MovieLens_Project

Leena Elsharkawy

08/06/2020

1. Introduction

Data Science is a field of studies which combines scientific methods and statistical analysis to extract information from data. The extracted information can then be cleaned up and presented or can be utilized to form inferences. The utilization of data to construct models which can then be used for prediction is called machine learning. Machine learning is a subset of data science in which a "training set" is used to predict outcomes. This can be quite useful for filtering, recommending, and predicting.

In the Movie Lens Project, machine learning principals are used to predict movie ratings for nearly 10,000 movies based on user provided data. The Movie Lens data is downloaded from a URL and is then extracted, tidied, and divided into sets. The information initially downloaded is very messy, therefore needs to be "cleaned up" and restructured to make it easily readable. Once this is done the data is divided into **edx** and **validation** set, where the validation set is not used for any training but rather only to validate the models. The edx set is further divided into a **train** and **test** set to allow for cross validation for methods that require variable optimization. Once this set is done the various methods can be carried out. The methods are compared based on root mean square error (RMSE) which compares predicted rating to the true rating for the validation set. The RMSE takes the mean of the error squared for each data point in the set. The goal is to minimize the RMSE as much as possible to provide a more accurate prediction, hence the various methods and the combination of ideas. In this project, 7 different methods are carried out to inspect the effect of different variables on the prediction as well as to combine and regularize the variables for the given data. The resulting RMSE is calculated and documented into a table for comparison.

2. Methodology

2.1. Data Preparation

The first step to start running the project after running the required libraries is to obtain the data and prepare it for utilization. There were 5 steps necessary to prepare the data for the movie lens project and they are as follow:

1. Download
2. Extract
3. Tidy
4. Partition
5. Inspect

First the data that will be utilized was downloaded from the URL link. Once the files are downloaded the next step is to unzip the files, read, and extract the necessary data. After having obtained the data it is important to clean it up or tidy it up. This step makes the information useful to both the user and

the program. In this case, this was done by restructuring the data into a data frame as well as providing meaningful column names. Additionally, the variable class are assigned to each variable. Once the data is rearranged in a way that makes it easy to read, it is then partitioned.

Prior to partitioning the data, the **seed** was set to 1 to ensure that the results will remain the same regardless of the computer used. Then the data was split into the **edx** and **validation** set with a 90-10 split. After the data was partitioned, it was essential to ensure that the validation set had **movies** and **users** which were also in the edx set since that is the sample data. The reviews that did not fall under both the same users and movies were then removed from the validation set and re-added to the edx set. The edx data set is the training set. However since some models require cross validation which cannot be done on the validation set, the edx set was split further into a **train** and **test** set at a 80-20 split. Once again, it was essential to ensure the overlap between the data, and reviews which had users and/or movies which were in the test set but not the train set were moved into the train set.

Now, the data was ready for modeling. However, what kind of scientist works on data without inspecting and understanding it first. Thus, the data was inspected, as shown below. The number of observations in each set was examined, as well as all the column names/variables available. Next columns were further inspected individually for the edx data set. It showed that there were nearly 10,000 different movies in this set with almost 70,000 users. The genre of a movie could fall under several categories not just one. The most common rating was 4 with whole number ratings being more likely than their partials (ie 4 more common than 4.5). After getting a deep understanding for the data, the procedural modeling could commence.

```
[1] This is the structure for the edx set:

'data.frame':   9000055 obs. of  6 variables:
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
 $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
 $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 838984885 83898
 $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
 $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Adve

[1] Number of Movies= 10677

[1] Number of Users= 69878

# A tibble: 10 x 2
   rating frequency
    <dbl>     <int>
 1    4     2588430
 2    3     2121240
 3    5     1390114
 4    3.5    791624
 5    2      711422
 6    4.5    526736
 7    1      345679
 8    2.5    333010
 9    1.5    106426
10    0.5     85374
```

2.2. Modeling

The first and most important step that was used for every method was the **RMSE** function. Thus, the function was introduced and defined in the program as the square root of the mean of the true rating minus

the predicted rating squared. Next a data frame called **rmse_df** (which would later be converted to a table named **RMSE_results**) was made to store all RMSE information for both the edx and validation set. The two columns are there to ensure that the data was not over trained (the edx data set has a ridiculously lower error while the validation set is significantly high).

The 7 models that were carried out could be divided into three groups, average, specific effect, and regularization. The average stands alone as it is simply just the average of ratings for all movies in the edx data set applied to all movies in the validation set. The specific effects examines the effect of movie and/or user on the rating in reference to the average. The regularization group take into account the size of data points gathered for a movie and/or user and uses that to predict to the average. The 7 methods in order are average, movie specific effect, user specific effect, movie and user specific effect, regularization of movie effect, regularization of user effect, and regularization of movie and user effect. The average was done first to show the large variation in movie ratings and that one number cannot predict due to other bias. Next, the movie and user effects were carried out individually as to see how each of them effects the RMSE independently of the other. This confirmed the hypothesis that there are movie and user bias, as some movies are better than others and some users are more likely to give a good rating rather than a bad one. After confirmation, logically the two bias were combined in one algorithm for prediction. Furthermore, another bias remained; some movies have many reviews while some have just 1 thus skewing the data. The same thing exists for users. To account for this, less weight must be given to ratings with fewer reviews. This method is called regularization, it was first carried out for the movie specific effect to provide less weight for bias when the movie had fewer ratings. Then the same thing was done to the user bias. Then they were combined together.

While performing the regularization, a specific number, lambda, has to be determined to be added to the size in the denominator in order to calculate the bias. To find the optimal value for lambda cross validation was carried out. Cross validation (cv) cannot be carried out in the same set that we are training otherwise the lambda value is likely to be 0 as that will provide us with the best prediction. Furthermore, to use cv with the validation set would be wildly inappropriate because that set only exists to check the work that has been done. Thus, this is why the edx set was split into a train and test set. The **train set** was used to train the algorithm of the bias(es) that exist, then the **test set** was used for cv to find the optimal value for lambda for each scenario. For the movie effect, we called the lambda **li** and for the user effect we called it **lu**. For the final method, since the movie and user effect were both accounted for cv was carried out again with a single lambda, **l**, to find the optimal point. Once cross validation is carried out, the lambda value found was then used to predict the ratings for both the edx and validation set and the RMSE was documented.

2.3. Optimization and Visualization

During this project, space was always an issue due to the size of the data set. To optimize size, once the information required was extracted from the downloaded files, the original files were then removed. This was also the case for temporary vectors and sets that were used to find something else. With a data set of the size, it is important to remove all unnecessary files as they take up part of the allocated space for RStudios to run. Furthermore, as seen by the supplementary code, after each method any data that is no longer of use is instantly remove to optimize space and performance. Although some might argue that it would be nice to keep the information from each method for longer, it is important to notice that the space restriction would not allow us to run the entire code if everything was to remain.

A great way to understand the data and the results is through visualization. Thus, visual tools were used to find the optimal lambda values, done through plot. The plots obtained for each lambda not only provide the optimal point but also show the trend and the effect of the constant in the calculation of RMSE. This tool is also beneficial with tracking and storing the results of each method. Although not another plot, the RMSE_results table provides a great visual for further analysis and understanding. The amount of visuals used in this project are limited due to data size. Although a plot with predicted vs actual ratings would be great, with almost 1 million points in the validation set it would be incredibly difficult to decipher any information from that. With a large data set, visualization of anything in general become more difficult as is seen if we were to try and print the edx data frame without using the str function to capture the essence.

3. Results and Discussion

When carrying out the various methods mentioned previously, it can be inferred that bias is something that definitely exists in this data set. Although we limit the bias to that of movieId and userId, there may still be more biases present. This can be identified via the RMSE values calculated. The **RMSE_results** table indicates that using the average to predict the ratings for all movies has high error. This can be attributed to the variation in movie ratings.

```
# A tibble: 7 x 3
  method                                RMSE_edx RMSE_validation
  <fct>                                    <dbl>           <dbl>
1 Average Rating                          1.0603          1.0612
2 Movie Specific Effect                   0.94235         0.94391
3 User Specific Effect                    0.97001         0.97834
4 Movie and User Effect                   0.85670         0.86535
5 Regularization for Movie Effect         0.94240         0.94385
6 Regularization for User Effect          0.97039         0.97794
7 Regularization for Movie and User Effect 0.85701        0.86482
```

Using the average as a standalone prediction is highly inaccurate because there are many factors which effect the rating of a movie. Firstly, it can be easily assumed that some movies are good while others are bad thus using the average 3.51247 can be inaccurate. Thus when conducting the next two methods, **movie specific effect** and **user specific effect**, we see a decline in RMSE value. The RMSE drops from 1.0612 to 0.94391 and 0.97834 for movie and user effects respectively. From this it can be inferred that the movie specific effect plays a larger role in controlling the rating than the user itself. Nonetheless, both provide a bias that needs to be accounted for. When combing the two effects the RMSE drops to 0.86535 which is a 0.19585 drop from the initial method. This decline further justifies the assumption that both effects need to be accounted for.

The next three methods cover regularization. To perform regularization a constant, lambda, must be optimized to minimize the RMSE. The lambda value which is added is used to minimize the effect of smaller groups (ie movies/users with few reviews). By minimizing the effect of smaller groups, we are essentially weighing the information differently based on the sizes. For each of the three methods cross validation was done to optimize the lambda values as discussed previously. The resulting plots can be seen in figures 1-3. In the first plot the **li**, lambda for movie specific effect, is plotted against RMSE. The value which minimizes the error is a lambda of 2.5. Please note that the lambda calculation was done for values from 0-10 every 0.25. The next plot provides the lambda for user effect, **lu**, which reaches the minimum RMSE for the test set at 5.5. The final lambda plot reaches its minimum error at **l** equals 4.75. Thus, showing that regularization has more of an effect on the user than the movie itself.

In theory, regularization should have a huge impact on the bias because some movies have very few reviews while others have thousands. The same holds true for users as well. In practice, however, it is evident that although regularization does help to enhance prediction it does not make a significant change. When regularized the errors dropped by 5.65328×10^{-5}, 4.00078×10^{-4}, and 5.28697×10^{-4} for movie, user, and combined effects respectively. There is a drop that can evidently be see in the figure above, however, is it significant enough? From this figure it can be seen that the method which provides the lowest RMSE is the **regularization of both movie and user specific effect**, at 0.86482.

4. Conclusion

The Movie Lens Project was carried out with the objective of predicting movie ratings for a large data set by reducing the RMSE. Throughout the project various methods were used to predict the ratings for a validation set. First the data was obtained, cleaned, and split to make ready for machine learning methods. The methods included averaging, accounting for bias, and regularization. Results from 7 methods were

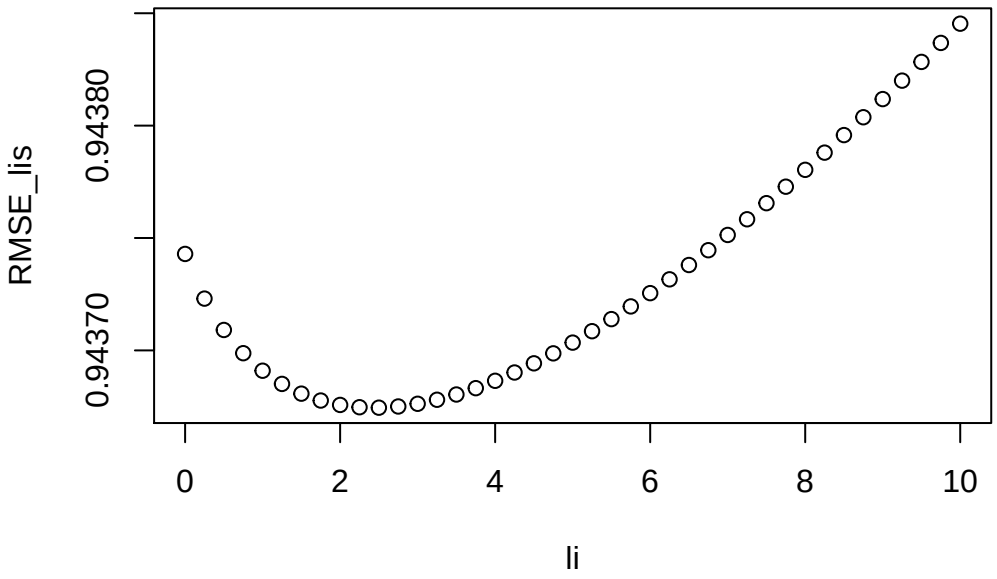

Figure 1: Lambda vs. RMSE

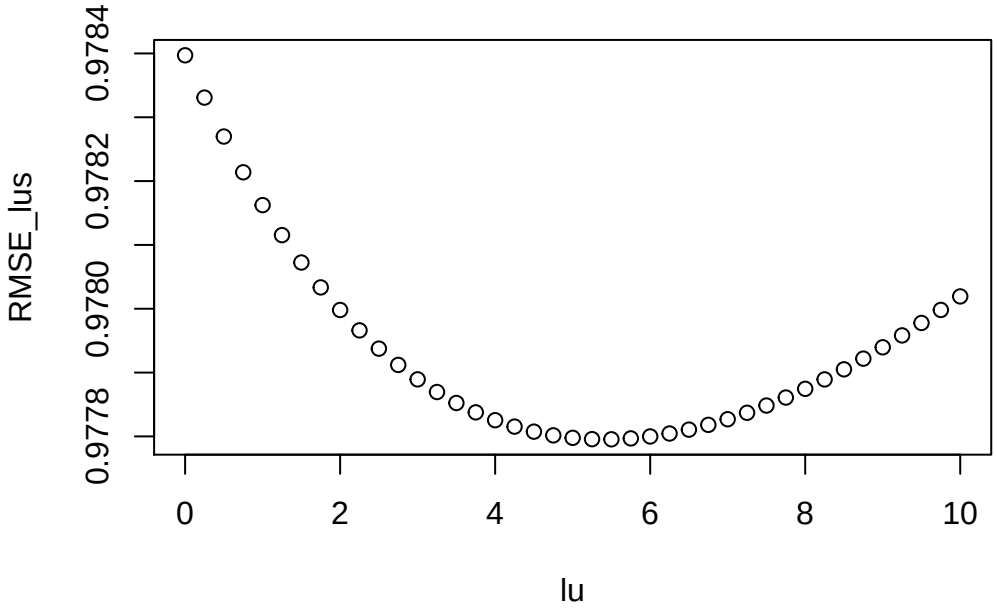

Figure 2: Lambda vs. RMSE

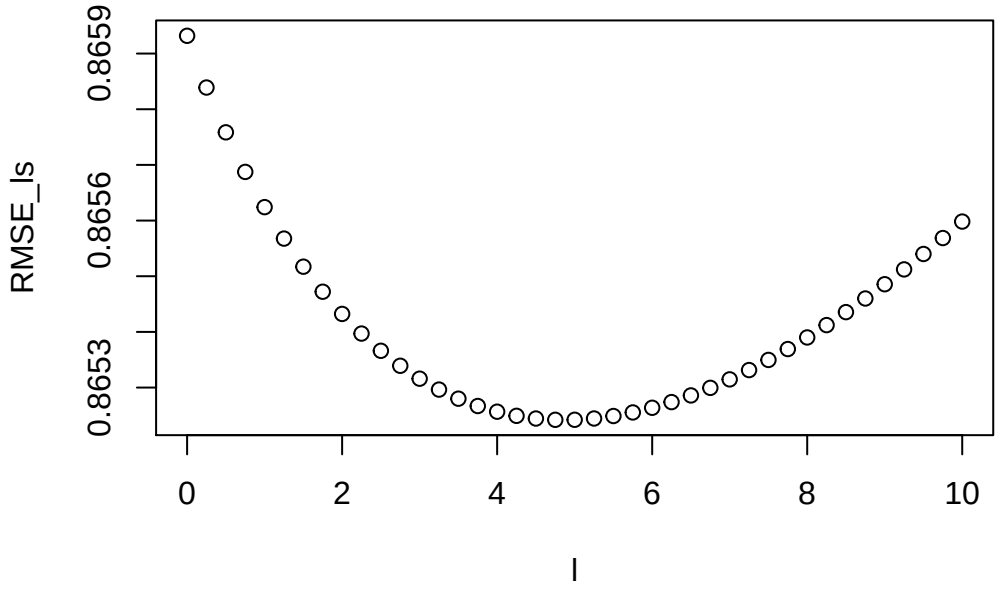

Figure 3: Lambda vs. RMSE

examined and showed that biases (movie and user) play a large role in the effect of the ratings. Furthermore, accounting for size per groups also seemed to decrease the error in estimation. It was found that the method providing the lowest RMSE of 0.86482 was the regularization for movie and user specific effects in reference to the overall average rating. Due to the large data size, not only was continuous removal of files necessary, but it also limited the methods utilized for the prediction. If the data provided was smaller or the computer used had larger capabilities, other machine learning methods could be applied such as knn, CART, and random forest. The methods used all use linear modeling, while the relationship might be more accurately predicted using binomial (glm). This is unknown because these methods were unable to be used due to data size. For future work on this dataset, I would recommend having a cut off to both limit the data points and the skewing of the data. By removing movies which have less than 10% of the mean number of reviews, and doing the same for the users. As a result, if a movie has a few vastly different ratings and is removed from the data, the error in the validation set will hopefully decrease.