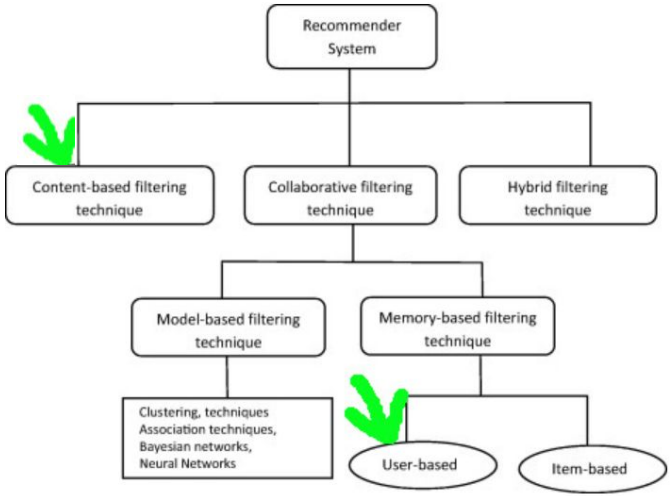
IR Final Project Presentation:

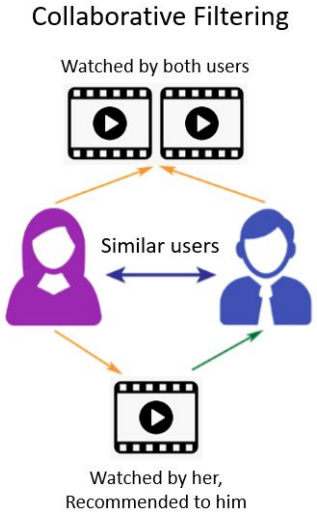
Conversational Movie Recommender Chatbot

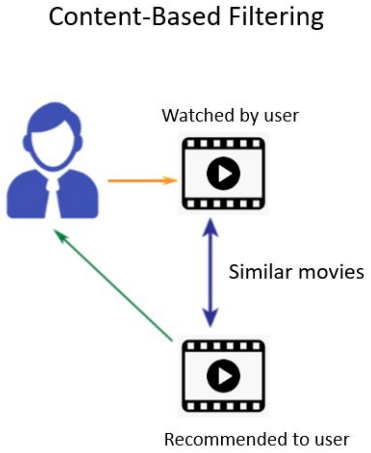
Presented By: Linyang Du, Jiajia Liang, Jie Fan

UVA ENGINEERING

Introduction







Model Overview

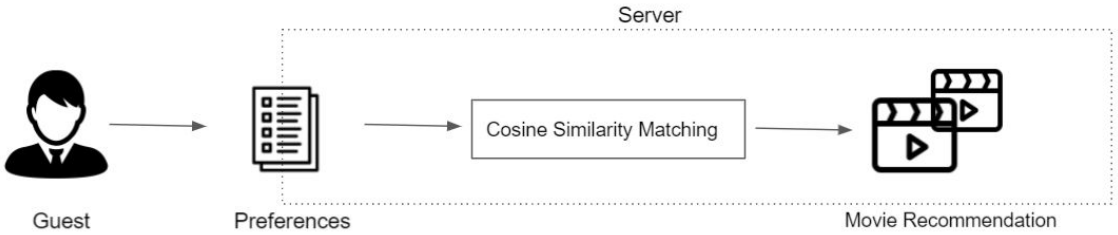


Fig (a)

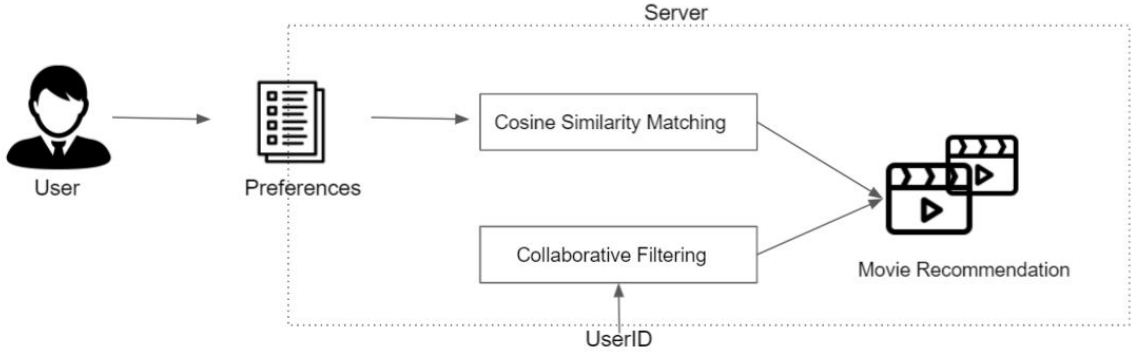


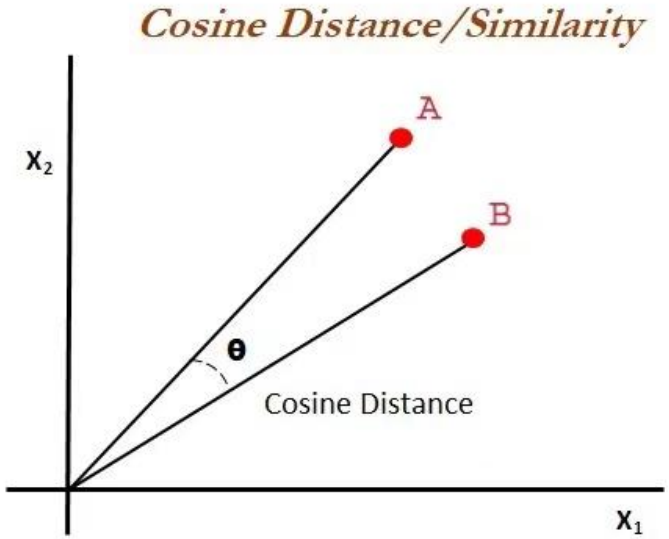
Fig (b)

Content Based-Count Vectorizer

Data = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', ' the', 'lazy', 'dog']

	The	quick	brown	fox	jumps	over	lazy	dog
Data	2	1	1	1	1	1	1	1

Content Based-Cosine Similarity



A: the vector for preference of user on movies

B: the vector for movie in our datasets

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}, \text{ where } \|A\| = \sqrt{\sum_{i=1}^{n} A_i^2} \text{ and } \|B\| = \sqrt{\sum_{i=1}^{n} B_i^2}.$$

Collaborative Filtering

Collaborative Filtering - problem definition

- **Recommend you movies that other users similar to you liked**

Problem 1. ***Predicted Rating*** *Given a user a, and movie m, output predicted rating $pred(a, m)$ of user a for movie m.*

$$pred(a, m) = \frac{\sum_{b \in N} sim(a, b) * (r_{b,m})}{\sum_{b \in N} sim(a, b)}$$

where:

- $sim(a, b)$ is the similarity between user a and user b.
- M is the set of common rated movies by user a and user b.
- $r_{b,m}$ is the rating of movie m by user b.

Collaborative Filtering - Similarity Computation

- Euclidean distance $$sim(a,b) = \sqrt{\sum_{m \in M} (r_{a,m} - r_{b,m})^2}$$
- Pearson Correlation $$sim(a,b) = \frac{\sum_{m \in M} (r_{a,m} - \bar{r_a})(r_{b,m} - \bar{r_b})}{\sqrt{\sum_{m \in M} (r_{a,m} - \bar{r_a})^2} \sqrt{\sum_{m \in M} (r_{b,m} - \bar{r_b})^2}}$$
- Cosine Distance $$sim(a,b) = \frac{\vec{a}\vec{b}}{|\vec{a}| * |\vec{b}|}$$

Collaborative Filtering - Similarity Evaluation

- Use RMSE as evaluation metric

$$RMSE = \sqrt{\left(\frac{\sum(\hat{y} - y)^2}{n}\right)}$$

- Use Pearson Correlation as final choice

Similarity Function	RMSE
Euclidean distance	1.0985
Pearson Correlation	1.0364
Cosine distance	1.0483

Table 1. Similarity Function and corresponding RMSE

Collaborative Filtering - Data Processing

- Data: rating.csv
  - Unique users: 270, 896
  - Unique movies: 45,115
  - Rating entries: 26,024,289
- Problem with large and sparse data
- Reduced data
  - Unique users: 2374
  - Unique movies: 19,930
  - Rating entries: 755,536

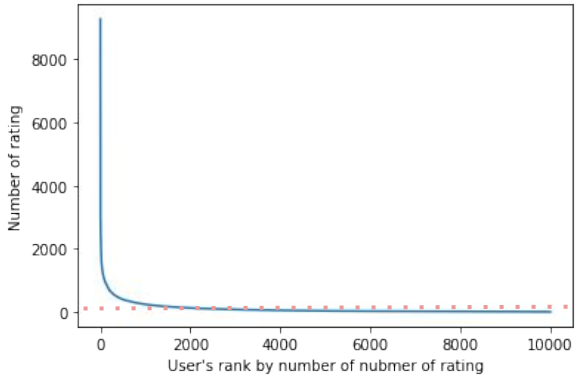


Fig 2. . User's rating frequency vs user's rank in the frequency table

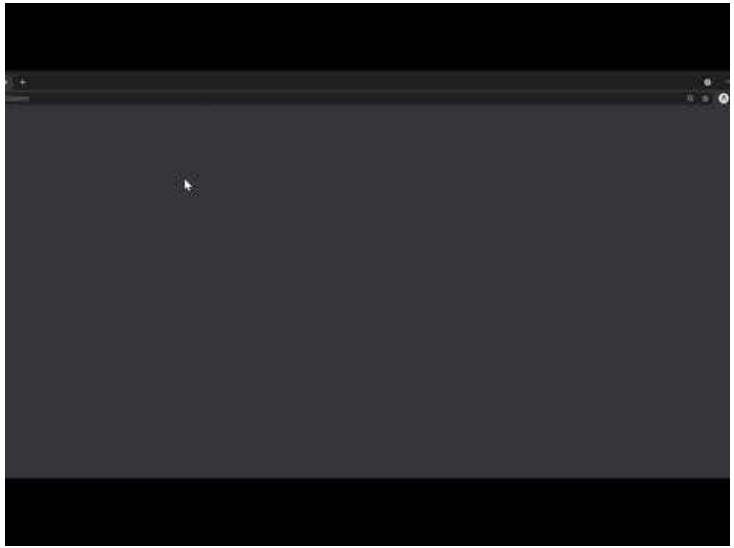
Collaborative Filtering - Example

- sim(you, b1) = 0.9 sim(you, b2) = 0.5
- Relevant movie returned by content-based filtering

User b1 Rating:	3	5
User b2 Rating:	5	4
Predicted Rating for you	3.71	4.64

$$pred(a,m) = \frac{\sum_{b \in N} sim(a,b) * (r_{b,m})}{\sum_{b \in N} sim(a,b)}$$

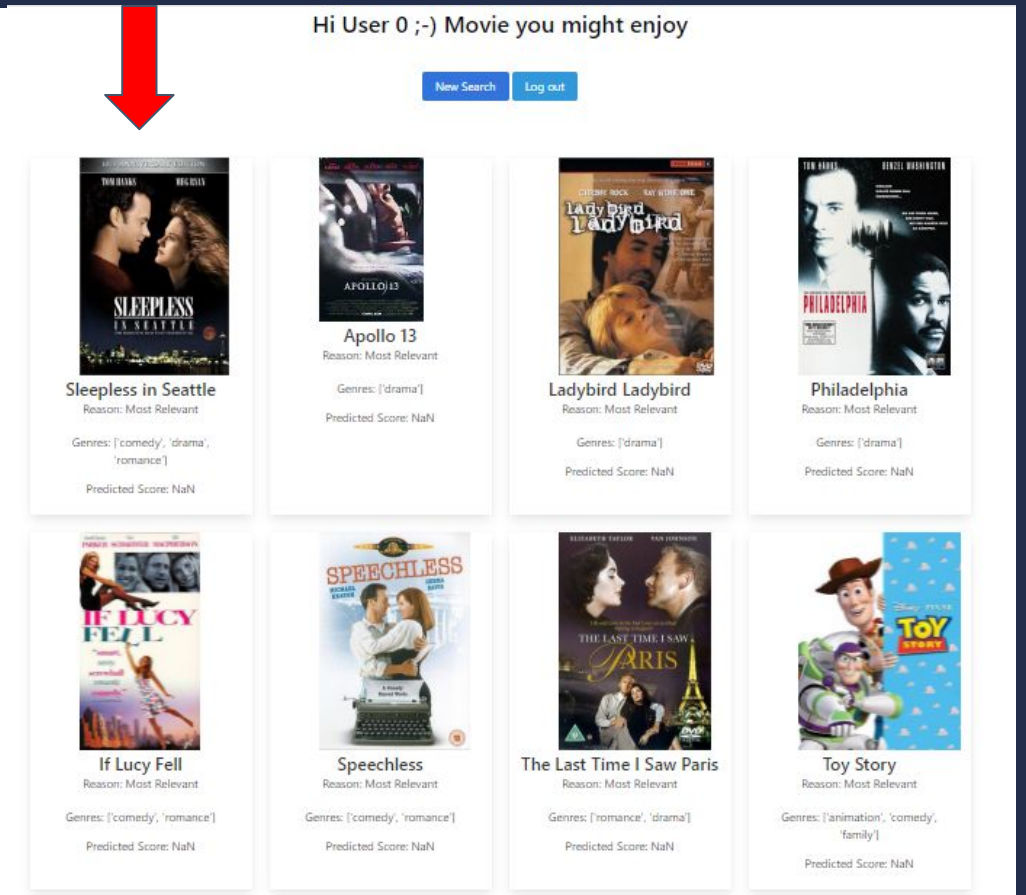
Web app- putting everything together



Case Study 1:

Search Using Content Based Recommender

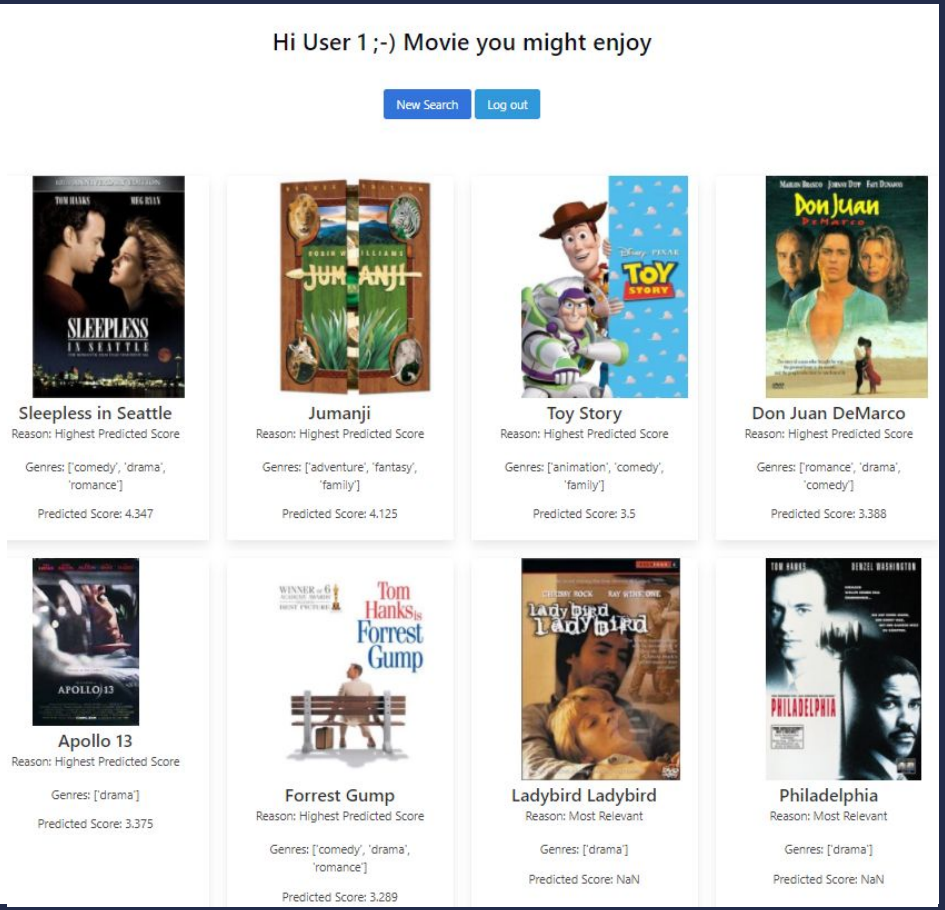
- Login with guest user. Searches using our cosine similarity content based recommender.
- Search keywords: {genre: love, actor: Tom Hanks, director: Noah Ephron}
- Expected result: List of movies that satisfies any of the three criteria. Should contain “Sleepless in Seattle” because it satisfies all three.



Case Study 2:

Search Using Collaborative Recommender With User 1

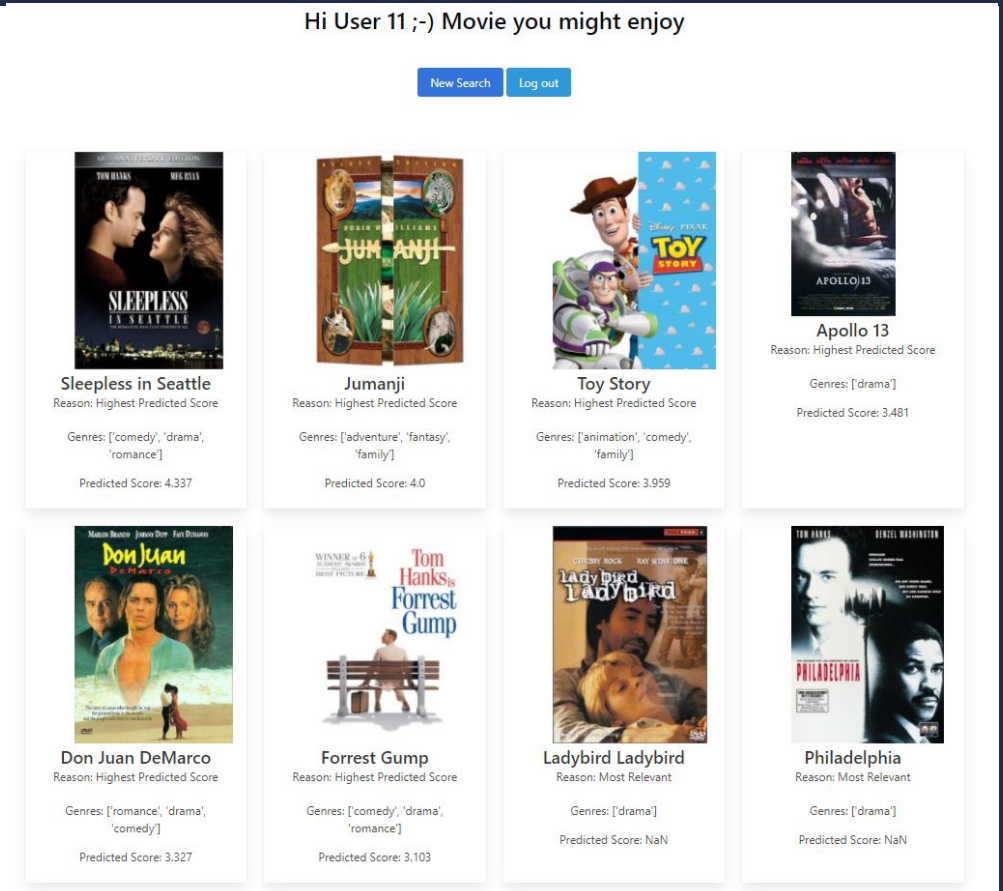
- Login with user 1. Searches using our collaborative filtering recommender.
- Search keywords: {genre: love, actor: Tom Hanks, director: Noah Ephron}.
- Expected result:
  - Movies are ranked with scores calculated from previous user 1 ratings and similarity of user 1 with other users.



Case Study 3:

Search Using Collaborative Recommender With User 11

- Login with user 11. Searches using our collaborative filtering recommender.
- Search keywords: {genre: love, actor: Tom Hanks, director: Noah Ephron}.
- Expected result: .
  - Should obtain different recommendation compared to user 1.

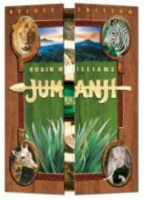


Case 2 v.s. Case 3

Hi User 1 ;-) Movie you might enjoy

New Search | Log out

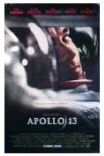
Sleepless in Seattle
Reason: Highest Predicted Score
Genres: ['comedy', 'drama', 'romance']
Predicted Score: 4.347



Jumanji
Reason: Highest Predicted Score
Genres: ['adventure', 'fantasy', 'family']
Predicted Score: 4.125

Toy Story
Reason: Highest Predicted Score
Genres: ['animation', 'comedy', 'family']
Predicted Score: 3.5

Don Juan DeMarco
Reason: Highest Predicted Score
Genres: ['romance', 'drama', 'comedy']
Predicted Score: 3.388



Apollo 13
Reason: Highest Predicted Score
Genres: ['drama']
Predicted Score: 3.375

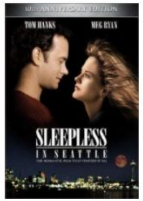
Forrest Gump
Reason: Highest Predicted Score
Genres: ['comedy', 'drama', 'romance']
Predicted Score: 3.289

Ladybird Ladybird
Reason: Most Relevant
Genres: ['drama']
Predicted Score: NaN

Philadelphia
Reason: Most Relevant
Genres: ['drama']
Predicted Score: NaN

Hi User 11 ;-) Movie you might enjoy

New Search | Log out



Sleepless in Seattle
Reason: Highest Predicted Score
Genres: ['comedy', 'drama', 'romance']
Predicted Score: 4.337

Jumanji
Reason: Highest Predicted Score
Genres: ['adventure', 'fantasy', 'family']
Predicted Score: 4.0

Toy Story
Reason: Highest Predicted Score
Genres: ['animation', 'comedy', 'family']
Predicted Score: 3.959

Apollo 13
Reason: Highest Predicted Score
Genres: ['drama']
Predicted Score: 3.481

Don Juan DeMarco
Reason: Highest Predicted Score
Genres: ['romance', 'drama', 'comedy']
Predicted Score: 3.327

Forrest Gump
Reason: Highest Predicted Score
Genres: ['comedy', 'drama', 'romance']
Predicted Score: 3.103

Ladybird Ladybird
Reason: Most Relevant
Genres: ['drama']
Predicted Score: NaN

Philadelphia
Reason: Most Relevant
Genres: ['drama']
Predicted Score: NaN

Future Work

- Deploy on public server.
- Incorporate natural language user input.
- Build explainable conversational recommendation system.

Conclusion:

- Implemented content based and collaborative filtering based recommender chatbot.
- Successfully recommend personalized list of movies based on user’s past ratings.

Questions?