Home Assignment 1

General requirements

- Ensure your practical work is reproducible and could be graded without doubt on your model quality and performance
  - all seeds are fixed and the notebook rerun returns the same evaluation result
  - the notebook runs top down without errors
  - all cells related to the model building, inference and evaluation are timed with the *%%time* magic
  - all necessary metrics are displayed.
- Clearly describe your approach, experiments, results and conclusions. Add visualizations to support your findings.
- Please do not cheat and do not share your code with the classmates.

1 Problem 1 (10 pts)

Theoretical Task

It is often stated, that pure content-based recommendation models provide very low level of personalization to users.

Prove this claim using a standard **regression-based** formulation for the case when a single global model is learned in the form:

$$\boldsymbol{r \approx \theta x + \epsilon},$$

where vector x encodes some features of both users and items (e.g., user attributes and item characteristics), and θ are the corresponding learnable weights of the regression model. Recall that top-n personalized recommendations task is stated as a selection of the top-n most relevant items for a user:

$$\mathbf{toprec}(\boldsymbol{u}, \boldsymbol{n}) = \mathbf{arg}\,\mathbf{\max_{\boldsymbol{i}}^{\boldsymbol{n}}}\, \boldsymbol{r_{ui}},$$

where r_{ui} is the relevance score assigned by the model to item i for user u.

Optional: Propose the other algorithms or feature preprocessing techniques, which could provide higher personalization level than the regression-based model described above. The algorithm should take vector x, which encodes some features of both users and items (e.g., user attributes and item characteristics) as an input and return corresponding relevance of the item for the user.

2 Problem 2 (20 pts)

Content-based models with personalization

In this problem you will train simple content-based model **for each user individually** in order to achieve some level of personalization. See important details below in *Features and data split* subsection and follow the general requirements from the beginning of the file. Try to avoid cycles in your code. Each of your recommendation pipelines (feature preprocessing + training + inference) **should complete in less than 4 hours** for the entire set of test users (time it!).

1. **Build** a collection of CB (content-based) models on the anime data from Seminar 1. Recommend the $top - n$, where $n = 10$ most relevant anime for each of the test users and **perform a standard evaluation** (using material from Lecture 3). You are free to use any content-based approach from Seminar 1 (*regression-* or *similarity-based*). Describe your model and explain your choice in terms of the dataset, available features and the recommendation task.

2. **Improve** your model by
   - adjusting how it processes content information (e.g., selecting different features and feature preprocessing techniques; switching to different similarities *for similarity-based models* or adjusting regularization and the other important parameters of a *regression-based* model).
   - changing the way user history is taken into account. In this case you can use
     - several most recent items
     - a random subsample of items of a fixed size.

   Report how evaluation metrics change with your adjustments. Use standard **HR@n**, **MRR@n**, and **Coverage@n** metrics for evaluation.

3. **Compare** your model with any two baselines from the list below. Report the results.

   Possible models: [*Random recommendations*, *Popularity-based*, *Popularity-Based with Baesian averaging*]

Features and data split

- Take the test users from Seminar 1. To get the test dataset, randomly select one liked item for each of test users; use the rest of the users' history for training.
- You could also use non-personalized anime scores and descriptions to enrich your model.
- You could use all available user and item features, except for the users' favorites.
- Keep in mind that you need to score all items for a user to get top-n personalized recommendations. Be careful using user-item pair features (e.g. user review text) as those features are not available for the entire set of user-item pairs.