Movie Recommender System

Carol Duan

Overview

- Problem Defining
- Data Exploring & Cleaning
- Modeling & Evaluation
- Recommender System
- Next Step

Problem Defining

What is a recommender system?

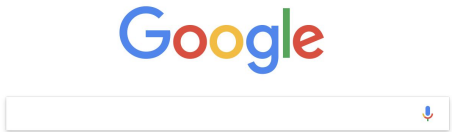
A system that is able to provide or suggest items to the end users

… long live the Age of Recommendation!

“We are leaving the age of information and entering the age of recommendation”

-- Chris Anderson in “The Long Tail”

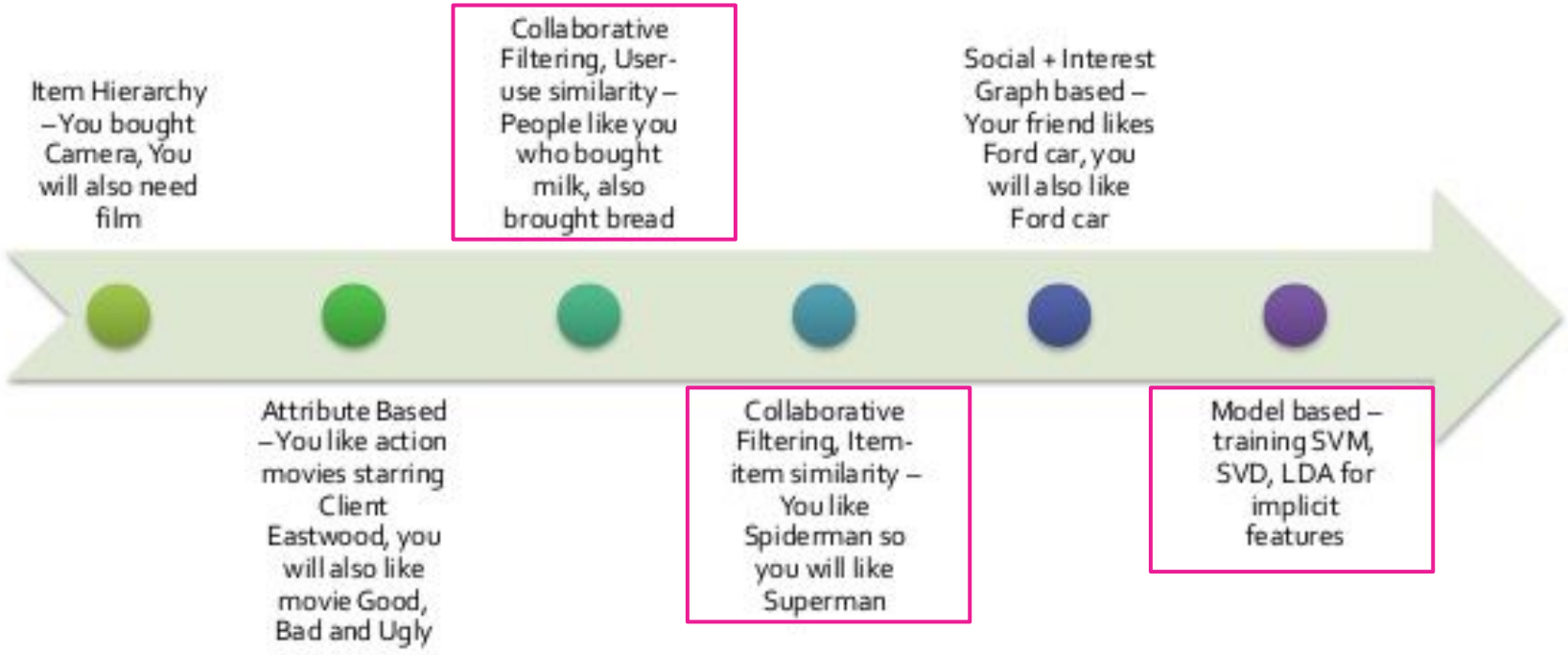
Almost every major tech company has applied recommender system in some form or the other…



Amazon: 35% sales from recommendations

Netflix: 2/3 of the movies watched are recommended

Evolution of Recommender System...



Collaborative / Content-based Recommender System

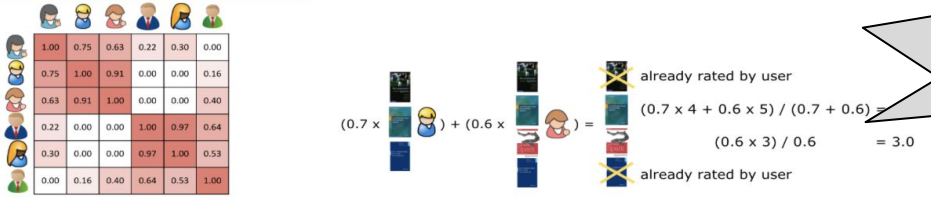
Predict ratings based on ratings from / of similar users or movies

	UserID	MovieID	Rating
0	1	1193	5
1	1	661	3
2	1	914	3
3	1	3408	4
4	1	2355	5

INPUT

Introduction to Recommender Systems
Machine learning Paradigms
Social Network-based Recommender Systems
Learning Spark
Recommender Systems Handbook
Recommender Systems and the Social Web

USER-BASED COLLABORATIVE FILTERING



ITEM-BASED COLLABORATIVE FILTERING

CONTENT-BASED FILTERING

How about other features?

OUTPUTS

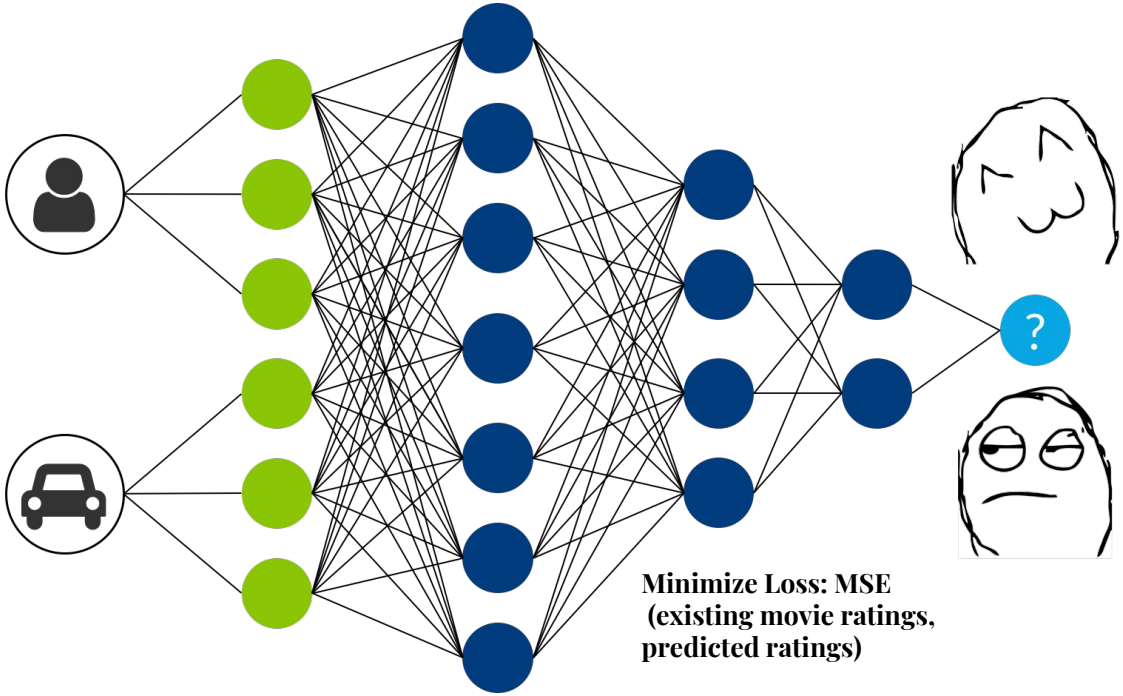
Model-based Recommender System (Neural Network)

User features

	UserID	Gender	Age	JobID
0	1	F	1	10
1	2	M	56	16
2	3	M	25	15
3	4	M	45	7
4	5	M	25	20

Movie features

	MovieID	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

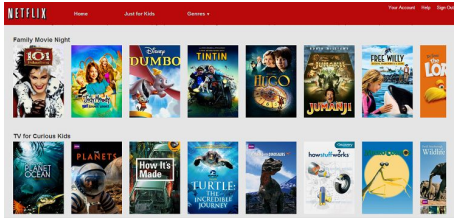


Minimize Loss: MSE
(existing movie ratings, predicted ratings)

Predicted ratings

	UserID	MovieID	Rating
0	1	1193	5
1	1	661	3
2	1	914	3
3	1	3408	4
4	1	2355	5

A Movie Recommender System



Goal: Build a movie recommender system

- For new user, recommend new similar movies by searching movie titles
- For existing user, predict their ratings for the unseen movies and give recommendations

Problem Type: Regression & Prediction

Workflow:

1. **Model:** Build a neural network using tensorflow to get user and movie feature vectors and predict movie ratings
2. **Recommender system:** Build a recommender system using the neural network model and implement the functions

Data Exploring & Cleaning

Data Explore

Database:

MovieLens 1M anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000.

- **Features (categorical)**
  - Users: UserID, Gender, Age, JobID
  - Movies: MovieID, Title, Genres
- **Target**
  - Ratings

Movie features

	MovieID	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

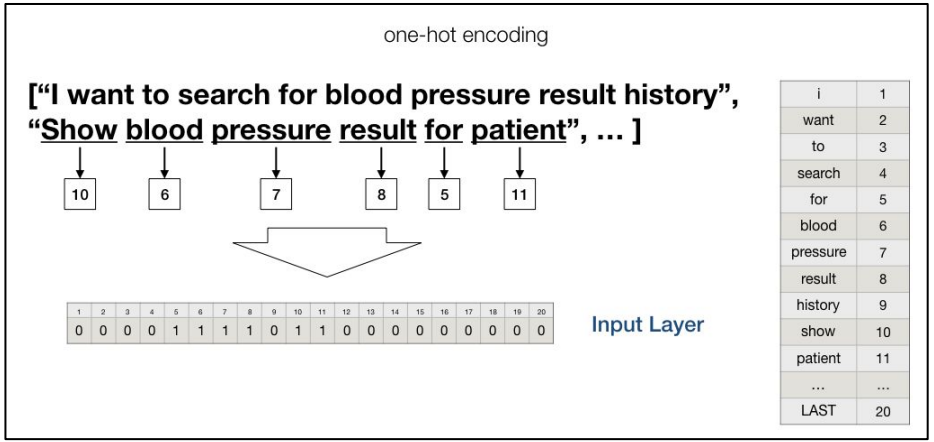
User features

	UserID	Gender	Age	JobID
0	1	F	1	10
1	2	M	56	16
2	3	M	25	15
3	4	M	45	7
4	5	M	25	20

Existing movie ratings

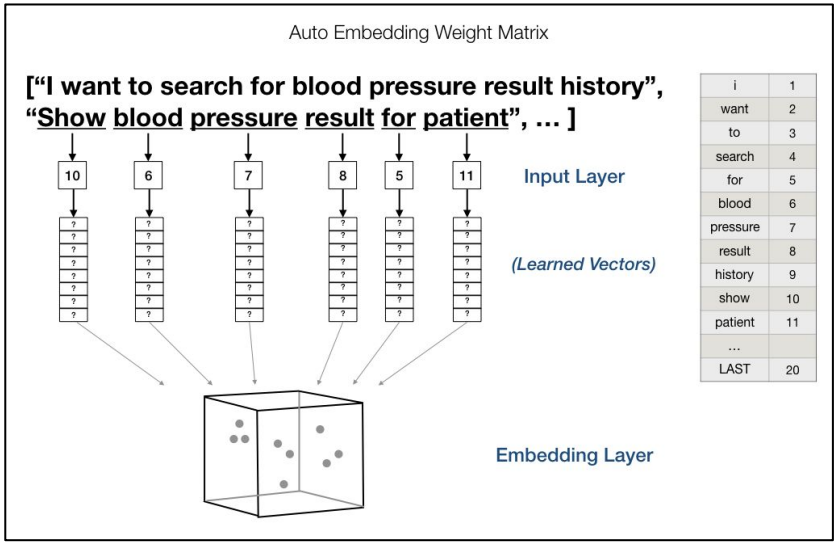
	UserID	MovieID	Rating
0	1	1193	5
1	1	661	3
2	1	914	3
3	1	3408	4
4	1	2355	5

Embedding Layer



One-hot encoding

Embedding layer



Map Categorical Data to Digits

- Gender: {'F': 0, 'M': 1}
- Age
- Movie Title
  - Create a word map for titles (add a value for blank “<PAD>”)
  - Turn it into a vector with length = max word numbers in a title
- Movie Genres

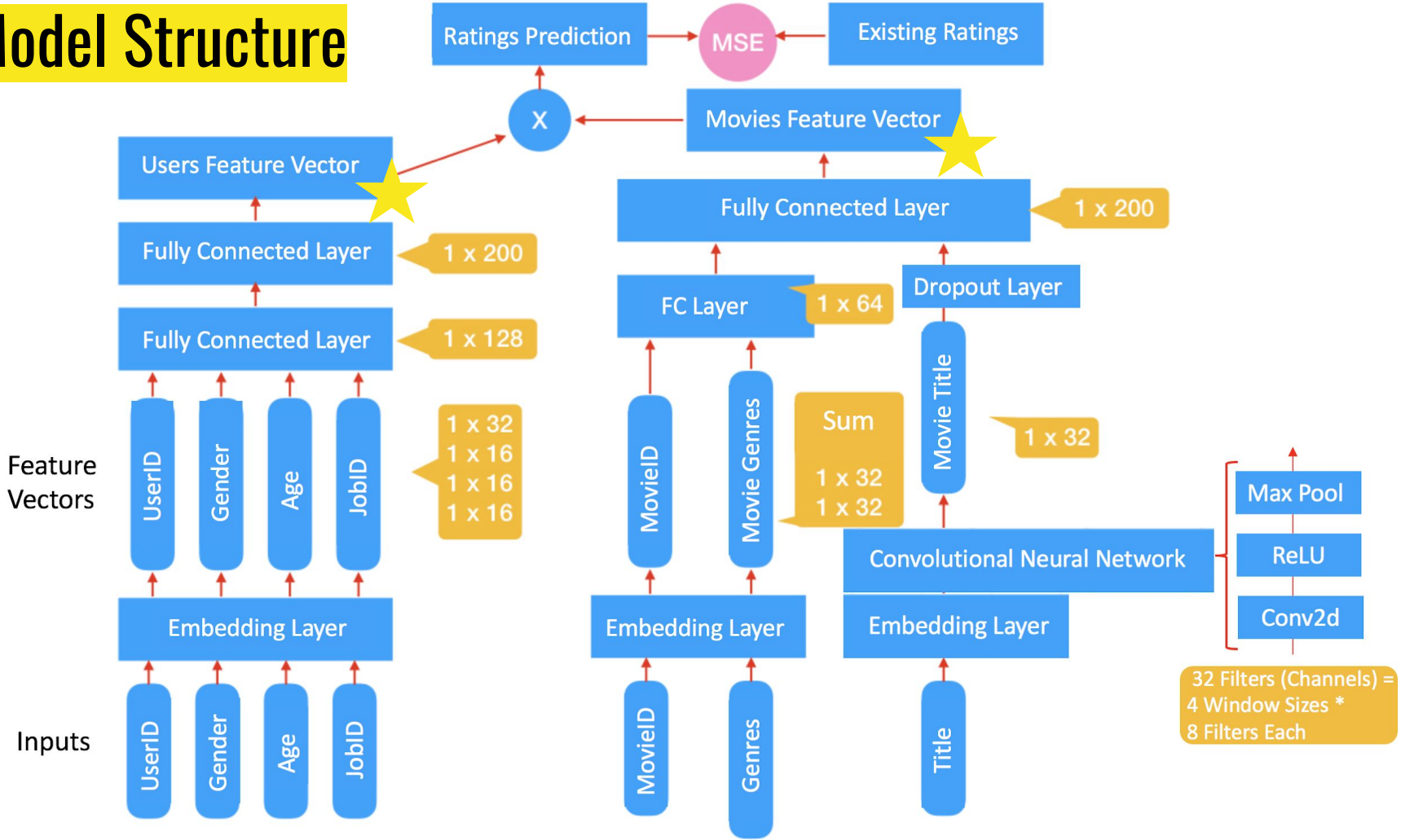
	UserID	MovieID	Rating	Gender	Age	JobID	Title	Genres
0	1	1193	5	0	0	10	[955, 2998, 92, 4067, 1298, 4266, 1170, 1170, ...	[4, 13, 13, 13, 13, 13]
1	2	1193	5	1	5	16	[955, 2998, 92, 4067, 1298, 4266, 1170, 1170, ...	[4, 13, 13, 13, 13, 13]
2	12	1193	4	1	6	12	[955, 2998, 92, 4067, 1298, 4266, 1170, 1170, ...	[4, 13, 13, 13, 13, 13]
3	15	1193	4	1	6	7	[955, 2998, 92, 4067, 1298, 4266, 1170, 1170, ...	[4, 13, 13, 13, 13, 13]
4	17	1193	5	1	3	1	[955, 2998, 92, 4067, 1298, 4266, 1170, 1170, ...	[4, 13, 13, 13, 13, 13]

Create Input Placeholders

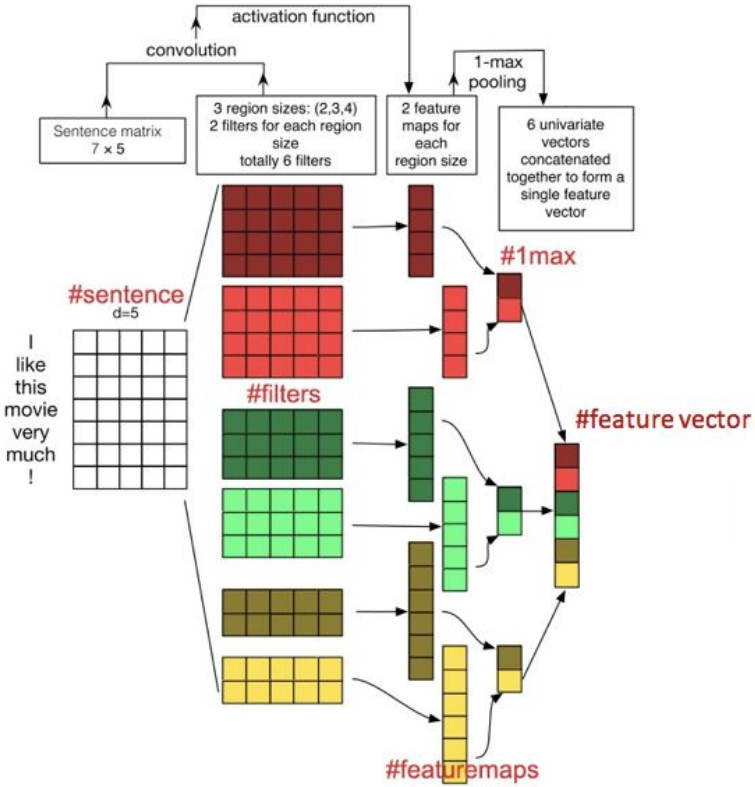
```
# inputs - placeholders
def get_inputs():
    uid = tf.placeholder(dtype=tf.int32, shape=[None, 1], name='uid')
    user_gender = tf.placeholder(dtype=tf.int32, shape=[None, 1], name='user_gender')
    user_age = tf.placeholder(dtype=tf.int32, shape=[None, 1], name='user_age')
    user_job = tf.placeholder(dtype=tf.int32, shape=[None, 1], name='user_job')

    movie_id = tf.placeholder(dtype=tf.int32, shape=[None, 1], name='movie_id')
    movie_genres = tf.placeholder(dtype=tf.int32, shape=[None, 6], name='movie_genres')
    movie_titles = tf.placeholder(dtype=tf.int32, shape=[None, 15], name='movie_titles')

    targets = tf.placeholder(dtype=tf.int32, shape=[None, 1], name='targets')

    learning_rate = tf.placeholder(dtype=tf.float32, name='learning_rate')
    dropout_rate = tf.placeholder(dtype=tf.float32, name='dropout_rate')

    return uid, user_gender, user_age, user_job, movie_id, movie_genres, movie_titles, \
           targets, learning_rate, dropout_rate
```

Modeling & Evaluation

Model Structure

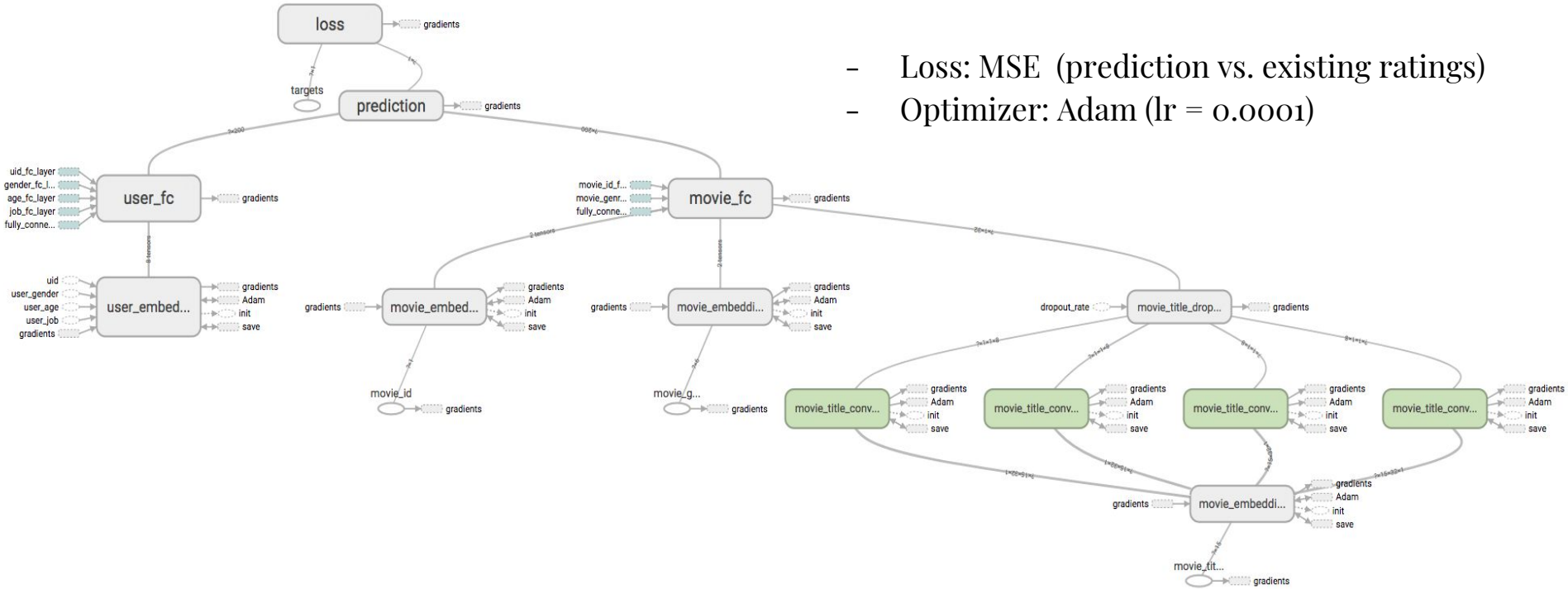


Movie Title NLP using Convolutional Neural Networks



- Input: movie title vectors after embedding layer (batch size, 15, 32)
- Convolutional 2d layer
  - Window size {2, 3, 4, 5}
  - Filter numbers: 8 (for each window)
  - Total 4*8=32 filters (channels)
- ReLU layer
- Max pooling layer
- Output: movie title feature vectors (1, 32)

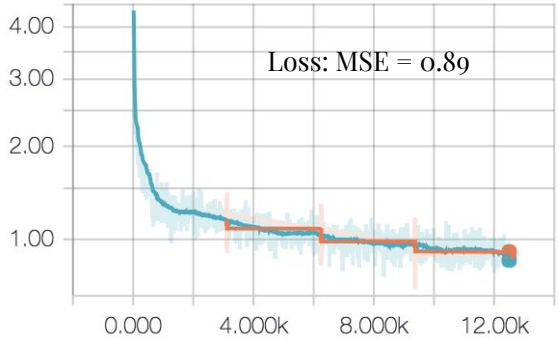
Model Implementation (TensorFlow)



- Loss: MSE (prediction vs. existing ratings)
- Optimizer: Adam (lr = 0.0001)

Model Evaluation and Comparison

- Train-Test Split (0.8 / 0.2)
- Model Training
  - Number of Epoch: 4
  - Batch Size: 256
  - Dropout rate: 0.5
  - Learning rate: 0.0001
- Model Testing



- Model Comparing (RMSE)
  - **Neural Network: 0.94**
  - User-based Collaborative Filtering: 699.96
  - Item-based Collaborative Filtering: 114.97

	Name	Smoothed	Value	Step	Time	Relative
●	prediction	0.8959	0.8959	12.50k	Fri Jul 13, 08:07:03	4m 19s
●	train	0.8148	0.8148	12.50k	Fri Jul 13, 08:07:03	5m 38s

Recommender System

Functions and Algorithm

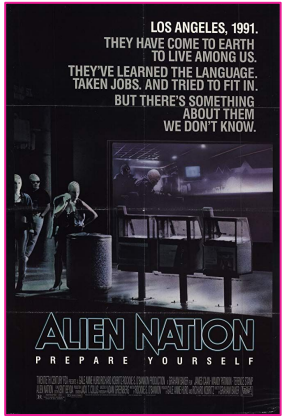
- For new users: recommend similar new movies by searching movie titles
  - Use **movie feature vectors**
  - Calculate cosine similarity between each movie feature vector - each row (1, 200)
  - For each movie, find the top 20 similar movies by ranking the cosine similarity numbers
  - Return 5 of the top 20 randomly each time (higher rank, higher possibility to show up)

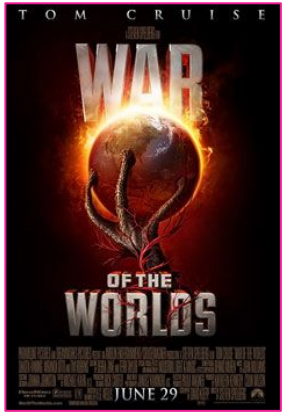
- For existing users: recommend new movies based on existing user data
  - For each user, use **user feature vector * movie feature vectors** (all movies)
  - Get the predicted movie ratings for new movies (for this user)
  - Find the top 20 recommended movies by ranking the predicted ratings
  - Return 5 of the top 20 randomly each time (higher ratings, higher possibility to show up)

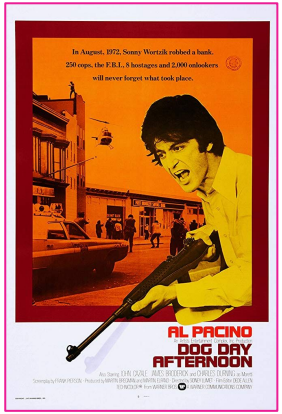
Showcase: New Users

Recommend similar new movies by searching movie titles

```
The movie you searched:

Movie ID:  1210
Movie Title:  Star Wars: Episode VI - Return of the Jedi (1983)
Movie Genres:  Action|Adventure|Romance|Sci-Fi|War
Average Rating:  4.0

Here are five movies you may like:

Movie ID:  2662
Movie Title:  War of the Worlds, The (1953)
Movie Genres:  Action|Sci-Fi|War
Average Rating:  3.9
---------------

Movie ID:  1372
Movie Title:  Star Trek VI: The Undiscovered Country (1991)
Movie Genres:  Action|Adventure|Sci-Fi
Average Rating:  3.4
---------------

Movie ID:  3701
Movie Title:  Alien Nation (1988)
Movie Genres:  Crime|Drama|Sci-Fi
Average Rating:  3.2
---------------

Movie ID:  3362
Movie Title:  Dog Day Afternoon (1975)
Movie Genres:  Comedy|Crime|Drama
Average Rating:  4.0
---------------

Movie ID:  969
Movie Title:  African Queen, The (1951)
Movie Genres:  Action|Adventure|Romance|War
Average Rating:  4.3
---------------
```



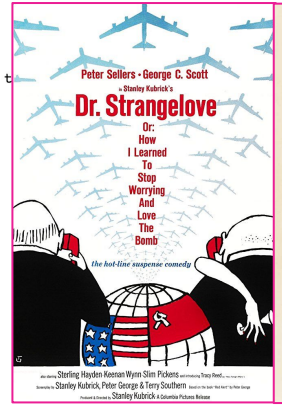


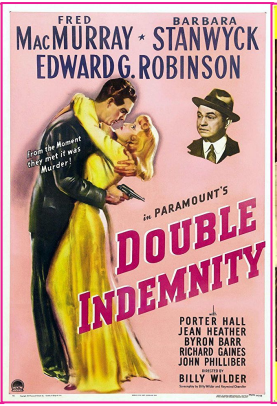


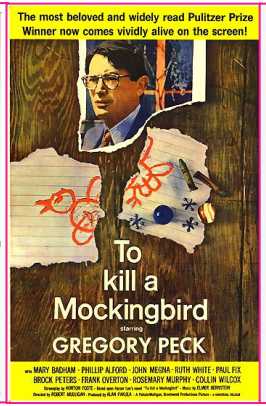
Showcase: Existing Users

Recommend new movies you may like based on your profile (user info / movie ratings)

```
Here is your profile:

User ID:  4
User Gender:  M
User Age:  45
Average Rating:  4.2

Here are five movies you may like:

Movie ID:  750
Movie Title:  Dr. Strangelove or: How I Learned t
Movie Genres:  Sci-Fi|War
Average Rating:  4.4
---------------

Movie ID:  913
Movie Title:  Maltese Falcon, The (1941)
Movie Genres:  Film-Noir|Mystery
Average Rating:  4.4
---------------

Movie ID:  3435
Movie Title:  Double Indemnity (1944)
Movie Genres:  Crime|Film-Noir
Average Rating:  4.4
---------------

Movie ID:  1207
Movie Title:  To Kill a Mockingbird (1962)
Movie Genres:  Drama
Average Rating:  4.4
---------------

Movie ID:  1284
Movie Title:  Big Sleep, The (1946)
Movie Genres:  Film-Noir|Mystery
Average Rating:  4.3
---------------
```



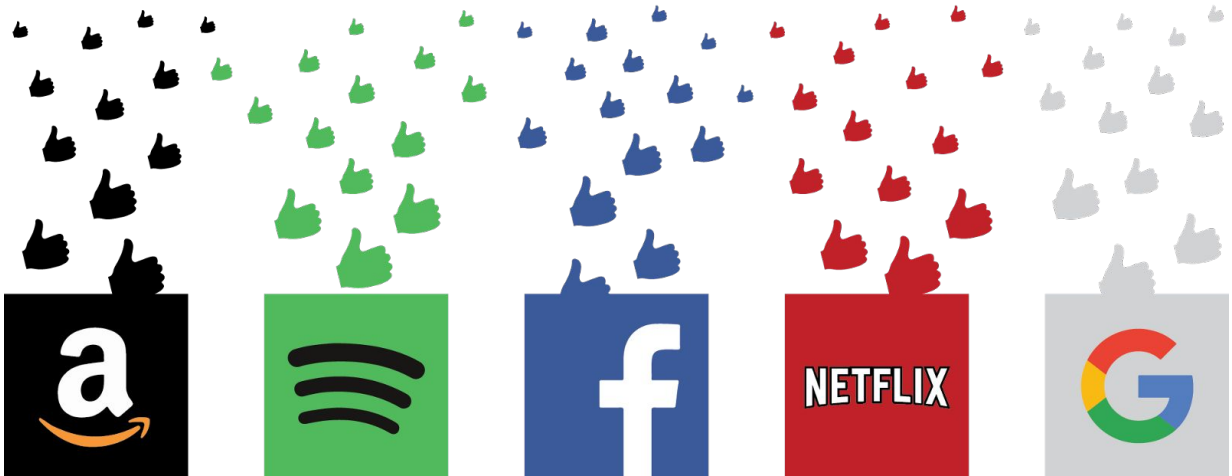




Next Step

Next Step

1. Train the model on a bigger dataset, add more movies
2. Explore other models (e.g. SVD) and compare the score
3. Build an user interface for better interactions



Thanks for Watching!