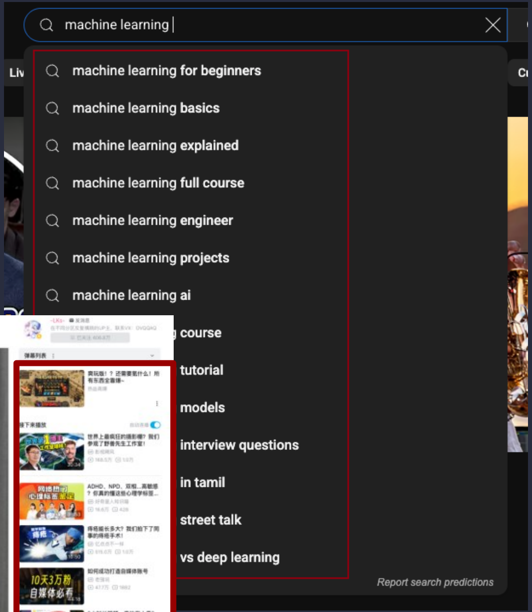
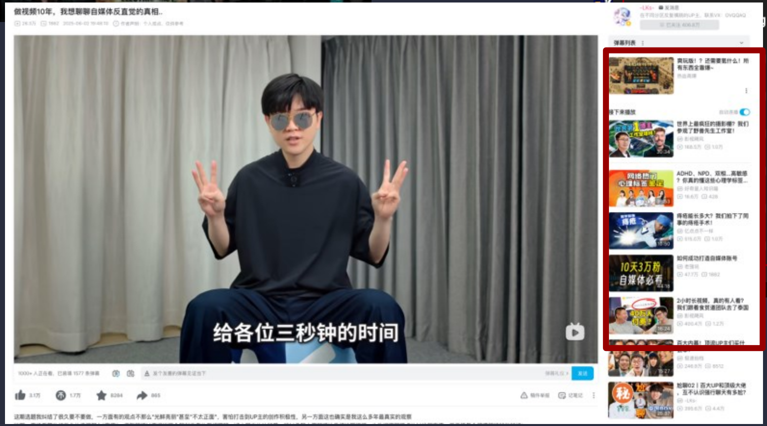
Building a Time-Aware Movie Recommendation System

CS 182 Final Project

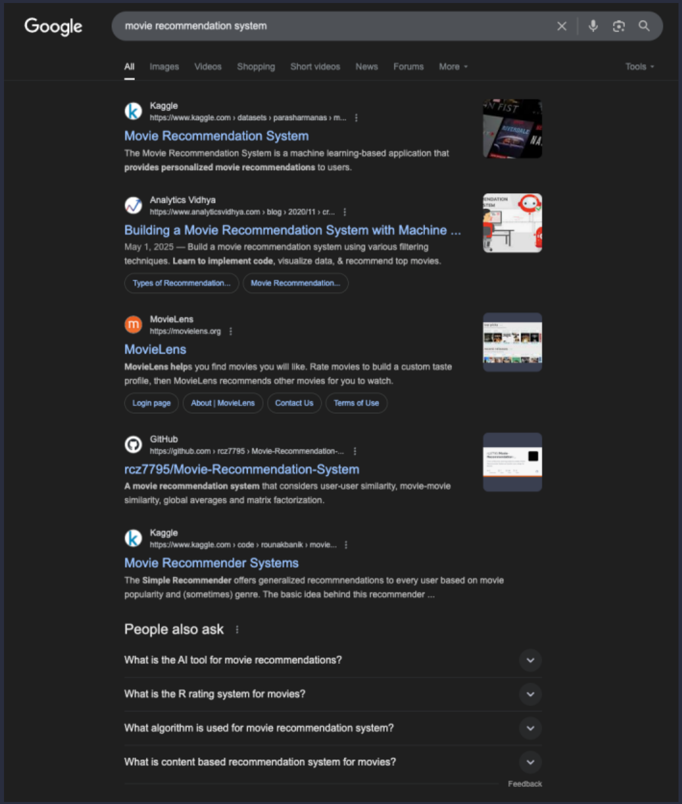
陈睦尧 夏鸣飞 张飏

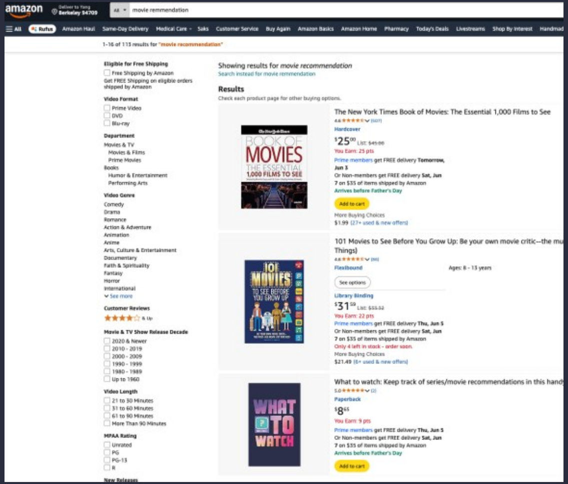
Introduction

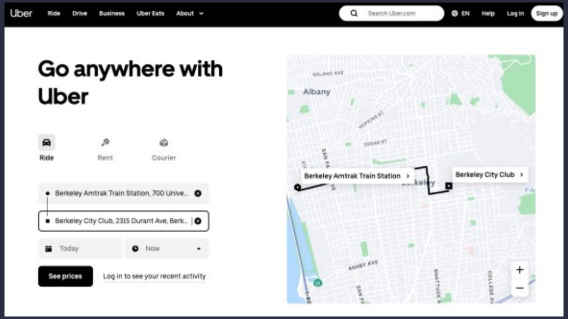




Not only for movies!







Not only for movies!

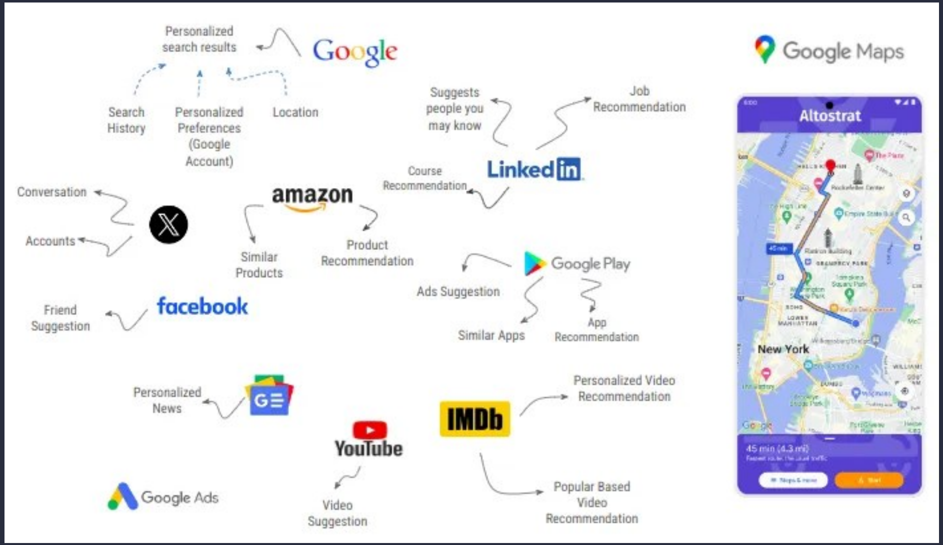


Figure from medium

Dataset

Movies: 3883 x 3

data > movies.csv > data

	movie_id	title	genres
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance
7	8	Tom and Huck (1995)	Adventure|Children's
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Advent
10	11	American President, The (1995)	C
11	12	Dracula: Dead and Loving It (1995	
12	13	Balto (1995)	Animation|Childre
13	14	Nixon (1995)	Drama
14	15	Cutthroat Island (1995)	Action|Ad
15	16	Casino (1995)	Drama|Thriller
16	17	Sense and Sensibility (1995)	D
17	18	Four Rooms (1995)	Thriller
18	19	Ace Ventura: When Nature Calls (1	

Users: 6040 x 5

data > users.csv > data

	user_id	gender	age	occupation	zipcode	age_desc	occ_desc
0	1	F	1	10	48067	Under 18	K-12 student
1	2	M	56	16	70072	56+	self-employed
2	3	M	25	15	55117	25-34	scientist
3	4	M	45	7	02460	45-49	executive/managerial
4	5	M	25	20	55455	25-34	writer
5	6	F	50	9	55117	50-55	homemaker
6	7	M	35	1	06810	35-44	academic/educator
7	8	M	25	12	11413	25-34	programmer
8	9	M	25	17	61614	25-34	technician/engi
9	10	F	35	1	95370	35-44	academic/educat
10	11	F	25	1	04093	25-34	academic/educat
11	12	M	25	12	32793	25-34	programmer
12	13	M	45	1	93304	45-49	academic/educat
13	14	M	35	0	60126	35-44	other or not sp
14	15	M	25	7	22903	25-34	executive/manag
15	16	F	35	0	20670	35-44	other or not sp
16	17	M	50	1	95350	50-55	academic/educat
17	18	F	18	3	95825	18-24	clerical/admin
18	19	M	1	10	48073	Under 18	K-12 studen

Ratings: 1000209 x 3

data > ratings.csv > data

	user_id	movie_id	rating	timestamp	user_emb_	
0	1	1193	5	978300760	0	1192
1	1	661	3	978302109	0	660
2	1	914	3	978301968	0	913
3	1	3408	4	978300275	0	3407
4	1	2355	5	978824291	0	2354
5	1	1197	3	978302268	0	1196
6	1	1287	5	978302039	0	1286
7	1	2804	5	978300719	0	2803
8	1	594	4	978302268	0	593
9	1	919	4	978301368	0	918
10	1	595	5	978824268	0	594
11	1	938	4	978301752	0	937
12	1	2398	4	978302281	0	2397
13	1	2918	4	978302124	0	2917
14	1	1035	5	978301753	0	1034
15	1	2791	4	978302188	0	2790
16	1	2687	3	978824268	0	2686
17	1	2018	4	978301777	0	2017
18	1	3105	5	978301713	0	3104

Our approach I: SVD

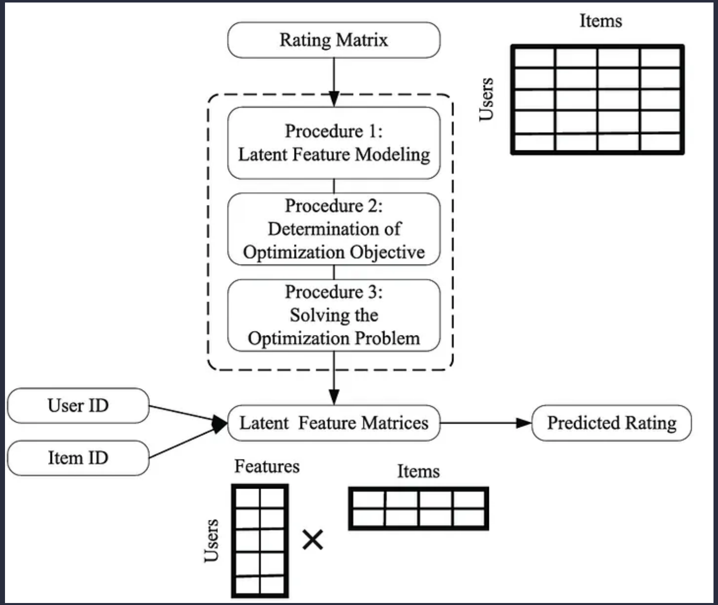
Assumption: there exists a low dimensional latent space of features in which we can represent both users and items and such that the interaction between a user and an item can be obtained by computing the dot product of corresponding dense vectors in that space.

SVD decomposes this matrix into three components:

$$M = U \cdot \Sigma \cdot V^*$$

U: A left singular vectors matrix representing user latent factors.
Σ: A diagonal matrix containing the singular values, which capture the importance of each latent factor.
V^T: A right singular vectors matrix representing item latent factors.

We **keep only the top 30 latent factors**, which capture the most important patterns in the data.
We then **reconstruct** the matrix using these factors → this gives us **predicted ratings** for movies the user hasn't rated yet.



Result

Test RMSE: 0.8736

Our approach II: Deep Learning (baseline)

Core Approach: Discover feature matrices for users and movies

Network Architecture: Embedding Layer → Dot Product Layer

- Each user and movie is assigned an embedding layer, representing their feature vectors

- Parameters are updated through training

By learning low-dimensional vector representations:

1. Similar users have close vectors; similar items have close vectors

2. High dot product between user and item vectors indicates preference

SVD Comparison:
- Finds global optimum but is static
- May not yield optimal predictions

Neural Network Advantage:
Better captures inherent characteristics of users and movies for prediction

Result

Test RMSE: 0.8616

Our approach III: Time-aware Model

Timestamp Utilization:
We explicitly leverage timestamp data recording when users rate movies

Objective:
Capture temporal distribution patterns in users' movie preferences

Model Architectures:

1. Time-Feature Additive Model:

- Processes time features separately
- Adds result to user-movie dot product

1. User Temporal Preference Model

- Adds time-attention layer on user embeddings
- Learns temporal biases (e.g., higher ratings on weekends/evenings)

1. User & Movie Temporal Preference Model

- Adds movie-specific time layer
- Captures preference shifts across eras

1. LSTM Temporal Evolution Model

- Incorporates LSTM to track preference trajectories
- Models sequential effects (e.g., action movie after action movie)
- Challenge: High complexity and computational cost (20 min for a training epoch)

Timestamp Processing:
Implemented three-tier hierarchical categorization:
1. Year level
2. Week level (weekday/weekend)
3. Day period (daytime/evening/early morning)

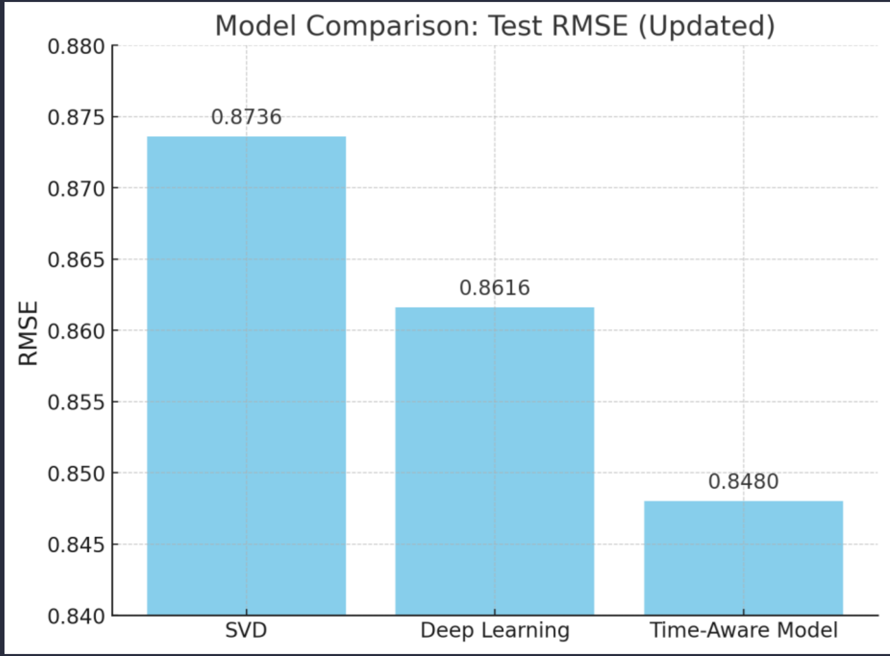
Result

Model 1 Test RMSE: 0.8530

Model 2 Test RMSE: 0.8532

Model 3 Test RMSE: 0.8480

Methods Comparison



Methods Comparison

Model	Accuracy (RMSE)	Pros	Cons
SVD	Worst 0.8736	Simple, efficient	Ignores nonlinear interactions
Deep Learning CF	Good 0.8616	Models nonlinear interactions, more expressive	Still lacks explicit time-awareness
Time-Aware CF	Best 0.8480	Captures evolving user preferences, best accuracy	More complex, needs user history data

Thank you