Luka Milekic
PA02 Movies_2
url to codeclimate: https://codeclimate.com/github/lmilekic/cs105b_movies_2
url to github: https://github.com/lmilekic/cs105b_movies_2

My prediction algorithm is pretty poor, but very very fast. It simply predicts that the user will give their average rating. The reason it is so efficient is that the average rating is a field in user, which is updated as more movies are added to their watch list. It's biggest drawback is it's accuracy. It's mean error is 0.9 on the u1 test set, with a standard deviation of 0.45. This certainly isn't ideal, but I think it could be much worse.

The ml-100k has a lot more data on the users than just the ratings and it also includes the genre of each movie. I think a better (although slower) algorithm would be to find users most similar to the given user in predict and see if they rated movies of the similar genre similarly.

Using ruby's built in Time.now to see how long run_test takes on u1 (8000 users) it says no time at all. As does run_test on ua (90000 users). The only way I can get it to show any difference is put Time.now before creating the MovieData object as the start and put Time.now after creating the MovieTest object, finding the mean, standard deviation, and root mean squared error. That time difference is 1/100th of a second. So all in all I think my algorithm is plenty fast, just quite inaccurate.