Building Recommendation Systems in Python

PyData Geneva

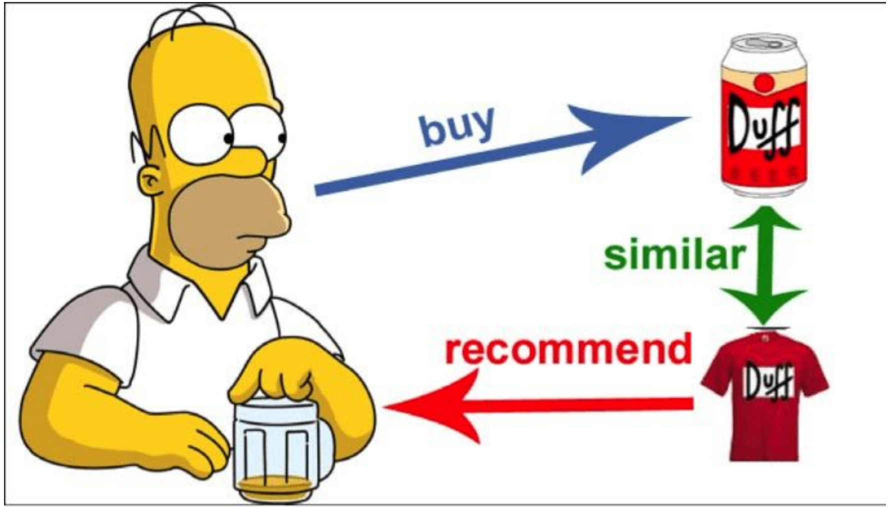
Vladimir Ilievski
5 April 2023

Outline

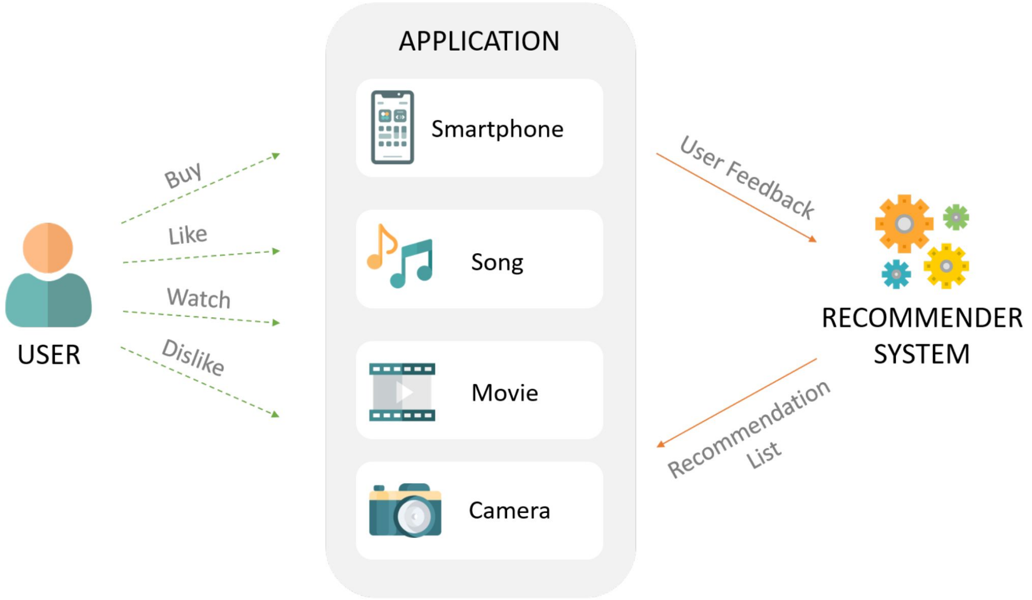
- RecSys Examples
- Intro to RecSys
- Collaborative Filtering
- Demo using RecBole

Intro to RecSys

Recommendations Everywhere



Recommendations Everywhere



Recommend Item **I** to the user **U**

The item could be anything:

- News
- Movies
- Videos
- Tweets
- Books
- ...

Let's see some examples

Recommendations Everywhere

Some Examples

In E-commerce

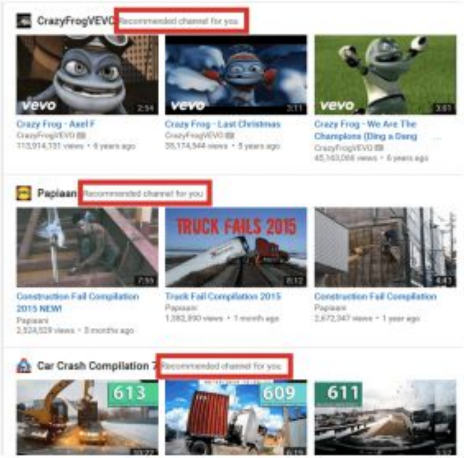
Recommendations Everywhere

Some Examples

Content Sharing

YouTube

Pinterest



Recommendations Everywhere

Some Examples

Social Media

What is a Recommendation System?

- Recommend the best possible set of **items** to a given **user**
- Based on the history of **interactions** of the users with the items
- We have 3 main sets:
  - U: set of all **users** uniquely identified with and ID
  - I: set of all **items** uniquely identified with and ID
  - R = U x I: matrix of user-item interactions

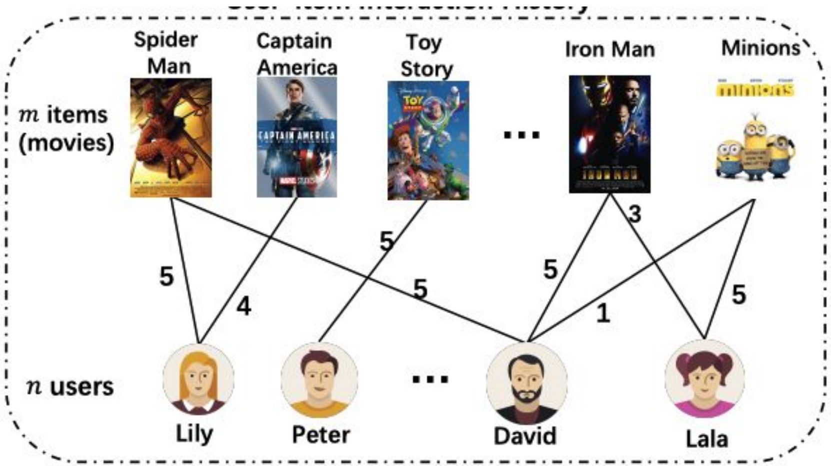
Type of Recommendation Systems

In general there are 3 types:

- **Content-based**: based upon the user and item descriptions or features
- **Collaborative filtering**: based upon the interaction of the users with the items
- **Hybrid**: combination of both

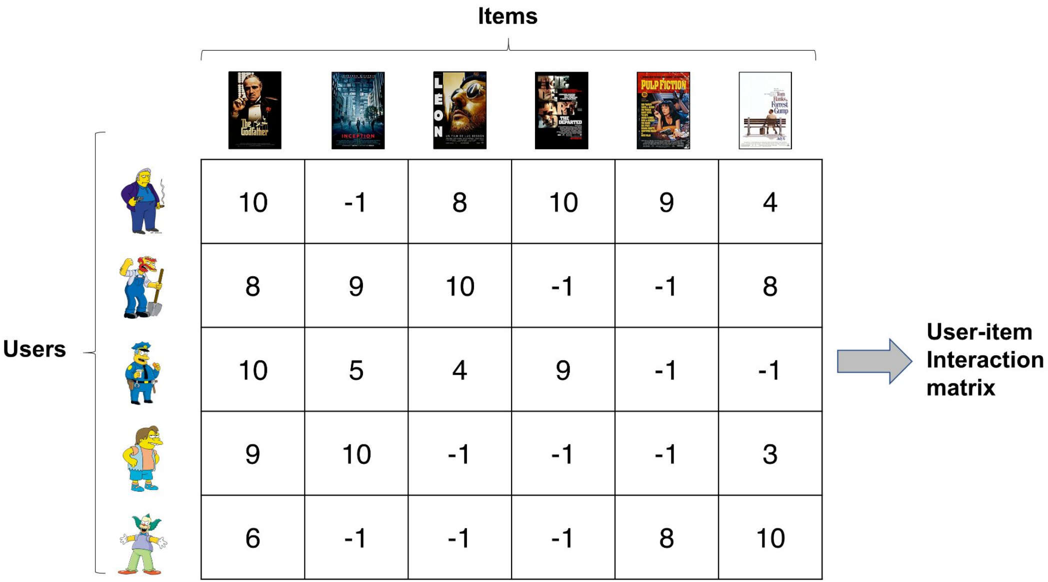
Collaborative Filtering

Learn from users preferences



Interaction Matrix

- Users in rows
- Items in columns
- Matrix elements as interaction records
- e.g. users rating movies



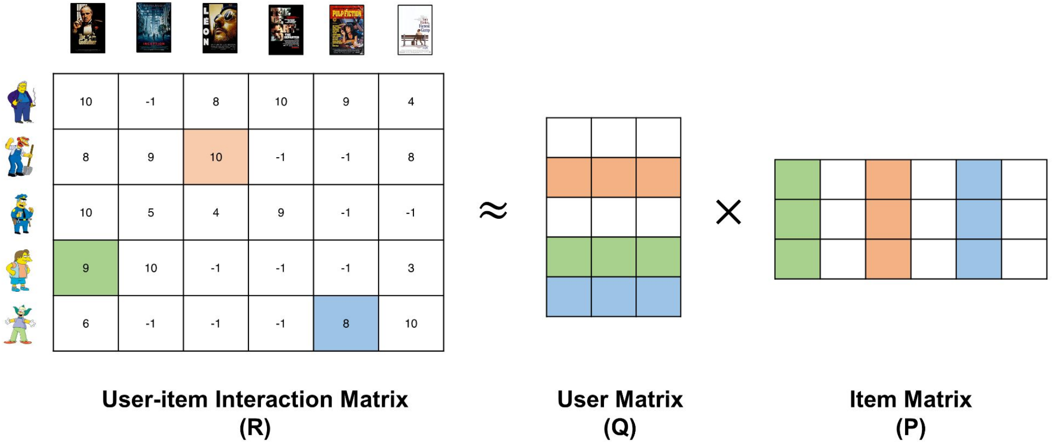
10	-1	8	10	9	4
8	9	10	-1	-1	8
10	5	4	9	-1	-1
9	10	-1	-1	-1	3
6	-1	-1	-1	8	10

How can we learn to make recommendations?

Matrix Factorization

- Decomposing the interaction matrix as a product of:
  - User embeddings
  - Item embedding
- We have to find i.e. to learn those embeddings
- The estimated ranking is then:

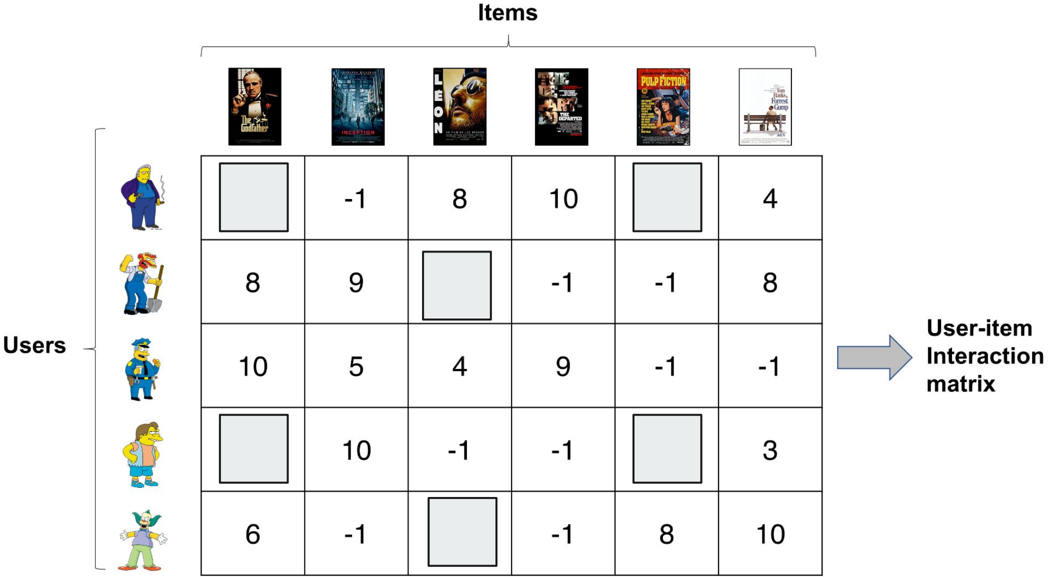
$$\hat{r}_{ij} = q_i p_j$$



10	-1	8	10	9	4
8	9	10	-1	-1	8
10	5	4	9	-1	-1
9	10	-1	-1	-1	3
6	-1	-1	-1	8	10

How to learn the embeddings?

- Predicting the “missing” values
- We are intentionally hiding some values from the interaction matrix
- Then the model should learn to predict those values



	-1	8	10		4
8	9		-1	-1	8
10	5	4	9	-1	-1
	10	-1	-1		3
6	-1		-1	8	10

How to learn the embeddings?

The simplest way is to minimize the squared difference between the true ratings and the estimated tanking:

$$minimize \sum_{i,j} (r_{ij} - q_i p_j)^2$$

a.k.a. the Funk algorithm

Many other models

Based on the same principle

- Singular Value Decomposition
- Alternating Least Squares
- Bayesian Personalized Ranking (BPR)
- Neural Collaborative Filtering (NCF)
- Restricted Boltzmann Machines (RBMs)
- GRU4Rec
- Wide and Deep
- And many more …

How to evaluate the recommendations?

Metrics

Scoring Metrics

- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- Area under the ROC curve (AUC)

Ranking Metrics

- Hit Rate
- Recall@K, Precision@k
- Mean Reciprocal Rank (MRR)
- Mean Average Precision (MAE)
- Discounted Cumulative Gain (DCG)

RecSys in Python

Numerous Open Source Libraries

- TorchRec
- RecPack
- RecBole
- Implicit
- Microsoft Recommenders
- Vowpal Wabbit Recommenders
- Cornac
- Surprise
- And many more …

RecBole

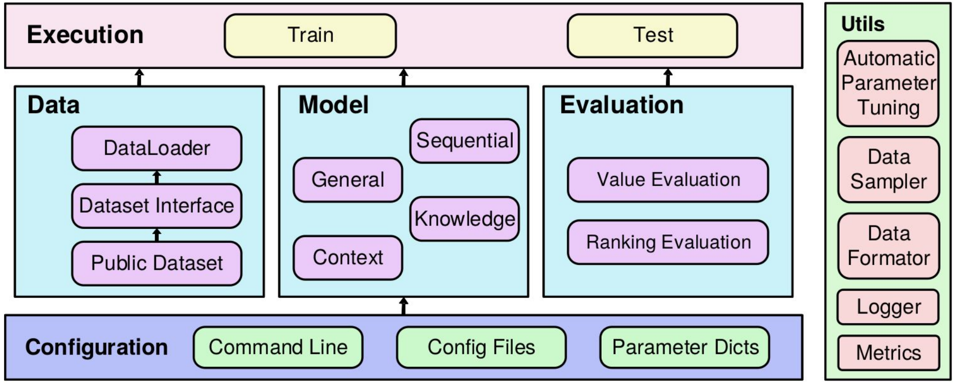
Building Movie Recommender

- It implements around 100 recommendation out-of-the-box models
- Divided in 4 categories:
  - General recommendation
  - Sequential recommendation
  - Context-aware recommendation
  - Knowledge-base recommendation
- It is very easy to use

Movie Recommender using RecBole

- Everything via the YAML configuration:
    - Select model
    - Select data
    - Select evaluation metrics and strategy

RecBole Architecture



You, 52 minutes ago | 1 author (You)

```yaml
## General
nproc: 1

## Model config
embedding_size: 64

## Dataset config : General Recommendation
USER_ID_FIELD: user_id
ITEM_ID_FIELD: item_id
load_col:
    inter: [user_id, item_id]

## Training
epochs: 500
train_batch_size: 512
eval_batch_size: 512

## Evaluation
metrics: ['Recall', 'MRR', 'NDCG', 'Hit', 'Precision']
topk: 10
eval_step: 2
valid_metric: MRR@10
```

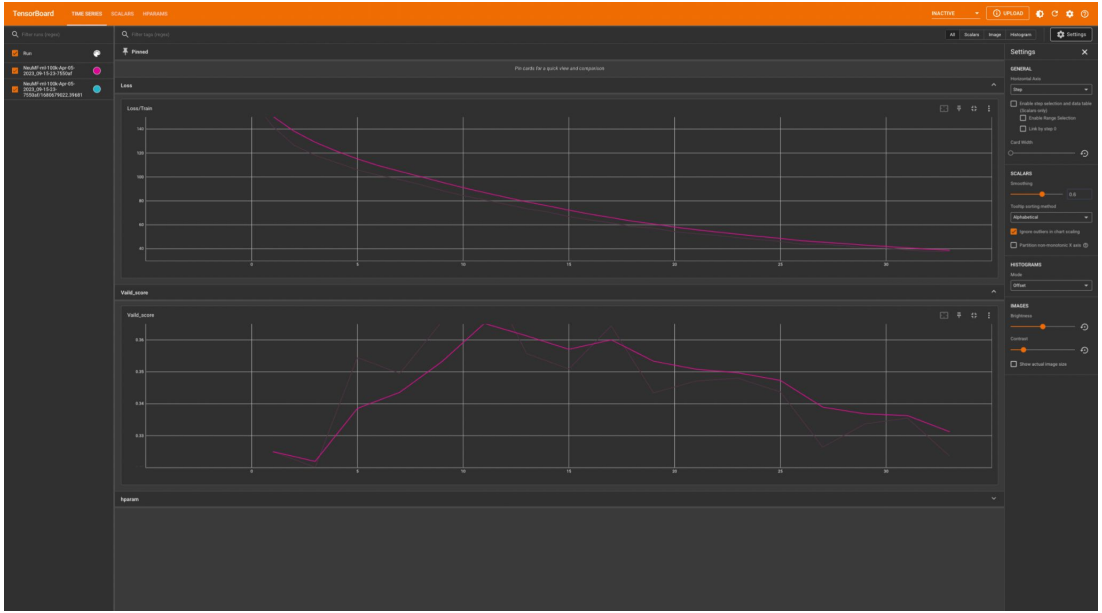
Movie Recommender using RecBole

- Train a simple RecSys using the NeuMF model
- Using the MovieLens 100K toy dataset
  - Useful list of RecSys Datasets
- Data format: three Atomic Files with the following extensions
  - *.user: data about the users*
  - *.item: data about the item*
  - *.inter: interaction between users and items*

Link to the GitHub repo

Movie Recommender using RecBole

Integrated TensorBoard logs:



Thank You for your attention!

Contact
Vladimir Ilievski
E-Mail: ilievski.vladimir@live.com
LinkedIn: https://www.linkedin.com/in/vilievski/