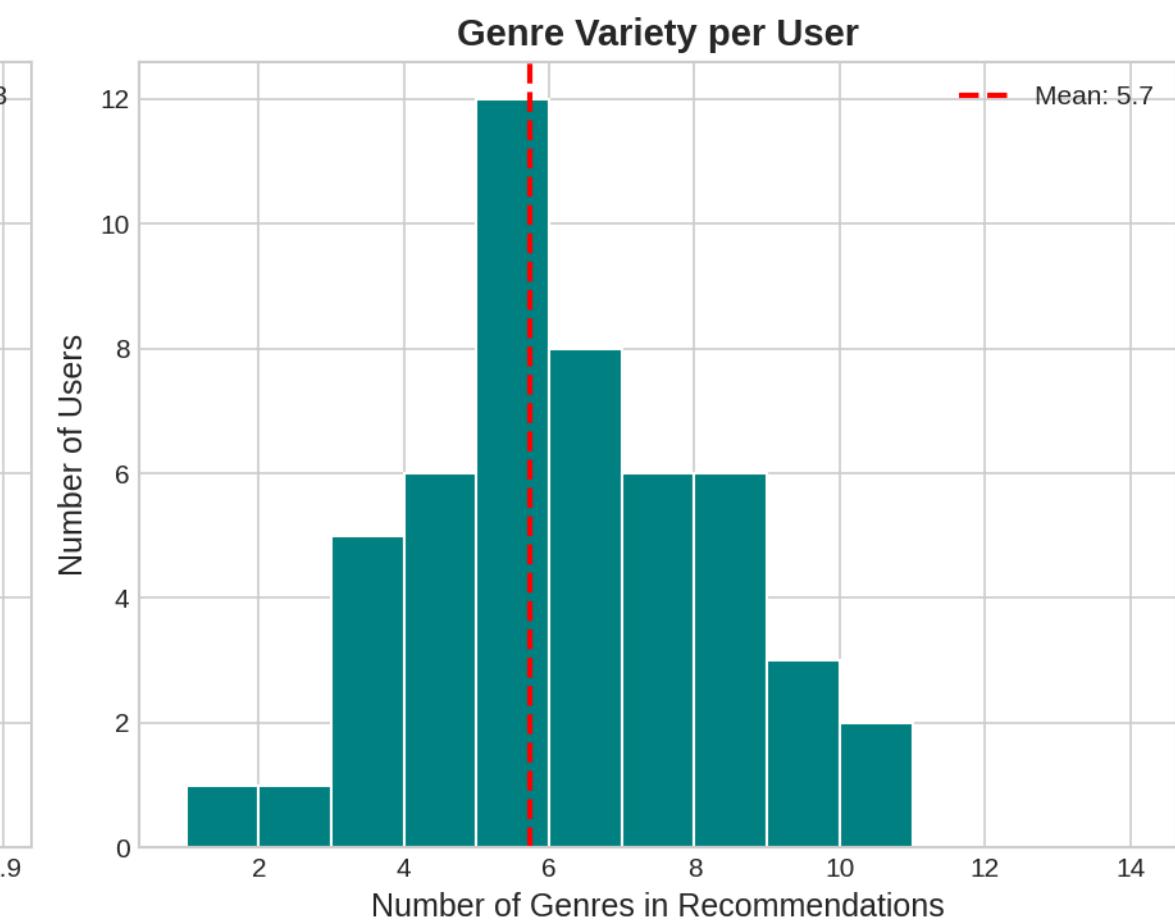
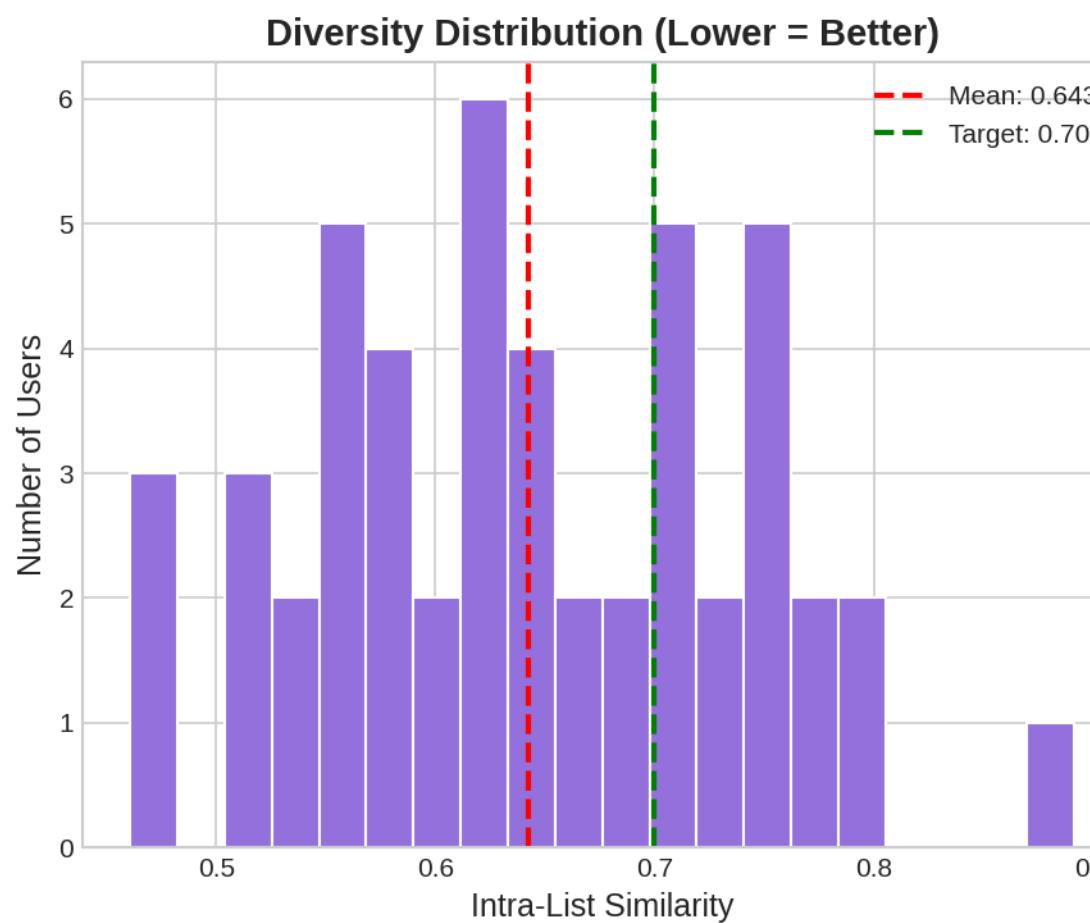
# The Final Test: Building a System That Avoids the Similarity Bubble



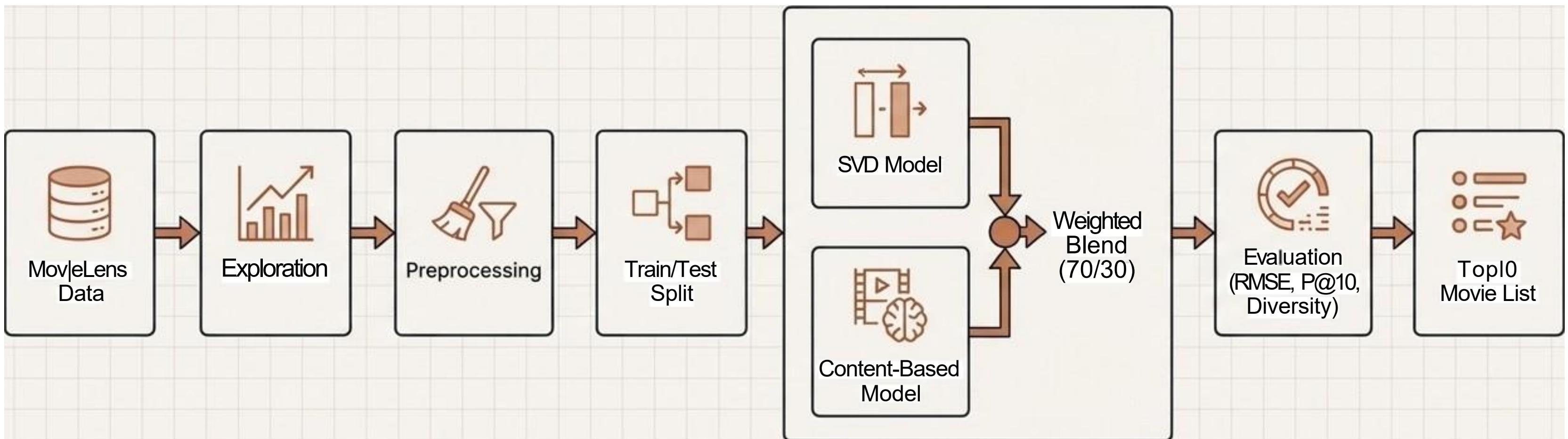
A purely accurate model might just recommend *iron Man 1, 2, 3, The Avengers, and Thor*. That's accurate, but it's a terrible user experience.

## How We Measure Diversity

**Metric: Intra-List Similarity (ILS).** A lower score means the items in a recommendation list are more varied and interesting.



# The Final Blueprint: **our** End-to-End Recommendation Pipeline



From 25 Million Raw Ratings to a Personalized Top-10 List.