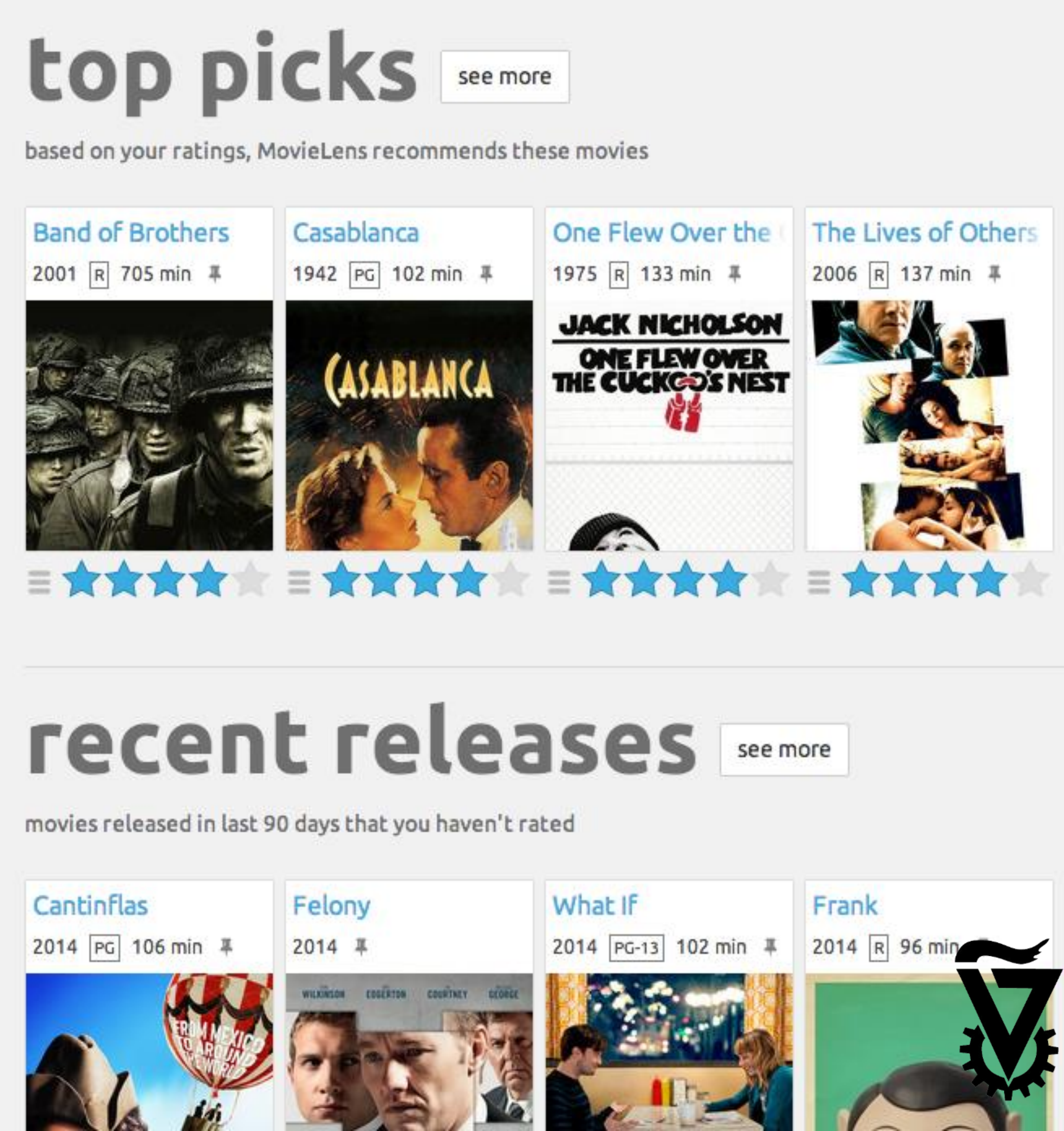
MovieLens Project

Recommendation System

Dean Shabi
Dedi Kovach
July 2019

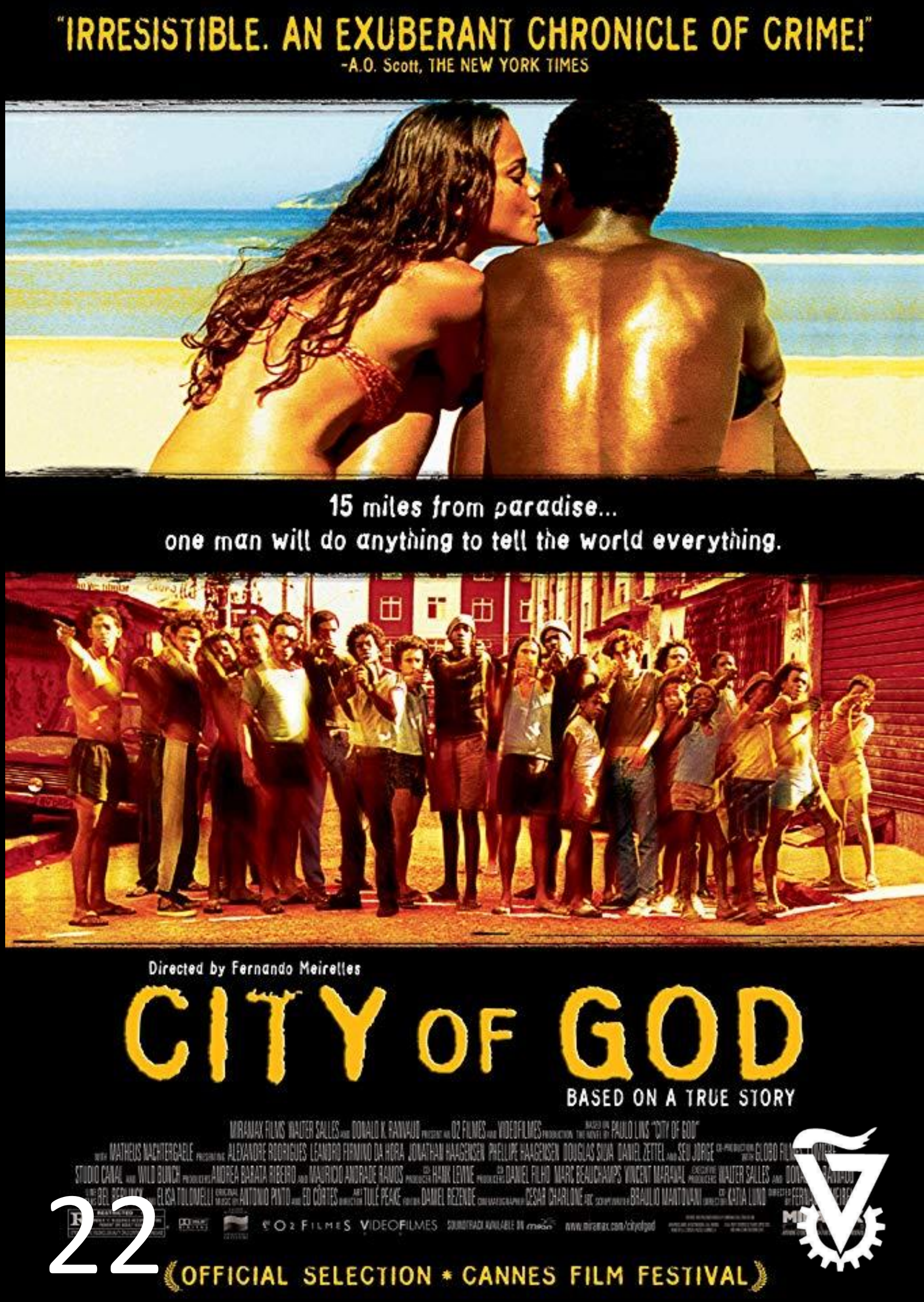
MovieLens

- **Movielens** - web service by GroupLens
- A research lab at the University of Minnesota
- Since 1997, Publishes Movielens recommender-system datasets



Datasets

- 100K Dataset, 1998 (original task)
  - (9K M, 600 U)
- 100K Dataset, 2018
- 1M, 10M Dataset, 2018
- **20M Dataset, 2016**
  - The benchmark for movie recommendations
  - 27K Movies
  - 138K Users
  - 465K Tags
  - 25K links to YouTube trailers



Target

Predict best movie for user based on user ratings

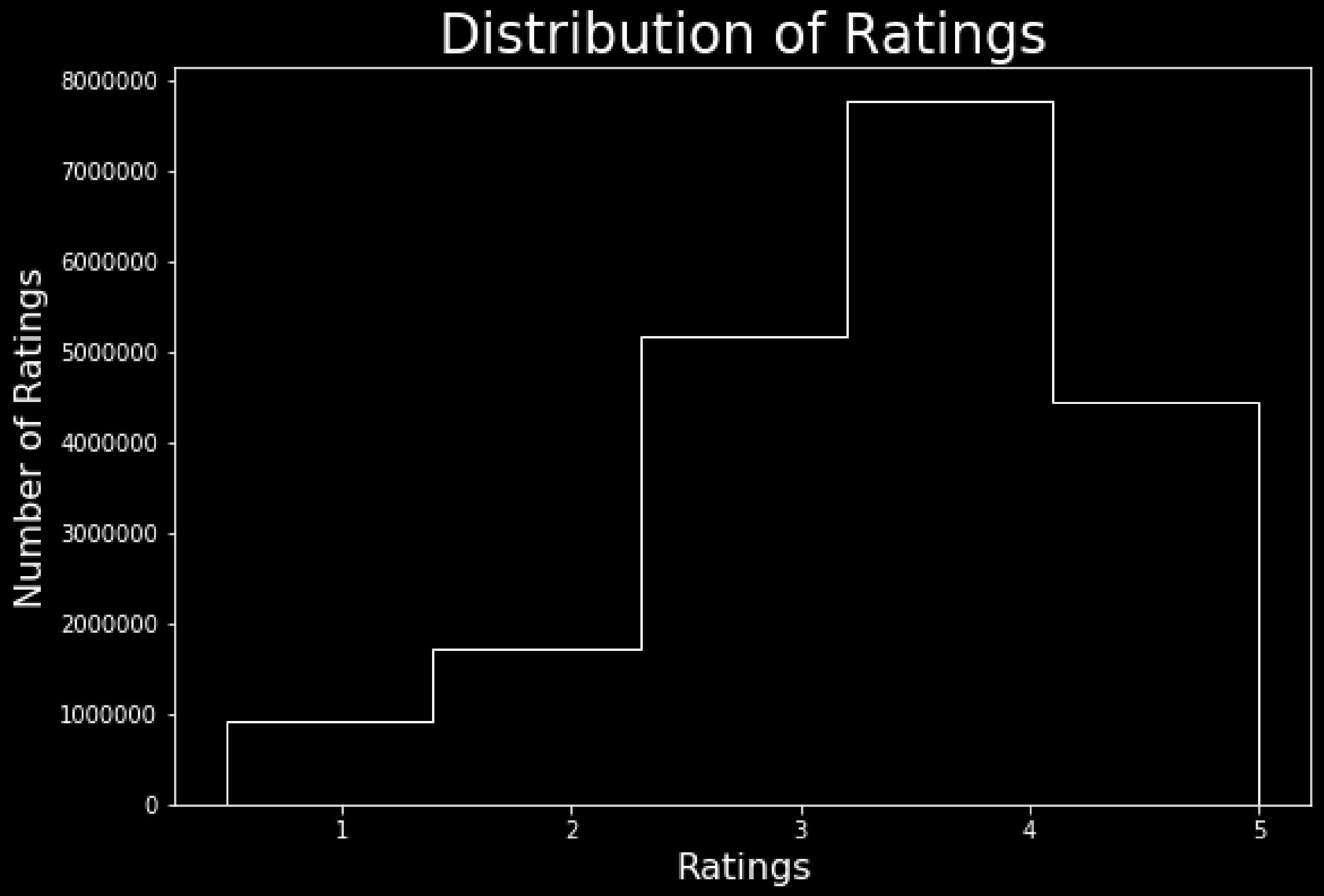
“You're no messiah. You're a movie of the week. You're a fucking t-shirt, at best.” David Mills

Objectives

Best Scores	Course Knowledge	Production
		

"There was an idea to bring together a group of remarkable people, to see if we could become something more". Nick Fury

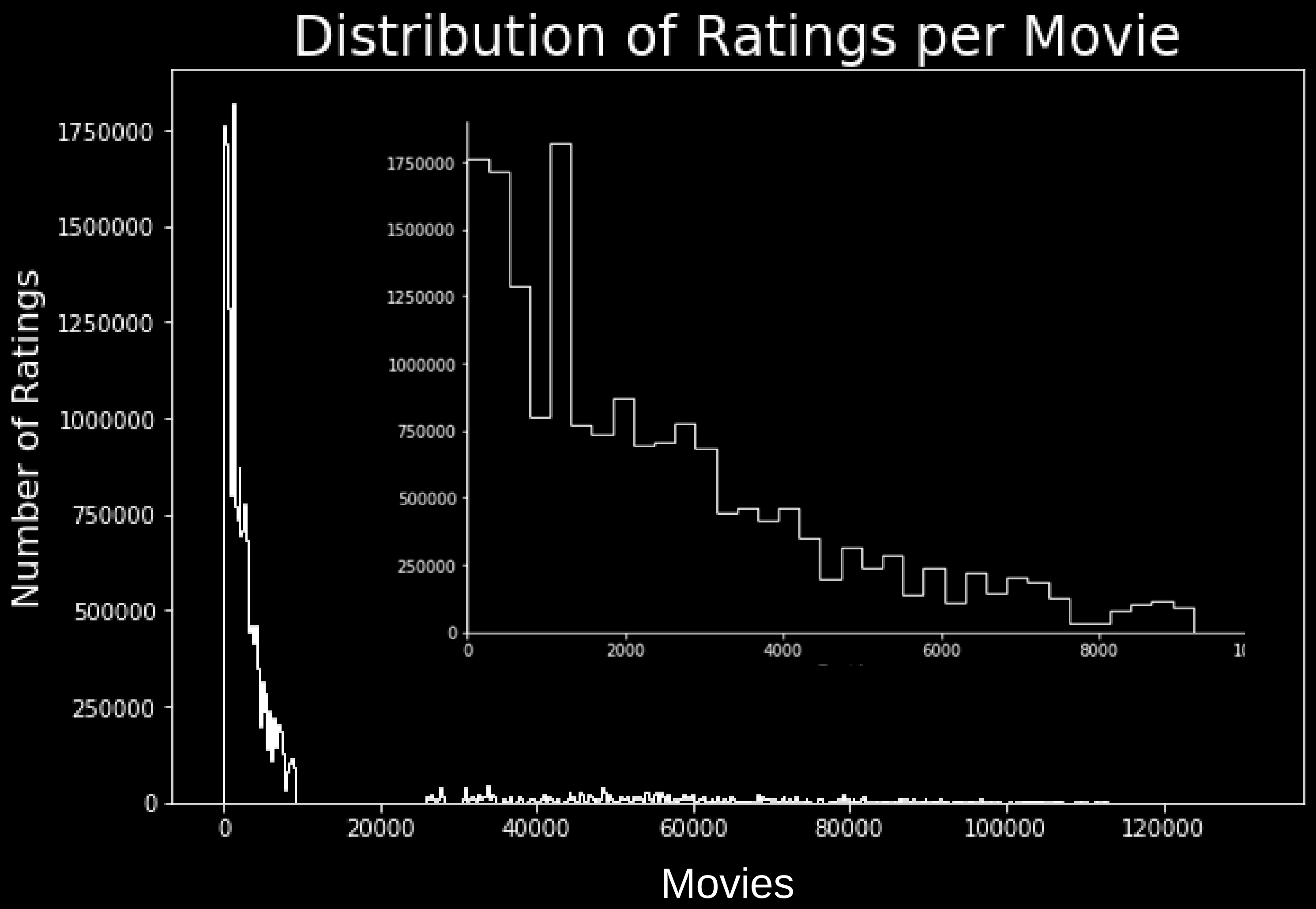
EDA



Worst rated:
Bratz: The Movie (2007) – 1.10, N=180
Glitter (2001) – 1.12, N=685

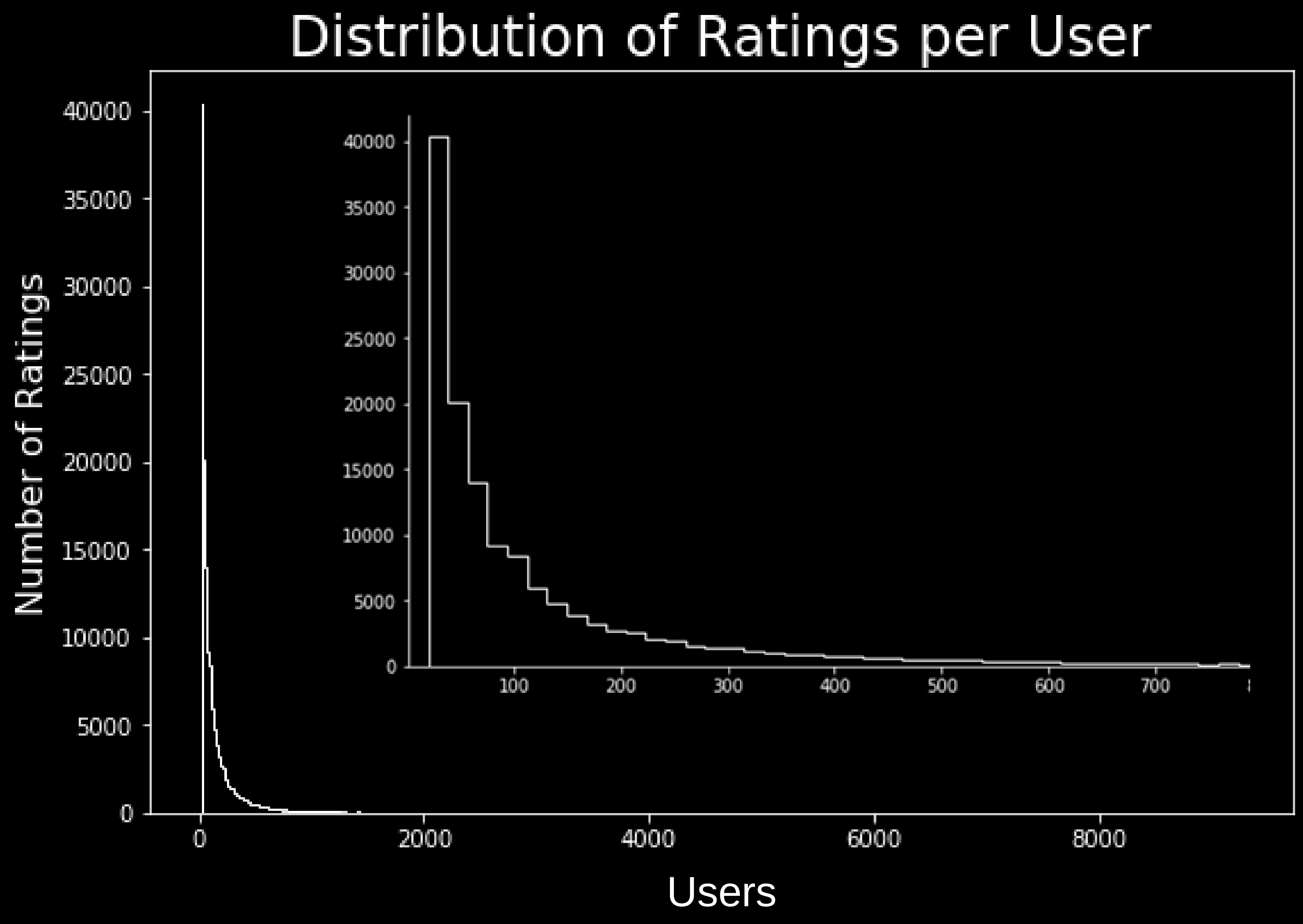
Best rated:
The Shashank Redemption (94) – 4.44, N=63K
The Godfather (72) – 4.36, N=41K

EDA



Most rated:
Pulp Fiction (94): N= 67K
Forrest Gump (94): N=66K
The Shashank Redemption (94): N=63K

EDA


Distribution of Ratings per User
Number of Ratings
Users
40000
35000
30000
25000
20000
15000
10000
5000
0
0
2000
4000
6000
8000
100
200
300
400
500
600
700


ROBERT DE NIRO
RAY LIOTTA
JOE PESCI
"As far back as I can remember, I've always wanted to be a gangster."
–Henry Hill, Brooklyn, N.Y. 1955.
GoodFellas
A MARTIN SCORSESE PICTURE
Three Decades of Life in the Mafia.
WARNER BROS.

Methods & Libraries

We used the following algorithms

- **Correlations** analysis
- **Content based** filtering
- **Sklearn** (CF)
- **Surprise** library (CF)
- FastAI (Embeddings)
- Keras - NN, Embeddings
- **Matrix Factorization CF** (Andrew Ng)
- **Singular Value Decomposition** (SVD)
- **Funk SVD**

"which one of you nuts has got any guts?" McMurphy

Collaborative Filtering

- **Collaborative Filtering (CF)** is a mean of recommendation based on users' past behavior.
- **Core assumption** here is that the users who have agreed in the past tend to also agree in the future
- **User-based:** measure the similarity between target users and other users.
- **Item-based:** measure the similarity between the items that target users rates/ interacts with and other items

"It's the job that's never started as takes longest to finish."

Matrix factorization:

CF weaknesses = Sparsity and Scalability

⇩

Decompose the original sparse matrix to low-dimensional matrices with latent factors/features and less sparsity

⇩

Matrix Factorization

"Unfortunately, no one can be told what The Matrix is. You'll have to see it for yourself." ***Morpheus***

SVD:

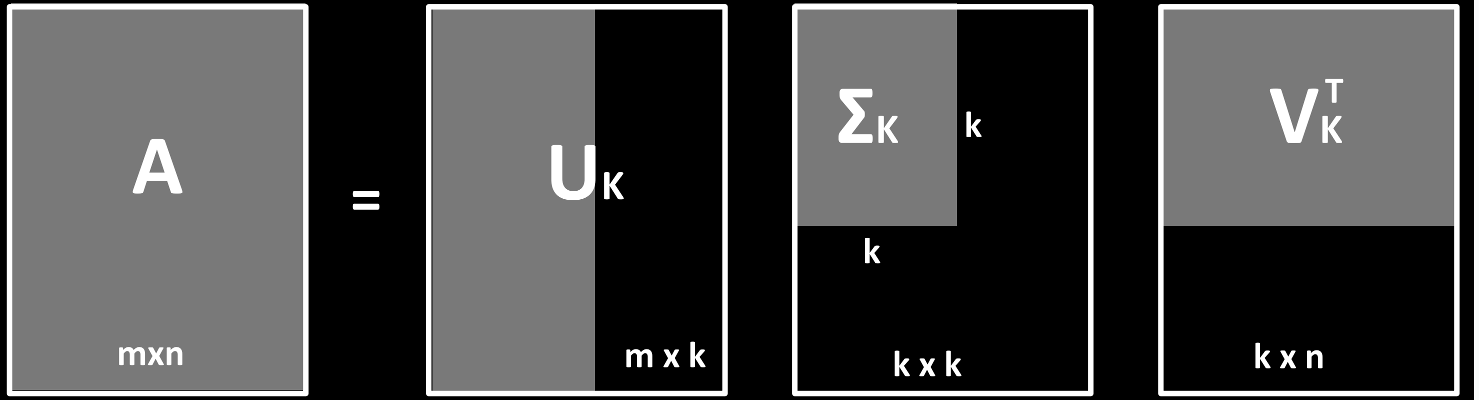
- SVD is an algorithm that decomposes a matrix R into the best lower rank approximation of the original matrix R.
- Mathematically, it decomposes R into 3 matrices: 2 unitary matrices and 1 diagonal matrix:

$$R=U\Sigma VT$$

"You mustn't be afraid to dream a little bigger, darling." Christopher Nolan

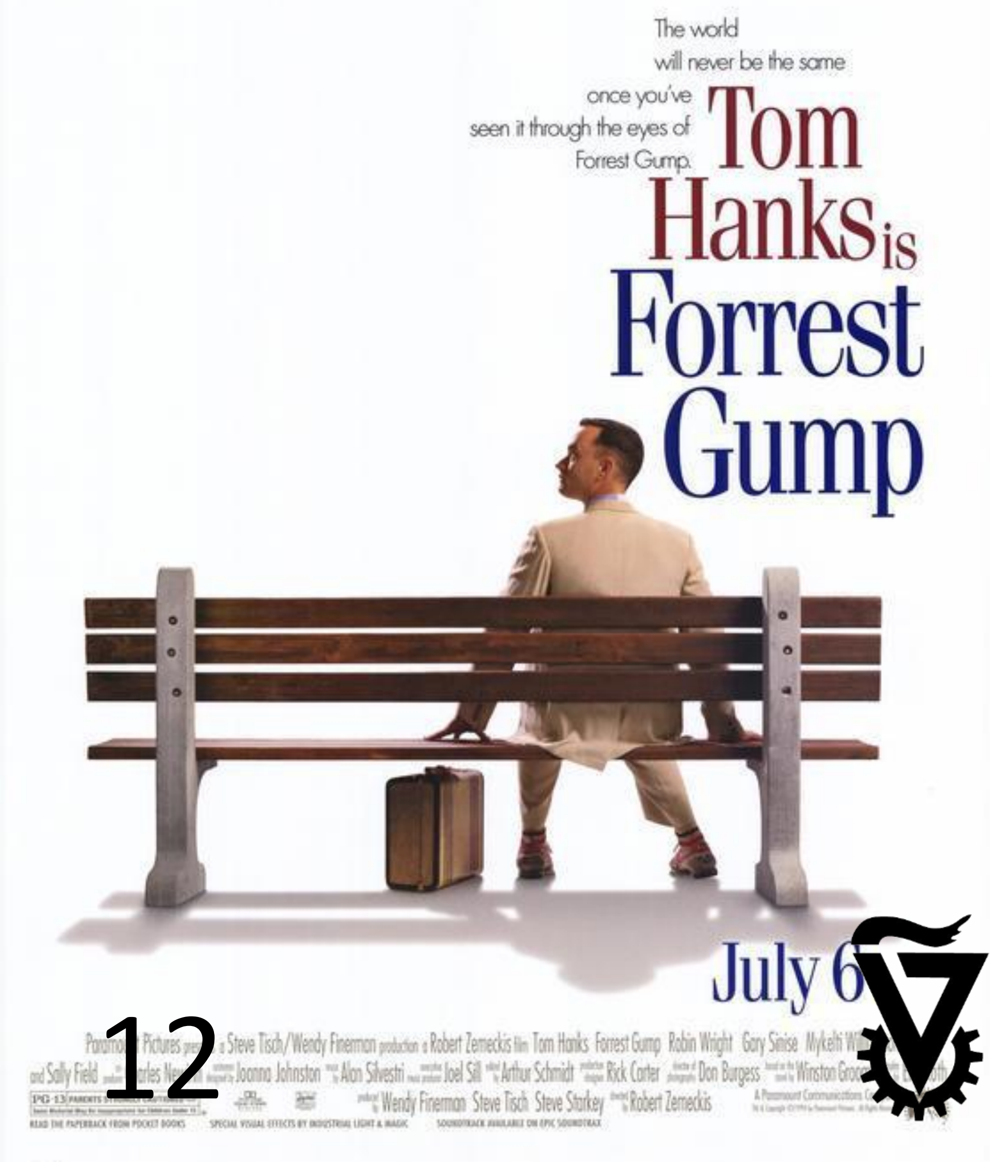
SVD:

R=UΣVT



$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}_{m \times n} \approx \begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}_{m \times r} \begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}_{r \times r} \begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}_{r \times n}$$

"Have you found Jesus yet, Gump?" – Lieutenant Daniel Taylor
"I didn't know I was supposed to be looking for him, sir." – Forrest Gump



Funk SVD:
Based on Simon Funk Algorithm

- **S**imple to use
- **V**ery fast
- **D**iamond class results

Got 3rd place in Netflix Prize
on April 23rd, 2007

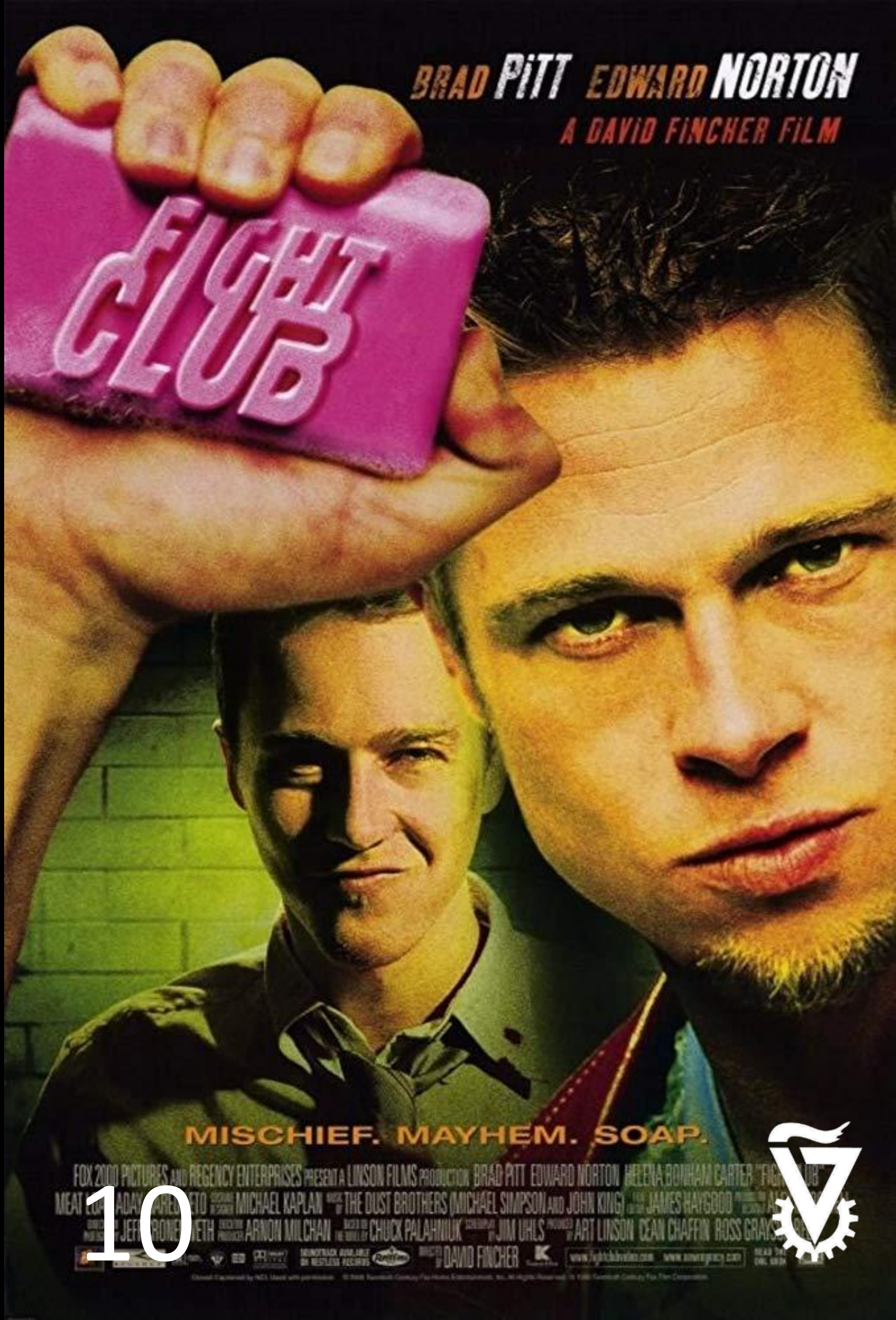
Funk SVD:

Based on Simon Funk Algorithm

Not an SVD but a MF method,
Fast for sparse rating matrices.

$$\begin{bmatrix} 1 & ? & 5 \\ ? & ? & 3 \\ 1 & 3 & 5 \\ 3 & 4 & 4 \end{bmatrix} \approx \begin{bmatrix} (\theta_1)^T \times x_1 & - & (\theta_1)^T \times x_3 \\ - & - & (\theta_2)^T \times x_3 \\ (\theta_3)^T \times x_1 & (\theta_3)^T \times x_2 & (\theta_3)^T \times x_3 \\ (\theta_4)^T \times x_1 & (\theta_4)^T \times x_2 & (\theta_4)^T \times x_3 \end{bmatrix}$$

“It's only after we've lost everything that we're free to do anything.” Chuck Palahniuk



Model:

Step 1 – Random Search

Sampling hyperparameters from a distributions and training the algorithm.

- **Iterations**: 100 (with Early Stopping on Val Loss)
- **Regularization**: Uniform(0.001, 0.1)
- **Learning Rate**: Uniform(0.001,0.1)
- **Factors** = Uniform(30,500)
- **n_epochs** = 100 trials

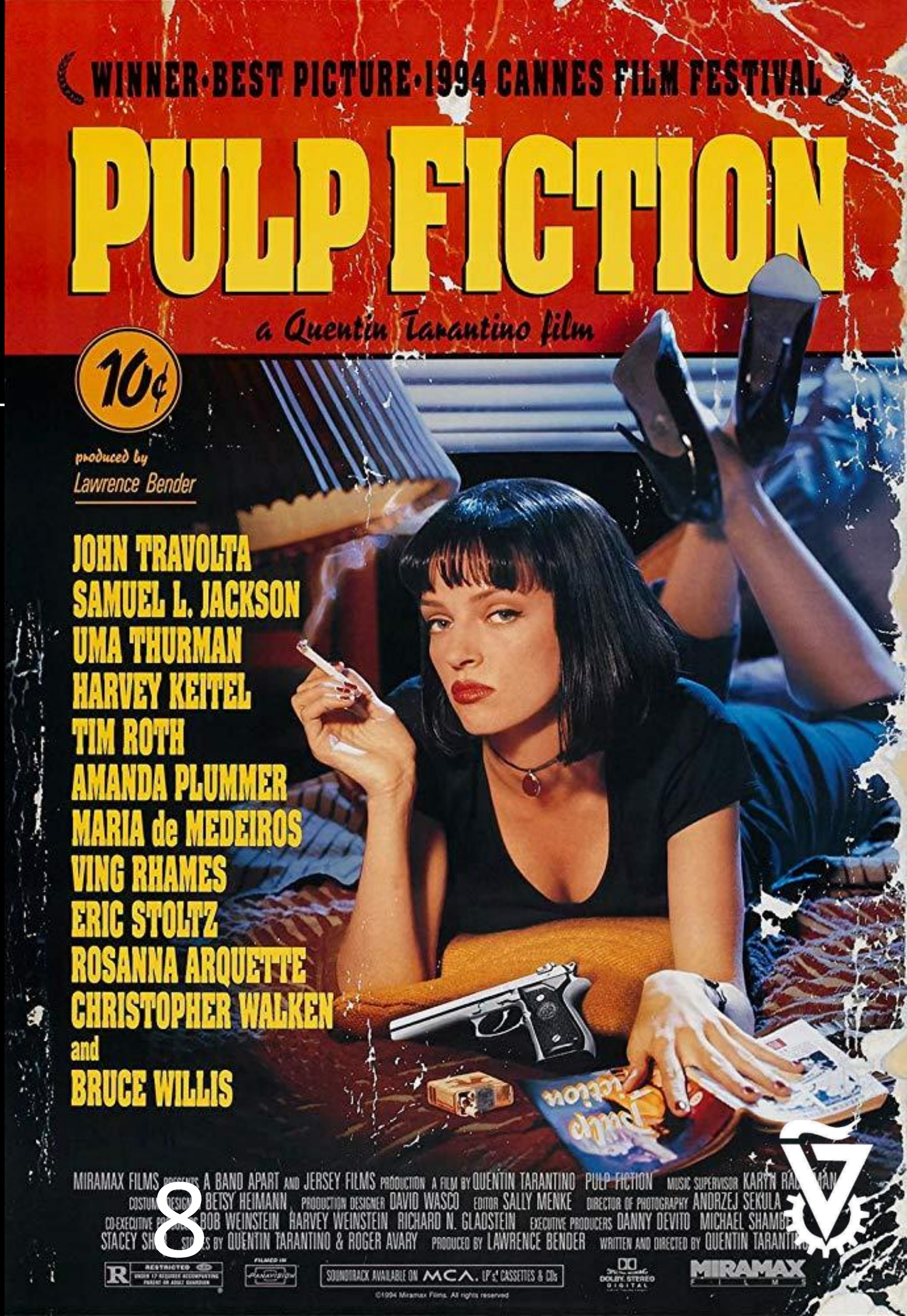
“Every gun makes its own tune.” The Good

Model:

Step 1 – Random Search

Epoch 1/200 | val_loss: 0.77 - val_rmse: 0.87 - val_mae: 0.67 - took 6.9 sec
Epoch 2/200 | val_loss: 0.75 - val_rmse: 0.86 - val_mae: 0.67 - took 6.2 sec
Epoch 3/200 | val_loss: 0.73 - val_rmse: 0.85 - val_mae: 0.66 - took 6.2 sec
Epoch 4/200 | val_loss: 0.71 - val_rmse: 0.84 - val_mae: 0.65 - took 6.1 sec
Epoch 5/200 | val_loss: 0.70 - val_rmse: 0.83 - val_mae: 0.64 - took 6.1 sec
Epoch 6/200 | val_loss: 0.68 - val_rmse: 0.83 - val_mae: 0.63 - took 6.1 sec
Epoch 7/200 | val_loss: 0.67 - val_rmse: 0.82 - val_mae: 0.63 - took 6.1 sec
Epoch 8/200 | val_loss: 0.66 - val_rmse: 0.81 - val_mae: 0.62 - took 6.1 sec
Epoch 9/200 | val_loss: 0.65 - val_rmse: 0.81 - val_mae: 0.62 - took 6.1 sec
Epoch 10/200 | val_loss: 0.64 - val_rmse: 0.80 - val_mae: 0.61 - took 6.1 sec
Epoch 11/200 | val_loss: 0.64 - val_rmse: 0.80 - val_mae: 0.61 - took 6.1 sec
Epoch 12/200 | val_loss: 0.63 - val_rmse: 0.80 - val_mae: 0.61 - took 6.1 sec
Epoch 13/200 | val_loss: 0.63 - val_rmse: 0.79 - val_mae: 0.61 - took 6.1 sec
Epoch 14/200 | val_loss: 0.62 - val_rmse: 0.79 - val_mae: 0.60 - took 6.1 sec

"Just because you are a character doesn't mean that you have character." -The Wolf



Model:

Step 2 – Training Best Model

- Training took 2 min and 32 sec
  - 90 latent features (factors)
  - Learning Rate: 0.007
  - Lambda: 0.03
- **Test MAE: 0.59**
- Test RMSE: 0.77

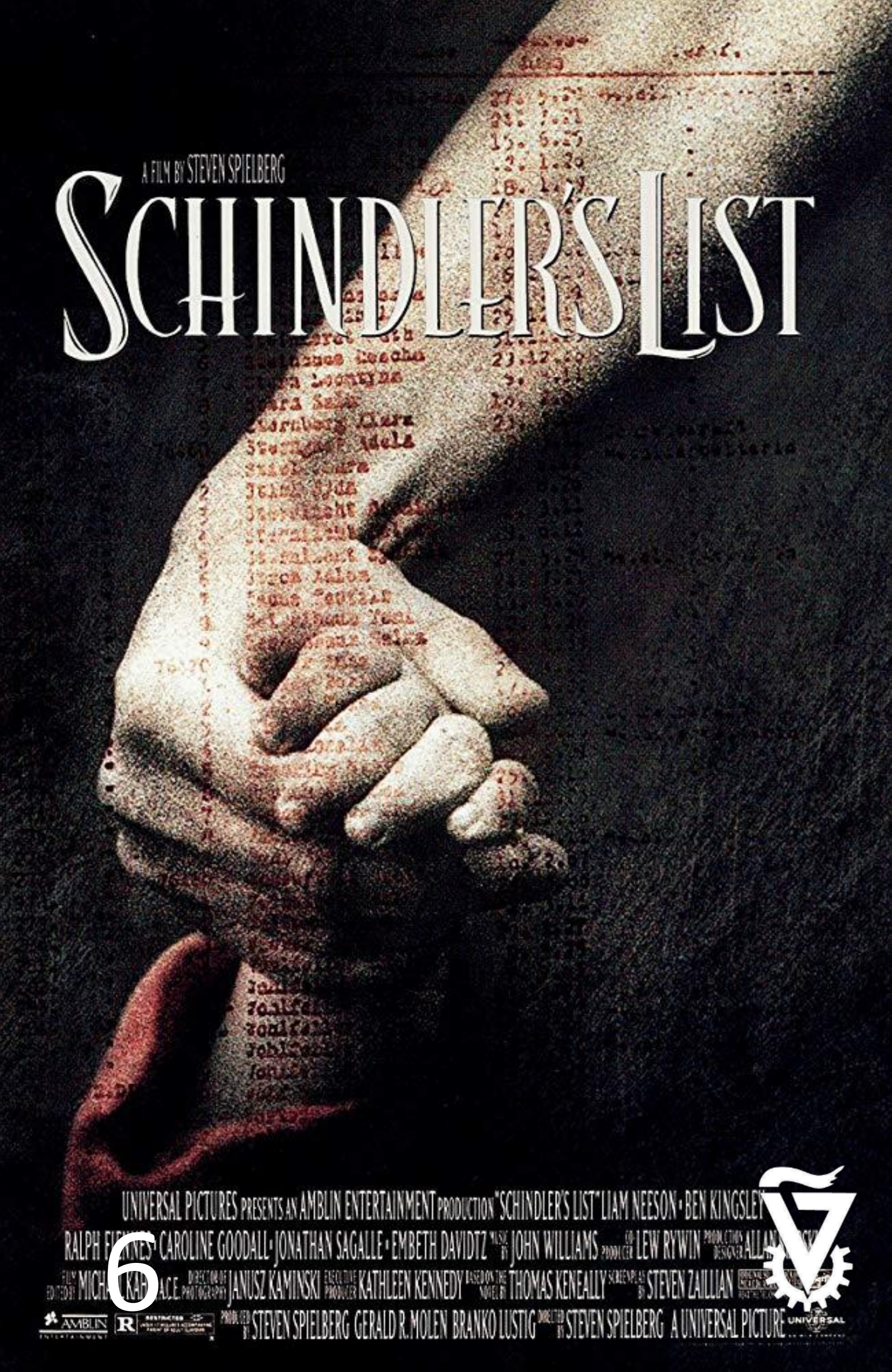
And getting recommendations.

Model:

Step 3 – Analyze Recommendations

Rated Movies	R
Star Wars: Episode IV - A New Hope (1977)	5
Monty Pythons Life of Brian (1979)	5
Star Wars: Episode V - The Empire Strikes Back (1980)	5
Star Wars: Episode VI - Return of the Jedi (1983)	5
Star Wars: Episode I - The Phantom Menace (1999)	5
Harry Potter and the Sorcerers Stone (2001)	5
The Lord of the Rings: The Fellowship of the Ring (2001)	5

Rated Movies	R
The Lord of the Rings: The Two Towers (2002)	5.0
Band of Brothers (2001)	5.0
The Lord of the Rings: The Return (2003)	5.0
Phone Box, The (Cabina, La) (1972)	5.0
Matrix, The (1999)	5.0
Star Wars: Ep. II - Attack of the Clones (2002)	5.0
Shawshank Redemption, The (1994)	4.98
Prime Suspect (1991)	4.97
Frozen Planet (2011)	4.97
Usual Suspects, The (1995)	4.96
Monty Python and the Holy Grail (1975)	4.94
Princess Bride, The (1987)	4.93
Indiana Jones and the Last Crusade (1989)	4.92
Jekyll (2007)	4.91
Braveheart (1995)	4.91
Cosmos (1980)	4.91
Day of the Doctor, The (2013)	4.90
Jim Gaffigan: Obsessed (2014)	4.89
Guardians of the Galaxy (2014)	4.89
Welfare (1975)	4.88
Raiders of the Lost Ark (1981)	4.87
Runaway Brain (1995)	4.86
Hollow Crown, The (2012)	4.85



From Model to Product:

Step 4 – Setting up the Survey

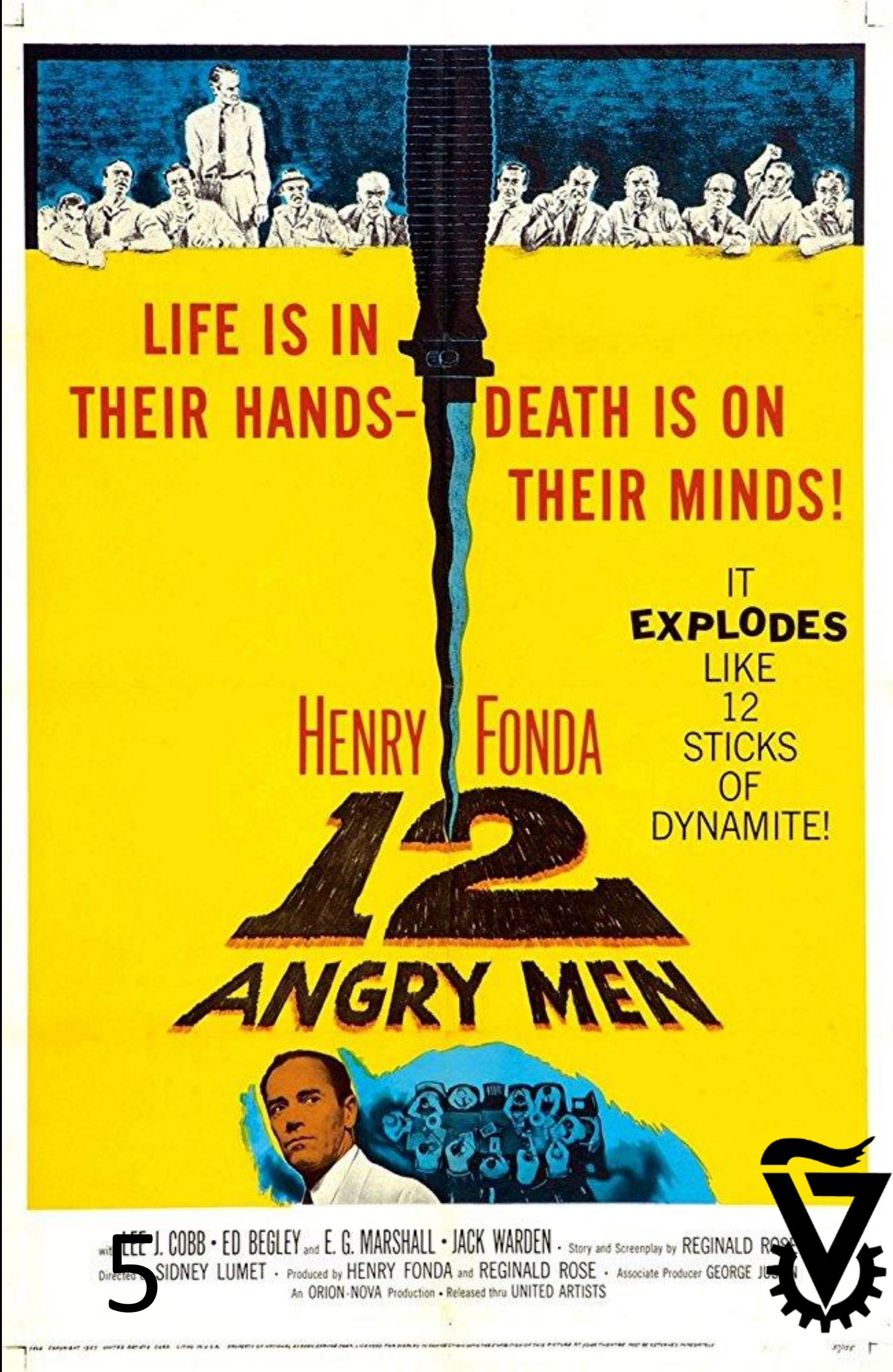
Movie List Criterions:

- Familiarity – N ratings > 100
- Variance – High STD
- Updated – Manually

Platform: Segmanta

Distribution:

- Social networks



From Model to Product

Step 5 – Analyze Survey

Movie	% Rate	Mean	STD
Fight Club (1999)	91%	0.3	1.5
Titanic (1997)	91%	-0.4	1.4
The Lion King (1994)	86%	0.3	1.0
Pirates of the Caribbean (2003)	86%	0.4	1.3
The Shawshank Redemption (1994)	82%	0.8	1.3
Star Wars: Episode I (1999)	82%	-0.3	1.4
Austin Powers (1997)	82%	0.3	0.9
Raiders of the Lost Ark (Indiana Jones) (1981)	82%	0.0	1.3
Jurassic Park (1993)	82%	-0.3	1.4
The Matrix (1999)	82%	0.5	1.3
Saving Private Ryan (1998)	82%	0.3	1.1
Lord of the Rings: The Return of the King (2003)	82%	0.3	1.5
Harry Potter and the Prisoner of Azkaban (2004)	82%	-0.0	1.5
Batman (1989)	77%	0.3	1.1
Men in Black (MIB) (1997)	77%	0.1	0.9
Mission: Impossible (1996)	77%	-0.0	1.0
X-Men (2000)	77%	-0.1	1.4
Spider-Man (2002)	77%	-0.1	1.5
Kill Bill: Vol. 2 (2004)	77%	0.5	1.3
The Incredibles (2004)	77%	-0.0	1.1
Back to the Future Part II (1989)	77%	0.3	1.1
The Dark Knight (2010)	77%	0.3	1.4
Iron Man (2008)	77%	-0.2	1.4
Borat (2006)	77%	0.5	1.1
Independence Day (a.k.a. ID4) (1996)	73%	-0.6	1.1

From Product to Glory

Step 6 – Getting prediction’s feedback

Trick to improve recommendations

- We got weird Soviet documentaries..
- Decided to recommend only popular movies (>100 ratings)
- Reduced the variety and hurts discovery a bit,
  but improves results substantially!

- Also done by the big guys!

So, who wants to get recommendations?

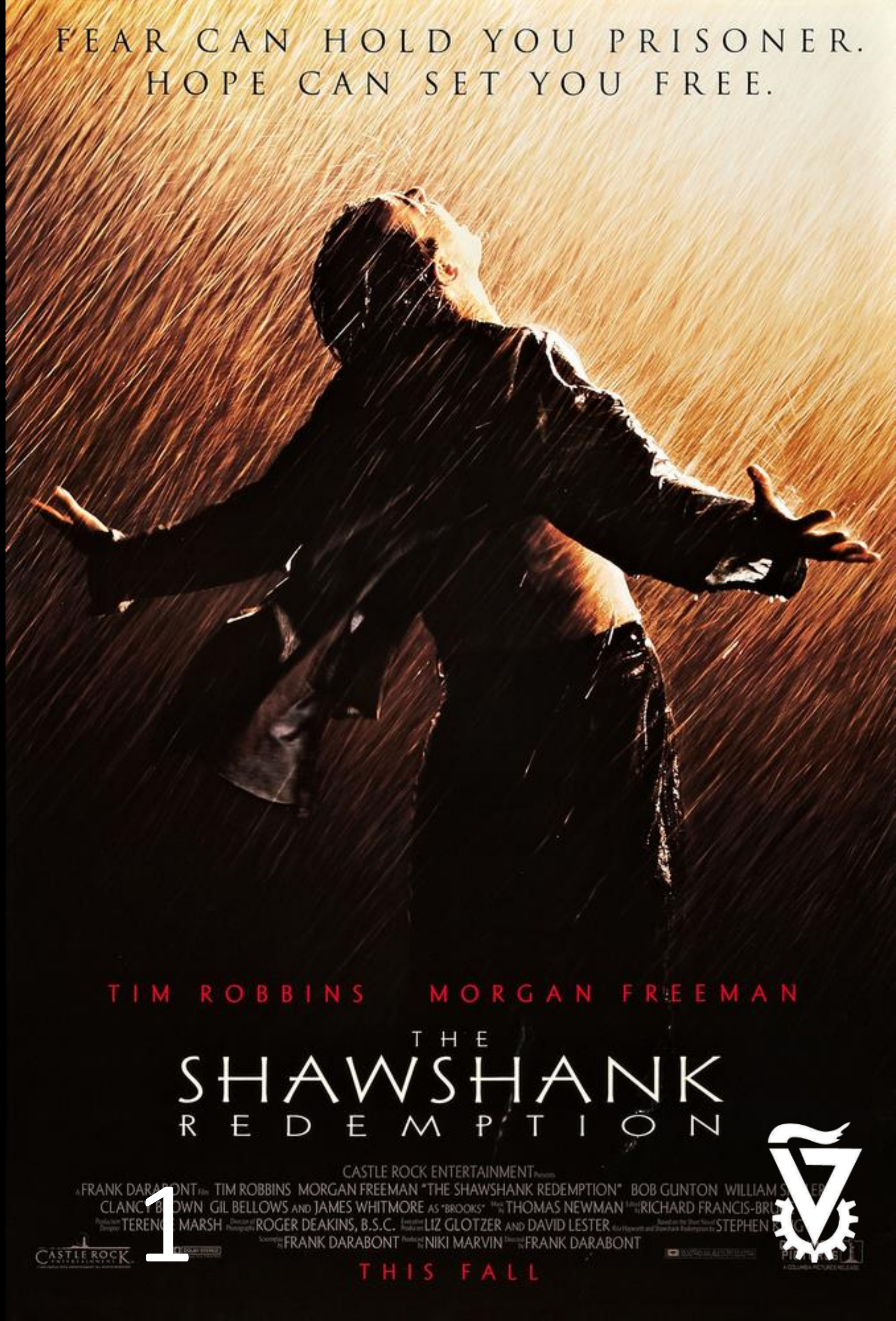
Some Insights:

- Talk to end-users & domain experts
- Find the right balance between Theory and Practice
- Inconsistency in ‘good predictions’ between users. Perhaps related to movies survey list.
- Make them an offer they can't refuse

Future work

- **Combine** Content-based with CF
- Predict **similar movies** by genre
- **Ensemble** recommendations from different models/hyperparameters
- Implement **Embeddings** as latent features, from tags/reviews

"Hope is a good thing, maybe the best of things, and no good thing ever dies." — Andy Dufresne



Thanks


he's just not that into you
TITANIC
PINK PANTHER
JOHNNY DEPP
IS
RANGO
NOTEBOOK
27 Dresses
Ab 14. Februar 2008. Nur im Kino!
JULIA ROBE
EAT PRAY LOVE
DEAR JOHN
CAMERON DIAZ
TONI COLLETTE
In Her SHOES
DodgeBall
KING KONG
AVATAR
GROWN UPS
THE SORCERER'S APPRENTICE
JULY 16
BIG MOMMAS
FINDING NEVERLAND
JOHNNY DEPP
ANGELINA JOLIE