**Exercise2 tasks:**

* information given sufficient for reproducibility:  
  workflow, data (subsets, test-train-split), random seeds, software (versions), hardware
* are reproduced results equal to paper?  
  if not: reasons?
* examine paper's experimental design:
  + have results been confirmed
  + CV
  + different random seeds compared
  + variance reported?
  + significance tests
  + flaws in the experimental design?
* documentation of workflow & tools  
  differences compared to paper

**Paper analysis**

**Material given for reproducing:**

* all material available as git-repository on github
* code as a single python file
* all settings for each dataset as config-files, that are loaded into the main code-file
* datasets included in repository:
  + already preprocessed
  + already split into train & test sets
* versions of used packages
* used CPU and size of RAM
* results and complete printed output for each dataset:
  + Recall@20, NDCG@20
  + configuration settings
  + results of each training epoch including train time, test time, loss, f1, precision, recall, NDCG
  + early stop at epoch number
  + number of best epoch
  + best recall and best NDCG
* exact evaluation measures

**Analysis of experiment setup:**

* Data:
  + ml-1m:
    - MovieLens 1M: 1 mio. movie ratings
    - Preprocessed data taken from another repo:  
      <https://github.com/Wenhui-Yu/LCFN>
    - Preprocessing consists only of parsing integer-values from .dat-file, selecting specific columns (user-id, movie-id) and reshaping the data matrix
    - Split size: train:test ~ 88:12
* early stop triggered:
  + if recall has not improved for *early\_stop\_count* of times
  + *early\_stop\_count* is defined in dataset-specific config-files (between 15-250)
* Sampling (main.py line 238)
  + for sampling data for each batch during training
  + no seeds for numpy random sampling defined  
    (maybe therefore slightly different results with our own run)
* choice of main measures (Recall, NDCG):
  + paper: “Recall@20 and NDCG@20 are chosen as the evaluation metrics as they are popular in the evaluation of GCN-based CF models.”
  + Recall@20:   
    How many of the top 20 relevant values are among the top 20 recommendations (focuses on the number of relevant items found)

<> Precision@20: How many of the recommended items are actually relevant (focuses on the average relevance of all recommendations)

* + NDCG@20:  
    Measures the overall relevance of the top 20 recommendations, where the results are normalized across all recommender-queries
* Comparison to other approaches done with significance testing:
  + Exact resulting p-values given
  + no code or details about implementation of tests
* total training time, number of training epochs needed and time per epoch compared to other approaches
* ablation study:
  + parameters and were used to disable certain features and investigate the resulting performance
    - reduced to matrix factorization training with BCE loss, no graph information, no item collaborations
    - no graph convolution
    - no learning of item-item relationships
    - Model performance:
  + Learning of item-item relationships: better performance with “augmenting positive user-item pairs” than with “optimizing between item-item pairs”
  + User-user co-occurrence information does not improve recommender performance
* parameter analysis:  
  influence of model parameters was investigated
  + K: number of selected neighbours
  + and

**Recall@*k*:**

For recommender systems.

Measure regarding (relevance of) the top ***k*** results recommended by the system.

e.g. of the top *k* recommendations, how many are truly relevant

**NDCG**

*Normalized discounted cumulative gain*

Measure for recommender systems or search algorithms based on the weighted relevance of recommended items

<https://en.wikipedia.org/wiki/Discounted_cumulative_gain>