Report

Data Preparation

Firstly in data preparation I needed to clean the initial dataset and make feature encoding. In u.data there were no missing values so I only needed to encode the timestamp value. For this I used cyclic feature encoding for day and month and made normalisation for year. Since year diapason can differ in other tables I decided to set an universal diapason from 1900 to 2000.

In u.item I firstly cleaned data since there was a column with NaN values and links. Then I found out that date values are not a movie release date but a movie announcement date. Also there is a release year in each title. So I decided to get years from the title and make it a separate feature. All dates I encoded the same way as in the previous dataset. For the title I used spacy to create embeddings for them.

In the u.user dataset I decided to remove the zip code column, since it requires resourceful encoding. To encode and use zip code properly it needs to be transformed in clusters and it also creates many new features that makes learning harder. For the occupation feature I used OneHotEncoding and for gender feature I used OrdinalEncoding. Also I needed to make a normalisation of the age feature.

To concatenate all features in one dataset I filtered rating duplicates, so that there were no ratings from one user for one film twice or more. Then I filtered u.data to remove unexisting users and movies.

Finally I combined all preprocessing to one function, so I could make this preprocessing to parts of the dataset separately to avoid data leaks.

Modelling

I decided to make a recommendation system based on prediction of user rating for movies. For this I needed to make a regression model. Since we have lots of features I decided to use a forest-like model. For that the best solution was to use XGBRegression because it is fast even on cpu and shows good results. To find the best parameters I used GridSearchCV. The best model was with parameters {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300} and showed the best learning score of MAE 0.24.

For the recommendation part I created a function that takes as input users that need prediction, using model predict ratings for movies and returns for each user N movies with highest predicted rating.

Evaluation

Since recommendations are based on rating prediction for evaluation of the system I can evaluate the quality of the prediction model. The results of model are Mean Squared Error (MSE): 0.94, Root Mean Squared Error (RMSE): 0.97 and Mean Absolute Error (MAE): 0.77. According to this metrics model, on average makes mistake in prediction in less than 1 star on a 5 star scale which I consider a good result.

A disadvantage of this system is that current realisation cannot make assumptions for users without watch history since it considers what ratings users have watched. So the

further development it is a good idea to also create a regression model that considers only user characteristics.