# Project 3: Recommender System

Goal: To get familiar with collaborative filtering recommender systems

Description: Recommender systems are ubiquitous these days from music and movies to shopping and online reading recommendations. Lots of companies are looking for ways to customize and maximize their user's experience through machine learning algorithms. For this project, you will implement a collaborative filtering recommender system.

Data: To start with, you'll need a dataset. There are many recommender data sets available online – I used MovieLens in my sample implementation. A Google search will pop up lots of results, but here are a few bloggers who have collected together links to a variety of recommender system data sets: Shuai Zhang and Julian McCauley and Alexander Gude.

Algorithm: You should implement the collaborative filtering algorithm yourself, even if an implementation is available through a library. If you wish to extend the algorithm beyond basic collaborative filtering, you are free to do so (e.g. if your dataset contains feature data, you could choose to incorporate content-based approaches with your collaborative filtering). You can use libraries for tasks like importing the dataset, performing linear algebra operations and storing structured data. You MAY NOT use a library implementation of your algorithm or core operations of your algorithm. As long as your code runs in less than 15mins, I don't mind whether you incorporate efficiency measures like vectorization.

You should incorporate mean normalization to allow a user who has provided no ratings to receive reasonable recommendations.

Your hyper-parameters -- alpha, lambda, etc. – should be easy to modify and should be set to good values for your dataset in the code that you submit.

Your code should display a plot showing how the squared error rate of your algorithm changes per iteration.

**(There are a number of example implementations of most ML algorithms available online. It goes without saying that you cannot copy one of them.)**

Submission: Please include instructions for running. Your code should have the option to feed it an alternative training file so that I can test it with a simple file with well-defined answers. Feel free to specify constraints on the format of the training file in the readme file.