

Project Tasks

RecSys Challenge 2019

Recommender Systems

VU 194.035, 2019S

Dimitris Sacharidis



FAKULTÄT
FÜR INFORMATIK

Faculty of Informatics



collaboration across groups

- contribute **code** that handles some basic **tasks**
- and make it public for other groups to use
- rewarded with **project points**
- tasks should **not** be recommendation approaches, but rather helper functions
- consider the following sample predefined tasks
- or propose other tasks

Task 1. Split into train, dev, test

- Sample sessions from the dataset, to create a smaller dataset
 - may be completely **random**, or based on some distribution
- Split into **test**, non-test datasets
- Create **ground truth** from test subset
 - used for the scoring
- Split non-test into **train** and **dev**
 - may implement **k-fold** validation

Task 2. Create a Ratings Matrix

- Parse the csv and construct a **session-item matrix**.
- In each session, note which items are interacted with, and count how many times
 - you may assign weights/points according to action types
 - e.g., clickout is 2 points, other actions 1 point
- A matrix entry contains the **points** for an **item** in a **session**

Task 3. Implement Ensemble

- Assume multiple submission files
- Consider a specific click-out for prediction
 - Each item in a ranking receives points (like votes)
 - e.g., top ranking item 1, next 1/2, then 1/3... like MRR (or like NDCG)
 - Sum the points and sort items by total points
- Or implement other ensemble methods

Task 4. Create Embeddings for Items

- An item is described (in `item_metadata.csv`) with a bunch of binary attributes
 - represented as a large sparse vector
- Learn a small dense vector (embedding)
- Introduce an appropriate learning task
 - e.g., autoencoder, word2vec
- Contribute the code and/or the vectors