Andrey Ambartsumov

MAIS202 – Deliverable 2

Deliverable Description:

1. I plan on making a movie recommendation model, where given a user makes a certain number of reviews, I can predict movies to watch.
2. I am using the MovieLens dataset. I currently am working on the MovieLens100k dataset, but I will probably opt for the one with 32M entries in order to increase precision. I kept the dataset as-is, besides adding a column in the movies.csv called *release_year* in order for the user to be able to filter by year within the web-app.
3. I decided to opt for *KNNWithMeans* algorithm, as after doing some research it is the best algorithm for this task, as I am doing user-based matching instead of item based. The algorithm finds the K nearest neighbors when comparing similarity matricies.
    a. I am using the *scikit-surprise* library. I have first used *GridSearchCV* in order to find the best hyperparameters. I ended up using Mean Squared Difference as the distance calculation between neighbors. Then, the minimum common movies that the users should have been 2 for the similarity matrices to be considered "neighbors". Further, I found that user-based collaborative filtering is better than item-based filtering, where users based will recommend movies outside of your genre, while item-based filtering will recommend movies that are similar to each other, thus preventing users from trying movies outside of their preferred genre.
    b. I decided to use *GridSearchCV* in order to fine tune my hyper parameters, as this is an optimizer library in order to use cross validation to find the best hyper parameters. Finally, I am doing 5-fold validation on my data, in order to find tune the weights, as this I found this would be the best method to avoid over-fitting and training on as much data as possible.
    c. I am using 5-fold cross validation in order to make sure it doesn't overfit the data by splitting the dataset into 5 chunks and iterating through them using *cross_validate()* which is a k-fold validation method from the *surprise library*.
    d. I have faced issues with deciding the possible hyper parameters to find the best ones using grid search. I wasn't sure what should be even the numbers for the minimum movies to be considered similar, as I tried out first with 20-30 and the precision was not good. Finally, after using grid search through all the different possibilities, I ended up using 2.

4. After running 5-fold validation, *surprise* calculates automatically the precision of the model when running *cross_validate()* by using the Root Mean Squared Error between the predicted rating and the actual rating. As such, here are the results I got from my most recent iteration:

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.9811	0.9683	0.9863	0.9701	0.9823	0.9823	0.0114
Fit time	0.01	0.02	0.01	0.01	0.02	0.02	0.00
Test time	0.06	0.07	0.06	0.06	0.06	0.06	0.00

5. The next steps are setting up the user interface using Flask and being able to merge new user reviews with the current dataset, as well as being able to properly display the top recommended movies to the user. Currently, with such precision, the project seems accomplishable. After that, if I have the time, I will try to modify it all to the MovieLens32M dataset, as it has much more data. I plan on possibly finding a dataset with the posters associated with each movie, in order to be able to display a picture to the user.