

# Movie Recommender System

## Sushmitha Macheri, Prachi Mardhekar, Tanvi Kulkarni

### BACKGROUND

Recommender systems have become ubiquitous in our lives. There are a wide variety of applications for recommendation systems. These have become increasingly popular over the last few years and are now utilized in most online platforms that we use. The content of such platforms varies from movies, music, books and videos, to friends and stories on social media platform We attempt to build a scalable model to perform the analysis on the Netflix movie dataset

### MOTIVATION

Questions emphasized during the analysis -  
 Why recommendation system?  
 What kind of recommendation system?  
 Which models to implement to provide the best result for the available features in our dataset?

### OBJECTIVE

Netflix, a media service provider and production company, offers its worldwide subscribers an ample amount of movies and TV shows to watch. It conducted an open competition for the best collaborative filtering algorithm that predicts user ratings for movies based on previous ratings alone. This project aims to use the dataset Netflix provided for this competition and recommend movies to users based on self and also other users' rating.

### DATA

**Netflix Prize Data**  
 480,000 customers in the dataset, each identified by a unique customer ID  
 Over 17,000 movies in the dataset, with title, movie ID and year of release  
 Contains over 100 million ratings

### TOOLS USED

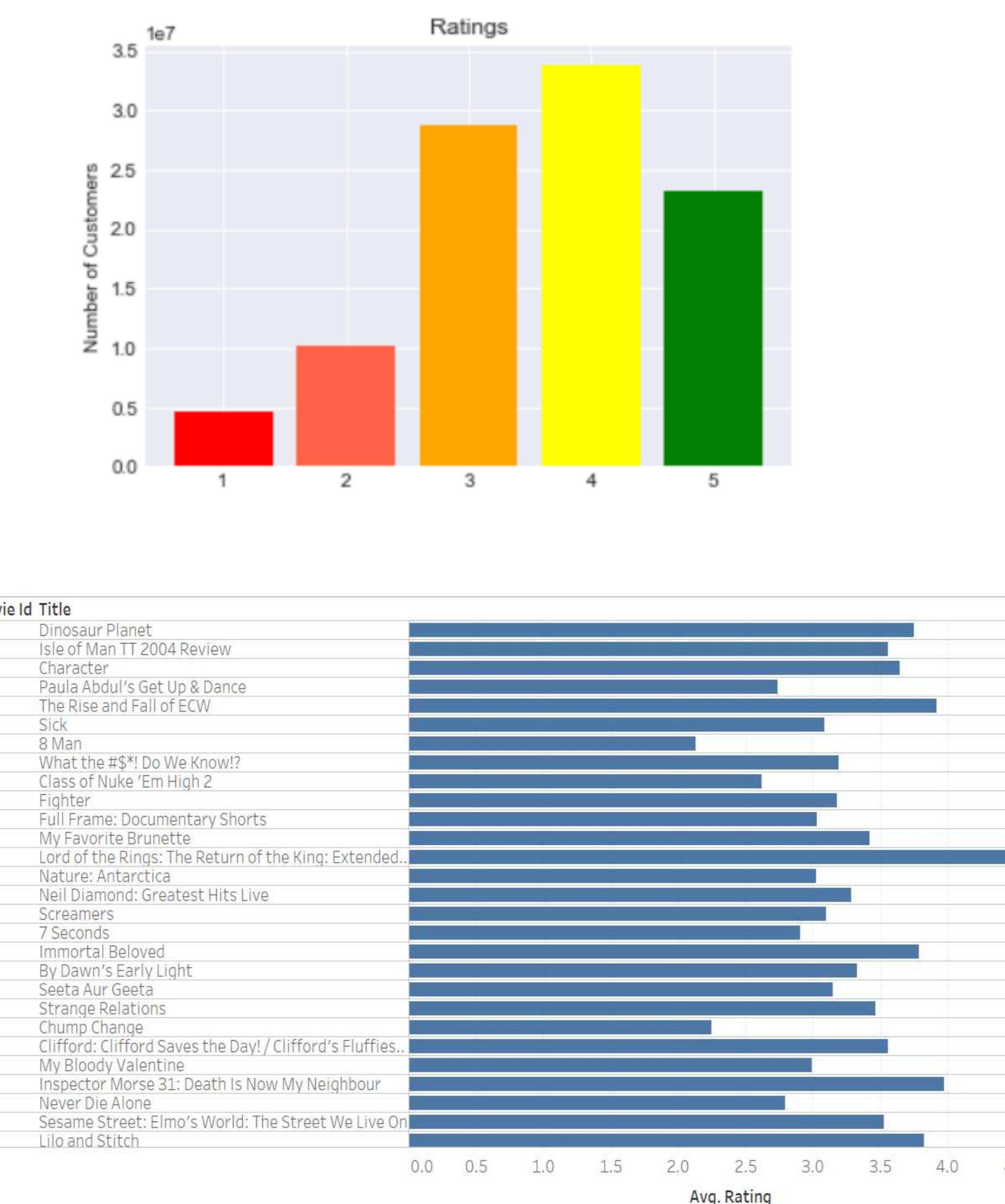
MS Excel for storing CSV files.  
 Python 3 for building and implementing the recommendation system  
 Libraries used in Python – Pandas, NumPy, Scipy, matplotlib, surprise  
 Tableau for Data Visualization

### TECHNIQUES

Collaborative filtering – Entirely based on the past behavior of the user. It's also based on the similarity in preferences and choices of two users.

Pearson Correlation - A correlation analysis provides information on the strength and direction of the linear relationship between two variables for predicting the values.

### ANALYSIS



Total Pool : 225 Movies, 283,670 customers, 999,774 rating given



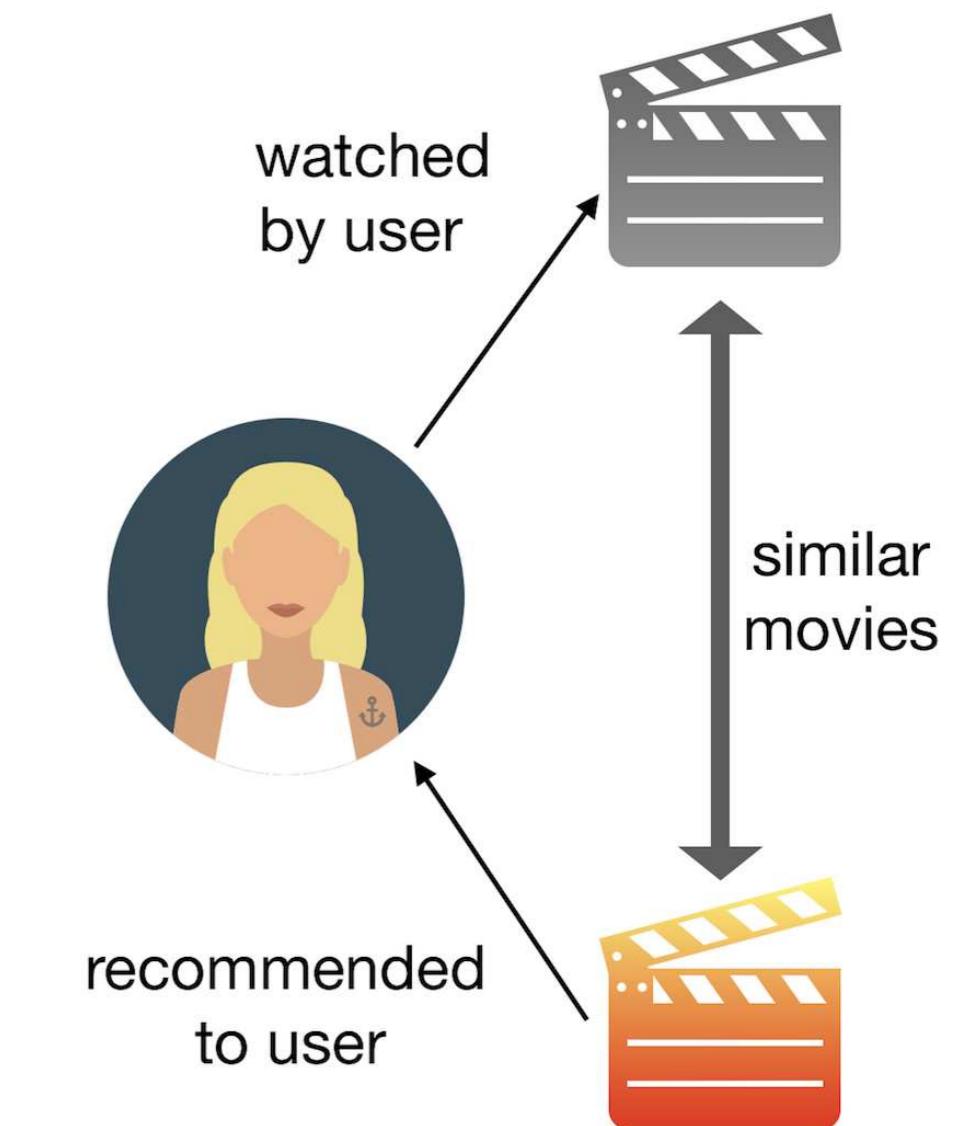
### RESULTS

Year	Name	Estimate_Score
0	2003.0	3.005931
150	2005.0	3.005931
141	2000.0	3.005931
143	2000.0	3.005931
144	1978.0	3.005931
145	1988.0	3.005931
146	1992.0	3.005931
148	2005.0	3.005931
149	2001.0	3.005931
152	2002.0	3.005931
139	1993.0	3.005931

recommend("Allergies: A Natural Approach")			
For movie (Allergies: A Natural Approach) - Top 10 movies recommended based on Pearson's R correlation -			
PearsonR	Name	count	mean
1.000000	Star Trek: Voyager: Season 1	6007	3.942234
0.548126	Star Trek: Deep Space Nine: Season 5	4373	4.160759
0.503780	Scratch	2418	3.833747
0.491445	Character	2012	3.641153
0.447624	Carandiru	1833	3.648663
0.407233	Maya Lin: A Strong Clear Vision	1561	3.770019
0.392089	The Last Shot	3558	2.600056
0.390802	Regular Guys	1931	3.372864
0.379042	Never Die Alone	5861	2.793721
0.373434	Bruce Lee: A Warrior's Journey	1393	3.885140

recommend("Laird: White Knuckle Extreme")			
For movie (Laird: White Knuckle Extreme) - Top 10 movies recommended based on Pearson's R correlation -			
PearsonR	Name	count	mean
1.000000	Husbands and Wives	8179	3.495171
0.400335	The Bad and the Beautiful	2350	3.266383
0.387720	Invader Zim	2216	4.142599
0.359785	Character	2012	3.641153
0.339487	Maya Lin: A Strong Clear Vision	1561	3.770019
0.327104	The Weather Underground	5147	3.757140
0.326547	Regular Guys	1931	3.372864
0.315479	Scandal	2159	3.033812
0.300660	Aqua Teen Hunger Force: Vol. 1	6890	4.168650
0.299222	The Powerpuff Girls Movie	2752	3.221294

<pre>df_785314 = dfFrame[(dfFrame['Cust_Id'] == 785314) &amp; (dfFrame['Rating'] == 5)] df_785314 = df_785314.set_index('Movie_Id') df_785314 = df_785314.join(df_title)[['Name']] print(df_785314)</pre>
<pre>Movie_Id 57      Richard III 175   Reservoir Dogs Name: Name, dtype: object</pre>



### FUTURE SCOPE

The current model can be expanded to build a recommendation system for the entire dataset of 100 million records and 17000 movies. Data can be further bifurcated as TV series and Movies in order to achieve more specifications. Methods like Matrix Factorization can be incorporated and the other features like actors or genres can be included to make a prediction.

# NETFLIX

### Collaborative Filtering

#### Watched By Both Users



#### Watched By Her, Recommended to Him