Movie Recommender

By Spencer Goble

Background

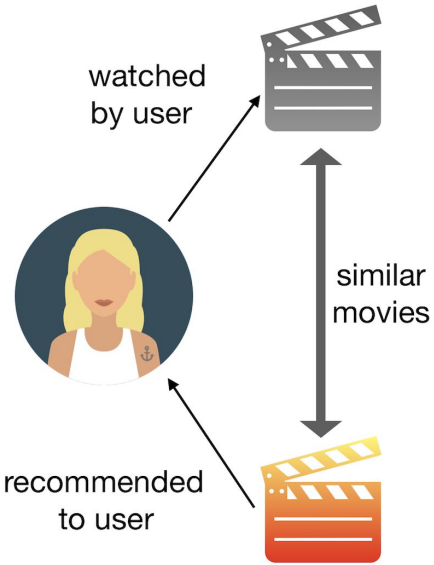
- The recent quarter end results showed a drop DVU (Daily View Per User) and MVU (Monthly Views Per User)
- The goal is to increase user retention by updating the recommendation system to engage current and future subscribers
- Are we **curating the *best* recommendations possible for customers**?

What

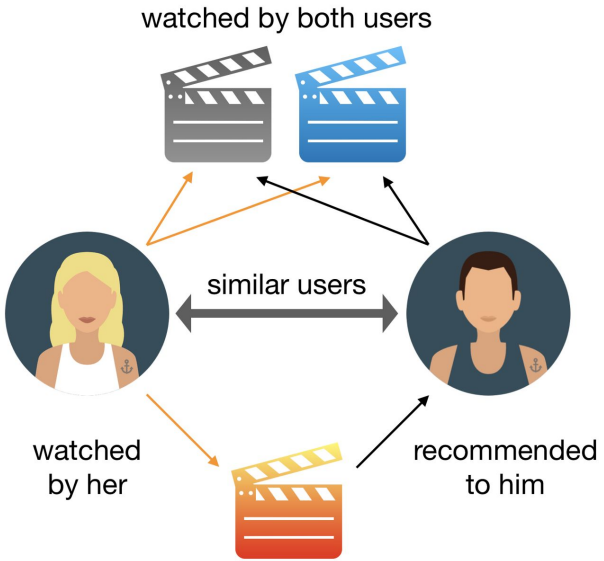
- The current recommendation system is content based
- Too much weight is put on movie ‘genre’, making the recommendation list is incohesive
- The correlation between films is too vague
- Currently users scroll endlessly through recommendations before logging off the service

Current Systems

Content Based



Collaborative Filtering



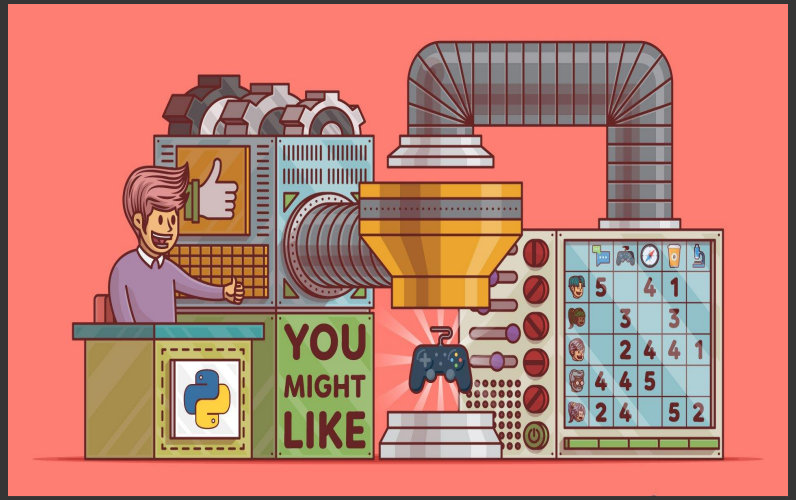
But How!?

- Upgrade the recommendation system with brilliant…

  Machine Learning techniques!

- With an improved system DVU's and MVU's will increase…

  10-15% by next quarter



Approach

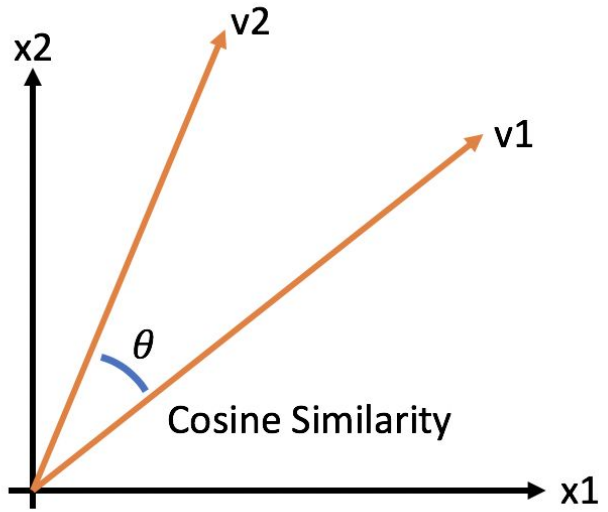
Content Based	Collaborative Filtering	Hybrid
❖ Use content metrics that shape the 'vibe' of films ➢ Director and lead actor ❖ Use keywords that refer to specific elements of the film ➢ 'marriage' for a comedy film with a wedding theme	❖ Refine metrics for determining the degree to which a user liked a film ❖ Improve the algorithm that predicts the rating a User will score a film	❖ Create a system that combines both Content based and Collaborative Filtering ❖ Order the suggestions with a priority on films with a higher predicted user score

Recommended Approach

Content Based Process

- Mine every bit of informational text in the dataset and create a long string of text for each film.
- The string for *Toy Story* looks like this:
  - ***jealousi toy boy friendship friend rivalri boynextdoor newtoy toycomestolif tomhanks timallen donrickles johnlasseter johnlasseter johnlasseter animation comedy family pixaranimationstudios***
  - It seems nonsensical for us mortals but the algorithm loves it!
- Add more weight to the director of the film by repeating his/her name 3 times within the text string
- Finally, create a text profile for each film which was then used to compare films

Cosine Similarity measures the distance/similarity between films (v1 and v2)

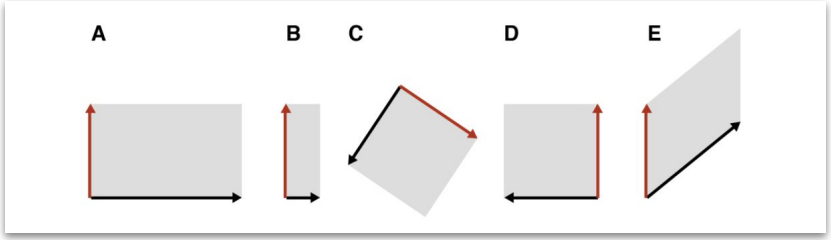


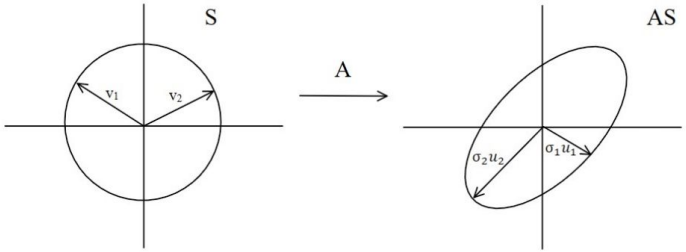
Collaborative Filtering Process

- Predict the rating a user will give a film based on how similar users have rated that film
- Compare several algorithms to see which performed the best

The answer? ***Singular Value Decomposition***

This method transforms shapes to find points that are similar





Hybrid System

- Yes, we can have the best of both worlds
- So ***User X*** likes ***Inception?***
  - Find the top 10 most similar films and rank them by how ***User X*** is *expected* to rate them (1-5)

Recommended results from ***Inception***

```
Memento 4.34
The Dark Knight 4.29
The Prestige 4.27
Interstellar 4.21
Batman Begins 4.06
The Dark Knight Rises 3.97
Don Jon 3.86
Following 3.82
Insomnia 3.75
Pacific Rim 3.72
```

Conclusions

- The customer retention rate is declining and the current recommendation system needs a makeover, let's increase DVU's and MVU's by 10 - 15%
- The current *content based* and *collaborative filtering* recommender systems are not optimized
- Employ the newly refined and more robust *Hybrid Recommender!*

Thank you

Additional Documents

White Paper:
https://github.com/LiftedAquatic/Movie-Recommender-System/blob/main/White%20Paper.pdf

Code:
https://github.com/LiftedAquatic/Movie-Recommender-System/tree/main/Notebooks

Project Repository:
https://github.com/LiftedAquatic/Movie-Recommender-System

Original Data:
https://github.com/LiftedAquatic/Movie-Recommender-System/tree/main/Original%20Data