MAIS 202

Andrey Ambartsumov

Deliverable 3

Final Training Results

Since the last deliverable, I tried to re-train the model using the MovieLens32M dataset, which instead of 100k datapoints contains 32M. After experimenting with various techniques such as batch processing or pre-polishing the data before the computations, but that I was unable to train the model on more than 1.6M Datapoints. However, once trained, I was able to asses the precision of the model using another set of 1.6M data points, and I calculated the precision using 2 methods: Mean Absolute Error and Root Mean Square Deviation, which gave the following precisions:

RMSE: 0.9781

MAE: 0.6763

Considering the ratings are out of 5 stars, this means that my model is off by a bit more than half a star, which is about 0.6763/5 = 13.5% which I consider a satisfactory precision for my model. However, this precision in reality gets smaller and smaller as a user leaves more and more reviews, as I will re-train the model with each provided rating the user leaves (or possibly on a daily basis, it's still TBD).

Final Demonstration Proposal:

I have decided to create a webapp using Flask, where each person could create a user, and upon a new account creation will leave about 10 ratings before getting any movie recommendations, as if not the results given by the model wouldn't mean anything. To accomplish that, I have created a MySQL database that will contain all the ratings, the movies and the users all on separate tables, and I will be able to perform all my operations using SQLAlchemy rather than using csv, which will make the process for efficient. Finally, I intend on using the TMDB Api, which is a free API that will allow me to get images of the posters as well as overviews of the movies that the model will recommend. For now, the entire thing is still in development, but I already got to set-up user authentication, (but without password hashing at the moment, I will do this in the next days) and I have made separate scripts in order to get the movie poster and movie descriptions based on the title and release year.

I have already experimented with Flask in the past, and I have already made multiple websites using MySQL, javascript and html, so I am comfortable with most technologies that I am using, for what I wasn't sure of, I brushed up on all the skills I needed using this youtube series: Before You Install Flask...Watch This! Flask Fridays #1. At this current stage, I now have to implement the actual recommendations for a user using my model, and then display them in an appropriate manner to the user. Currently the website looks like this: