

Final project by:
Sukhorukov Nikita
Nuzhnov Mark
Ivanov Nikolay
Alexandrov Bogdan
Viskov Vasily

Anime Recommendation Bot

- Problem Statement

Design and create a recommender system based on the telegram bot that will recommend users anime based on their rating history or on their personal information (age, gender).

- Dataset description

Dataset was taken from [kaggle](#). It is built with information from [myanimelist](#) (MAL) and contains user information (username, userid, gender, location), interactions (rating given to anime by users and anime info such as genre, type, source, score, members of fandom) Original version contains more than 80M interactions between users and anime. Cleaned version has truncated all users with ridiculously large number of episodes in anime which obviously don't have that much episodes, watched episodes larger than number of episodes in individual anime were fixed and seen episodes and watch time were recalculated accordingly. For some users, last online was 1900 year, just weird values, so their last activity was inferred from their last animelist update timestamp. Many users incorrectly filled number of rewatched episodes. For anime where more episodes have been watched than that anime has episodes, watched episodes have been rewritten to number of episodes in that anime. Watch time and number of watched episodes have been fixed accordingly. Users too young and too old obviously were truncated too. After the cleaning we were left with 20M interactions.

- Splitting Strategy

For evaluating different models we split our dataset. The split is performed based on the timestamp column of data, using 0.95 as the quantile value. The holdout dataset contains only the immediate interactions following the fixed timepoint for each test user from the testset. The set of users in training is disjoint with the set of users in the testset, which implements the warm-start scenario.

- Baseline models

First of all we started with popularity based model and random recommendation model. They were chosen as one the most commonly used baselines for such kind of problems. Comparing with this models allows us to be sure that we are actually predicting something reasonable.

- Models

We started with scaled SVD. Performed the grid search to estimate the best parameters. It easily solves warm-start problem that we have via folding-in but useless in cold-start scenario. That is why we decided to implement lightFM, since it uses items' and users' features alongside their interactions. In case of lightFM we solve the warm-start problem also via folding in, but we implemented it ourselves in the following manner:

```
def warm_start(model, user_preferences, item_features):
    item_feature_bias, item_feature_factors = model.get_item_representations()
    item_factors = (item_features @ item_feature_factors)
    item_bias = (item_features @ item_feature_bias)
    item_factors_inv = np.linalg.pinv(item_factors)

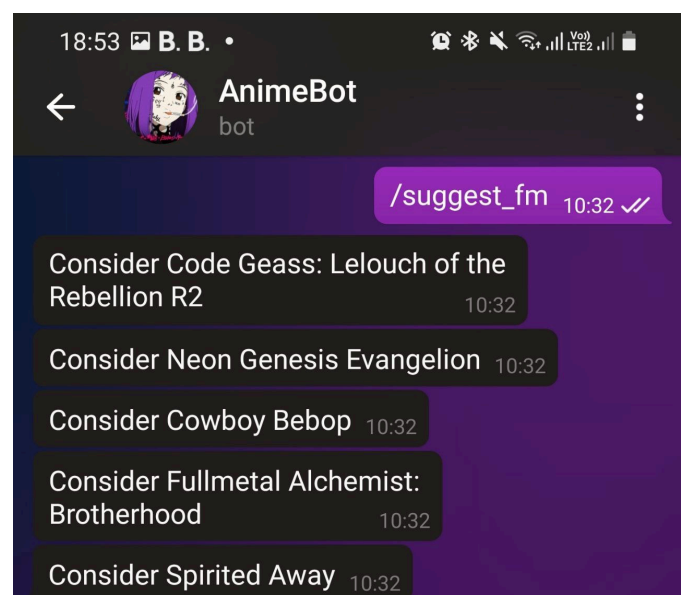
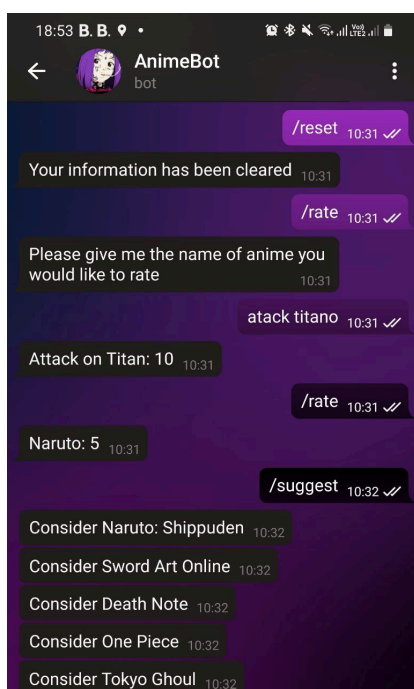
    new_user_to_item = user_preferences

    user_factors = (item_factors_inv @ new_user_to_item.T).T
    predictions = user_factors @ item_factors.T + item_bias
    return predictions
```

For comparison metrics we've chosen HR@5 and coverage. We've chosen the number of recommendation equal to 5 since it seems to be the most reasonable number for person to comprehend via the telegram chat. The results are in the table below:

	HR@5	COV
Random	0.000261	1.0
Popular	0.00197	0.00083
SSVD	0.0484	0.463
LightFM	0.0382	-

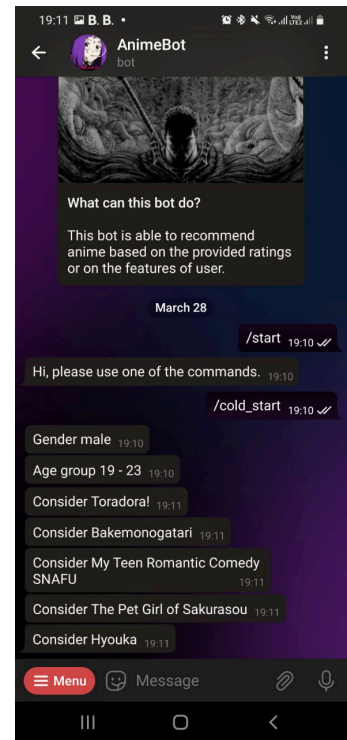
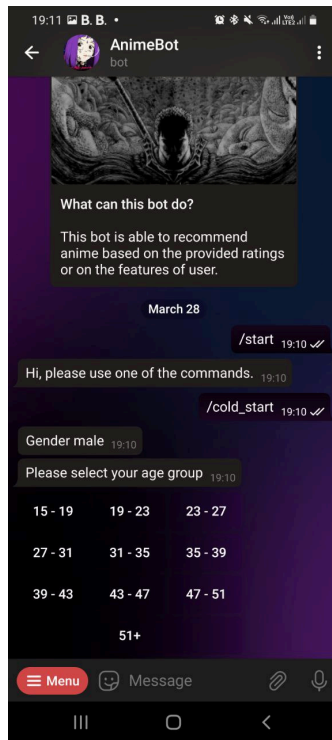
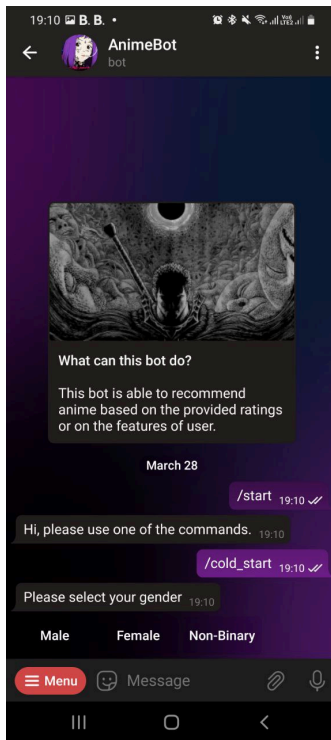
Despite LightFM having worse HR@5 it solves the problem SVD has when we recommend user something similar to disliked item.



Above is an example when we rated Naruto with 5/10 and still got recommended something from Naruto by SSVD, however LightFM didn't recommend anything related to it.

- Cold-start

To solve cold-start problem when user haven't watched any anime we use LightFM and ask user to provide his gender and age group:



- Future development

For the future development we plan to item cold start (be able to recommend new anime that has no ratings), use newer dataset (ours is from 2020). Make better and faster search

- Contributions

Sukhorukov Nikita - models and experiments

Nuzhnov Mark - telegram bot

Ivanov Nikolay - dataset and presentation

Alexandrov Bogdan - experiments, dataset processing

Viskov Vasily - experiments, report

- [github](#)

- supervisor's feedback

