RecSys course Challenge PoliMi 2023

Claudio Moriello : 10869147
Jody Roberto Battistini : 10691667

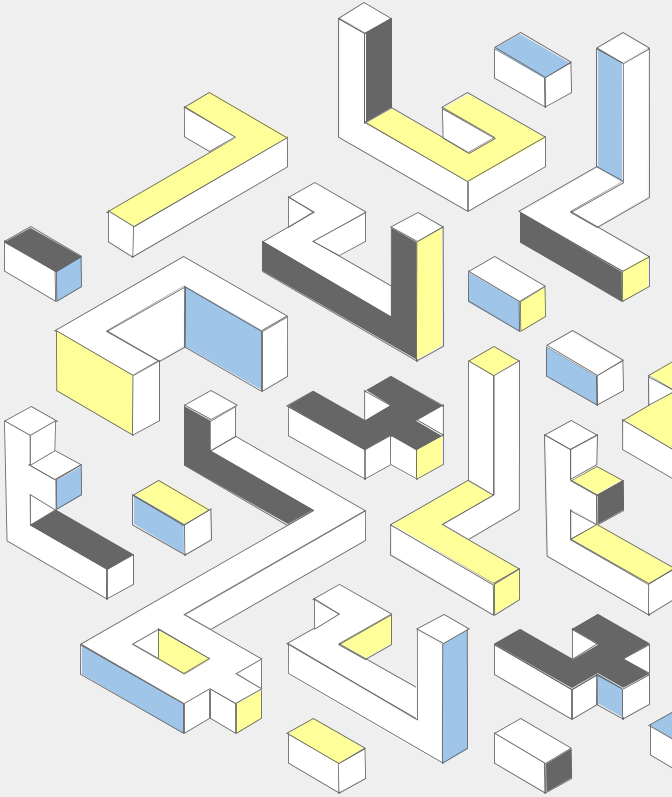


Table of contents

01

Recommenders

- ItemKNNCF.
- RP3Beta.
- SLIMElasticNet.
- IALS.
- MultVAE.

02

Hybrid

Experiments with different hybrids: normalized, not normalized and one by one.

03

Experiments

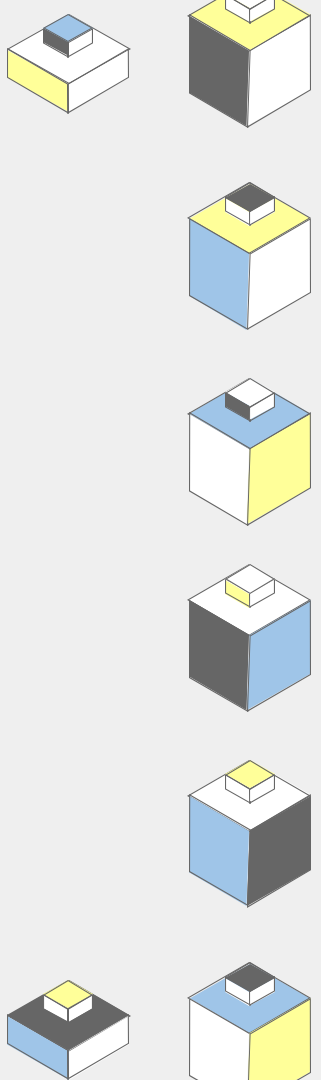
- Data Augmentation.
- Tail Boost.
- User interaction based hybrid.
- XGBoost (Ranker and Classifier).

ItemKNNCF

Collaborative Filtering Recommender.

First ever submission and pretty good MAP even alone.
Hyperparameter tuning pretty straightforward.

Tversky similarity much better than cosine similarity, because it takes into account elements present in only one of the two sets, which might be significant in an implicit feedback context.

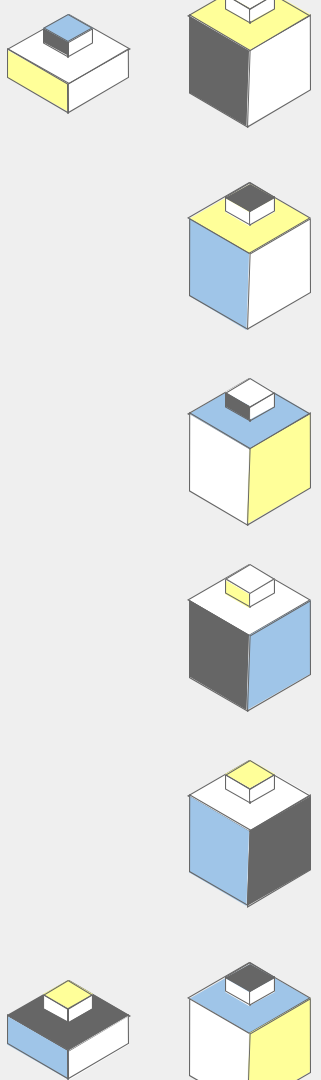


RP3Beta

Graph based recommender.

Good MAP alone and great improvement on hybrid recommenders.
Hyperparameter tuning pretty straightforward.

Very robust recommendations in our case of implicit ratings.

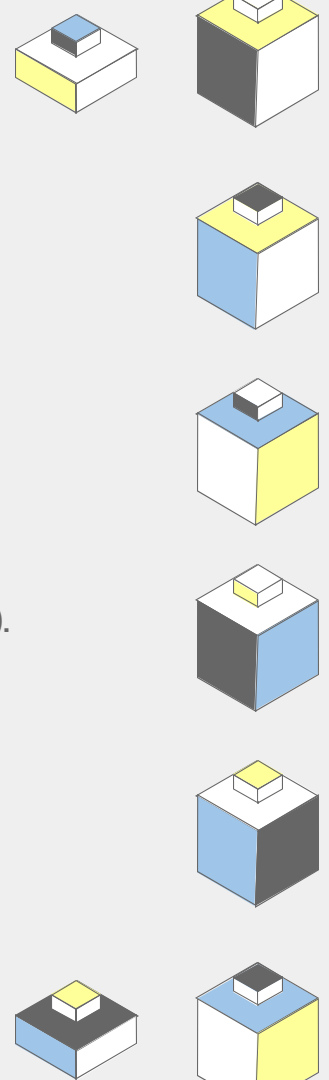


SLIMElasticNet

Sparse linear methods recommender.

Best MAP alone.
Hyperparameter tuning extremely difficult, need high level CPU(Hetzner).

Hyperparameter tuning done with multi-thread algorithm and then use the same weights in the standard one.

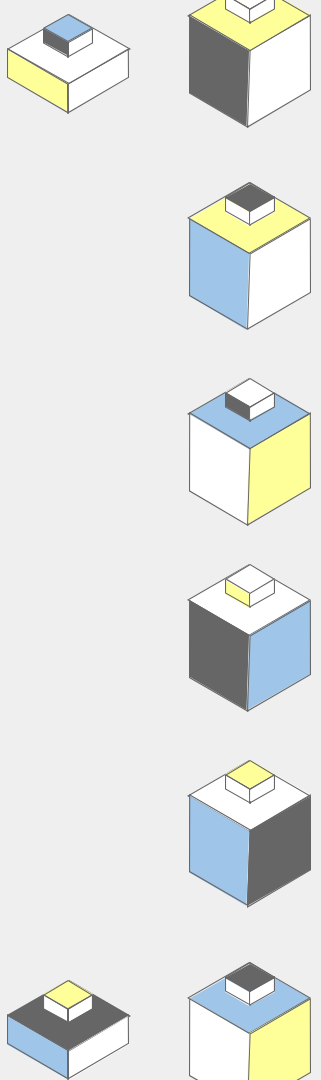


iALS

Matrix factorization recommender.

Very low MAP if used alone and low improvement if used in a hybrid and only with a negative weight.
Hyperparameter tuning extremely difficult, need GPU (colab).

Slightly better than the ALS alternative from external libraries.



MultVAE

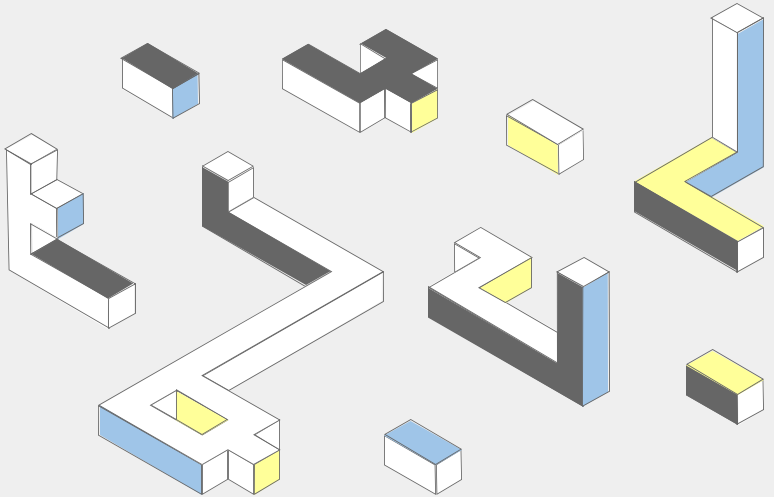
Neural Recommender (Variational Autoencoder)

Low MAP if used alone but high improvement if used in a hybrid.
Hyperparameter tuning extremely difficult, need GPU (colab).

Randomness due to its nature, so we can't save and load the model.
Best improvement for hybrid recommender, solved the cold start problem.

02

Hybrid



Process

1

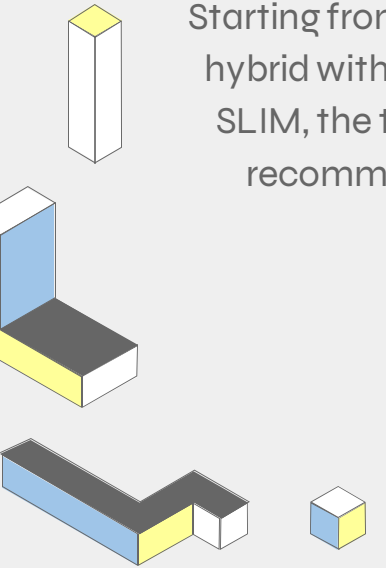
Starting from a simple hybrid with RP3 and SLIM, the two best recommender.

2

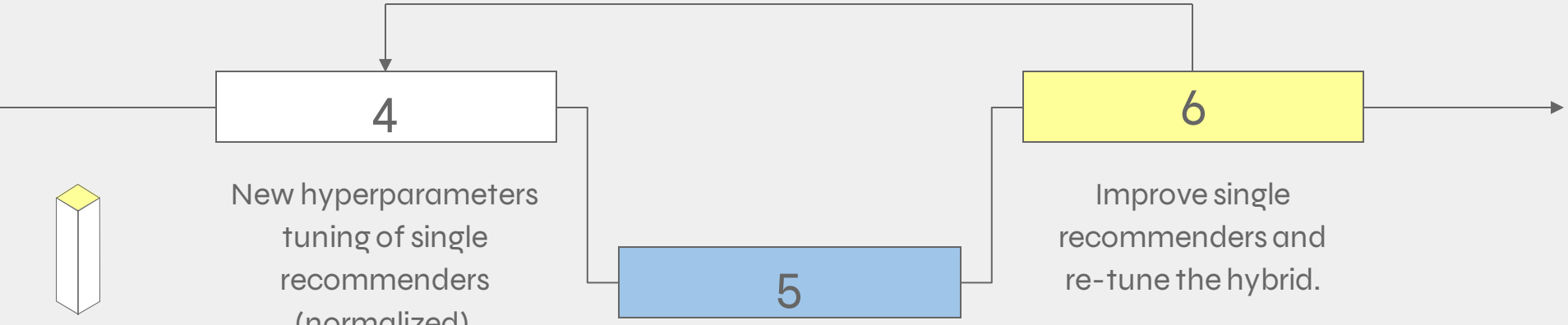
Various attempts, adding more recommender.

3

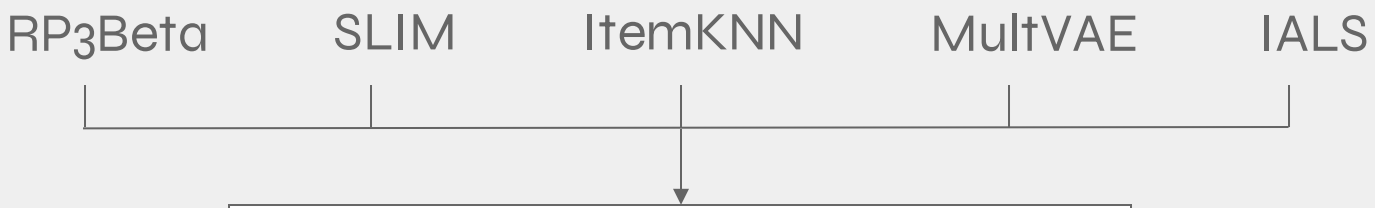
Great improvement when adding MultVAE.



Process


4
New hyperparameters tuning of single recommenders (normalized).
5
Hyperparameter tuning of hybrid (normalized).
6
Improve single recommenders and re-tune the hybrid.

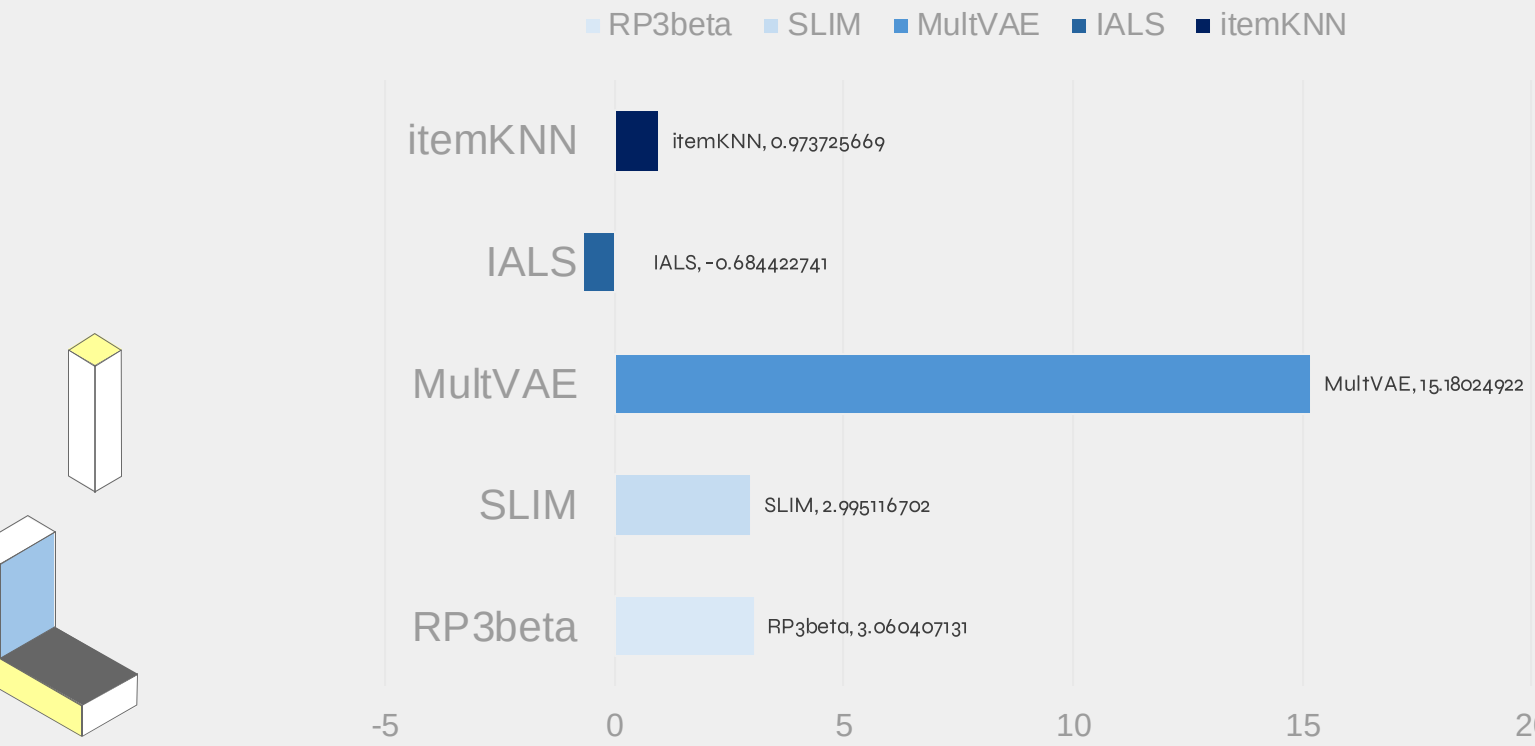
Best Hybrid




```
mean = np.mean(rec_weights)
std = np.std(rec_weights)
weights[rec_id] = (rec_weights - mean) / std
```

0.14192

Recommenders' weight



Experiments on hybrids

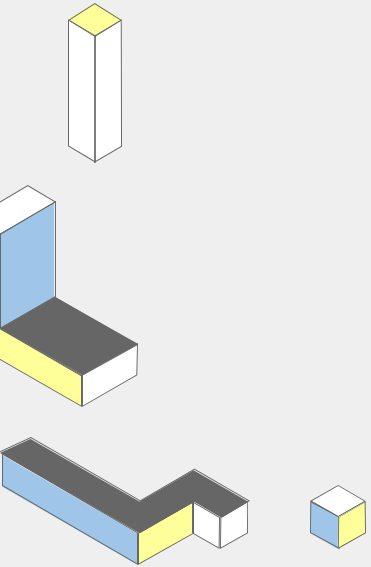
Add and remove recommenders.

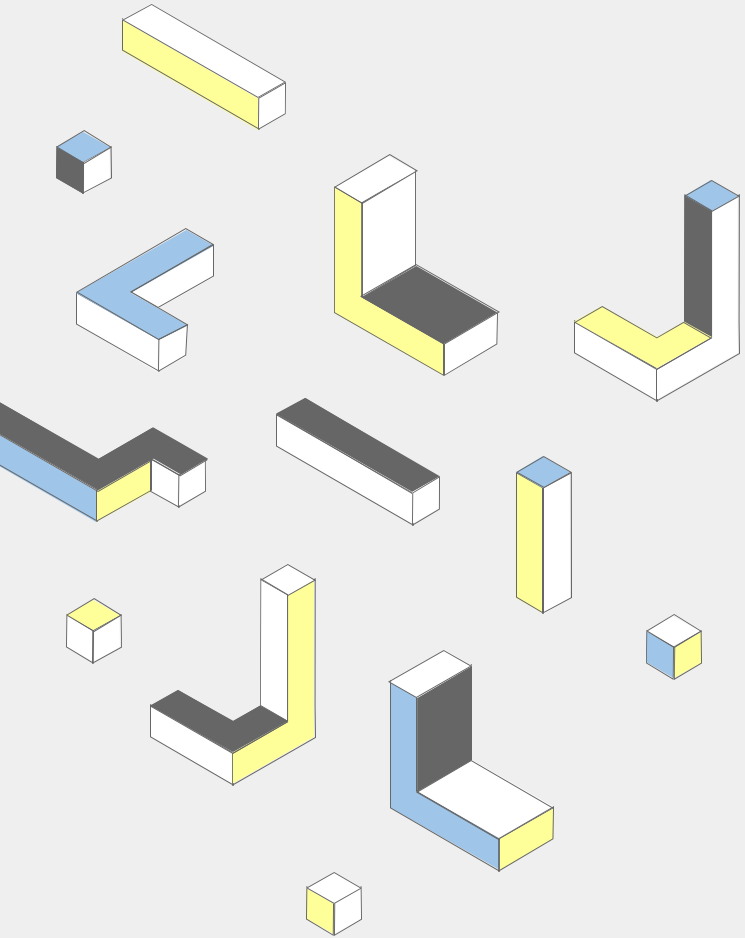
Build the hybrid one recommender at a time (overfitting).

Use different types of normalization.

Put togheter hybrids of recommenders of the same type.

Hybrid of two hybrids : normalized and not.





03

Experiments

Failed Experiments

- Data Augmentation.
- Tail Boost.
- Remapping User IDs.
- User Interaction Based Recommender.
- XGBoost Ranker.
- XGBoost Classifier.