

# Netflix Recommendation System

Stephen Fok  
Erasmus  
Ka Ho Li



# Project Aim

To create a recommendation system for movie lovers <> better predict what kind of movies a user may enjoy based on their past ratings

# Business Value

- 1 | Increase engagement with users and the platform by tailoring user experiences with personalization and customization**
- 2 | Reviews of the recommendations and customer feedback also help the platform to improvise future selection among movies**
- 3 | Improve user loyalty and lower the possibilities of User Churn**

Section 1

# Data Collection & Preprocessing

# Where is the data coming from

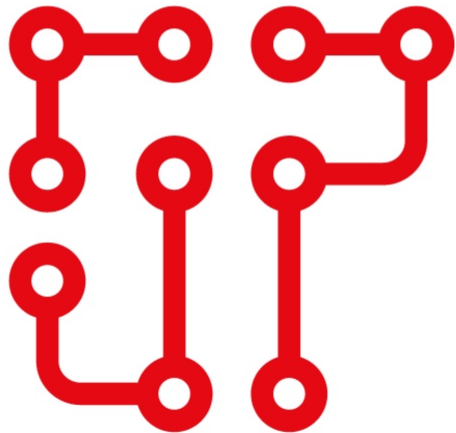
Dataset is from Kaggle (Netflix provided)

The Kaggle logo, featuring the word "kaggle" in a light blue, lowercase, sans-serif font. Below the text is a stylized, low-poly mountain or iceberg shape composed of various shades of blue triangles.

- Movie IDs
- Customer IDs
- Ratings
- Dates

# Data Pre-processing

Rebuild the structure of the dataframe



## Library Used:

- pandas (data)
- numpy (data)
- scikit-surprise (model & function)
- random (example)

**Section 2**

# **EDA**



# There are total 5 questions we would like to answer

- 1 TOP 5 highest number of movie release year
- 2 Which movie has the highest mean rating
- 3 Which movie has the highest no. of rating
- 4 Which customers are the most active in rating
- 5 Which period have the most ratings



## Section 3

# Model

# Recommender Systems

- **Content-based recommendations**

Recommendations based on a user profile (specific interests) or previously consumed content

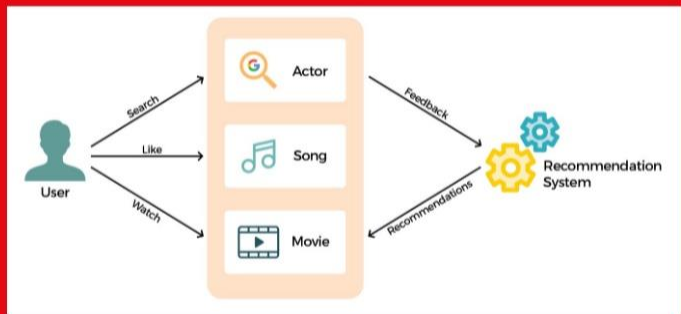
- **Collaborative filtering**

Recommendations based on the content preferences of similar users

- **Hybrid approaches**



# Model Explanation



- **Principle behind: Collaborative Filtering System**

For instance, user\_1 rated movie A and movie B, and rated movie C. When user\_2 has rated movie A and movie B, and recommendation system will calculate the expected rating of movie C with respects to user\_2, subject to similarity between user\_1 and user\_2.

- **Library used: SVD from surprise library**

- **Advantage**

Different to content-based recommendation system, recommendation will not be only limited to users' movie watching history. New potential high rating movie will be recommended to user based on other users' pattern of watching movies.

- **Examples**

- Movie/TV recommendation (Netflix, Hulu, iTunes)
- Product recommendation (Amazon)

# Mathematical Principle behind Collaborative Filtering System

STATFLIX

## COLLABORATIVE FILTERING

IDEA: PAST  
SIMILAR PREFS  
CAN INFORM  
FUTURE PREFS

	1	2	3	4	5	6	7	8
$u_1$	5	1	4				2	1
$u_2$	5		3	2	5		2	1
$u_3$	1	4	2	5	2		5	4

Q: NEXT REC FOR  $u_1$ ?

Prior:  $\bar{S}_4 = \bar{S}_5 = 3.5$

BUT... WHO IS  $u_1$  MORE SIMILAR TO?

$u_1$	5	4	2	1
$u_2$	5	3	2	1

$u_1$	5	1	4	2	1
$u_3$	1	4	2	5	4

Cosine Similarity:

$$\text{Sim}(u_i, u_j) = \frac{u_i \cdot u_j}{\|u_i\| \|u_j\|} = \cos(\theta)$$

$$\text{Sim}(u_1, u_2) = 0.99, \text{Sim}(u_1, u_3) = 0.57$$

$$\hat{r}_{1,4} = \frac{S_{12} \times 2 + S_{13} \times 5}{S_{12} + S_{13}} = 3.1$$

$$\hat{r}_{1,5} = 3.9 \Rightarrow \text{Rec } S_5$$

Section 4

# Insights

# Challenges & Limitations

There are huge amount of data and they are separated in different files

i.e.

total over 1 Billion rows if we take all the records into account

It refers to a recommendation engines inability to generate decent predictions for new items / users, or users / items which have a low amount of interactions,

**Data are not fully taken**

**It's hard to define "good recommendation"**

**Cold start problem**

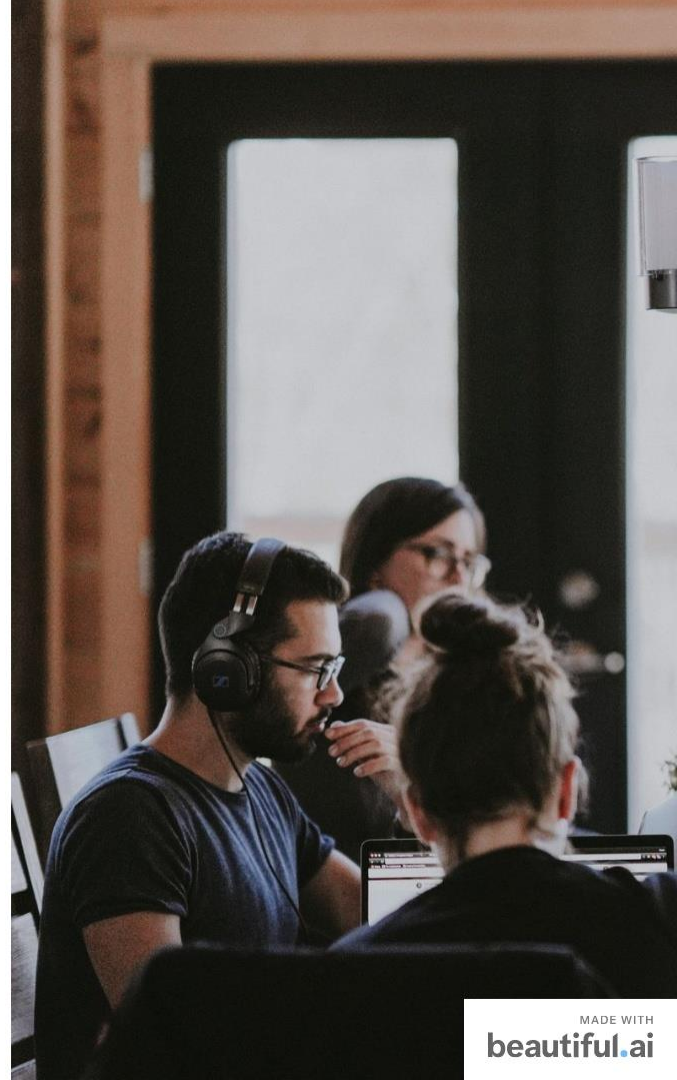
As recommendations are machine-based, yet movies are kind of "art" and "subjective", algorithm may not provide the best suggestions to some users

# Next Steps

**Get feedback data to  
improvis the model**

**Better selection of the  
movies based on reviews**

**Create user feedback loop**





# Conclusions

- 1 | Nowadays, users aim for more customization and personalization from service provider, a good recommendation system certainly is a great help to improve user experiences
- 2 | The more feedback we get from users, the more accurate the recommender system become, which will create a positive user feedback loop
- 3 | Hence, more engagement between users and the platform will result in better user loyalty and lower User Churn



# NETFLIX

Thanks!  
Q&A