Tuning Baseline Singular Value Decomposition and comparing benchmark algorithms

Lakshmi Harika G (lg22k@fsu.edu)
Arambhumi Reddy (ar22br@fsu.edu)
Nov 29th 2023

Outline

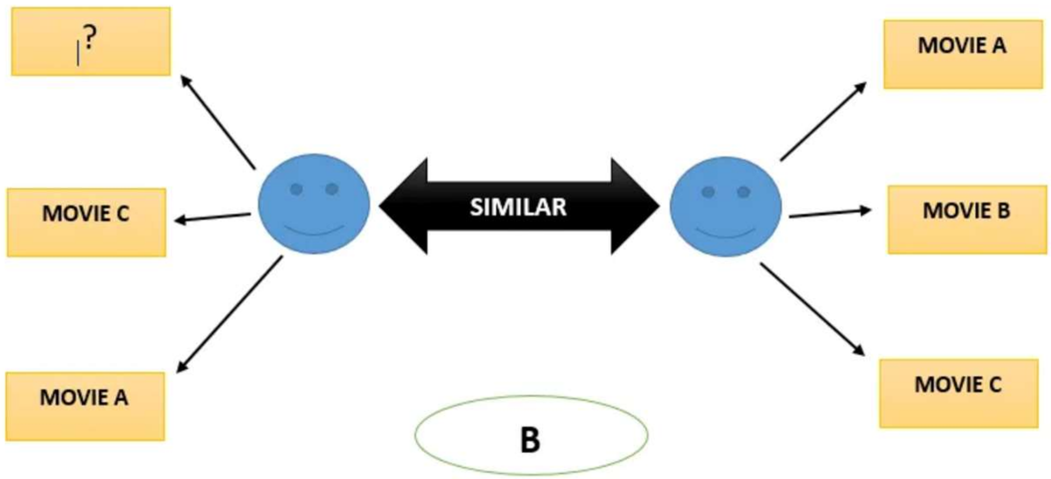
- Introduction
- Related Work
- Data Analysis
- Model Design
- Evaluation
- Discussion
- Conclusion

Introduction

Amazon Product recommender System

Simple User-based Collaborative Filtering

- Create User-item interaction matrix
- Find similar users based on Pearson correlation and cosine similarity
- Predict ratings of unseen items based on similar user's ratings
- Make recommendations based on higher rating predictions for items.



Matrix Factorization techniques

- But due to the curse of dimensionality, the user-item interaction matrix can be difficult to handle and measure similarity for every user and item.
- To reduce the dimensions of the matrix, we use a model-based approach called Matrix Factorization.
- Assumes that every user and item is associated with latent factors.

$$R \approx P \times Q^T$$

User	God father	Termin-ator	Money game	Titanic	Back to the future	...	X-men
Alice	5	1	4	4	3	...	2

User	Horror	Sci-Fi	Humanity	Drama	...
Alice	0.3	2.1	6.1	4.7	...

User	Horror	Sci-Fi	Humanity	Drama	...
God father	2.3	0.4	4.9	5.7	...

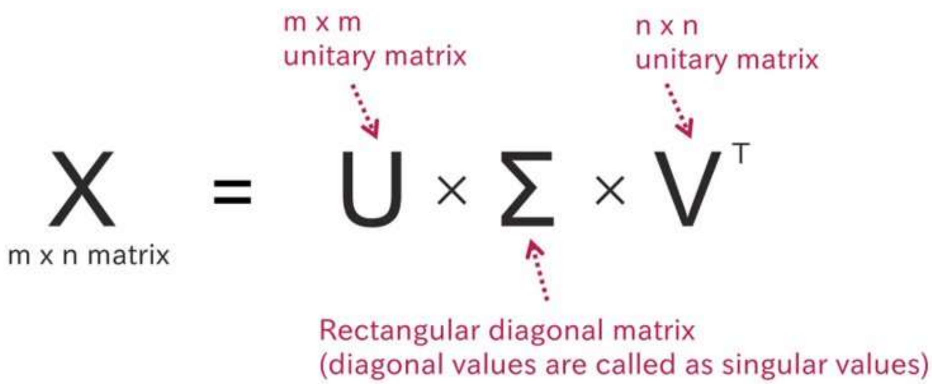
- The rating matrix is decomposed from the latent factors and has a lower dimension than the matrix derived from number of users and items.

Matrix Factorization technique :- SVD

- Latent User-item matrix using SVD:

$$\mathbf{X = U \Sigma V^T}$$

- We can approximate a given matrix by focusing on the largest singular values and the vectors of U and V that correspond to these values.



U – User latent matrix
V – Item latent matrix
Σ - diagonal matrix with singular values

SVD

- For Example

$$X = U \times \Sigma \times V^{T}$$

$$\begin{bmatrix} 1 & 3 & 3 & 3 & 0 \\ 2 & 4 & 2 & 2 & 4 \\ 1 & 3 & 3 & 5 & 1 \\ 4 & 5 & 2 & 3 & 3 \\ 1 & 1 & 5 & 2 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} -0.369 & -0.325 & 0.282 & 0.343 & 0.749 \\ -0.459 & 0.448 & -0.339 & -0.584 & 0.364 \\ -0.468 & -0.359 & 0.544 & -0.459 & -0.381 \\ -0.563 & 0.491 & 0.07 & 0.562 & -0.348 \\ -0.341 & -0.569 & -0.71 & 0.118 & -0.202 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 13.368 & 0 & 0 & 0 & 0 \\ 0 & 4.708 & 0 & 0 & 0 \\ 0 & 0 & 2.792 & 0 & 0 \\ 0 & 0 & 0 & 1.586 & 0 \\ 0 & 0 & 0 & 0 & 0.904 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.325 & 0.341 & -0.101 & 0.682 & -0.55 \\ -0.562 & 0.345 & 0.273 & 0.153 & 0.684 \\ -0.468 & -0.642 & -0.577 & 0.125 & 0.141 \\ -0.504 & -0.327 & 0.601 & -0.324 & -0.416 \\ -0.324 & 0.496 & -0.47 & -0.626 & -0.189 \end{bmatrix}$$

SVD

- We can approximate a given matrix by focusing on the largest singular values and the vectors of U and V which corresponds to the singular values.

$$U=\begin{bmatrix} -0.369 & -0.325 & 0.282 & 0.343 & 0.749 \\ -0.459 & 0.448 & -0.339 & -0.584 & 0.364 \\ -0.468 & -0.359 & 0.544 & -0.459 & -0.381 \\ -0.563 & 0.491 & 0.07 & 0.562 & -0.348 \\ -0.341 & -0.569 & -0.71 & 0.118 & -0.202 \end{bmatrix}$$

$$\Sigma=\begin{bmatrix} 13.368 & 0 & 0 & 0 & 0 \\ 0 & 4.708 & 0 & 0 & 0 \\ 0 & 0 & 2.792 & 0 & 0 \\ & & & 1.586 & 0 \\ 0 & 0 & 0 & 0 & 0.904 \end{bmatrix}$$

Largest singular values

$$V=\begin{bmatrix} -0.325 & 0.341 & -0.101 & 0.682 & -0.55 \\ -0.562 & 0.345 & 0.273 & 0.153 & 0.684 \\ -0.468 & -0.642 & -0.577 & 0.125 & 0.141 \\ -0.504 & -0.327 & 0.601 & -0.324 & -0.416 \\ -0.324 & 0.496 & -0.47 & -0.626 & -0.189 \end{bmatrix}$$

SVD

$$U_2 = \begin{bmatrix} -0.369 & -0.325 \\ -0.459 & 0.448 \\ -0.468 & -0.359 \\ -0.563 & 0.491 \\ -0.341 & -0.569 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 13.368 & 0 \\ 0 & 4.708 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} -0.325 & 0.341 \\ -0.562 & 0.345 \\ -0.468 & -0.642 \\ -0.504 & -0.327 \\ -0.324 & 0.496 \end{bmatrix}$$

$$U_2 \times \Sigma_2 \times V_2^T$$

$$= \begin{bmatrix} 1.08 & 2.24 & 3.29 & 2.98 & 0.84 \\ 2.72 & 4.17 & 1.52 & 2.41 & 3.04 \\ 1.46 & 2.93 & 4.02 & 3.71 & 1.19 \\ 3.24 & 5.03 & 2.04 & 3.04 & 3.59 \\ 0.57 & 1.64 & 3.86 & 3.18 & 0.15 \end{bmatrix} \approx \begin{bmatrix} 1 & 3 & 3 & 3 & 0 \\ 2 & 4 & 2 & 2 & 4 \\ 1 & 3 & 3 & 5 & 1 \\ 4 & 5 & 2 & 3 & 3 \\ 1 & 1 & 5 & 2 & 1 \end{bmatrix} = X$$

Related Work

Improving regularized singular value decomposition for collaborative filtering. [paper]

[Arkadiusz Paterek Institute of Informatics, Warsaw University]

Improving Default SVD by adding biases to the baseline algorithm and combining predictors.

RSVD - y^ij = uiT * vj ;

RSVD2 - y^ij = ci + dj + uiT * vj

Predictor	Test RMSE with BASIC	Test RMSE with BASIC and RSVD2	Cumulative test RMSE
BASIC	.9826	.9039	.9826
RSVD	.9094	.9018	.9094
RSVD2	.9039	.9039	.9018
KMEANS	.9410	.9029	.9010
SVD_KNN	.9525	.9013	.8988
SVD_KRR	.9006	.8959	.8933
LM	.9506	.8995	.8902
NSVD1	.9312	.8986	.8887
NSVD2	.9590	.9032	.8879
SVD_KRR * NSVD1	—	—	.8879
SVD_KRR * NSVD2	—	—	.8877

Table 1: Linear regression results - RMSE on the test set

Related Work

- Incremental singular value decomposition algorithms for highly scalable recommender systems. paper

[Badrul Sarwar, GeorgeKarypis, Joseph Konstan and John Riedl, GroupLens Research Group, University of Minnesota]

- To scale SVD for large datasets, the model is trained on a sample subset, followed by incrementing the samples using k-folding technique and building full trainsets.

Dataset

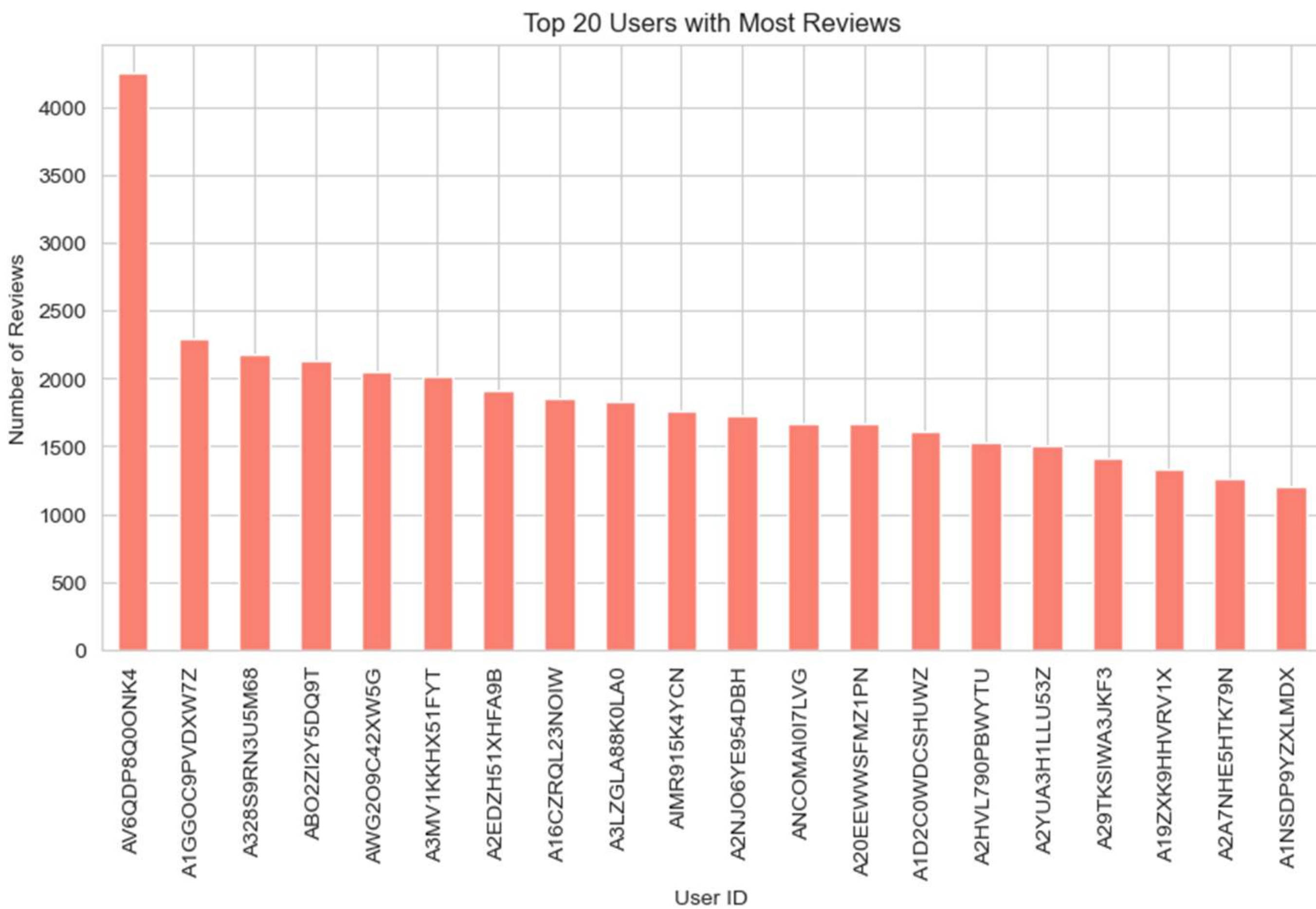
Amazon Product recommender System

Dataset used – Movies and TV ratings

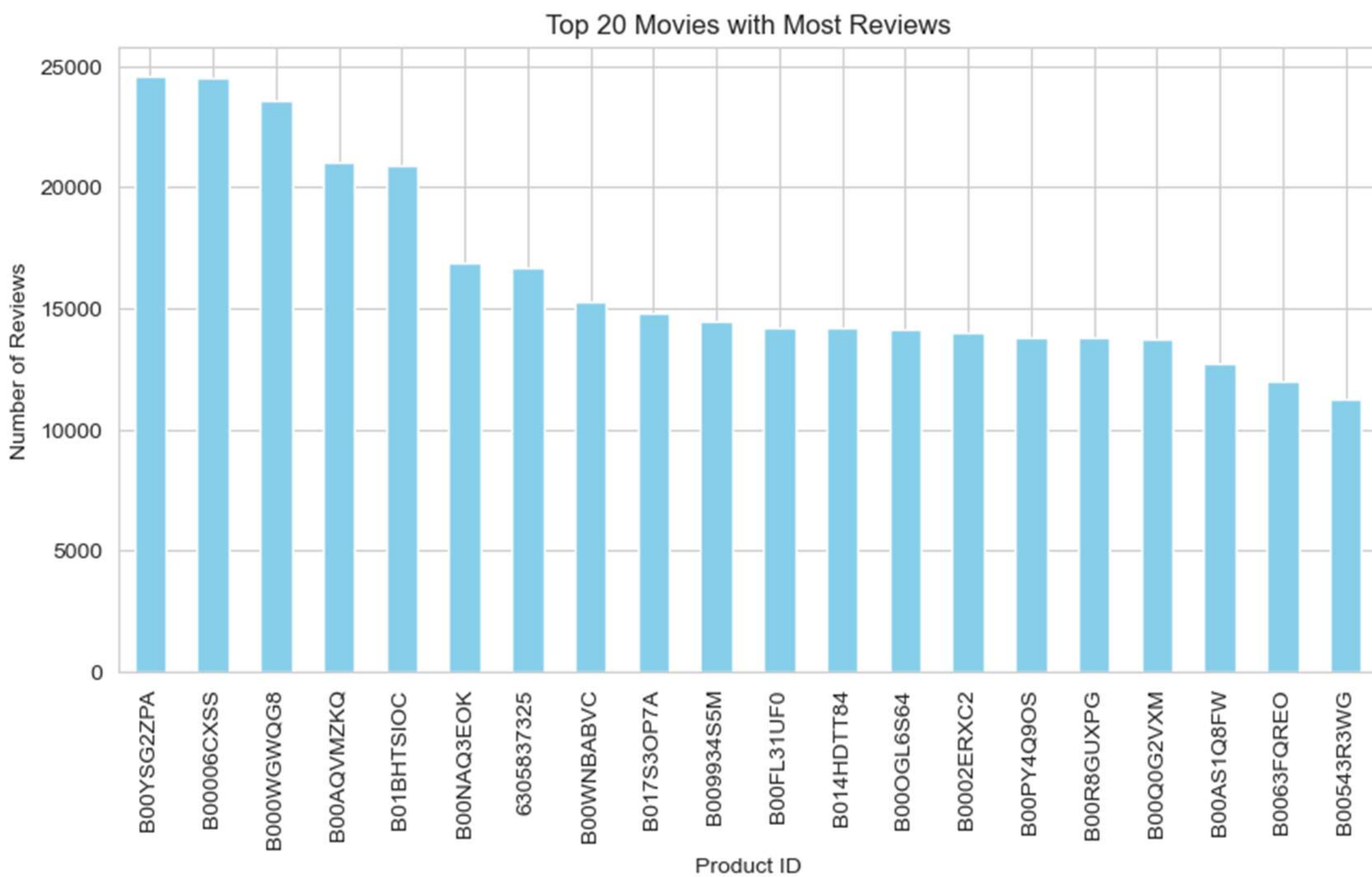
- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714
- rating – overall rating of the product
- Title – Title of the Movie or TV show

	asin	reviewerID	rating	title
0	0001527665	A3478QRKQDOPQ2	5.0	Peace Child VHS
1	0001527665	A2VHSG6TZHU1OB	5.0	Peace Child VHS
2	0001527665	A23EJWOW1TLENE	5.0	Peace Child VHS
3	0001527665	A1KM9FNEJ8Q171	5.0	Peace Child VHS
4	0001527665	A38LY2SSHVHRYB	4.0	Peace Child VHS
5	0001527665	AHTYUW2H1276L	5.0	Peace Child VHS

Exploratory Data Analysis



Exploratory Data Analysis



Model Selection

We have tried to improve the performance of the BaselineSVD by incorporating various algorithms like KNNBaseline, SVDpp, KNNwithMeans, KNNBasic etc.,

We then performed 3-fold cross-validation of the above algorithms and calculated the RMSE

The list is then sorted in ascending order of RMSE to choose the best features and select the optimal algorithm.

	test_rmse	fit_time	test_time
Algorithm			
BaselineOnly	1.022461	0.091306	0.057847
SVD	1.023334	0.487221	0.067836
SVDpp	1.025483	33.117089	0.881494
KNNBaseline	1.031946	0.090931	0.076367
KNNWithMeans	1.084568	0.014641	0.082035
SlopeOne	1.087040	8.726848	0.363178
KNNBasic	1.090521	0.007004	0.077622
CoClustering	1.091252	2.511543	0.054965
NMF	1.100628	1.620182	0.066588
NormalPredictor	1.373919	0.034633	0.087059

Hyperparameter Tuning

We have tried to improve the performance of the SVD algorithm by adding features and tuning the hyperparameters as:

```
SVD(n_factors=100, n_epochs=20, lr_all=0.007, reg_all=0.04, ran
dom_state=10)
```

We noticed that the tuned predictions obtained a slightly lower RMSE, hence proceeded to choose tuned predictions to provide recommendations for users.

	Default SVD	Tuned SVD
RMSE	1.0042	0.9956
MAE	0.9336	0.7070

Model Evaluation

We have tested the predictions on a sample subset of 10000 rows, in a 5-folding splits.

Higher Precision on Full Dataset Predictions:
Possible Reasons:

- The model might be making more accurate predictions on the full dataset, leading to a higher precision. This could be due to a larger amount of training data, allowing the model to learn more robust patterns.
- The recommendations on the full dataset might align better with user preferences, leading to fewer false positives.

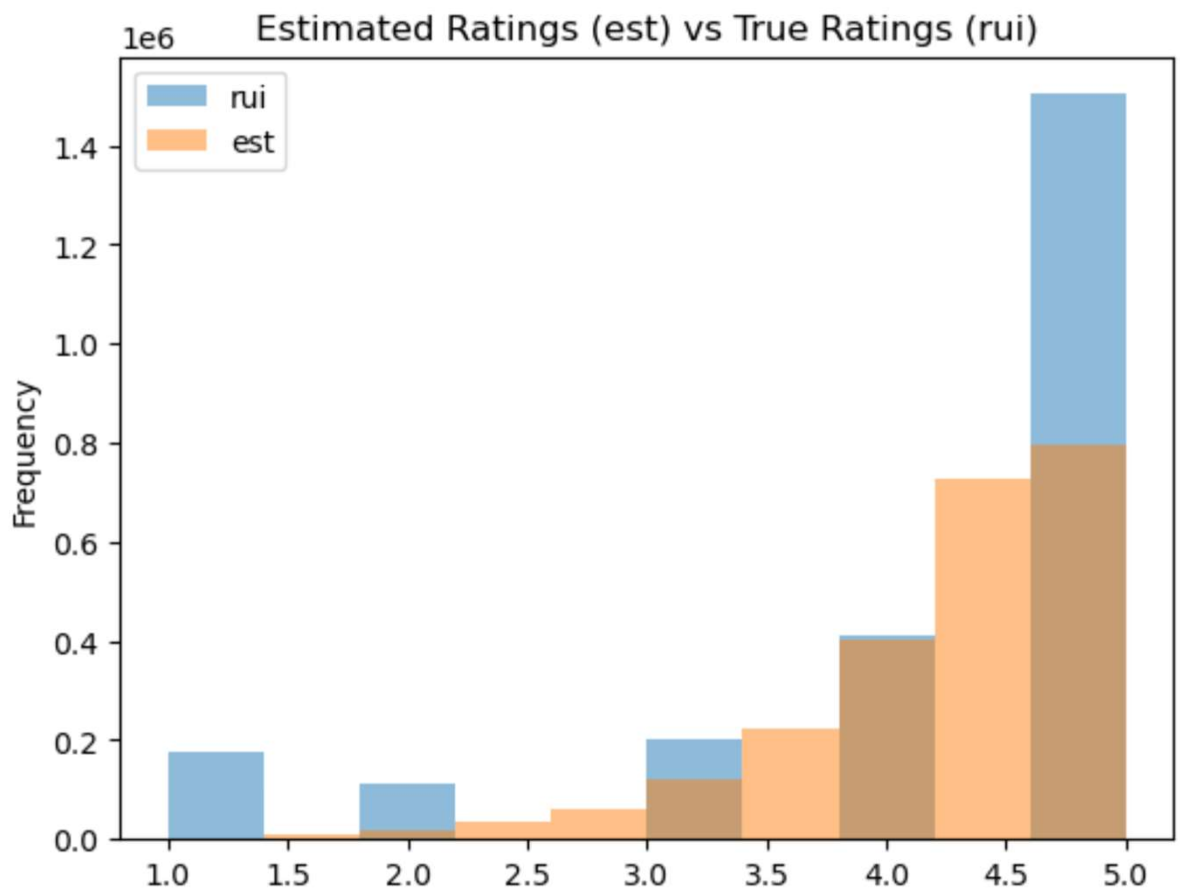
Higher Recall on Sample Subset Predictions:
Possible Reasons:

- The sample subset might contain a more diverse set of user preferences or cover specific user segments where the model performs well.
- The smaller size of the sample might introduce variability, affecting precision, but leading to higher recall.

```
Full Dataset

KFold Test #1:
Precision: 0.9223250937668417
Recall: 0.8641948160156651

KFold Test #2:
Precision: 0.9223464618392903
Recall: 0.8641980112520732

KFold Test #3:
Precision: 0.9224052025135365
Recall: 0.8638809873359804

KFold Test #4:
Precision: 0.9223479080416341
Recall: 0.8638508874119581

KFold Test #5:
Precision: 0.9225106550554031
Recall: 0.864692299410979
```

```
Sampled 10000 Ratings

KFold Test #1:
Precision: 0.8167001003009027
Recall: 0.970912738214644

KFold Test #2:
Precision: 0.8087349397590361
Recall: 0.9706325301204819

KFold Test #3:
Precision: 0.7946920380570857
Recall: 0.9664496745117677

KFold Test #4:
Precision: 0.8343373493975904
Recall: 0.9754016064257028

KFold Test #5:
Precision: 0.8167587476979742
Recall: 0.9673530889000502
```

Estimated ratings vs True Ratings



Recommendations

We have then used the algorithm to obtain top-10 movies/TV recommendations for a given user

```
The top 10 product recommendations with estimated ratings for user A3478QRKQDOPQ2 is:
0   Touched by an Angel: Season 8  :  4.03078239289202
1   Gone  :  3.6651529781356724
2   Gran Torino  :  3.639651038026172
3   Peace Child VHS  :  3.604098289289142
4   Reaching From Heaven  :  3.5719288495937476
5   Bamboo In Winter  :  3.5171483183461185
6   I Love Lucy: Season 1, Vol.6  :  3.3879513852253926
7   The Pretender / The Daylight Zone / Crime Of the Age  :  3.376563113770536
8   Pilgrim's Progress - The Immortal Story of Pilgrim's Journey to the Celestial City  :  3.3619258696166323
9   Climb a Tall Mountain VHS  :  3.3451336795163487
```

Challenges

- SVD does not take rating score range into consideration.

- SVD can be computationally intensive to build and train on anti-test set.

- We are training on a HPC cluster to train on a GPU to obtain more optimised results. We submitted our job to a backfill queue and we hope to receive better rating predictions on unseen data for all users.

Conclusion

- In conclusion, we have improved the predictions by comparing various matrix factorization algorithms' test RMSEs and choosing SVD as the optimal recommender.

- We have further lowered the errors by tuning the parameters of SVD, and training a sample dataset, followed by the k-folding approach on the full dataset.

- This way, we have improved the overall precision and recall scores and obtained the top-10 recommended movies for a given user.

THANK YOU