April 21, 2025
Berkin, Sarah

Capstone Project Report – Movie Recommendation System

1. Problem Statement

According to a survey conducted by UserTesting [1], the average American spends **over 110 hours per year** scrolling through streaming platforms, struggling to find something to watch. Additionally, **52% of users** say a streaming service's browsing experience significantly influences their decision to subscribe, and **56%** cancel subscriptions after completing a single show. [2] The objective of this project is to build a **movie recommendation system** that **reduces scrolling time**, **improves user satisfaction**, and **lowers churn rates** for streaming platforms.

2. Data Description

The data for this project originated from the GroupLens research group at the Department of Computer Science and Engineering at the University of Minnesota. They built a movie platform, called MovieLens, in order to gather data and further build out ML models of their own. Their **MovieLens 1M dataset [3]** contains:

- 1,000,209 anonymous ratings
- 6,040 MovieLens users who joined MovieLens in 2000
- 3,900 movies

User Features:

- Gender
- Age Group
- Occupation

Movie Features:

- Genres (multi-label, one-hot encoded)
- Release Year

The target variable for prediction was **user rating** on a scale of 0.5 to 5.0.

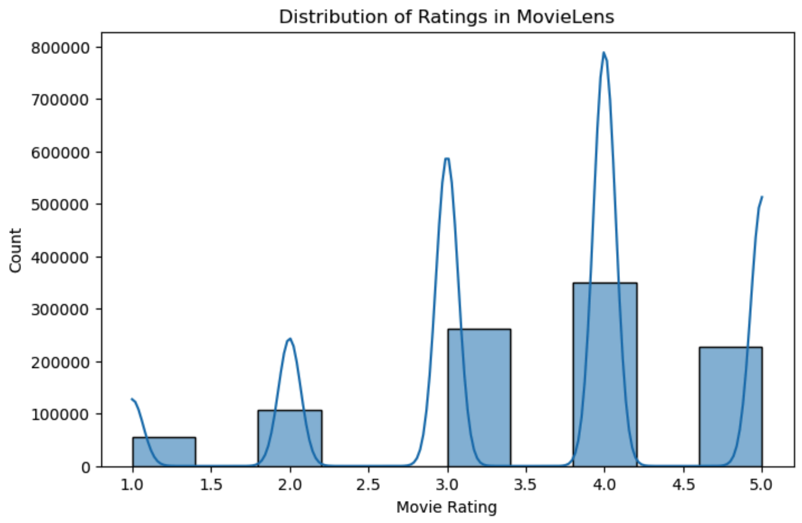
3. Methodology

The analysis followed these key steps:

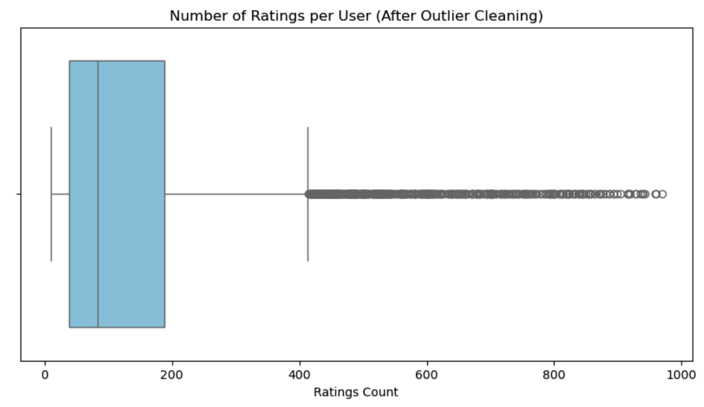
- **Data Cleaning**: Prior to running any analyses, I examined my data to find any issues that would cause problems later. Here's what I found and the action I took to solve them:

- ○ **Problem 1:** The title field contained mixed text (movie names and release years) which was unsuitable for direct modeling.
- ○ **Solution 1:** Extracted the release year from movie titles using regular expressions and created a separate year feature.
- ○ **Problem 2:** The genres field contained multiple genres in a single string (e.g., "Action|Adventure|Sci-Fi"), making it difficult to use directly in modeling.
- ○ **Solution 2:** Applied one-hot encoding to expand genres into individual binary columns.
- ○ **Problem 3:** User demographic fields (gender, occupation) were categorical and not machine-readable.
- ○ **Solution 3:** Applied one-hot encoding to these fields and created an age_group feature by binning ages into logical buckets (e.g., 18-24, 25-34).

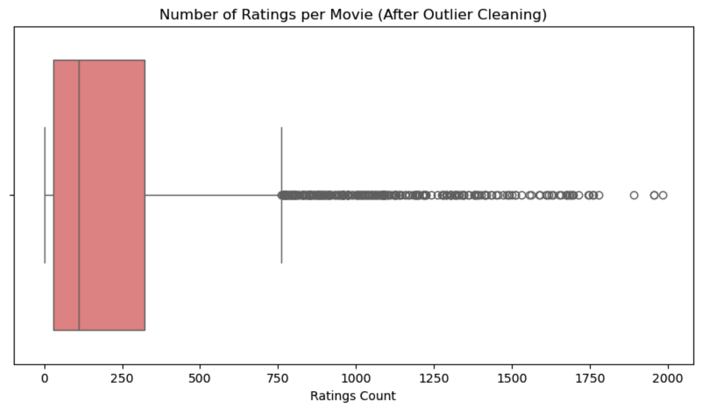
- **Exploratory Data Analysis (EDA)**:
  - Histograms visualized rating distributions. If there's an extreme on either side (ex. ~90% above or below 3.5), then the dataset may be imbalanced. Results showed 57% of ratings were 3.5 or higher.



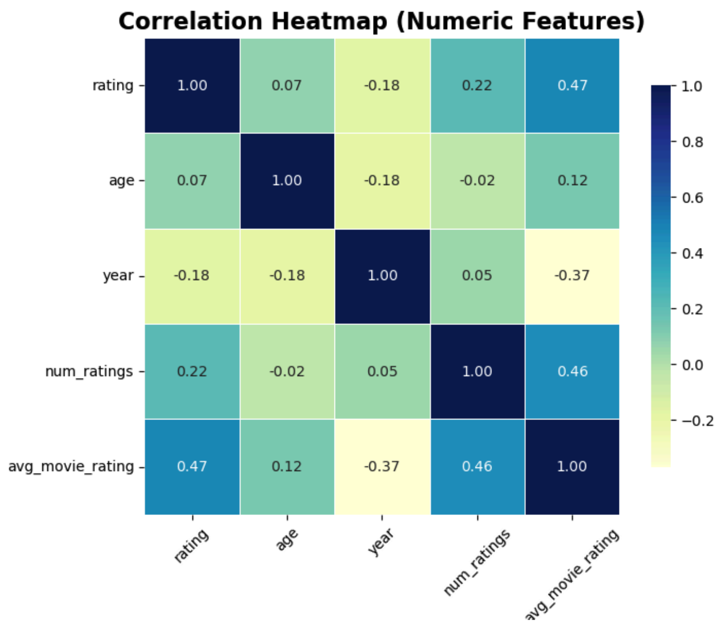
  - Boxplots detected rating count outliers. Extreme outliers existed in the number of ratings per user and the number of ratings per movie (super-users and blockbuster movies), skewing the data. Trimmed extreme outliers based on IQR method and boxplot analysis:
    - ■ Trimmed users with more than 1,000 ratings (approx 5x the 75th percentile 208, to preserve 5,999 active users)



- Trimmed movies with more than 2,000 ratings (approx 6x the 75th percentile 329, to preserve 3,661 movies)



- Correlation heatmaps identified weak relationships between user age, movie release year, and ratings.



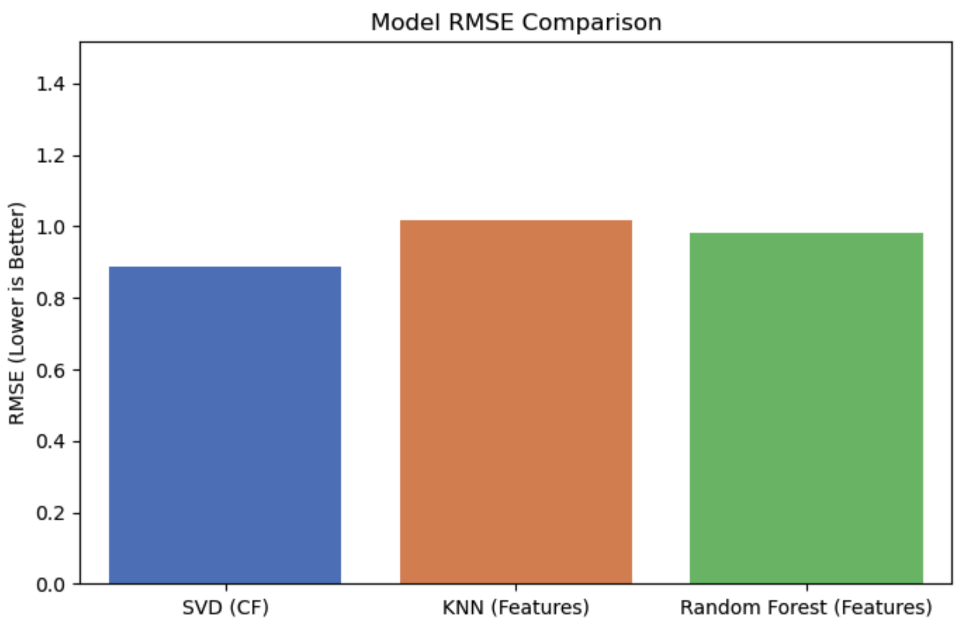
	rating	age	year	num_ratings	avg_movie_rating
rating	1.00	0.07	-0.18	0.22	0.47
age	0.07	1.00	-0.18	-0.02	0.12
year	-0.18	-0.18	1.00	0.05	-0.37
num_ratings	0.22	-0.02	0.05	1.00	0.46
avg_movie_rating	0.47	0.12	-0.37	0.46	1.00

    - Checked how many low (1-2 score) ratings were present to see if it makes sense to explore this as a proxy for scrolling frustration. Behavior patterns suggested moderate scroll frustration in raw data, as 16.37% were low scores.

- **Feature Engineering & Pre-Processing**:
    - **Problem 1:** The categorical user features (gender, age_group, occupation) were non-numeric and could not be directly used by machine learning models.
    - **Solution 1:** Applied one-hot encoding to convert these categorical features into binary numeric columns.
    - **Problem 2:** Feature magnitudes varied widely (e.g., age vs. number of ratings), which could bias distance-based models like KNN.
    - **Solution 2:** Applied StandardScaler to standardize all numeric features (mean = 0, standard deviation = 1) for consistent scaling across features.
    - **Problem 3:** The dataset contained a mix of final numeric features and dummy variables, requiring a clean split into modeling inputs and target labels.
    - **Solution 3:** Separated the features (X) from the target variable (y, user rating) and performed an 80/20 train-test split to prepare for modeling and evaluation.

- **Modeling**: I used surprise **[4]**, scikit-learn **[5]**, and matplotlib **[6]** libraries for training and visualizing my recommendation system. Three models were trained and evaluated:

Model	**Description**
SVD (Collaborative Filtering)	Matrix factorization using userId and movieId
KNN Regressor	Content + demographic feature-based distance model
Random Forest Regressor	Ensemble model using full feature set
Train-test split: 80/20	



4. Findings

- Behavior simulation showed that without recommendations, users frequently rated multiple movies poorly before finding one they liked.
- With the SVD model, the top-10 recommendations included 10/10 highly-rated movies, demonstrating a significant improvement in the browsing experience.
- **Winning Model: The SVD collaborative filtering model outperformed both feature-based models.**

Model	RMSE (Lower = Better)
SVD (Collaborative Filtering)	**0.8869 (Best)**
KNN Regressor	**1.0165**
Random Forest Regressor	**0.9813**

5. Recommended Business Use Cases

1. **Reduce Subscription Churn by Improving Content Discovery**
   - Many users cancel subscriptions after finishing a specific show, reporting frustration with browsing for new content. [2]
   - Streaming platforms can deploy the recommendation system to personalize the homepage feed for each user immediately after finishing a series or movie. By

serving highly-rated, personalized recommendations instantly, the platform can keep users engaged longer and reduce cancellation rates.

2. **Increase User Engagement with Personalized Watchlists and Push Notifications**
    - Users often spend too long scrolling and may log off without watching anything.
    - Use the model to generate personalized watchlists and push recommendations via email or app notifications ("Because you liked *Movie X*, you might love *Movie Y*!"). Personalized, high-confidence recommendations based on previous high ratings will increase session starts and watch time.

3. **Targeted Upsell Opportunities Based on User Preferences**
    - Upselling premium content or bundles is more effective when aligned with user taste.
    - Leverage the model to predict when users are likely to enjoy premium content (e.g., exclusive movies or new releases) and target upsell campaigns ("Upgrade to Premium to access movies you'll love!"). Only users with high predicted enjoyment (≥4 rating) receive the upsell offer, making campaigns more effective and less annoying.

6. Future Work

To further enhance performance and business impact:

- **Quantify Scroll Time Reduction**: Scroll frustration was approximated using low user ratings as a proxy for difficulty in finding enjoyable content. While this provided directional evidence that the recommendation system improved the user experience, the project did not directly measure individual baseline scroll times or quantify the specific reduction achieved. Future work could incorporate scroll behavior tracking (e.g., time to first positive engagement, number of movies browsed before selection) to more precisely measure the model's impact on reducing user browsing time and improving satisfaction.

- **Explore Hybrid Models**: Combine collaborative and content-based recommendations to better handle new users or new movies ("cold start" problem).

- **Incorporate Implicit Feedback**: Add features like watch time, partial watches, and browsing behavior.

- **Expand User Feature Set**: Integrate additional user demographic data (e.g., location, watch history patterns) to refine personalization.

- **Conduct A/B Testing**: Evaluate real-world engagement improvements and quantify the model's effect on churn reduction through live experimentation.

7. Sources

- [1] https://www.usertesting.com/
- [2] https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=167376
- [3] https://grouplens.org/datasets/movielens/1m/
- [4] https://surpriselib.com/
- [5] https://scikit-learn.org/stable/
- [6] https://matplotlib.org/