Predictive Recommendation Engine
Using MovieLens Dataset

Final Project
Milestone #4

By:

Timur Zambalayev
Joshua Coffie

Harvard University

CS-109A, Fall 2016
Introduction to Data Science

For Milestone #4, we referenced the "The BellKor Solution to the Netflix Grand Prize" document by Yehuda Koren from 2009 (Link: Source). In this documentation, he references the use of baseline predictors, which can be used to gather additional data on users and movies through comparison. For instance, we can find that a user tends to consistently rate movies lower than the average reviews that those movies typically receive or that a particular movie is better than the average movie rating. Keeping this in mind, we're able to build a baseline of using means and then build upon those models by decoupling both the user and item parameters.

To create those baselines, we have that we must first model the mean user rating and the mean movie rating. Any model that we build needs to perform better than these baseline. To create these baselines, we use the equation: $b_{ui} = \mu + b_u + b_i + i$ where b_u and b_i "indicate the observed deviations of user u and item i." $\boldsymbol{\mu}$ is the overall average rating and b_{ui} is the baseline prediction for an unknown rating, which "accounts for the user and item effects." [2]

Before we begin building a model however, we need to determine the base μ for both movie rating and user reviews. The average movie rating of approximately 100k movie ratings was 3.29, with a standard deviation of .88.

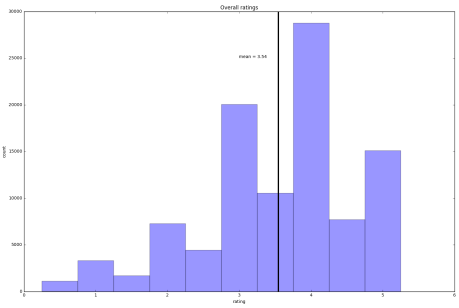

Figure 1: Histogram of overall ratings

This mean is the value that R^2 (which is a measure of how well our x variable(s) explain the y variable) will use to calculate how well our model performs for the model. Not surprisingly, the model that uses the mean movie rating scored a 0 for the training set and very nearly a 0 for the test set. However, when we use the mean movie rating as a predictor for our model, we perform better than the total mean model with an R^2 of .12.

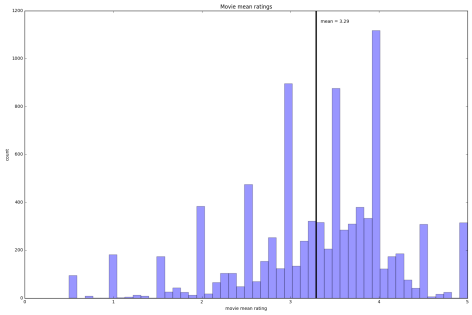

Figure 2: Histogram of movie ratings

This is still a rather low performing model and can be explained by the fact that of the ~9k movies, approximately 3k of them only have a single rating and 1,200 have only 2 ratings. This means that 47% of the movies in our data have 2 or less ratings. For this reason, we are compelled to examine another method in which to build our model – beginning with using mean user rating as a predictor.

Of our 671 unique reviewers in the dataset, the mean user rating was 3.66, with a max user mean rating of 4.95 and a minimum user mean rating of 1.33. The distribution of mean user ratings appears to be normally distributed with a standard deviation of .47.

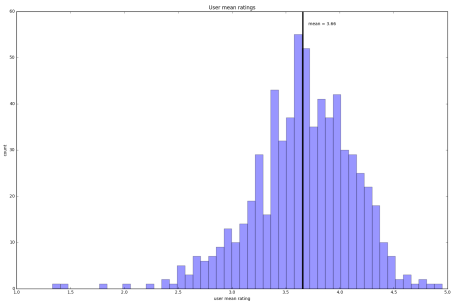

Figure 3: Histogram of user mean ratings

When we build a model using mean user ratings as a predictor, we see that our R^2 is .17, which is better performing than all of the other models so far.

For the third model in this milestone, we decided to decouple the item and user variables and then take into account both item (movie) and user effects, which is the same baseline predictor method used in the article write-up referenced above. This method uses a regularization parameter, lambda, which we tuned to provide the highest R^2 possible. Using this final baselining method, the score on the test set is .28, which is higher than the model that uses only the user mean review (R^2 =.17). This becomes the baseline in which any model we create for the final project must outperform.

NOTE: The remainder of our visualizations are available within our Python Jupyter Notebook on our repo. Please visit this site to view the extent of the analysis, outside of this summary submission. Thanks!

Github repo: https://github.com/starbuck10/CS109a_DataScience_UserRatings_Team_Project

Milestone #4 Folder:
https://github.com/starbuck10/CS109a_DataScience_UserRatings_Team_Project/tree/master/Milestone%20%234/MovieLens

Jupyter Notebook direct link:
https://github.com/starbuck10/CS109a_DataScience_UserRatings_Team_Project/blob/master/Milestone%20%234/MovieLens/Baseline_Models.ipynb