Movie recommendation

- Amish Popli

Problem statement : Creating a Recommendation Model for an OTT Platform -> Personalized Movie Suggestions

Dataset: User ratings for movies and metadata for movies (only genre)

userId	movieId	rating	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thriller	War	Western
1	1193	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	661	3	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	914	3	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
1	3408	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	2355	5	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Formulating an ML problem

Step 1 -> ML model - Build a regression (*or classification model assuming rating are discrete values from 1 to 5*) model to predict the rating of a movie given a user, movie and genre.

Step 2 -> Use the model to predict the movie rating for all movies and show the one with highest rating (Naive way - will discuss more further)

Step 3 -> Set up a feedback look (*A/B test or something else*) to assess the model and make improvements

Step 4 -> Make 💵💵💵

Formulating an ML problem

Step 1 -> ML model - Build a regression (*or classification model assuming rating are discrete values from 1 to 5*) model to predict the rating of a movie given a user, movie and genre.

Step 2 -> Use the model to predict the movie rating for all movies and show the one with highest rating (Naive way - will discuss more further)

Step 3 -> Set up a feedback look (*A/B test or something else*) to assess the model and make improvements

Step 4 -> Make 💵💵💵

We explored 3 methods to predict the rating

1. Dummy classifier (Baseline model)

2. Matrix Factorization

3. Deep learning based approach

1. Dummy classifier (Setting the baseline)

- Used a **uniform distribution** to assign values for a rating.
- MAE -> 3.7

2. Matrix Factorization

movieId	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
userId																																					
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	4.0	0.0	4.0	0.0	3.0	0.0

2. Matrix Factorization (Cont)

Item

User	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

Rating Matrix

=

User		
A	*1.2*	*0.8*
B	*1.4*	*0.9*
C	*1.5*	*1.0*
D	*1.2*	*0.8*

User Matrix

X

W	X	Y	Z
1.5	*1.2*	*1.0*	*0.8*
1.7	*0.6*	*1.1*	*0.4*

Item Matrix

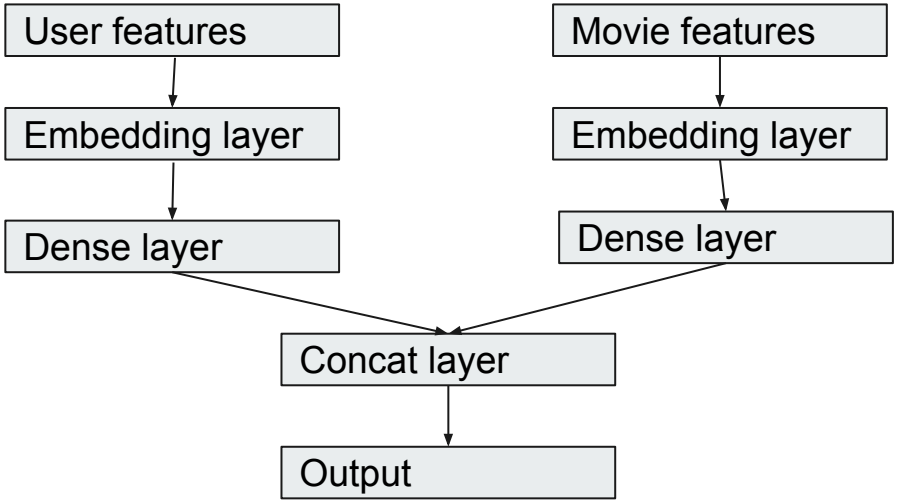
2. Matrix Factorization (Cont)

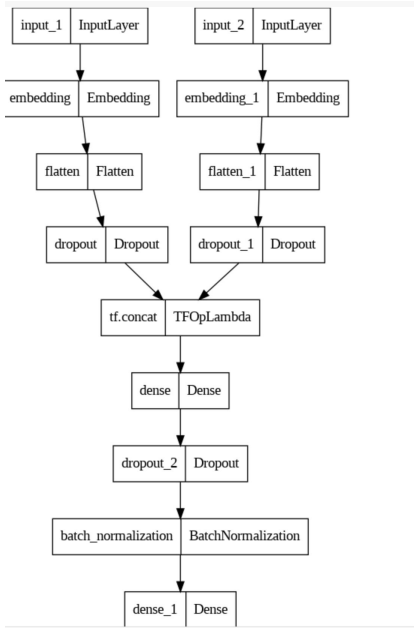
- MAE -> 3.1

Cons:

- Matrix factorization can suffer from overfitting and underfitting
- Cannot recommend for users/movies outside the dataset. (the split of test vs train causes the issue)

3. Deep learning model - Two tower architecture





3. Deep learning model (Cont)

- Model parameters
  - Adam
  - Sparse categorical entropy
  - 20 epochs
  - 80% train
  - 10% validation
  - 10% test

- **MAE -> 0.79**

Embeddings for FREE !!!

- Embeddings can open up other use cases:
  - User similarity (simple K means)
  - Movie similarity
  - Use vector databases to quickly find closest match for a user
  -

Questions ?