RECOMMENDER SYSTEM

Sachin K Manjhi

Feb 2020

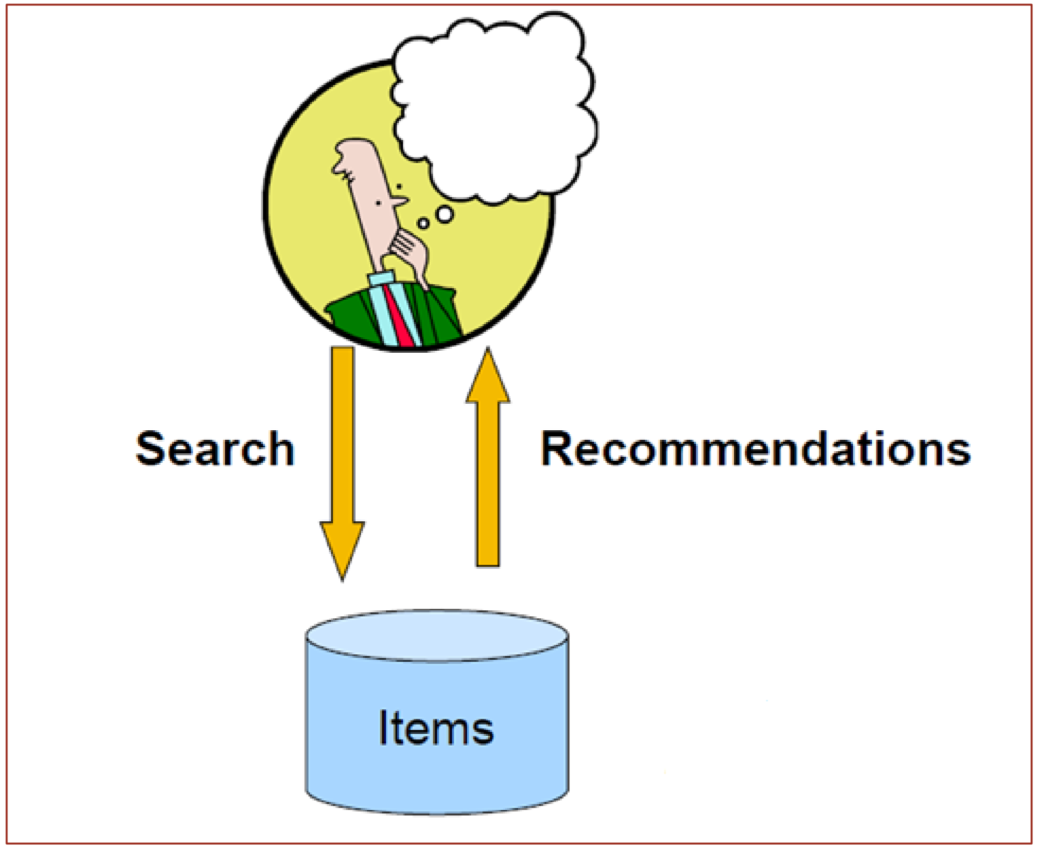
B R User Group

AGENDA

- Recommender Systems Definition
- Application of Recommender System
- Types of Recommender Systems
- Content Based RS
  - Mathematical Model
  - Pros & Cons
- Collaborative Filtering
  - Mathematical Model
  - Types of CF
  - Pros & Cons
- Latent Factor Method
- Implementation
- Netflix Competition – KDD 2007

RECOMMENDER SYSTEM – WHAT IS IT?

- A **Recommender System**(engine) : is a subclass of information filtering system , that seeks to predict the "rating" or "preference" a user would give to an item.



QUESTION 1

- What are some practical applications of Recommender System?

APPLICATION OF RECOMMENDER SYSTEM

Product Recommendation

Playlist Generation

Content Recommendation

Speciality Recommendation

? Recommendation

CORE OF RECOMMENDATION- DATA MINING

Data Preparation

- Gathering User and Item features
- Gathering Ratings
  - Explicit Vs Implicit
- Data preprocessing
  - outlier removal, denoising, removal of global effects

Data Mining

- Extrapolate unknown ratings from known ones
- Evaluate extrapolation method

Post Processing

- Don't recommend items already used/bought
- Expand the user's taste into neighboring areas by improving the obvious
- Showing the recommended items based on context

TYPES OF RECOMMENDER SYSTEM

1

Editorial and hand curated
- ✓ Editor's Choice
- ✓ List of favorites
- ✓ List of must-have

2

Simple aggregates
- ✓ Weekly Top 10
- ✓ Recently Trending
- ✓ All Time favorites

3

Customization

Traditional Methods

- **Content Based**
- **Collaborative Filtering**
- Clustering
- Association

Non-Traditional Methods

- **Factorization**
- Ranking
- Deep Learning (RBM)
- Reinforcement Learning
- Graph Models

CONTENT BASED RECOMMENDATION

- In *content-based* recommendations, the system tries to recommend items **similar** to those a given user has liked in the past, based solely on the profile built up by **analyzing the content** of items.

Mathematical Model:

- Define Item(s) Profile:
  - i = [w1, w2, w3 …. ,wn]
- User(s) Profile:
  - Weighted average of rated item profiles
  - **Variation:** weight by difference from average rating for item
- **Prediction heuristic:** Given user profile ***x*** and item profile ***i***, estimate cosine similarity:

  $u(\boldsymbol{x},\boldsymbol{i}) = \cos(\boldsymbol{x},\boldsymbol{i}) = \boldsymbol{x}\cdot\boldsymbol{i}/||\boldsymbol{x}||\cdot||\boldsymbol{i}||$

QUESTION 2

- What are some of the issues with Content based recommendation?

PROS AND CONS – CONTENT BASED RECOMMENDATION

- PROS:
  - No need for data on other users (cold start or sparsity problems)
  - Able to recommend to Users with unique tastes
  - Able to recommend new & unpopular items (first rater problem)
  - Able to provide explanations by listing content features

- CONS:
  - Requires content that can be encoded as meaningful features (difficult for movies, songs, webpages)
  - Users' tastes must be represented as a learnable function of these content features
  - Doesn't work for New user with no existing ratings
  - Overspecialization
    - Doesn't exploit quality judgements of other users
    - Easy to overfit (e.g. for a user with few data points we may "pigeon hole" her)
    - Difficult to implement serendipity

COLLABORATIVE FILTERING

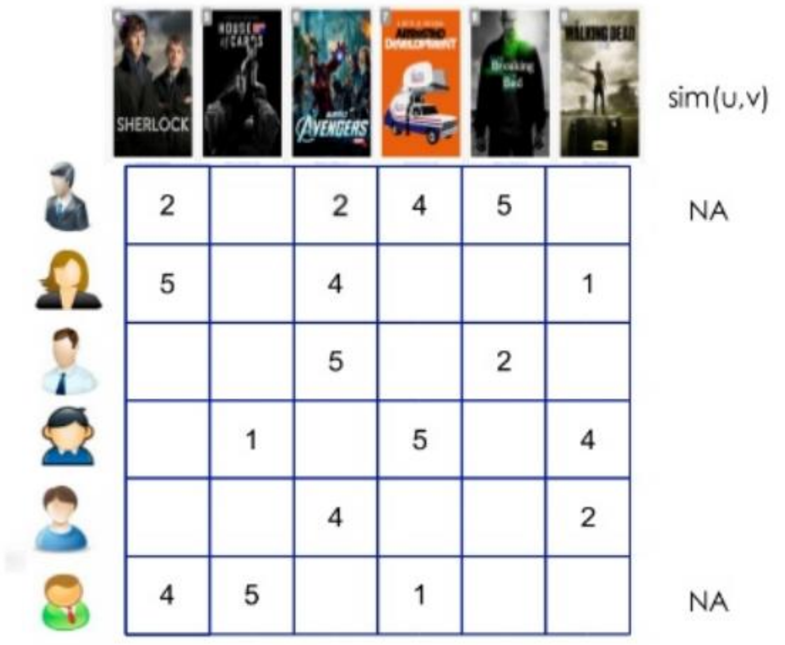
- **Collaborative Filtering:**
  - Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past.
  - The system generates recommendations using only information about rating profiles for different users or items

- **Mathematical Model:**
  - X = set of users/customers
  - S = set of items (products, movies, web pages)

  The goal is to define the mapping
  - **Utility function** $\boldsymbol{u} : \boldsymbol{X} \times \boldsymbol{S} \to \boldsymbol{R}$
  - $\boldsymbol{R}$ is a totally ordered set e.g.,**0 - 5** stars, real number in**[0,1]**

- **Types of Collaborative Filtering:**
  - User based Collaborative Filtering
  - Item based Collaborative Filtering

Utility Matrix


SHERLOCK
AVENGERS

						sim(u,v)
2		2	4	5		NA
5		4			1	
		5		2		
	1		5		4	
		4			2	
4	5		1			NA

(Huge, Sparse)

STEPS OF COLLABORATIVE FILTERING

- The basic steps:
  1. Identify set of ratings for the **target/active user** (Neighborhood creation)
  2. Identify set of users most similar to the target/active user
     1. Jaccard Similarity (Doesn't take rating into consideration)
     2. Cosine Similarity
     3. Pearson Correlation
  3. Identify the products these similar users liked
  4. **Generate a prediction** - rating that would be given by the target user to the product - for each one of these products (Average; Weighted Average)
  5. Based on this predicted rating recommend a set of top N products

QUESTION 3

- Which method works better in general?
  - User Based Collaborative Filtering
  - Item Based Collaborative Filtering

- Answer: Item Based
  - Items are “Simpler”
  - Item similarity is more meaningful that User similarity.

PROS AND CONS – COLLABORATIVE FILTERING

- PROS:
  - Requires **no feature extraction of items**
    - Users and products are symbols without any internal structure or characteristics
  - Produces good-enough results in most cases

- CONS:
  - Requires a large number of **reliable** “user feedback data points” to bootstrap (Cold start)
  - Assumes that **prior behavior determines current behavior** without taking into account “context”
  - Requires products to be **standardized** (users should have bought exactly the same product)
  - Cannot recommend an item that has not been previously rated (First rater)
  - Tends to recommend popular items more than less popular items (popularity bias)
  - Highly time consuming with millions of users and items in the database.

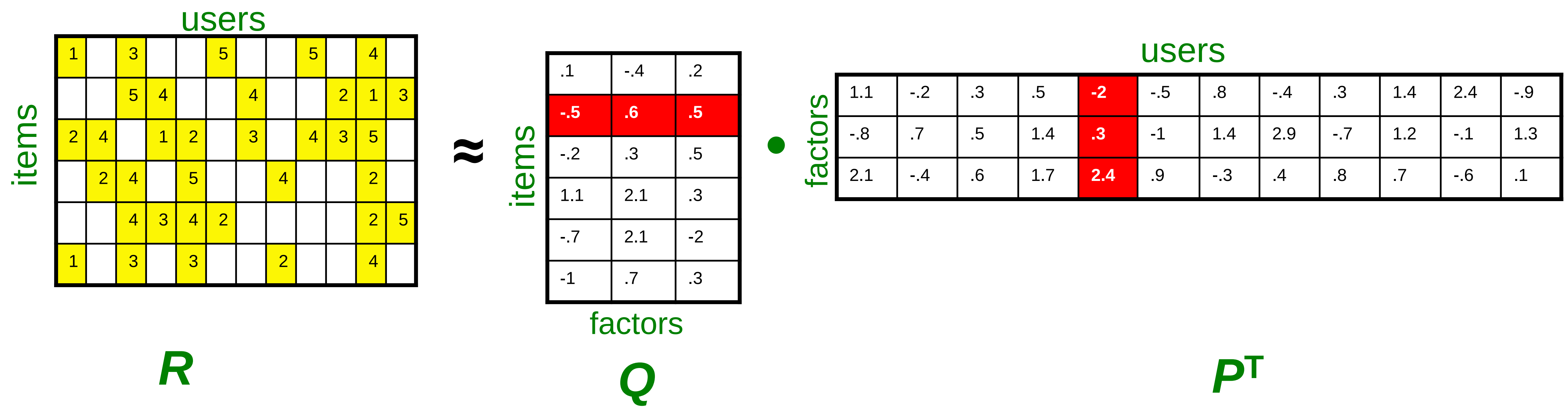
QUESTION 4

- Which method will lead to poor accuracy if we store pre-computed similarity measure?
  - Item based
  - User based

- Ans: User based CF:
  - User-based CF – similarity between users is dynamic, pre-computing user neighborhood can lead to poor predictions.
  - Item-based CF – similarity between items is static.

LATENT FACTOR METHOD

- Rating as Product of Factors:
  - Assume we can approximate the rating matrix ***R*** as a product of "thin" matrices $\boldsymbol{Q} \cdot \boldsymbol{P}^{\mathrm{T}}$



R

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

Q

.1	-.4	.2
-.5	**.6**	**.5**
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

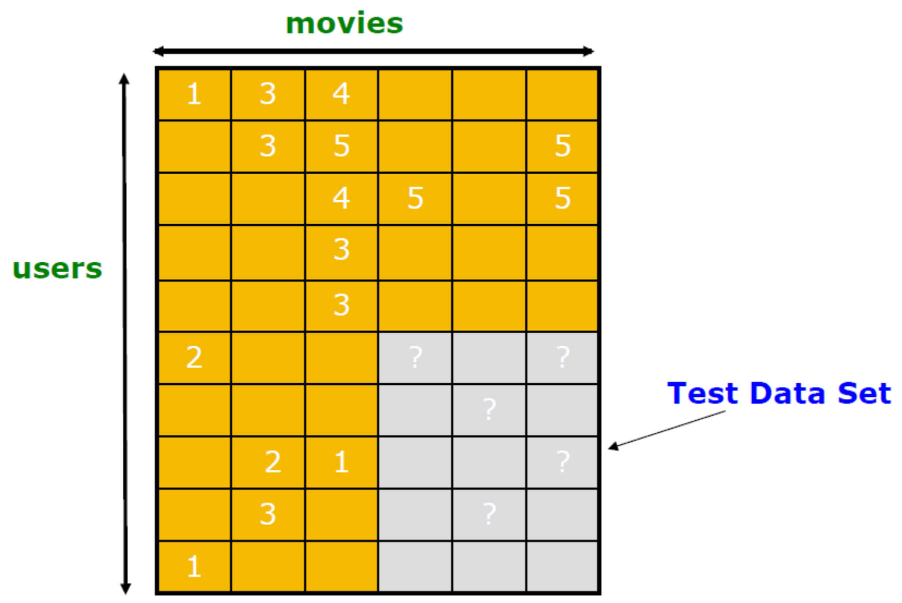
$\boldsymbol{P}^{\mathrm{T}}$

1.1	-.2	.3	.5	**-2**	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	**.3**	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	**2.4**	.9	-.3	.4	.8	.7	-.6	.1

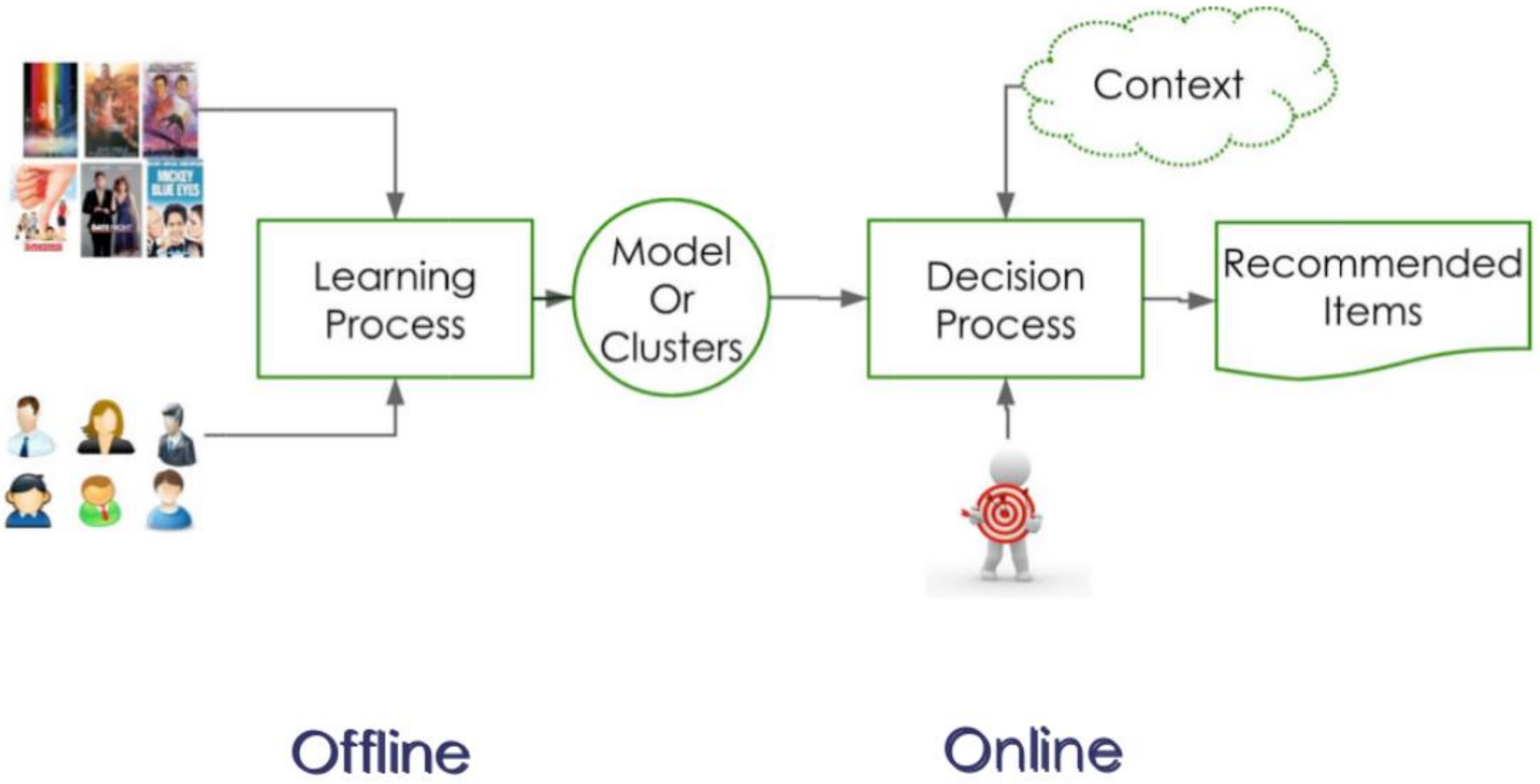
- ***R*** has missing entries but let's ignore that for now! Basically, we will want the reconstruction error to be small on known ratings and we don't care about the values on the missing ones
- Use Regularization and Gradient Descent to find the factors
- Addition of "User Bias" and "Movie Bias" can further improve performance

PERFORMANCE EVALUATION

- **Compare predictions with known ratings**
  - **Root-mean-square error** (RMSE)
    - $\sqrt{\sum_{xi}(r_{xi}-r_{xi}^{*})^2}$ where r_{xi} is predicted, r_{xi}^{*} is the true rating of x on i
  - **Precision at top 10**:
    - % of those in top 10
  - **Rank Correlation**:
    - Spearman's *correlation* between system's and user's complete rankings
- **Another approach: 0/1 model**
  - **Coverage:**
    - Number of items/users for which system can make predictions
  - **Precision:**
    - Accuracy of predictions
  - **Receiver operating characteristic** (ROC)
    - Tradeoff curve between false positives and false negatives



IMPLEMENTATION

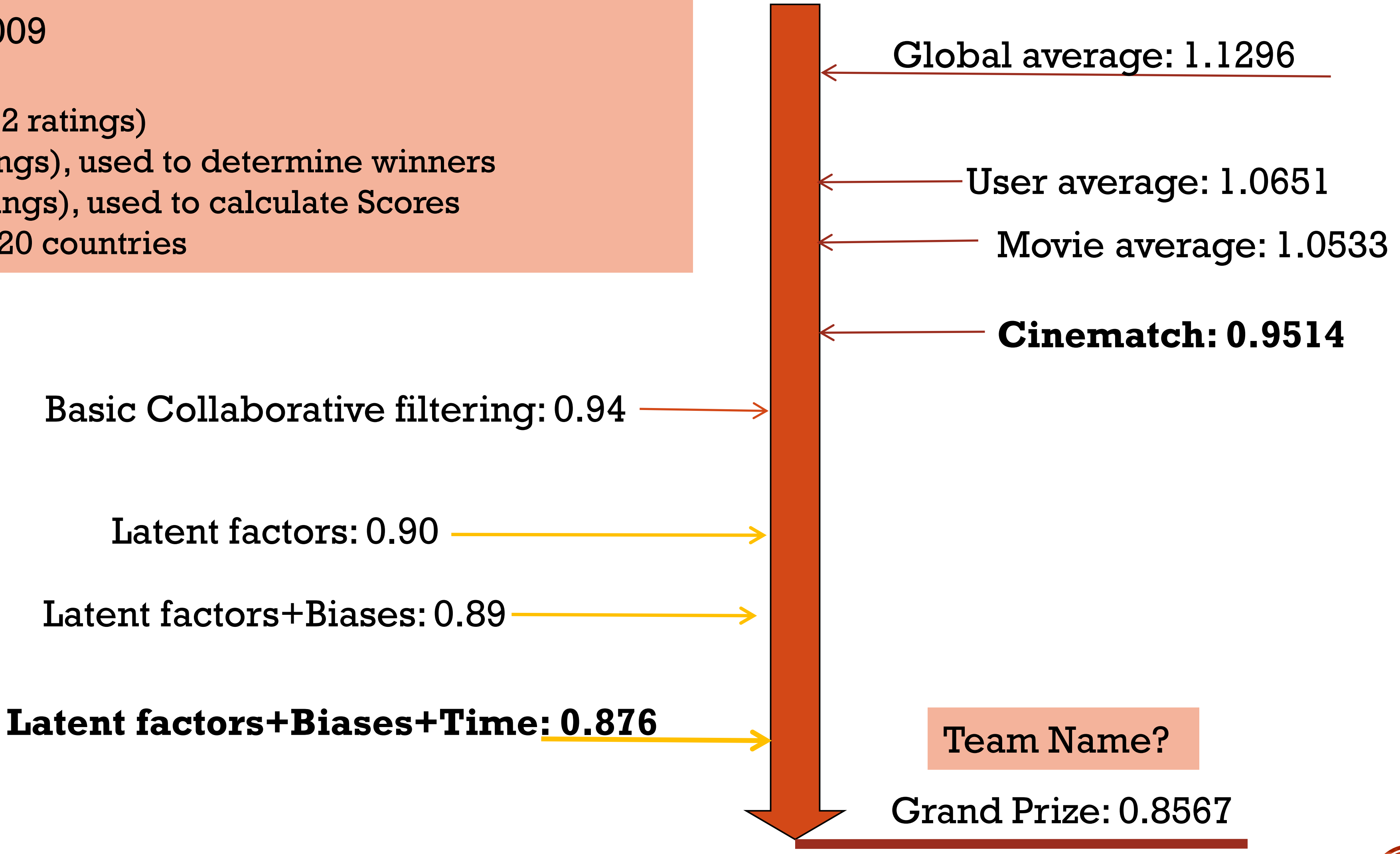


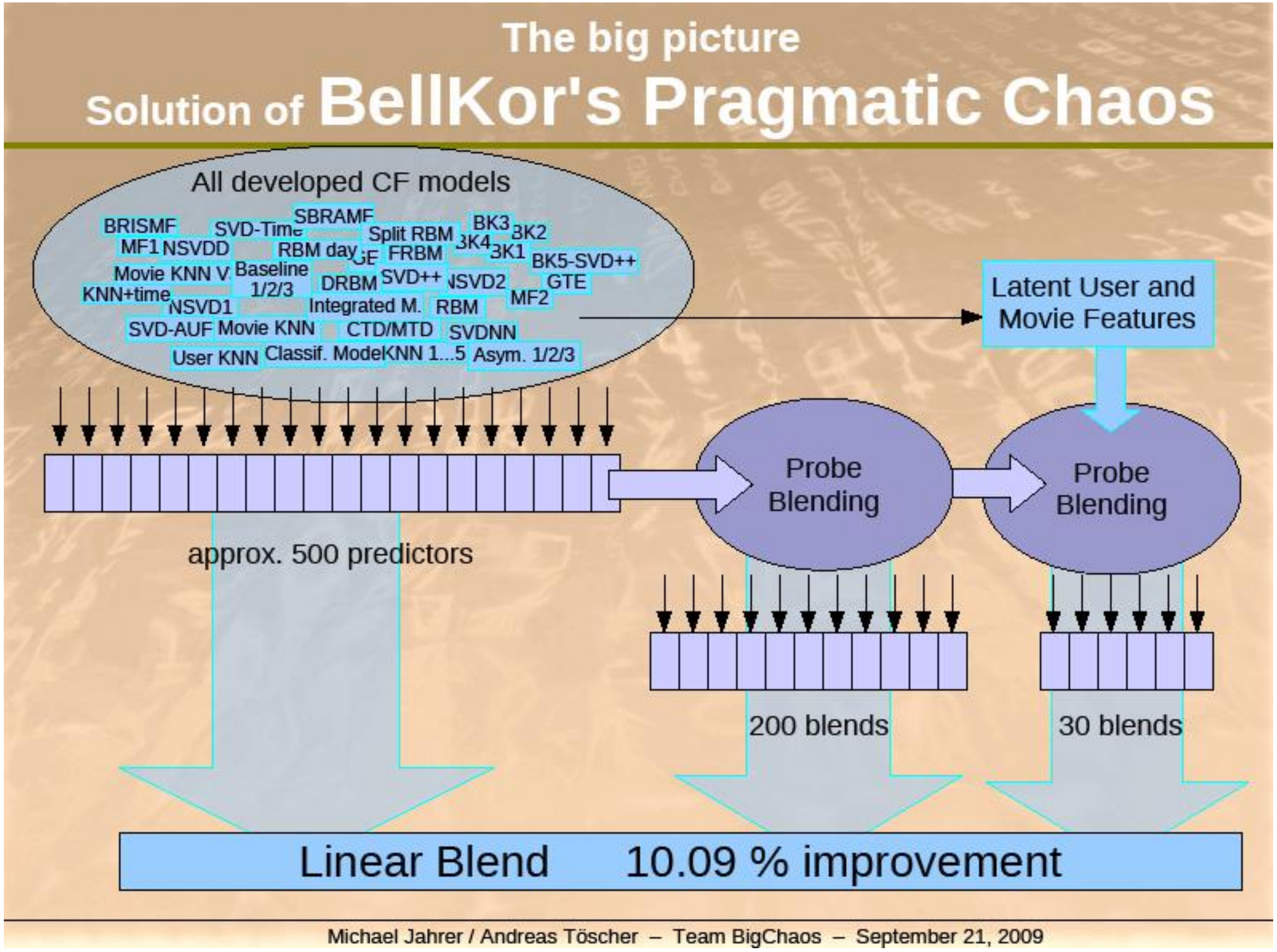
Source: Xavier Amatriain – July 2014 – Recommender Systems

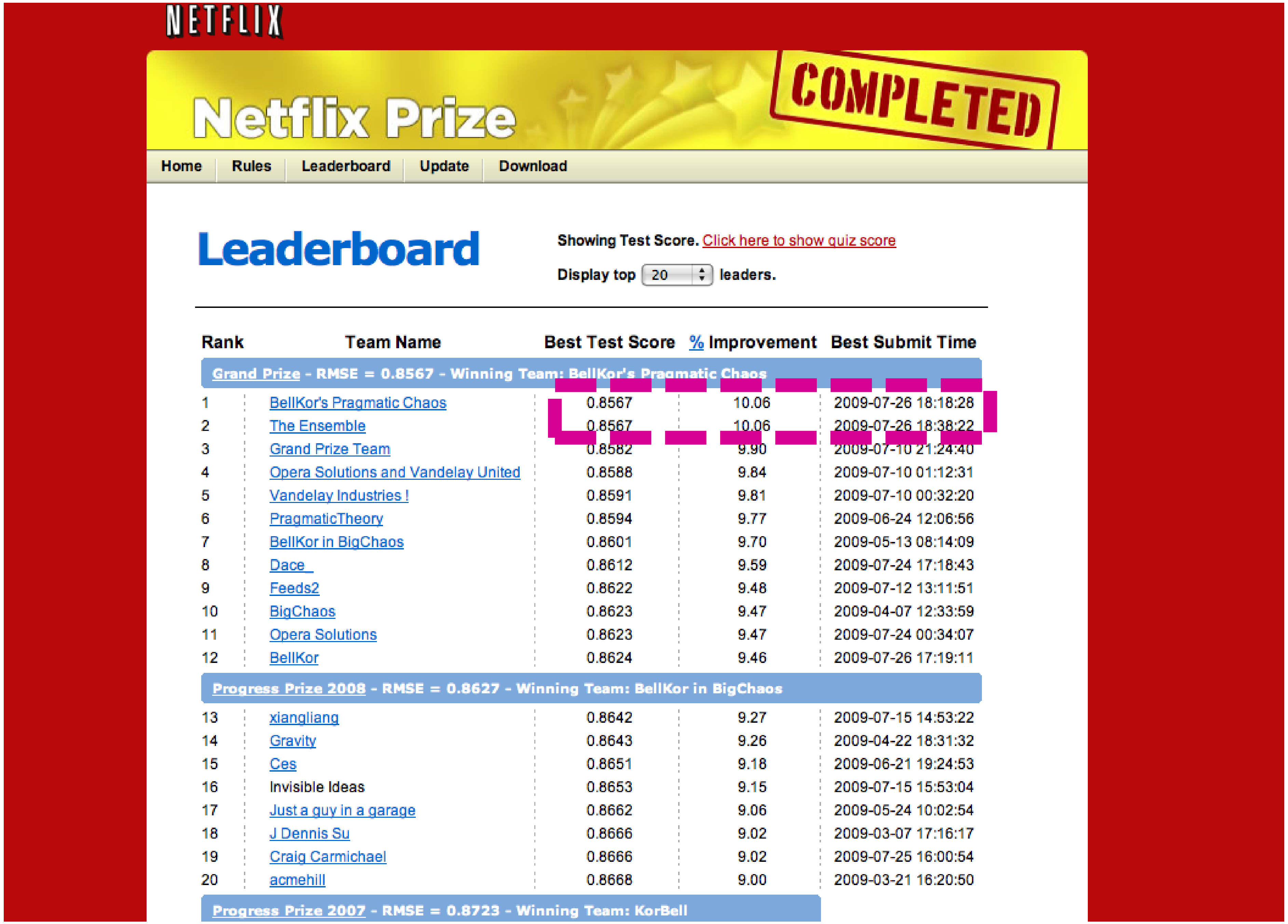
NETFLIX COMPETITION

About the Competition: $1 Million in Prize

- Time : Oct ,2006 – July 2009
- Data:
    - Training set (99,072,112 ratings)
    - Test set (1,408,789 ratings), used to determine winners
    - Quiz set (1,408,342 ratings), used to calculate Scores
- Teams: 20,000 teams from 120 countries




The big picture
Solution of BellKor's Pragmatic Chaos
All developed CF models
BRISMF
SVD-Time
SBRAMF
Split RBM
BK3
BK2
MF1
NSVDD
RBM day
GE
FRBM
BK4
BK1
BK5-SVD++
Movie KNN V.
Baseline 1/2/3
DRBM
SVD++
NSVD2
GTE
KNN+time
NSVD1
Integrated M.
RBM
MF2
SVD-AUF
Movie KNN
CTD/MTD
SVDNN
User KNN
Classif. Mode
KNN 1...5
Asym. 1/2/3
Latent User and Movie Features
approx. 500 predictors
Probe Blending
Probe Blending
200 blends
30 blends
Linear Blend
10.09 % improvement
Michael Jahrer / Andreas Töscher – Team BigChaos – September 21, 2009



Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22
14	Gravity	0.8643	9.26	2009-04-22 18:31:32
15	Ces	0.8651	9.18	2009-06-21 19:24:53
16	Invisible Ideas	0.8653	9.15	2009-07-15 15:53:04
17	Just a guy in a garage	0.8662	9.06	2009-05-24 10:02:54
18	J Dennis Su	0.8666	9.02	2009-03-07 17:16:17
19	Craig Carmichael	0.8666	9.02	2009-07-25 16:00:54
20	acmehill	0.8668	9.00	2009-03-21 16:20:50
Progress Prize 2007 - RMSE = 0.8723 - Winning Team: KorBell				

REFERENCES

- Mining Massive Datasets - J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org
- Recommender Systems - Collaborative Filtering and other approaches by *Xavier Amatriain, Research/Engineering Director @ Netflix*
- Recommenderlab R Package - https://cran.r-project.org/web/packages/recommenderlab/recommenderlab.pdf
- Google Crash Course on Recommendation: https://developers.google.com/machine-learning/recommendation
- Coursera: https://www.coursera.org/learn/recommendation-models-gcp
- Netflix Competition:
  - https://www.netflixprize.com/assets/ProgressPrize2007_KorBell.pdf
  - https://en.wikipedia.org/wiki/Netflix_Prize