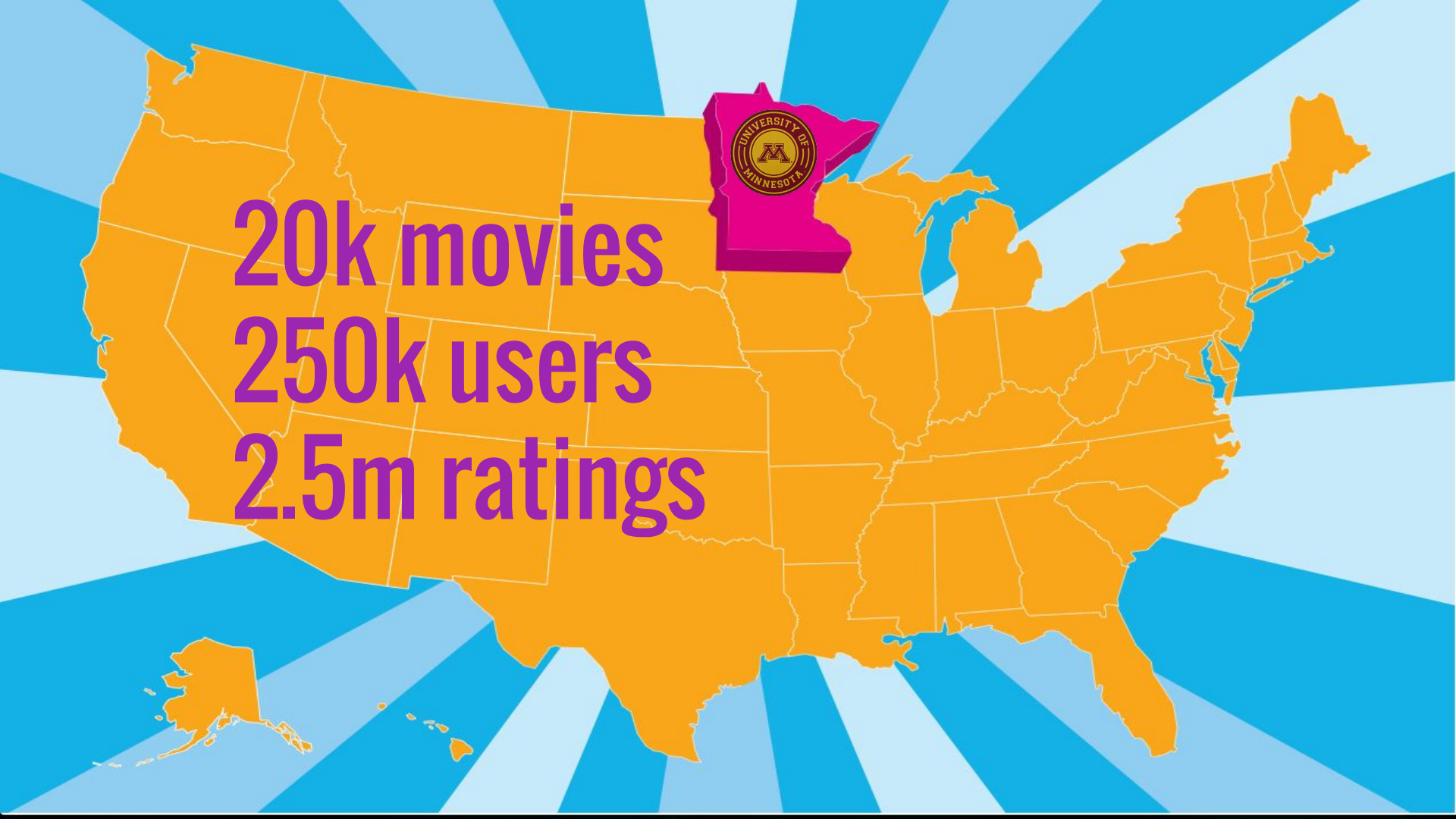
Movies!

Zhanna Rozenberg
May 13, 2016


20k movies
250k users
2.5m ratings
UNIVERSITY OF MINNESOTA

Performance Criteria

kaggle Host Sign up

Wiki (Beta) »

Root Mean Squared Error

Root Mean Squared Error (RMSE)

The square root of the mean/average of the square of all of the error.

The use of RMSE is very common and it makes an excellent general purpose error metric for numerical predictions.

Compared to the similar Mean Absolute Error, RMSE amplifies and severely punishes large errors.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$


BULLS
23
Who's good?

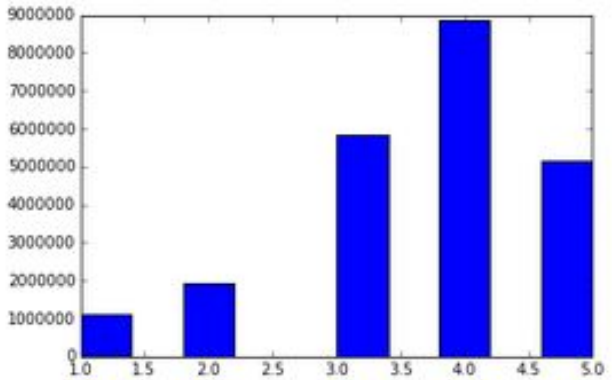
Spectrum of Benchmarks

Major League

Netflix Prize Winners

Little League

Choose Most Popular
(All 4's)



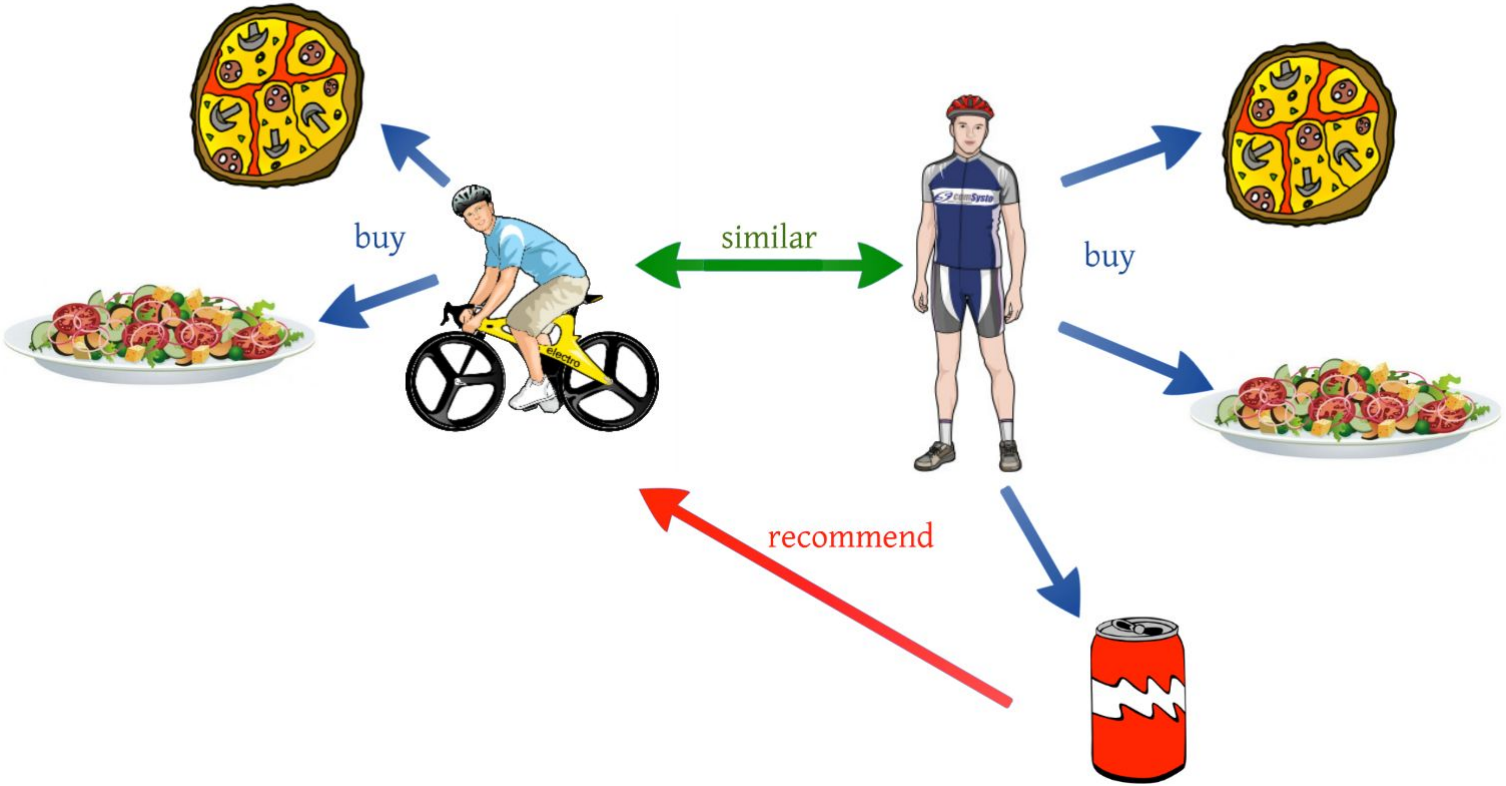
Spectrum of Benchmarks

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Netflix Prize Winners	0.87 (Grand Prize) 0.95 (Netflix Prior)
Choose Most Popular (All 4's)	3.35

Feature Engineering

Collaborative Filtering ….



My Results

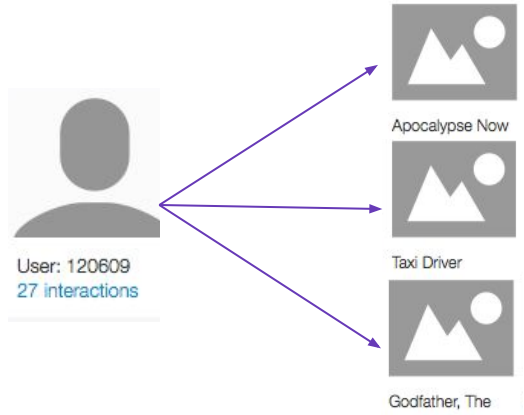
My RMSE:
0.876

Ranking	Team	RMSE
1	Bellkor	0.856
2	Gravity	0.873
3	Dinosaur Planet	0.875
4	ML@UToronto A	0.875
5	Arek Paterek	0.879
6	basho	0.879
7	NIPS Reject	0.881
8	Ensemble Experts	0.884
→	**Netflix Benchmark**	**0.9410**

Three Applications

Find similar users

Find similar movies



Make recommendations

Switch to demo...

Thanks

APPENDIX

Future Work

Better features like
Text analysis on plot
Image analysis across trailer or
movievisual sentiment analysis

Precision / Recall Overall

```
model_precision_recall["precision_recall_overall"]
```

cutoff	precision	recall
1	0.154340836013	0.0097045333192
2	0.139335476956	0.0176682629843
3	0.130046445159	0.0243574065097
4	0.128081457663	0.0322400572214
5	0.124330117899	0.0390843342264
6	0.11914969632	0.0448888778413
7	0.116521206553	0.0518017014863
8	0.113210075027	0.0581317660175
9	0.109324758842	0.0618955764172
10	0.106645230439	0.0657890466091

[18 rows x 3 columns]

Note: Only the head of the SFrame is printed.

You can use print_rows(num_rows=m, num_columns=n) to print more rows and columns.

Recommending for a given user….

```
model_with_rating_target.recommend([110],k=20).join(items[["movieId","title","genres"]], on="movieId").sort("rank")
```

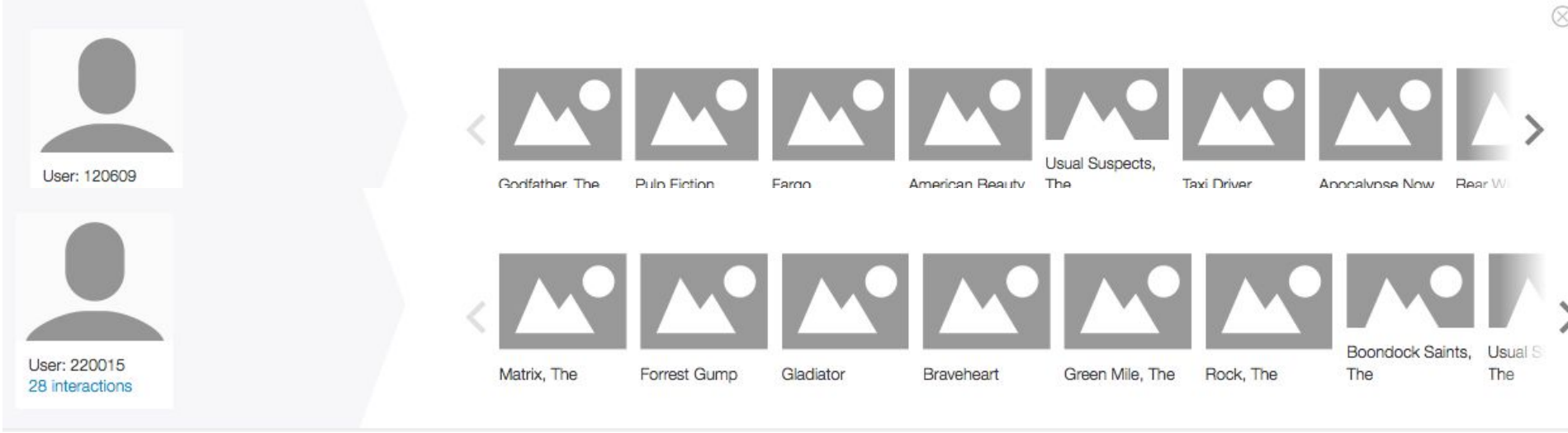
userId	movieId	score	rank	title	genres
110	50	4.78415803823	1	Usual Suspec	[Crime, Mystery, Thriller] ...
110	858	4.71632398877	2	Godfath	[Crime, Drama]
110	527	4.69358701471	3	Schindler	[Drama, War]
110	260	4.61135787729	4	Star Wars: Episode IV - A N ...	[Action, Adventure, Sci-Fi] ...
110	2571	4.60840608392	5	Matr	[Action, Sci-Fi, Thriller] ...
110	47	4.57308262054	6	Seven (a.k.a.	[Mystery, Thriller]
110	1198	4.55400207851	7	Raiders of the Lost Ark (Indiana Jones and the ...	[Action, Adventure]
110	1196	4.53955073867	8	Star Wars: Episode V - The Empire Strik ...	[Action, Adventure, Sci-Fi] ...
110	1221	4.53557963882	9	Godfather: Part	[Crime, Drama]
110	356	4.48818037126	10	Forre	[Comedy, Drama, Romance, War] ...

[20 rows x 6 columns]

Note: Only the head of the SFrame is printed.

You can use print_rows(num_rows=m, num_columns=n) to print more rows and columns.

Users & Their Recomendations



National Lampoon's Senior Trip

Jerky Boys, The

Bio-Dome

Halloween: The Curse of Michael Myers (Halloween 6: The Curse of Michael Myers)

Bulletproof

Black Sheep

High School High

Dracula: Dead and Loving It

Airhead

Sisterhood of the Traveling Pants, The

Shall We Dance?

Catch and Release

Sex and the City

Bridget Jones: The Edge of Reason

Just Like Heaven

Must Love Dogs

Music and Lyrics

Rumor

Find Similar Users...

```
model_with_rating_target.get_similar_users([330])
```

```
Getting similar users completed in 0.0143
```

userId	similar	score	rank
330	129962	0.966024935246	1
330	61401	0.964756786823	2
330	231529	0.963108241558	3
330	86971	0.960080862045	4
330	134967	0.957589387894	5
330	61514	0.956067562103	6
330	191169	0.955652832985	7
330	197800	0.953986406326	8
330	115595	0.953759908676	9
330	123468	0.953629910946	10

[10 rows x 4 columns]

		Best Score	% Improvement
	Grand Prize Barrier	**0.8563**	
1	BellKor	0.8728	8.26
2	Gravity	0.8750	8.03
3	Dinosaur Planet	0.8753	8.00
4	ML@UToronto A	0.8787	7.64
5	Arek Paterek	0.8789	7.62
6	basho	0.8805	7.45
7	NIPS Reject	0.8808	7.42
8	Ensemble Experts	0.8841	7.07
	2007 Progress Prize Barrier	**0.9419**	

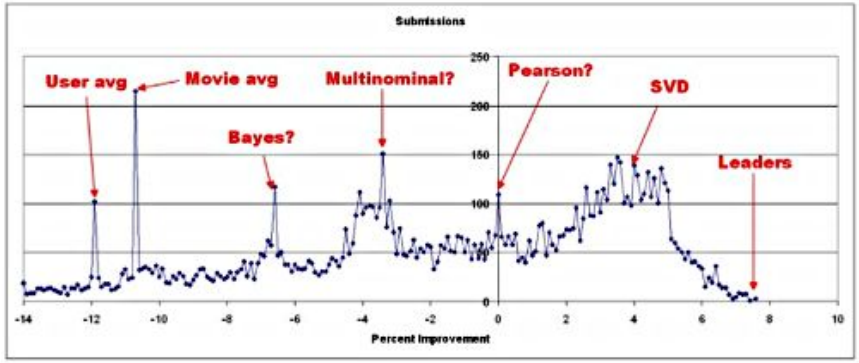
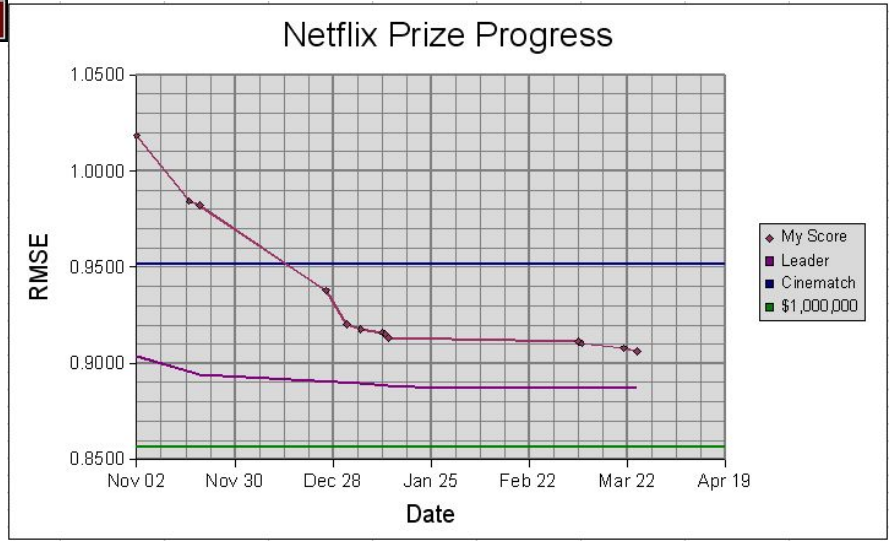


Figure 2: Detail of distribution of leading submissions indicating possible techniques



Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(2)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}, \theta^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

Collaborative Filtering