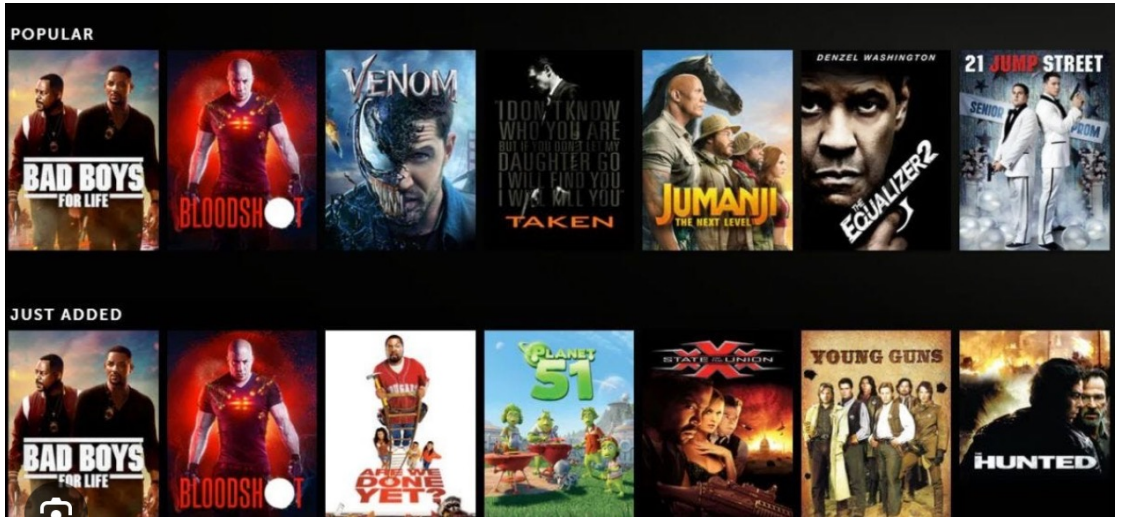
Movie Recommendation System

Motivation & Background
- Content Based & Collaborative Based
- Memory based & Model Based

Data Cleaning & Exploration & Modeling
- Machine Learning
- Model Selection
- Hyperparameter Tuning
- Generating recommendations

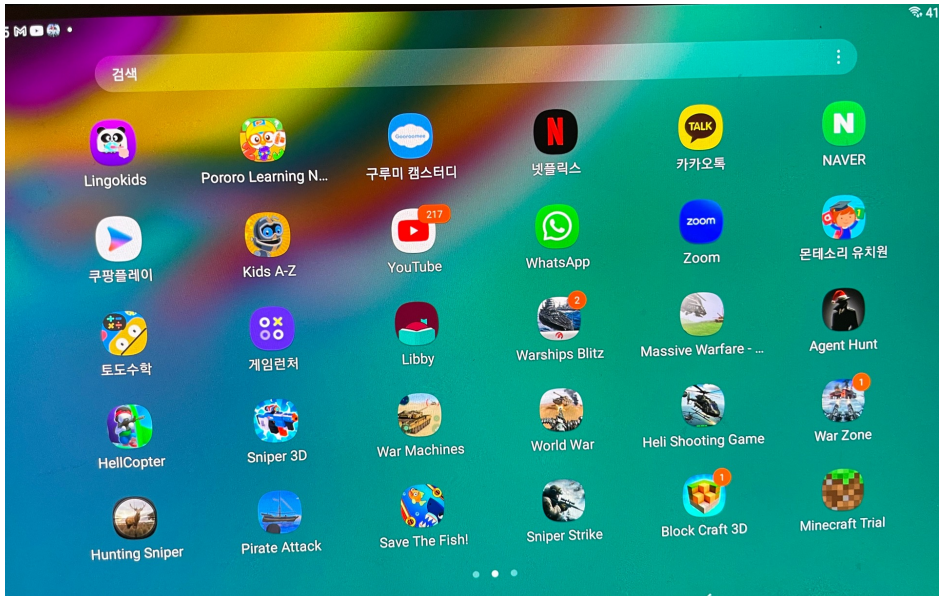
Conclusions & Next future directions



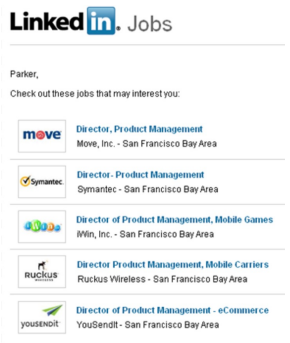
2631_ML
Group6
Minseok Oh
Ajinkya Desai

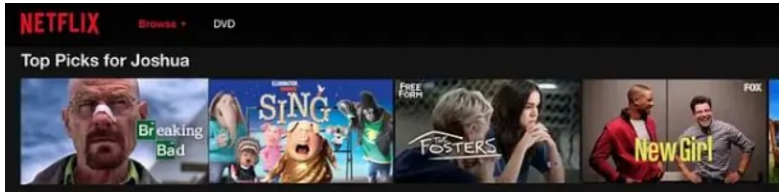
How people use and recognize the recommendation system?

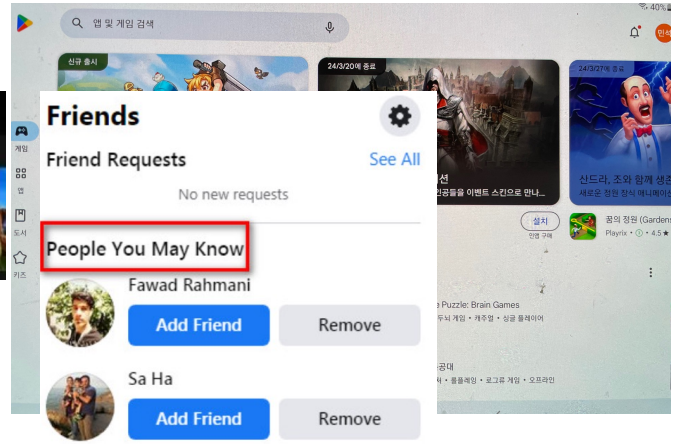
Netflix : movie recommendation
LinkedIn : job recommendation
Facebook : friends recommendation
Instagram : posts recommendation
Youtube : contents recommendation
Appstore : App recommendation
....

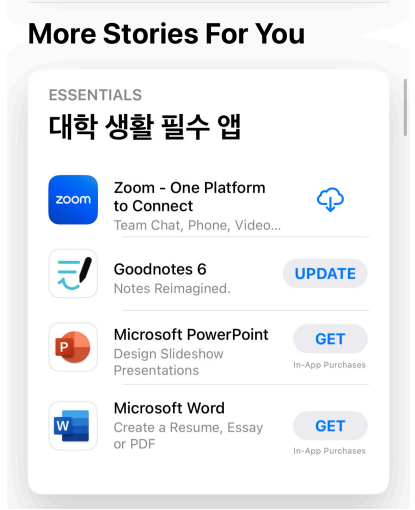


Almost everywhere people already used it!







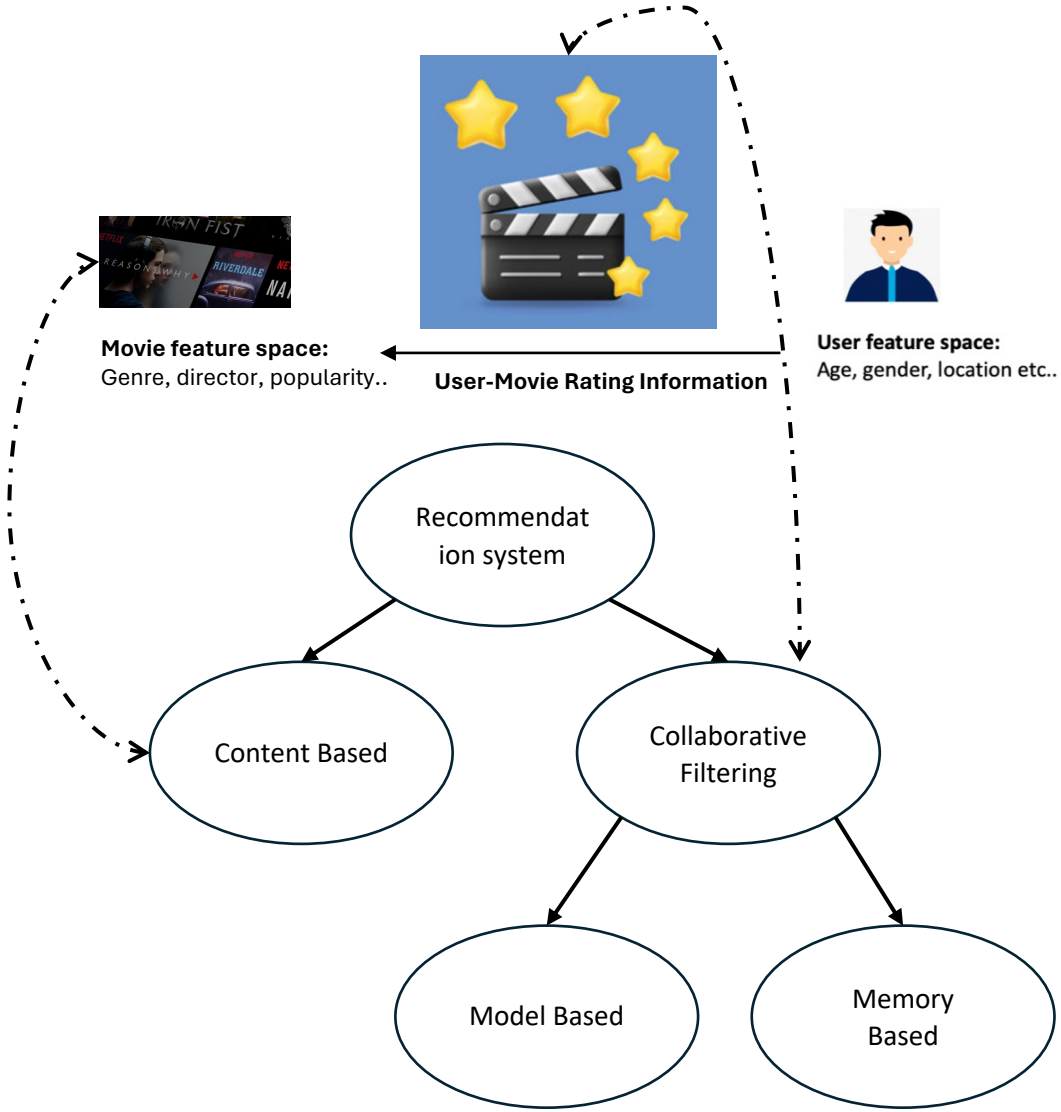


Recommendation Systems: One of the largest application areas of ML.
Enable tailoring personalized content for users, thereby generating revenue for businesses

1) Content based recommendations
Based on user past likes & dislikes
– System recommends items similar to items users have liked based on item feature space

2) Collaborative based recommendations
More robust & widely used.
Disregards item & user feature space & solely based on how different users rate different items

- **Memory based approach**
- **Model based approach**

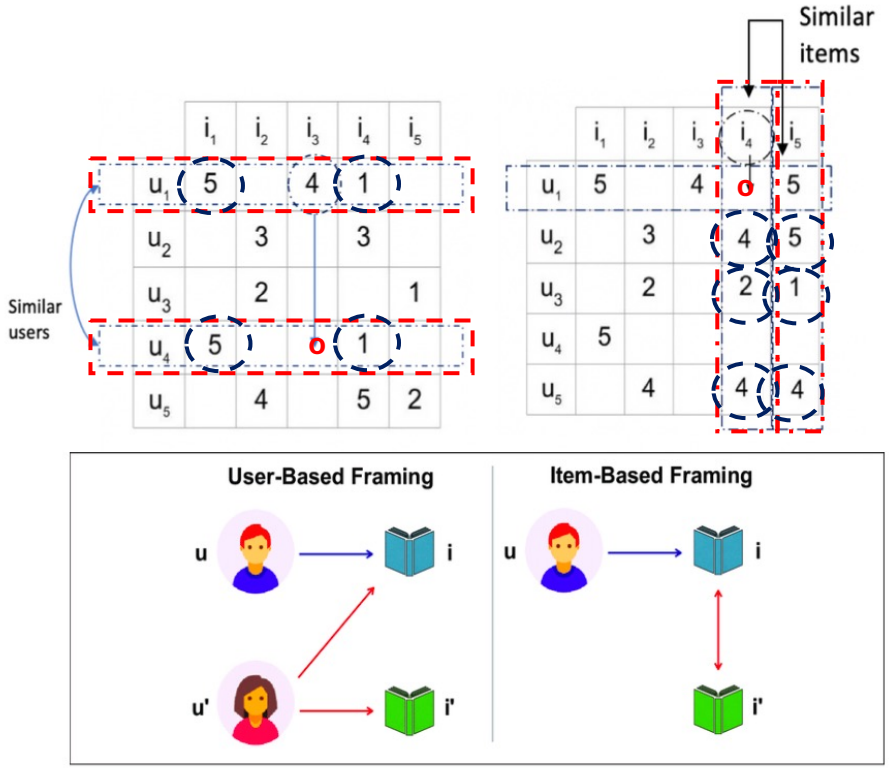


Collaborative filtering - Memory based
- e.g. KNN (K-Nearest Neighbors)

Utilizes entire **user-item rating** information to calculate **similarity scores between users or items**

In **user based collaborative filtering**, two user's are considered similar, if they rate items in a similar manner. An item is recommended to a user, if another user i.e., similar to the user in question has liked the item

In **item based collaborative filtering**, two item's are considered similar, if users rate them in a similar manner. An item is recommended to a user, that is similar to the items the user has rated in the past



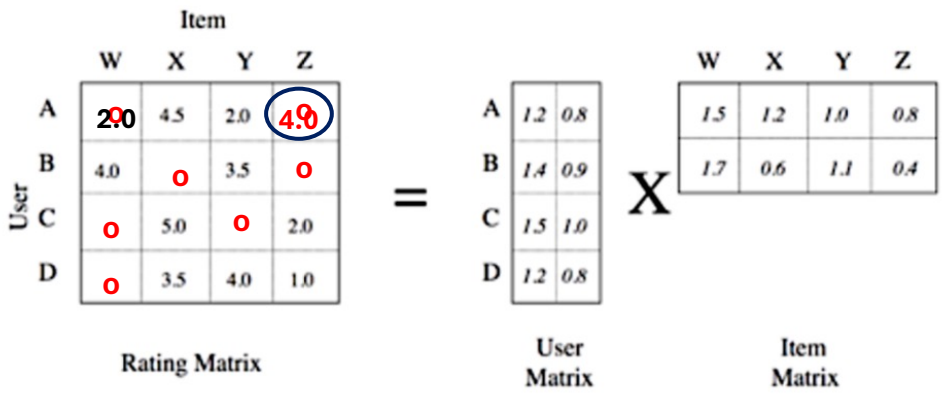
u: User
i: Moive

Model based Collaborative filtering
- e.g. Singular Value Decomposition (SVD)

Singular Value Decomposition, or SVD, is a powerful technique used in model-based collaborative filtering.

Not all items are rated by all users, leading to a lot of missing information. SVD helps us find patterns in the ratings that we do have, to predict the missing ones.

By applying SVD to user-item ratings matrix, we can predict how much a user might like an item they haven't tried yet.



Content-based filtering

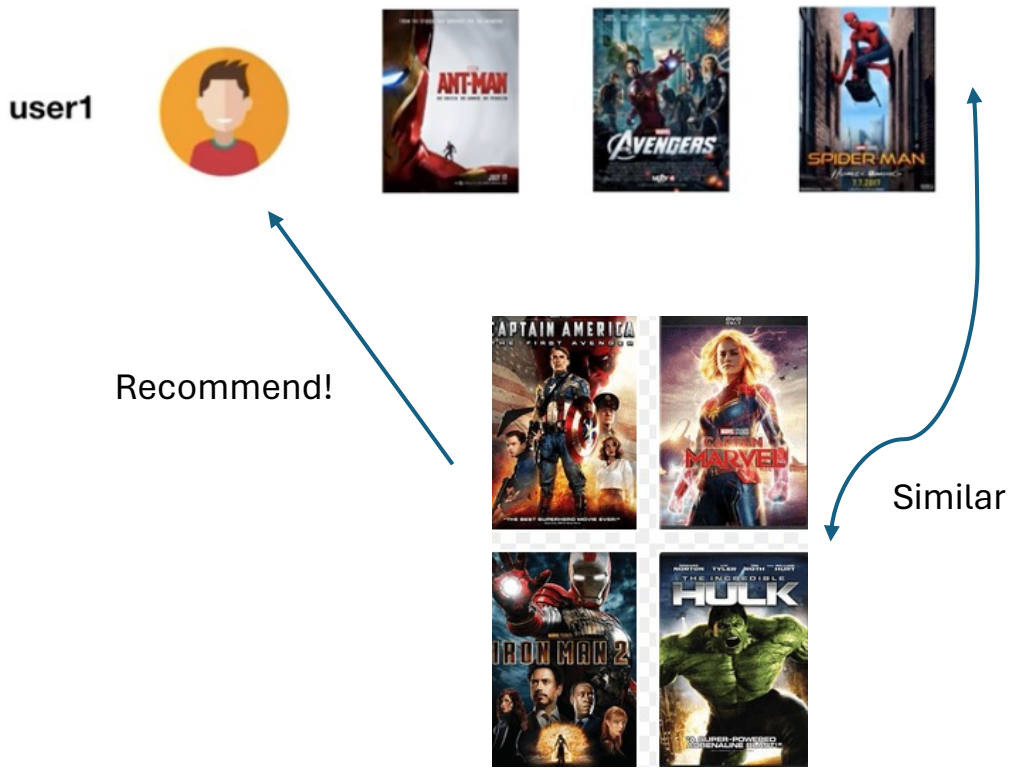
Content-based filtering is a recommendation system method that utilizes item features to recommend additional items similar to what the user likes, based on their previous actions or explicit feedback.

It analyzes the items users have interacted with in the past to construct a profile of their interests.

This profile is then used to recommend new items that share similar characteristics, such as genre, director, or any other item attributes.

Unlike collaborative filtering, content-based filtering doesn't require other users' data, making it more privacy-centric and able to handle new items effectively.

Title, Genre, director, budge, revenue, popularity, production_company..



Data Exploration & Cleaning

- Movie data set collected by Kaggle

https://www.kaggle.com/datasets/parasharmanas/movie-recommendation-system/data

Collaborative filtering

	userId	movieId	rating	timestamp
0	1	296	5.0	1147880044
1	1	306	3.5	1147868817
2	1	307	5.0	1147868828
3	1	665	5.0	1147878820
4	1	899	3.5	1147868510

25,000,095 rows × 4 columns

*Merge ratings table with movies table using 'movieId' to include **'movie title'** instead of 'movieId' for interpretability*

	userId	title	rating
0	548	Toy Story (1995)	4.5
1	626	Toy Story (1995)	4.5
2	847	Toy Story (1995)	4.0
3	997	Toy Story (1995)	4.5
4	1401	Toy Story (1995)	4.5

1,128,299 rows × 3 columns

Check for missing values & duplicated entries – Drop them from the data

Content-based filtering

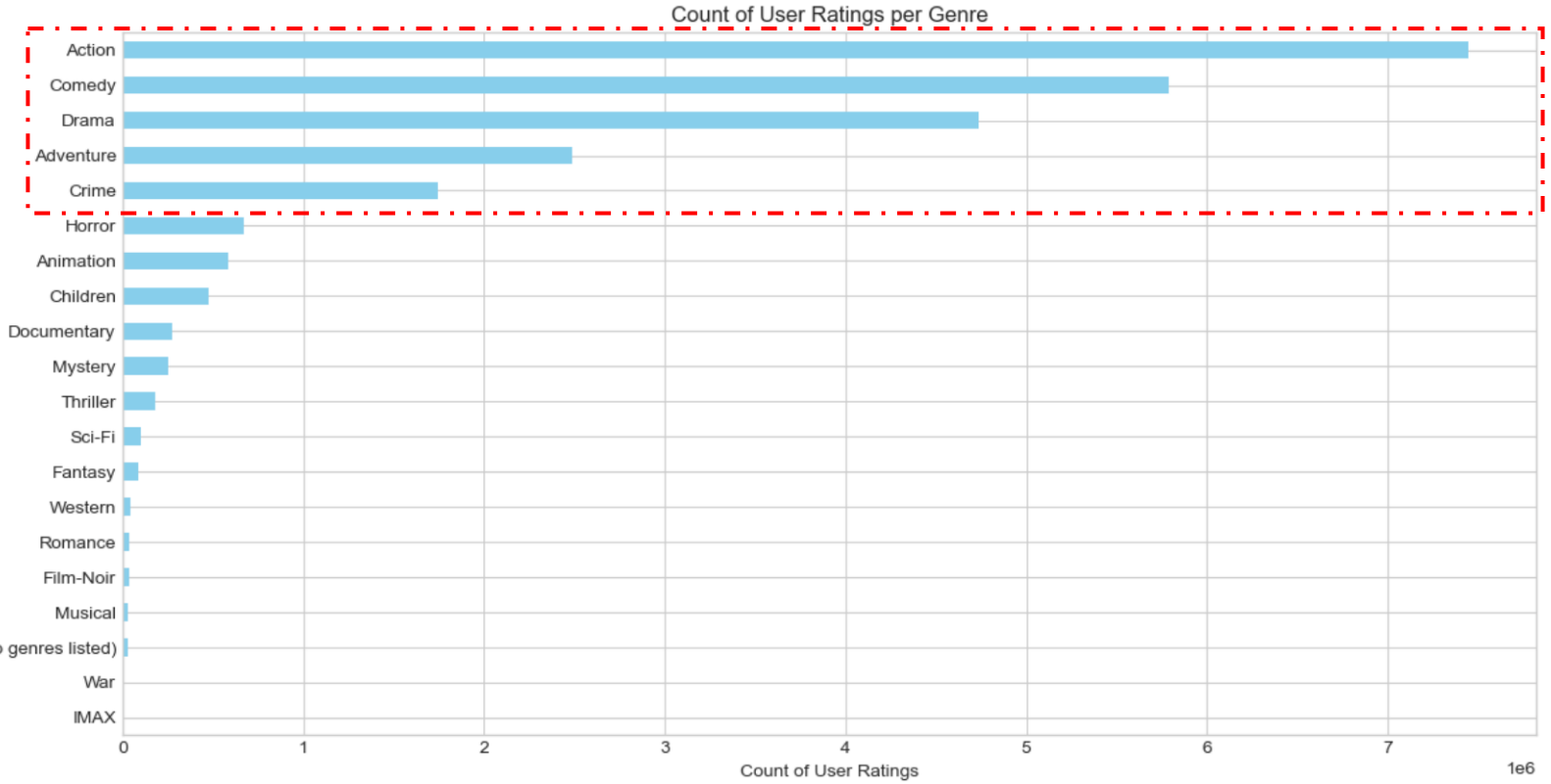
	title	budget	original_title	overview	popularity	production_companies	production_countries	release_date	revenue	runtime	spoken_languages	status	tagline	vote_average	vote_count
0	Toy Story	30000000	Toy Story	Led by Woody, Andy's toys live happily in his ...	21.946943	[{'name': 'Pixar Animation Studios', 'id': 3}]	[{'iso_3166_1': 'US', 'name': 'United States o...	1995-10-30	373554033.0	81.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	NaN	7.7	5415.0
1	Jumanji	65000000	Jumanji	When siblings Judy and Peter discover an encha...	17.015539	[{'name': 'TriStar Pictures', 'id': 559}, {'na...	[{'iso_3166_1': 'US', 'name': 'United States o...	1995-12-15	262797249.0	104.0	[{'iso_639_1': 'en', 'name': 'English'}, {'iso...	Released	Roll the dice and unleash the excitement!	6.9	2413.0
2	Grumpier Old Men	0	Grumpier Old Men	A family wedding reignites the ancient feud be...	11.7129	[{'name': 'Warner Bros.', 'id': 6194}, {'name'...	[{'iso_3166_1': 'US', 'name': 'United States o...	1995-12-22	0.0	101.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	Still Yelling. Still Fighting. Still Ready for...	6.5	92.0

70,039 rows × 15 columns

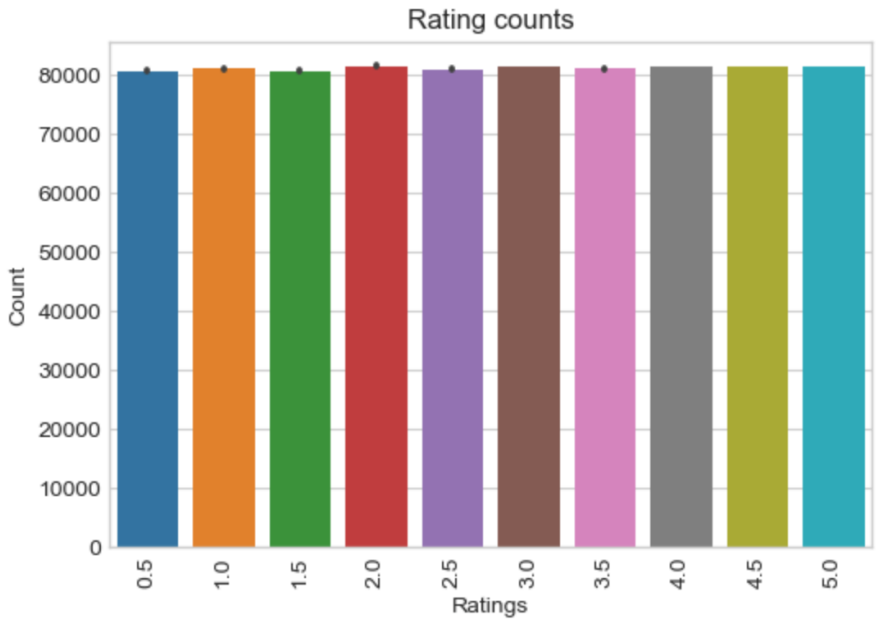
	movieId	budget	popularity	revenue	runtime	vote_average	vote_count	status_binary	Animation	Comedy	...
	1	30000000.0	21.946943	373554033.0	81.0	7.7	5415.0	1	1	1	...
	2	65000000.0	17.015539	262797249.0	104.0	6.9	2413.0	1	0	0	...
	3	9500000.0	11.712900	17405415.0	101.0	6.5	92.0	1	0	1	...
	4	16000000.0	3.859495	81452156.0	127.0	6.1	34.0	1	0	1	...
	5	9500000.0	8.387519	76578911.0	106.0	5.7	173.0	1	0	1	...
	...	...	...	...	...	...	...	...	...	...	...
	208800	9500000.0	12.994387	93600000.0	76.0	6.9	1241.0	1	1	0	...
	208861	70000000.0	9.417460	76066841.0	96.0	5.9	880.0	1	0	0	...
	208887	9500000.0	5.460270	17405415.0	82.0	5.5	11.0	1	0	1	...

41422 rows × 30 columns

Exploratory data analysis

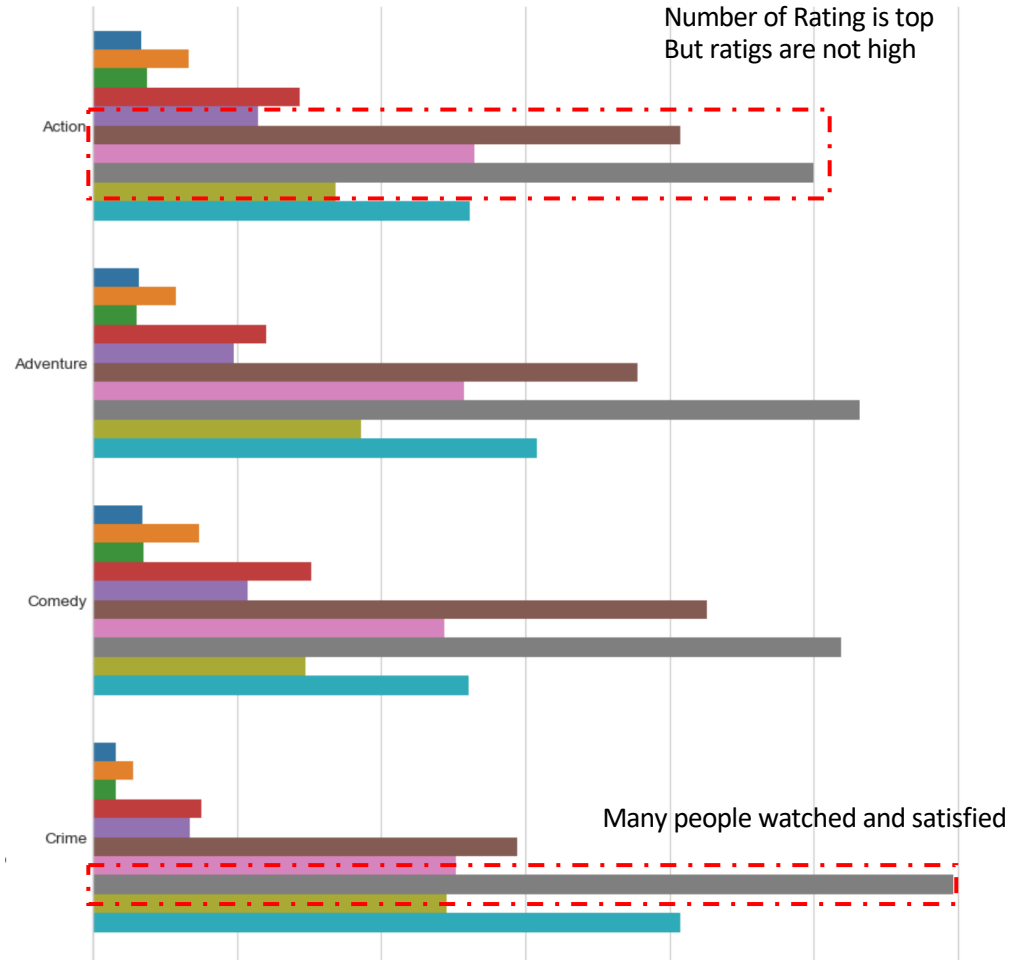


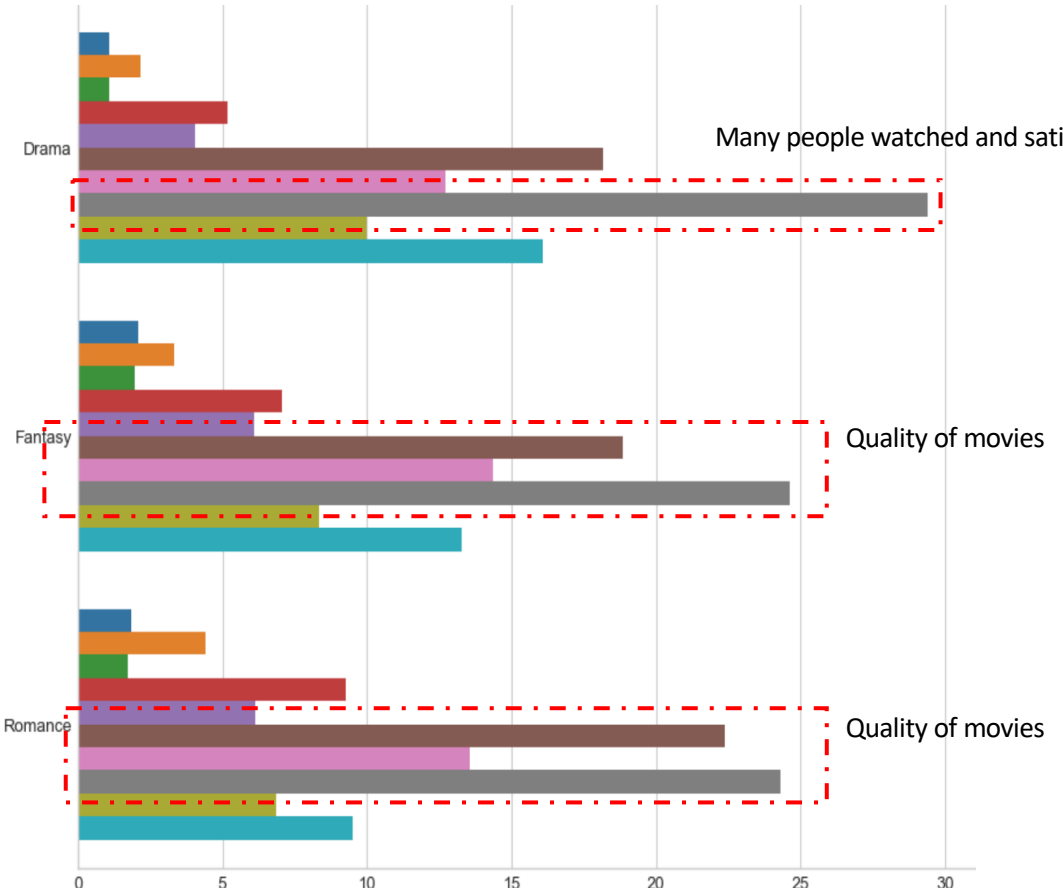
Action, Comedy, Drama Adventure are the most popular movies genre



Ratings are evenly distributed

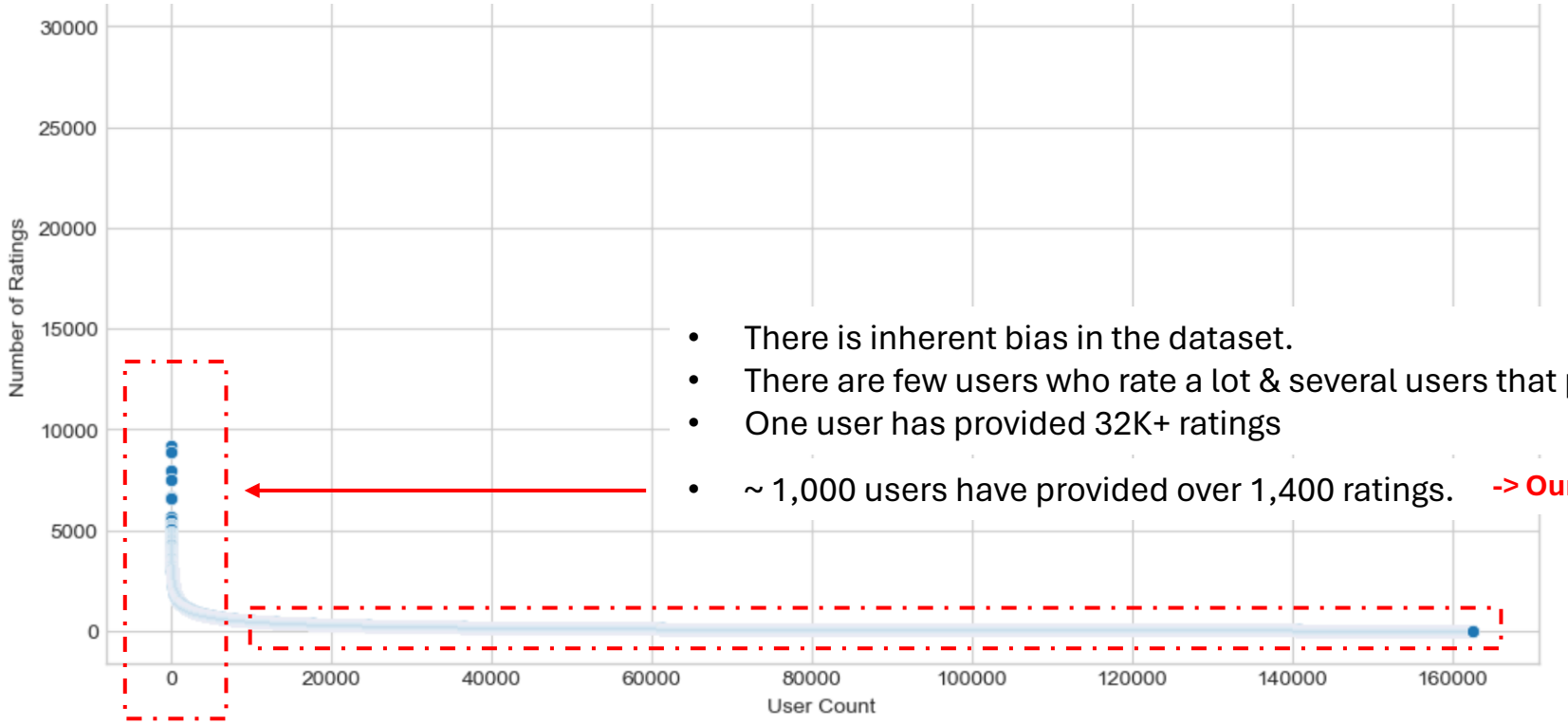
Exploratory data analysis





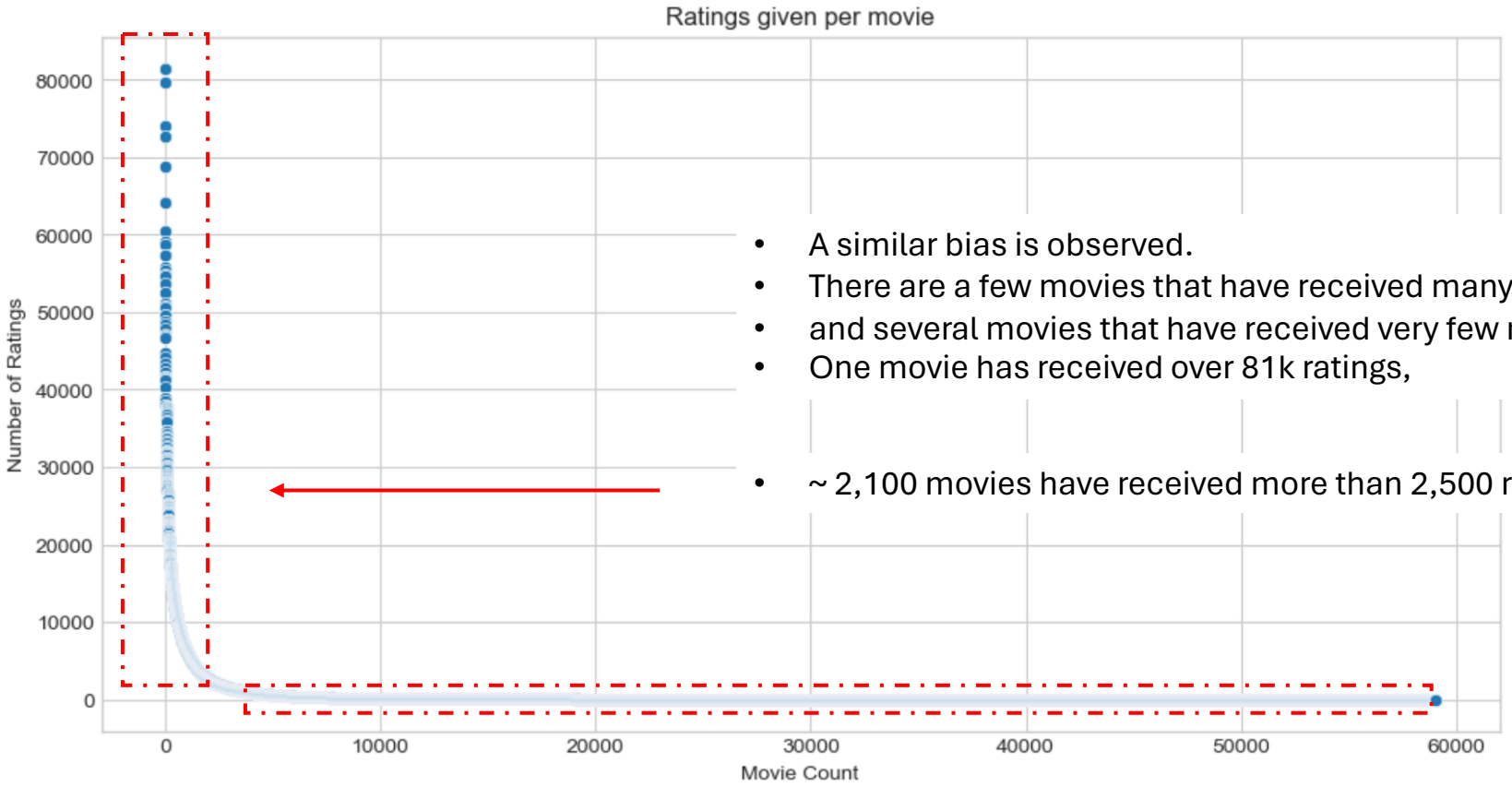
* Percentage count of user ratings

Exploratory data analysis



- There is inherent bias in the dataset.
- There are few users who rate a lot & several users that provide very few ratings.
- One user has provided 32K+ ratings
- ~ 1,000 users have provided over 1,400 ratings. **-> Our Target Users (>1,400 ratings)**

Exploratory data analysis



- A similar bias is observed.
- There are a few movies that have received many ratings
- and several movies that have received very few ratings.
- One movie has received over 81k ratings,

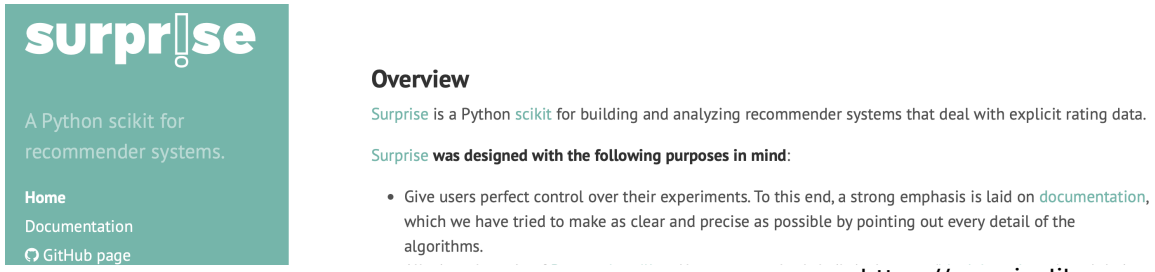
- ~ 2,100 movies have received more than 2,500 ratings **-> Our Target Movies (>2,500 ratings)**

Machine Learning – Model selection

- Memory-based Collaborative filtering
  - **KNNBaseline**
  - **KNNWithMeans**
  - **KNNWithZScore**
  - **KNNBasic**

- Model-based Collaborative filtering
  - **SVD**
  - SVD++
  - PMF
  - NMF

- Content-based filtering
  - **Cosine-similarity metric**



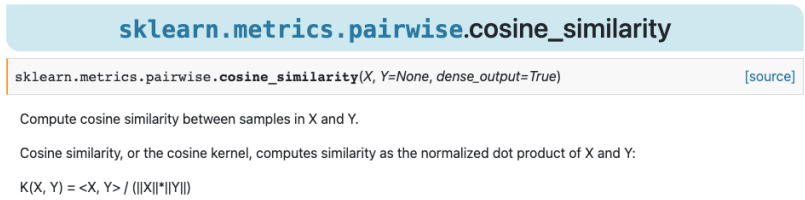
https://surpriselib.com

```
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import cross_validate

# Load the movielens-100k dataset (download it if needed).
data = Dataset.load_builtin('ml-100k')

# Use the famous SVD algorithm.
algo = SVD()

# Run 5-fold cross-validation and print results.
cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```



Machine Learning – Model selection (KNN Variants in Collaborative Filtering)

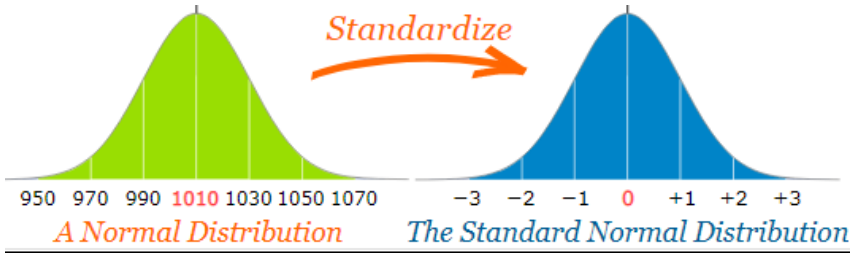
- **KNNBasic:**
  - Straightforward approach using raw ratings
  - Fast and simple, suitable for consistent rating patterns

- **KNNBaseline:**
  - Balances user/item bias with baseline ratings
  - Often more accurate due to adjustment for biases

- **KNNWithMeans:**
  - Accounts for mean ratings in similarities
  - Personalizes by adjusting for users' rating scales

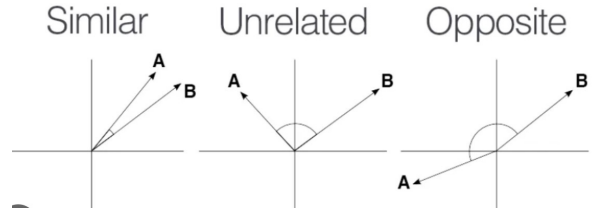
$$\hat{r}_{ui} = \mu_i + \frac{\sum_{j \in N_u^k(i)} \text{sim}(i,j) \cdot (r_{uj} - \mu_j)}{\sum_{j \in N_u^k(i)} \text{sim}(i,j)}$$

- **KNNWithZScore:**
  - Incorporates z-score for normalization
  - Fine-tunes recommendations by considering variance



Machine Learning – Similarity metrics (Cosine, Pearson, MSD)

- **Cosine Similarity:**
  - Measures how similar the items are in terms of user preference independent of scale.



- **Mean Squared Difference Similarity (MSD):**
  - Assesses similarity by the average of squared differences, emphasizing larger variances

- **Pearson Similarity:**
  - Evaluates the degree of linear relationship between two rating sets, adjusting for shifts in rating scale

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

- **Pearson Baseline Similarity:**
  - Refines Pearson by accounting for user and item biases, beneficial for data with diverse rating scales

Machine Learning – Model Performance & selection

< Randomly reduced rows based on unique userId >

- After data cleaning, there were 2,675,000 rows left
  - 2675 unique users who have rated
  - 42355 unique movie titles that have received

	test_mae	RMSE	fit_time	test_time
matrix_factorization.SVD	0.582711	0.769208	7.535558	1.426858
knns.KNNBaseline	0.611728	0.804061	9.008862	40.986354
knns.KNNWithMeans	0.623648	0.816428	8.853945	38.661634
knns.KNNWithZScore	0.618435	0.819372	9.104844	39.903479
knns.KNNBasic	0.640653	0.846059	8.669252	39.292688

< Reduced rows based on number of ratings > - Removing biased data

- After data cleaning, there were **1,128,299** rows left
  - 1102 unique users who have rated > **1,400** movies each
  - 2112 unique movie titles that have received > **2,500** ratings each

	test_mae	RMSE	fit_time	test_time
matrix_factorization.SVD	0.555202	0.735065	2.738692	0.464876
knns.KNNBaseline	0.596420	0.784245	2.127610	16.939502
knns.KNNWithMeans	0.602326	0.789270	2.086609	15.593563
knns.KNNWithZScore	0.599377	0.795918	2.095154	16.388760
knns.KNNBasic	0.614752	0.811507	2.016142	15.457511

* Better performance & 4 times better fit_time & 2 times test_time

Machine Learning – Hyperparameter tuning with GridSearchCV

KNNBaseline Model Hyper parameters:

- k: **Maximum number of neighbors** to take into account for aggregation
- name: **Similarity metric** used to calculate the distance or similarity between users or items. **(Cosine, Pearson, MSD)**
- user_based: **True: User-based, False: Item-based**
- method: The baseline ratings, initial guess at the rating a user would give an item **(SGD, ALS)**
- n_epochs: **the number of iteration cycles** to run for optimizing the algorithm

RMSE Score: 0.7308

{'k': **20**, 'sim_options': {'name': **'pearson_baseline**', 'user_based': **False**}, 'bsl_options': {'method': **'als**', 'n_epochs': **10**}}

Unbiased testing Performance:

RMSE Score: 0.72

Generalized well!

Machine Learning – Hyperparameter tuning with GridSearchCV

SVD Model Hyper parameters:

- n_factors: The number of latent features, typically, more factors can capture more complex patterns (10, 100, 20)
- n_epochs: The number of iterations (5, 10, 20)
- lr_all: Determines the step size at each iteration (0.002, 0.005)
- reg_all: Regularization adds a penalty for complexity in the model (0.2, 0.5)

RMSE Score: 0.7240

{' n_factors': '**70'**, n_epochs ': '**30**', lr_all ': '**0.01'**, reg_all ': **0.04**' }

Unbiased testing Performance:

RMSE Score: 0.7187

Generalized well!

Recommendations

Case1: if ratings are existed

The outputs (above & below) shows the top 10 recommendations for the user ‘548’ using KNNBaseline & SVD

Both algorithms recommended movies which seem to be a similar genre lending confidence in interpretability of recommendations

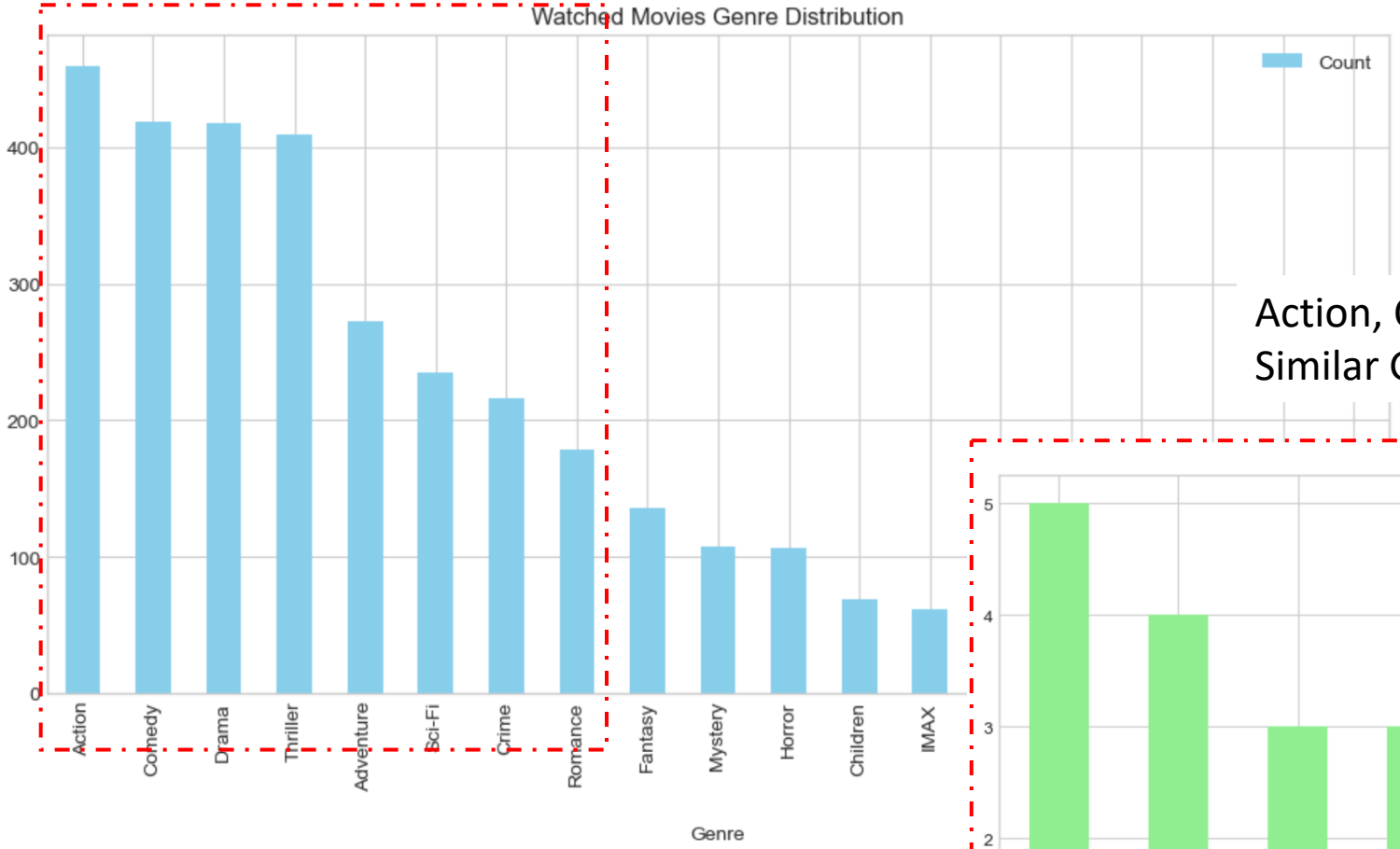
Recommended the list of movie by KNNBaseline

	movieId	title	genres
87	88	Black Sheep (1996)	Comedy
597	605	One Fine Day (1996)	Drama|Romance
1335	1371	Star Trek: The Motion Picture (1979)	Adventure|Sci-Fi
1370	1408	Last of the Mohicans, The (1992)	Action|Romance|War|Western
2321	2412	Rocky V (1990)	Action|Drama
2442	2533	Escape from the Planet of the Apes (1971)	Action|Sci-Fi
2712	2804	Christmas Story, A (1983)	Children|Comedy
2969	3062	Longest Day, The (1962)	Action|Drama|War
6266	6385	Whale Rider (2002)	Drama
9423	30707	Million Dollar Baby (2004)	Drama
18896	98491	Paperman (2012)	Animation|Comedy|Romance

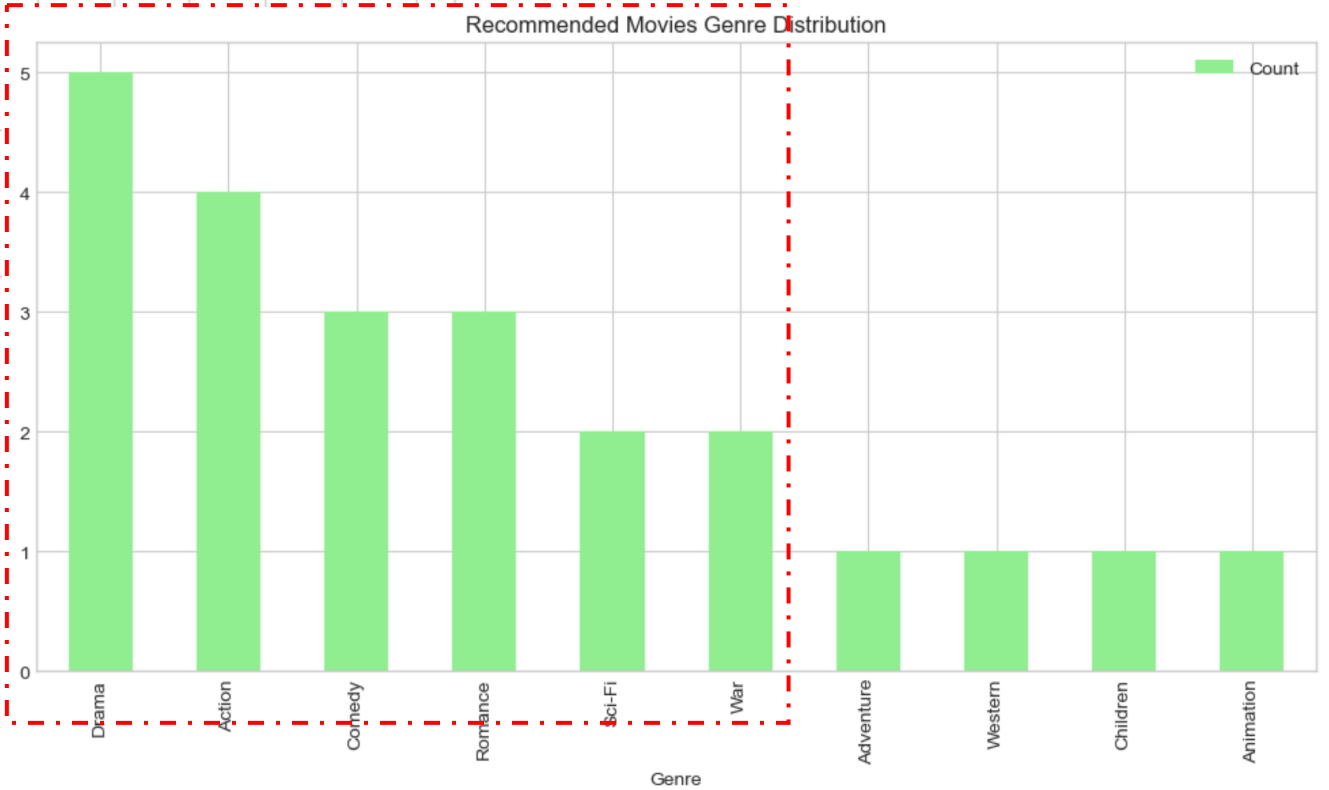
Recommended the list of movie by SVD

movieId	title	genres
750	Dr. Strangelove or: How I Learned to Stop Worr...	Comedy|War
858	Godfather, The (1972)	Crime|Drama
904	Rear Window (1954)	Mystery|Thriller
908	North by Northwest (1959)	Action|Adventure|Mystery|Romance|Thriller
912	Casablanca (1942)	Drama|Romance
922	Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)	Drama|Film-Noir|Romance
1193	One Flew Over the Cuckoo's Nest (1975)	Drama
1203	12 Angry Men (1957)	Drama
1221	Godfather: Part II, The (1974)	Crime|Drama
2019	Seven Samurai (Shichinin no samurai) (1954)	Action|Adventure|Drama

User ‘548’ watched movies versus recommended movie genres



Action, Comedy, Drama, Romance, Sci-Fi
Similar Genre recommended!



Recommendations

Case2: No rating history

If there was no history of ratings,
e.g. New users, new movies
Old movies

Use Content-based filtering to compare various numerical features such as genres, budget, revenue, runtime

Normalized various movie features using `StandardScaler` from `sklearn.preprocessing`

Computed similarity scores between movies in multi-dimensional feature space using `cosine_similarity` From `sklearn.metrics.pairwise`

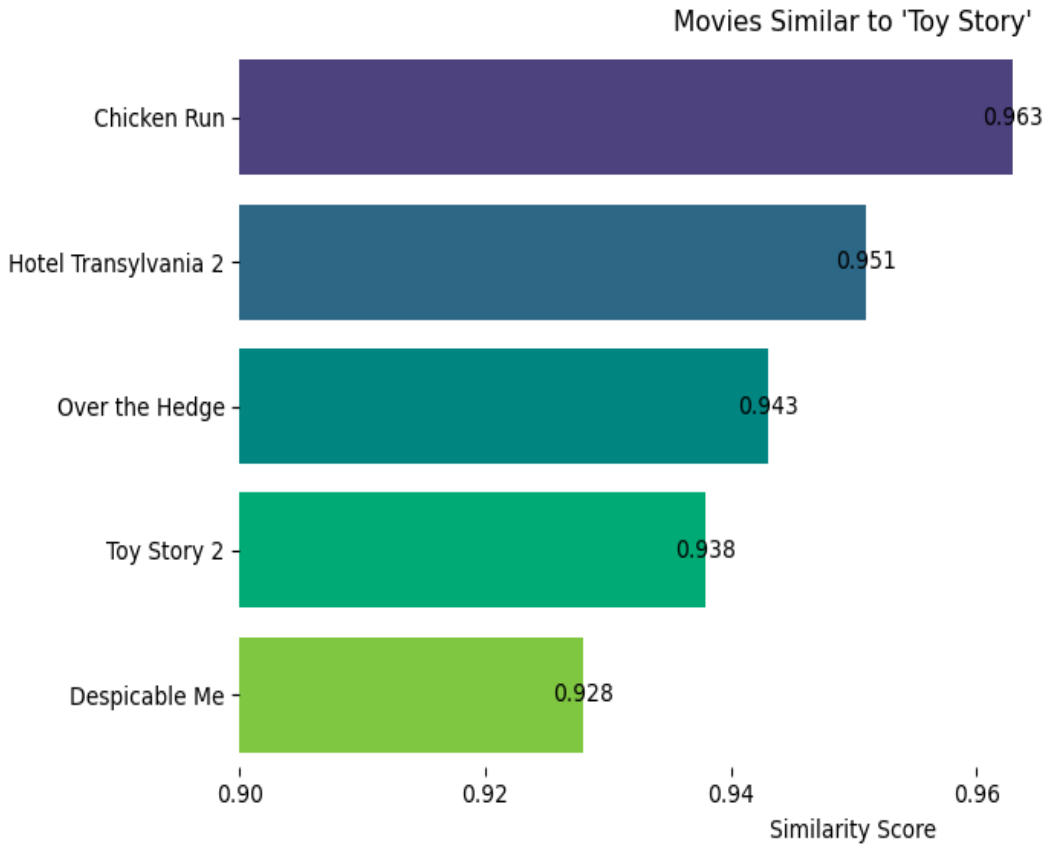
Recommended the list of movie by Content-based filtering

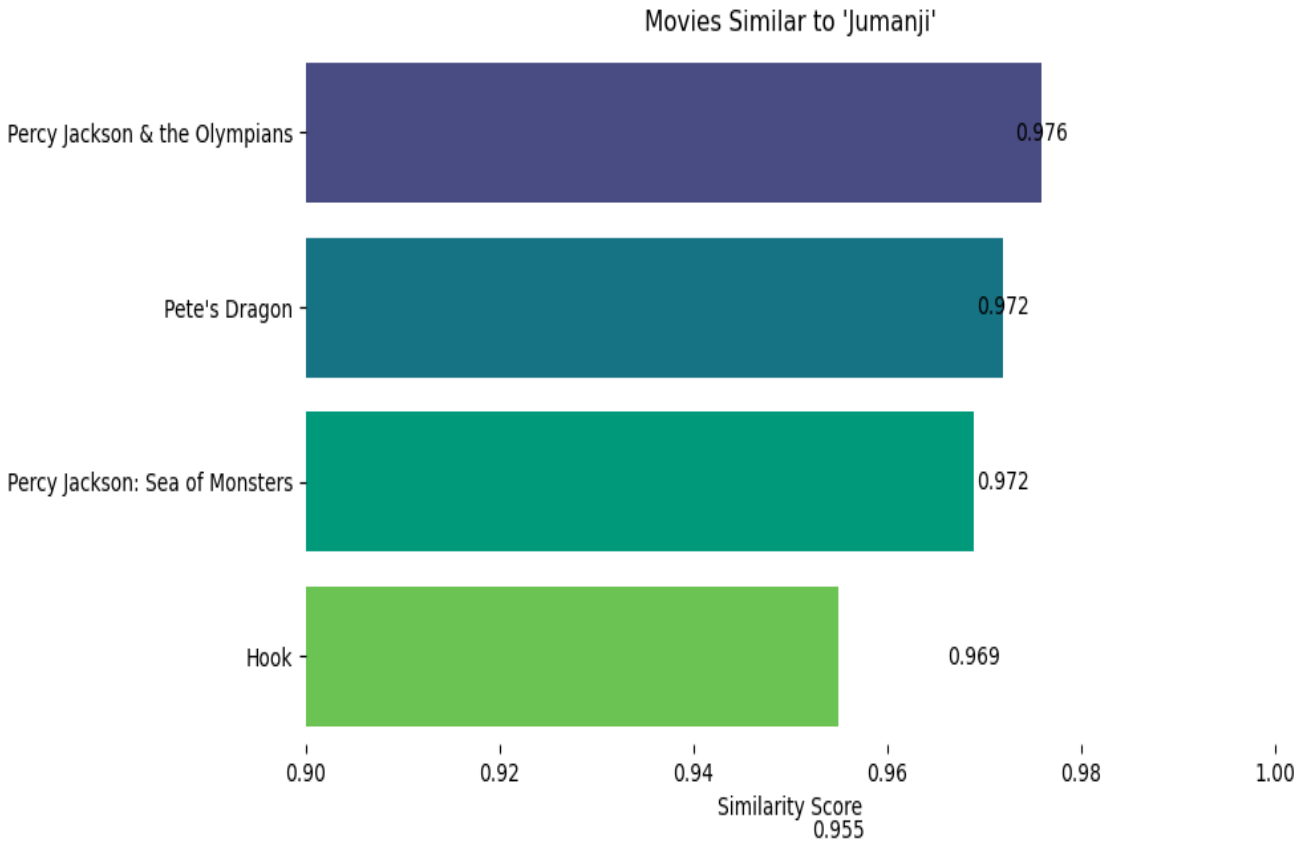
movieId	budget	revenue	runtime	status_binary	Animation	Comedy	Family	Adventure	Fantasy	...		Science Fiction	War	Foreign	TV Movie	Music	Documentary	Western
1	30000000.0	373554033.0	81.0	1	1	1	1	0	0	...	0	0	0	0	0	0	0	0
2	65000000.0	262797249.0	104.0	1	0	0	1	1	1	...	0	0	0	0	0	0	0	0
3	9500000.0	17405415.0	101.0	1	0	1	0	0	0	...	0	0	0	0	0	0	0	0
4	16000000.0	81452156.0	127.0	1	0	1	0	0	0	...	0	0	0	0	0	0	0	0
5	9500000.0	76578911.0	106.0	1	0	1	0	0	0	...	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
208800	9500000.0	93600000.0	76.0	1	1	0	1	0	0	...	0	0	0	0	0	0	0	0
208861	70000000.0	76066841.0	96.0	1	0	0	0	0	0	...	0	1	0	0	0	0	0	0
208887	9500000.0	17405415.0	82.0	1	0	1	0	0	0	...	0	0	0	0	0	0	0	0
208945	9500000.0	17405415.0	111.0	1	0	0	0	0	1	...	0	1	0	0	0	0	0	0
209143	9500000.0	11229.0	76.0	1	1	0	0	0	1	...	0	0	0	0	0	0	0	0

```
Movies similar to 'Toy Story':
Despicable Me: 0.971
Monsters, Inc.: 0.950
Toy Story 2: 0.931
Ice Age: 0.929
The Lion King: 0.924
```

```
Movies similar to 'Jumanji':
Percy Jackson & the Olympians: The Lightning Thief: 0.970
Percy Jackson: Sea of Monsters: 0.958
Hook: 0.946
Charlie and the Chocolate Factory: 0.924
Night at the Museum: Secret of the Tomb: 0.919
```

Similar genre movies (Animation, Adventure, and Sci-Fi) are recommended with high cosine similarity scores!

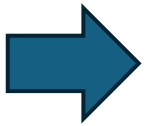




Project Summary & Future Directions

- **Optimization & Techniques**

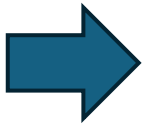
Optimizing data size and selecting effective metrics for KNN and SVD



Utilized Bayesian optimization for data and metric efficiency.

- **Session-based recommendation System**

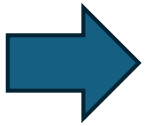
Currently, we do not factor in the temporal dimension of user interactions. we may overlook evolving trends in user preferences.



The identification of individual user sessions (periods of activity), can be particularly useful for sequential recommendation.

- **Performance & Innovation**

KNNBaseline with Pearson Baseline Similarity excelled in handling rating biases.



Exploring deep learning to further refine our system, capturing intricate user preferences

Thank You!