# Project Description

The project implements a scalable version of a Latent Factor based recommendation engine. The data used with the algorithm is the MovieLens: 20M dataset, however due to resource constraints, only the first 250000 records are used in the model, while the first 1M records are included in the file for testing.

## Files

* pre\_process.py: The script implements a data\_processing class, which extracts all the relevant information from the data needed to predict movies to users.
* engine.py: The script implements the recommender class, which is a latent factor recommendation engine using scipy sparse matrices with linear time complexity.
* data/movies.csv: csv file containing the movie names with the corresponding movie\_id.
* data/ratings.csv: csv file containing the movie, user and rating pairs in transaction format.

## Source

The implementation heavily relies on the following lectures:

* https://www.youtube.com/watch?v=E8aMcwmqsTg
* https://www.youtube.com/watch?v=E8aMcwmqsTg
* https://www.youtube.com/watch?v=GGWBMg0i9d4
* https://www.youtube.com/watch?v=HY3Csl52PfE

## Performance

The training happens in linear time (Number of lines : runtime in minutes).

* 1000 : 0.02
* 5000 : 0.12
* 10000 : 0.25
* 50000 : 1.20
* 100000 : 2.44
* 200000 : 4.68
* 300000 : 7.84
* 1000000 : 24.42

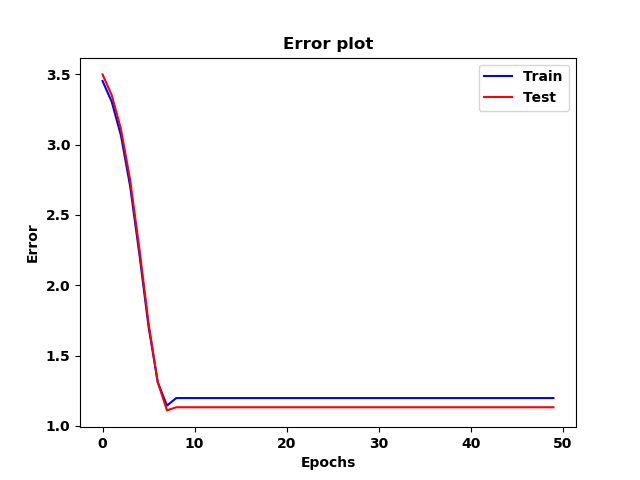
## Comments:

Almost every line of the code is commented. However just in case the input variables are the following in the same order as they are required in each class:

* data\_processor: csv file with the ratings, csv file with the movie-user-rating transactions, and number of records to use.
* recommender.predict: number of latent factors to use, learning rate, number of epochs to run, lambda regularization term, time it or not, verbose or not, plot the errors or not.
* recommender.recommend: user\_id for which the movie predictions are made.

## Known issues & improvement:

Once the minimum is found with the given learning rate, and the training error starts “exploding”, the learning rate is set to zero. However there is a bug in the logic, where it does not stop at the minimum, rather one step above:



Early stopping needs to be implemented, once the learning rate is set to zero.