
Book
Recommender
Project by AV David

Outline

The Problem

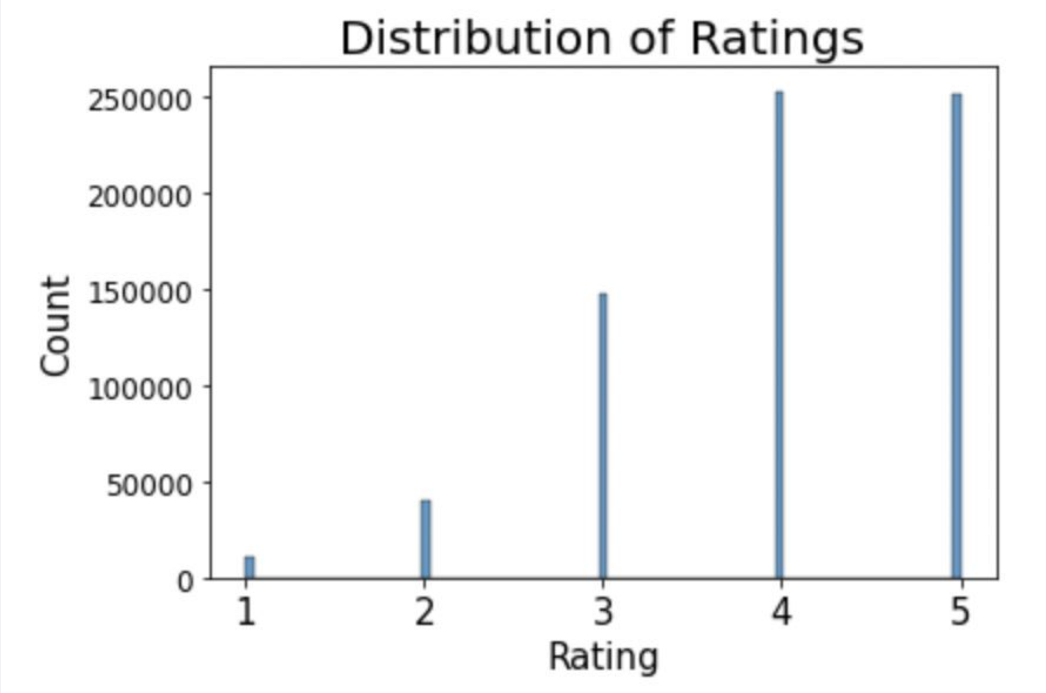
Create an algorithm that recommends books to readers.

The Data

	Books (Items)	Readers (Users)	Ratings
Goodreads Dataset	2,360,655	876,145	104,551,549
Genre: Children's Books	124,082	90,381	703,527
Sample used for modeling	3,512	7,684	91,567

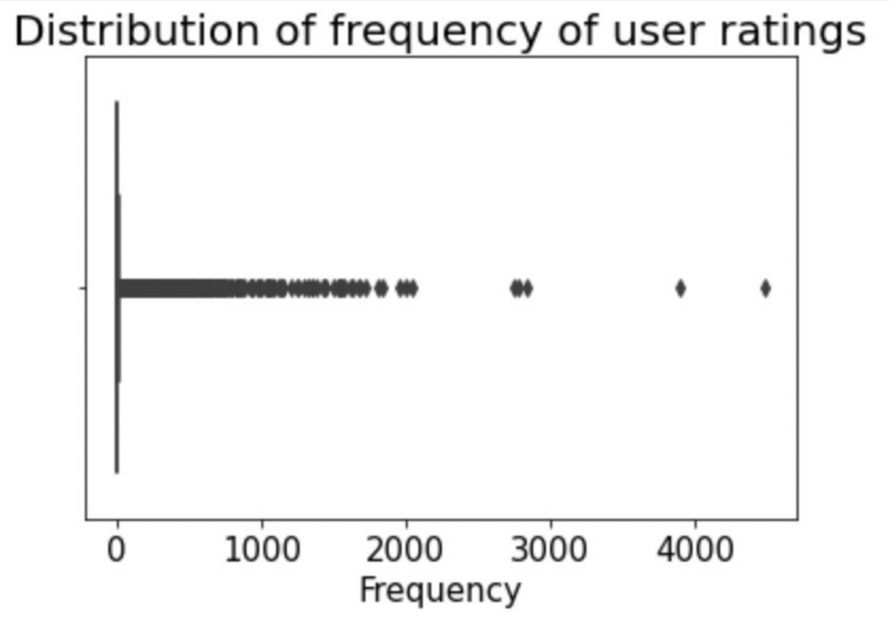
Children's Books Ratings

	Rating
count	703,527
mean	3.987
min	1.00
25%	3.00
50%	4.00
75%	5.00
max	5.00



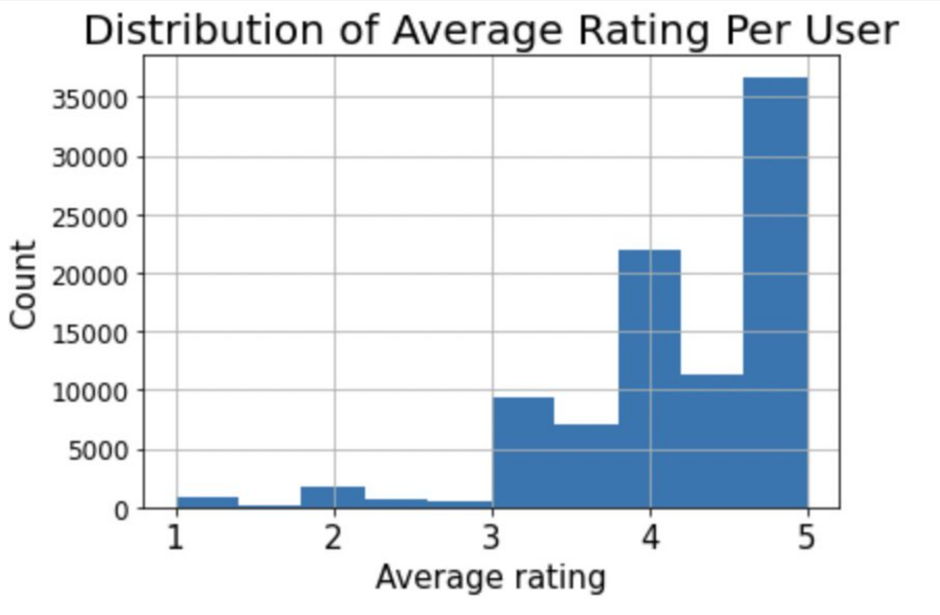
User Rating Frequency

	Rating Frequency of Users
count	90,381
mean	7.78
min	1
25%	1
50%	2
75%	4
max	4,481



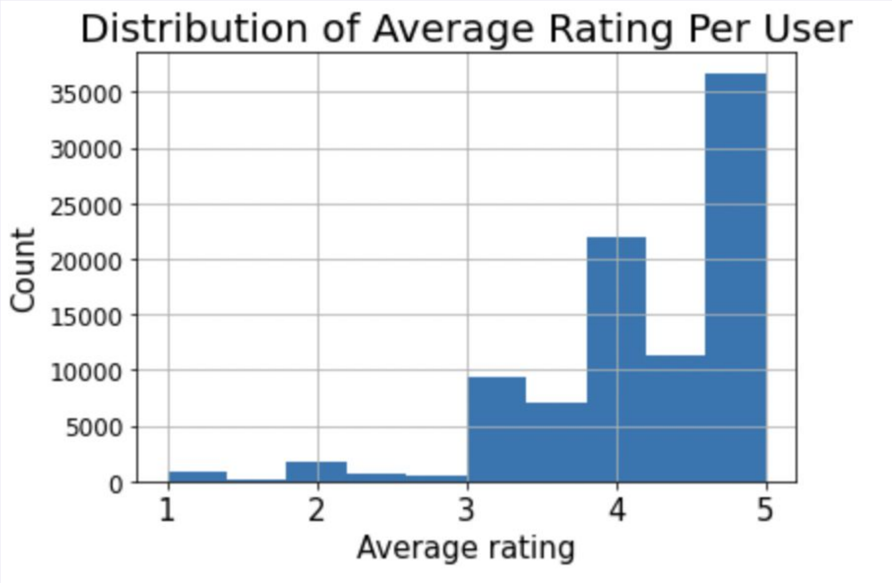
User Rating Average

	Average Rating Per User
count	90,381
mean	4.22
min	1.00
25%	4.00
50%	4.33
75%	5.00
max	5.00

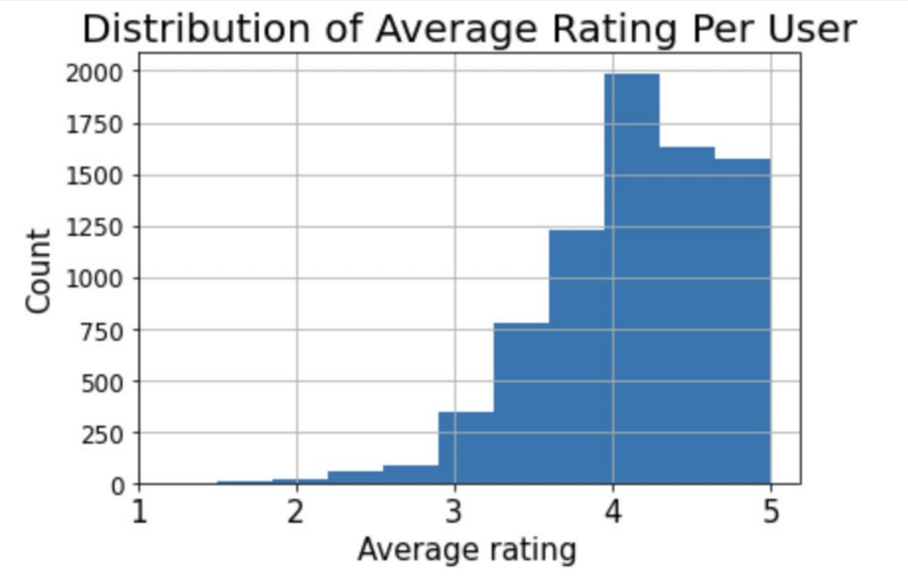


	Original Data: Average Rating Per User	**Sample: Average Rating Per User**
count	90,381	7,684
mean	4.22	4.13
min	1.00	1.50
25%	4.00	3.77
50%	4.33	4.17
75%	5.00	4.52
max	5.00	5.00

Original Data



Sample



Very Sparse: User by Item Matrix

	Item 1	Item 2	Item 3	Item 3512
User 1	?	4	?	?
User 2	3	?	?	3
User 3	2	1	?	?
User 7684	?	?	5	?

0.34 % sparsity

Collaborative Filtering

K-Nearest Neighbors (KNN)

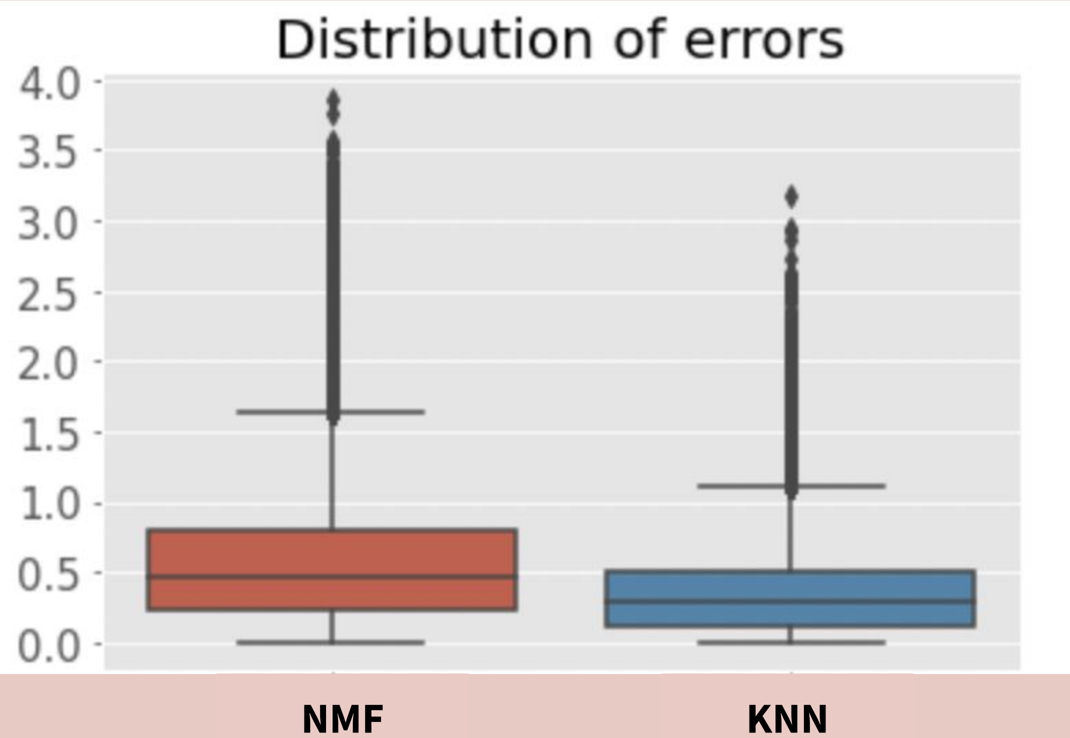
- Memory-based
- Similarity matrix

Non-negative matrix factorization (NMF)

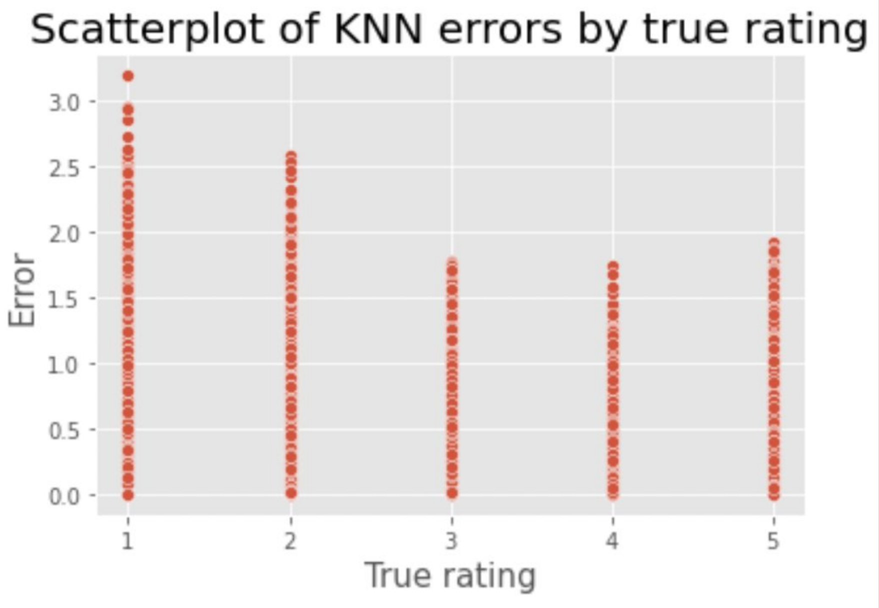
- Model-based
- “Latent factors”

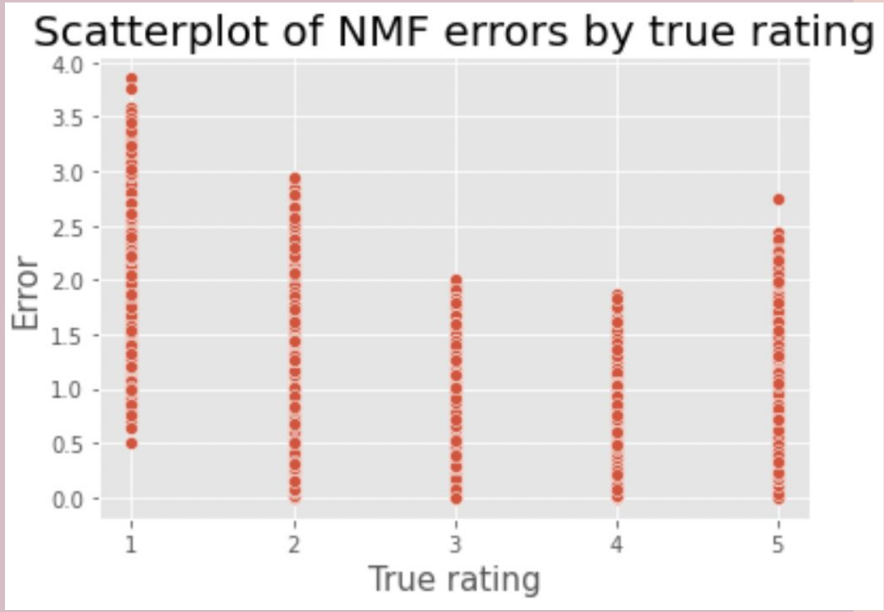
Comparing the two models

	NMF	KNN
count	91567	91567
mean	0.56	0.35
min	0.00	0.00
25%	0.23	0.12
50%	0.47	0.29
75%	0.79	0.51
max	3.85	3.19

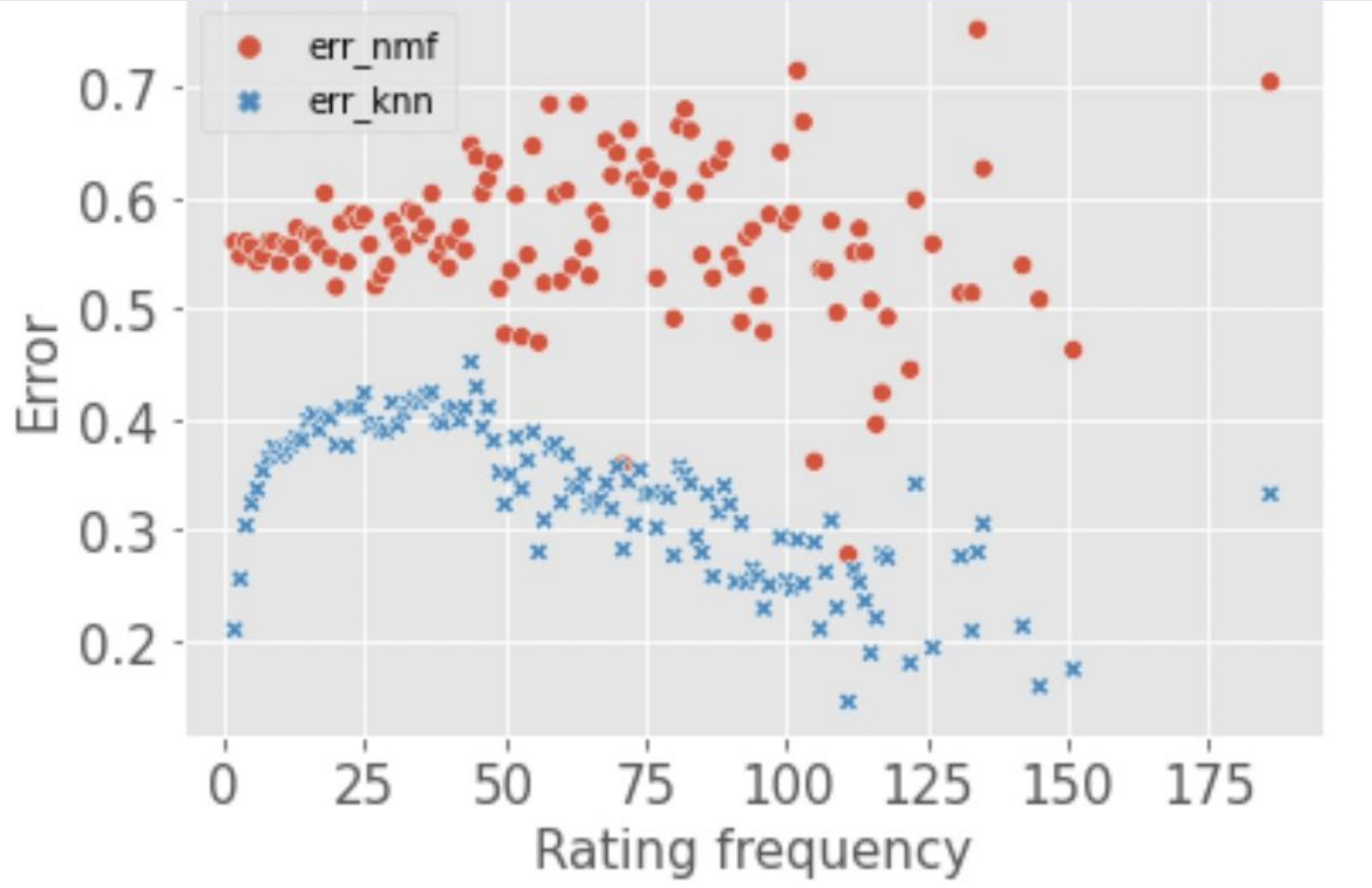


Comparing the two models

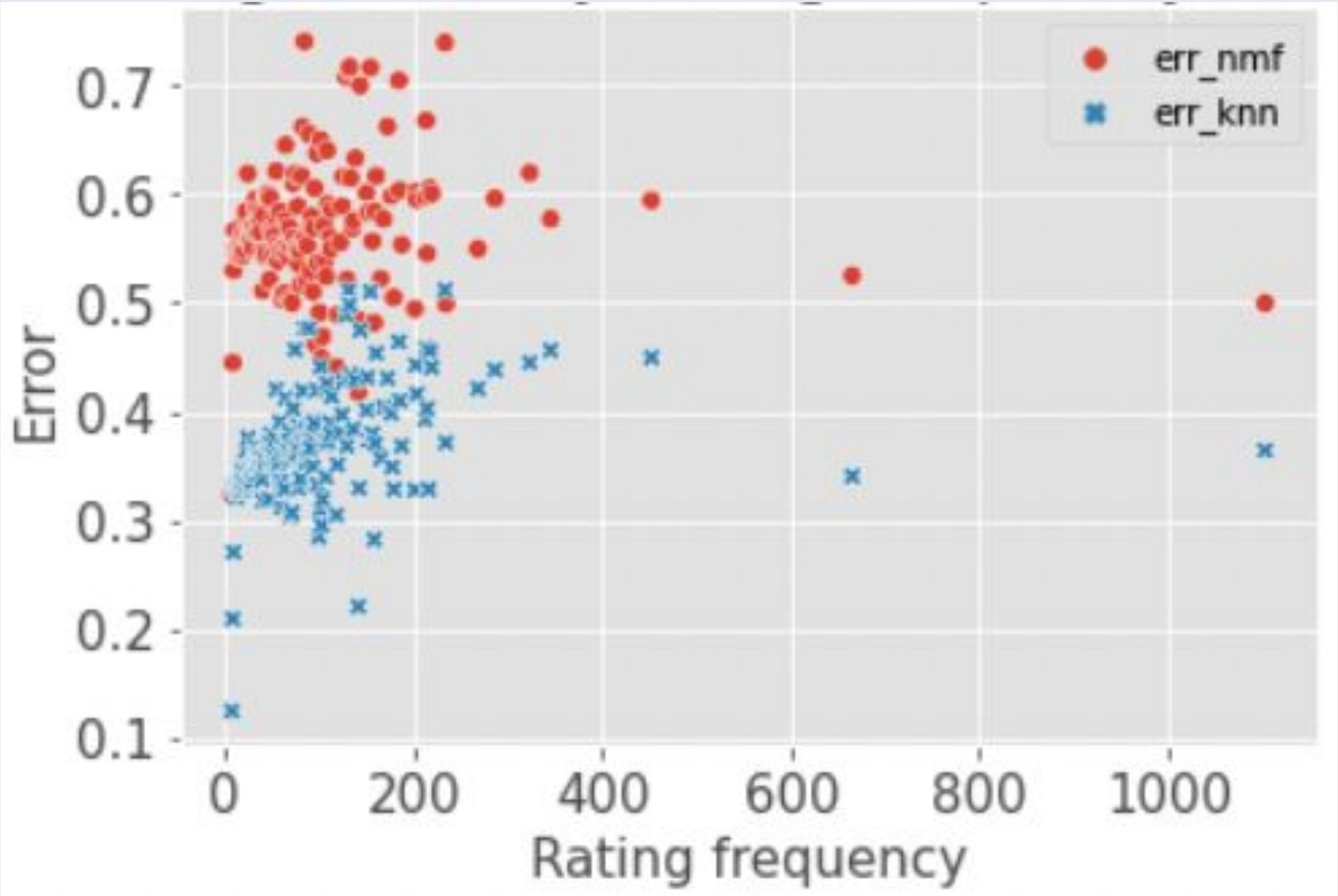




Average error by rating frequency of user



Average error by rating frequency of item



Conclusion

- Hybrid approach
  - New users: KNN
  - Existing users: NMF
- Considerations
  - Accuracy
  - Processing time
  - File size

Demo

Recommendations for further work

- Look into book series
- Input keywords to search for actual titles
- Other model-based algorithms
  - Neural nets
- Other libraries
  - fastai

Thanks

Questions?

auroravhd@gmail.com
www.linkedin.com/in/avdavid/

Appendix

Algorithm	RMSE
KNNBasic	0.8839
KNNWithMeans	0.8326
KNNWithZScore	0.8537
KNNBaseline	**0.7942**
NMF	**0.7961**
SVD	0.7964

Parameters

KNNBaseline

```
bsl_options = {'method': 'sgd',
        'reg': .08,
        'learning_rate': .005,
        'n_epochs': 40}

sim_options = {'name': 'msd',
        'min_support':1,
        'user_based': False}

algo_knn = KNNBaseline(k=40, min_k=2, sim_options = sim_options, bsl_options = bsl_options)
```

NMF

```
algo_nmf = NMF(n_factors=8, n_epochs=40, biased=True,
        reg_pu=0.8, reg_qi=2,
        reg_bu=.03, reg_bi=0.3)
```