Movielens report

Deeksha RV

1/4/2022

Contents

1 Abstract:

The following report entails details on a data science project that I took up as part of the *HarvardX PH125.9x Data Science: Capstone* final course, about creating a machine learning algorithm for a movie recommendation system. The algorithm is expected to have an RMSE value of 0.86490 or lesser, implying good accuracy. The creation of this algorithm was originally tasked to data scientists across the globe, as part of the Netflix Challenge 2006. The dataset used for this analysis was the 10M version of the Movielens dataset, the links of which were accessed directly from the edx course. The given dataset was split into 2 sub-datasets called the **edx** and **validation** sets. The **edx** set was further sub-divided into the test and training sets and the algorithm was built on these datasets accordingly. Finally, the final model was tested on the validation dataset and its RMSE was recorded.

2 Introduction:

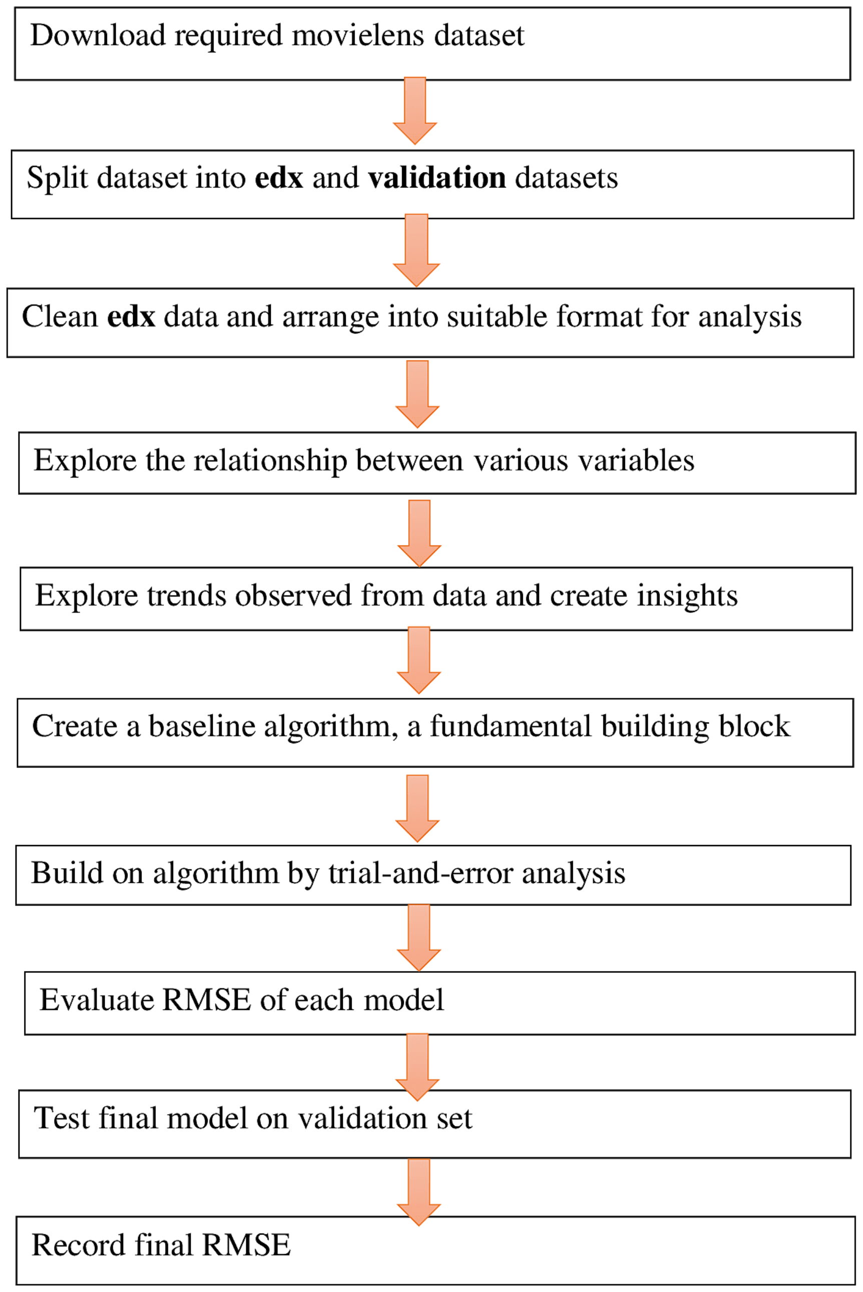
A Movie recommendation System is a filtration program whose prime goal is to predict the “rating” or “preference” of a user towards a movie. Therefore the main focus of a recommendation system is to filter and predict only those movies which a user would prefer given some data about the user himself/herself.
In general, recommendation systems use Machine Learning to understand a user’s preferences. Items are ranked according to their relevancy, and the most relevant ones are shown to the user. A recommendation system continuously improves itself as the user progressively makes decisions on various domains. The same can be applied for a movie recommendation system, which becomes progressively better when users choose movies over a broad range, and make a lot of ratings. In today’s world, lots of industries use recommendation systems to create business-oriented decisions. To name a few, we have *Netflix*, the popular streaming service for movies and other shows of varied genres and languages, *Amazon*, another widely used application that helps users make suitable choices based on their shopping history, and other e-commerce companies.

Every algorithm that is built needs to be tested on some data before it can be used commercially. A popular method for evaluating the accuracy of an algorithm is the **Root Mean Square Method** or simply, **RMSE**. In R, a function called RMSE() is provided in the *caret* package that takes true values and predicted values as input, and generates an output score, whose value determines the accuracy of the algorithm. RMSE is commonly used in supervised learning applications.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (true_i - pred\hat{icted}_i)^2}{N}}$$

where N is the total number of observations

3 Workflow of Project


Download required movielens dataset
Split dataset into **edx** and **validation** datasets
Clean **edx** data and arrange into suitable format for analysis
Explore the relationship between various variables
Explore trends observed from data and create insights
Create a baseline algorithm, a fundamental building block
Build on algorithm by trial-and-error analysis
Evaluate RMSE of each model
Test final model on validation set
Record final RMSE

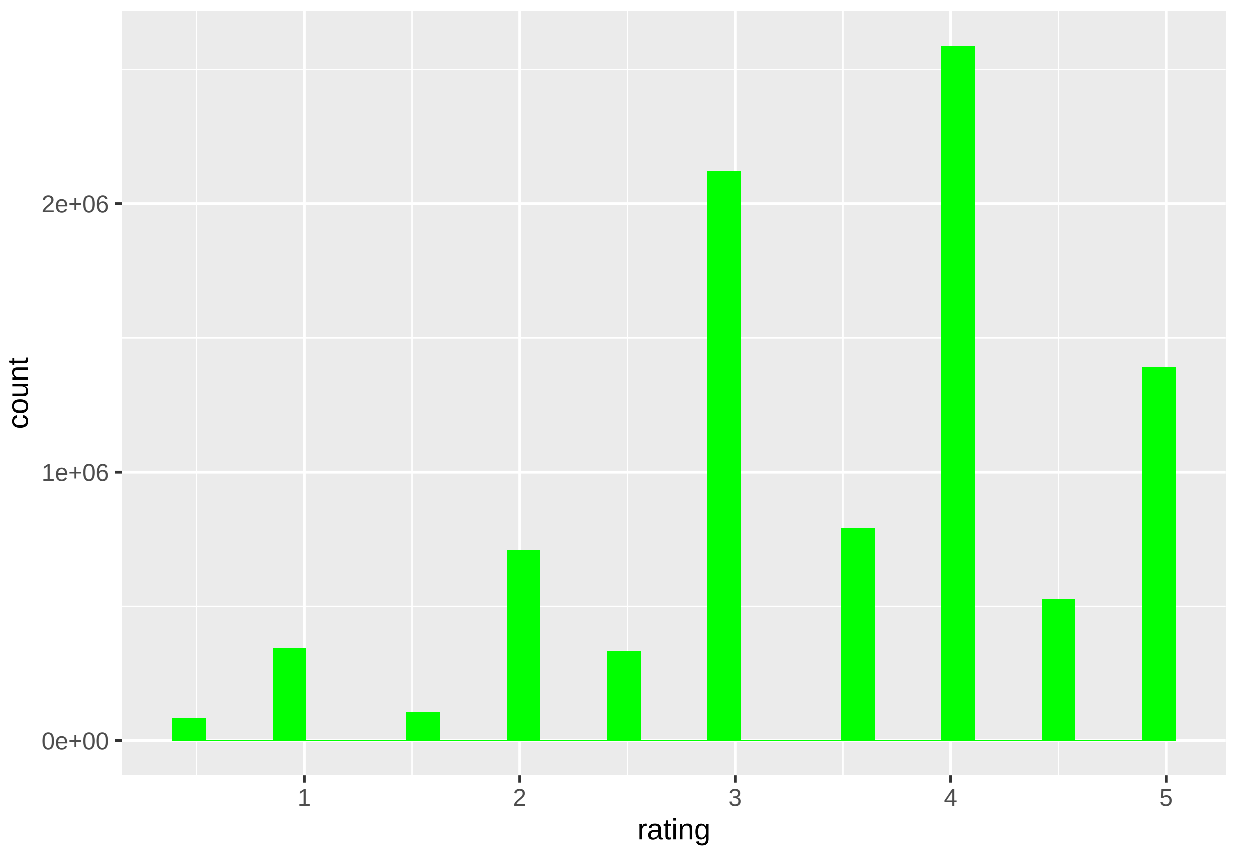
4 Data Analysis and Cleaning

To start with, lets view the dataset that we will primarily work with:

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

The above shown table shows the first 6 rows of the edx dataset, along with all the variables.

We will now play around with the variables, to see how each relates to the other. We start with finding out how rating is spread across the dataset:



From the given histogram, we can observe that 4 is the most popular rating given to movies in the edx dataset. Also, comparing the type of ratings, we observe that users prefer to give movies more whole-star ratings than half-star ratings.

Next, we see that the genres describing each movie are a mix; for example, the first movie in the above shown edx dataset table is “Boomerang(1992)”, which has been put under a mixed genre of “Comedy” and “Romance”. Similarly, some other movies such as “Star Trek: Generations(1994)” fall under a wider variety of genres such as “Action”, “Adventure”, “Sci-Fi” as well as “Drama”. We now find out all the unique genres that describe every movie:

```
##  [1] "Comedy"             "Romance"            "Action"
##  [4] "Crime"              "Thriller"           "Drama"
##  [7] "Sci-Fi"             "Adventure"          "Children"
## [10] "Fantasy"            "War"                "Animation"
## [13] "Musical"            "Western"            "Mystery"
## [16] "Film-Noir"          "Horror"             "Documentary"
## [19] "IMAX"               "(no genres listed)"
```

As we can see, 20 unique genres exist in total.

We now find out how each unique genre has been rated. The following list contains how many ratings each genre has:

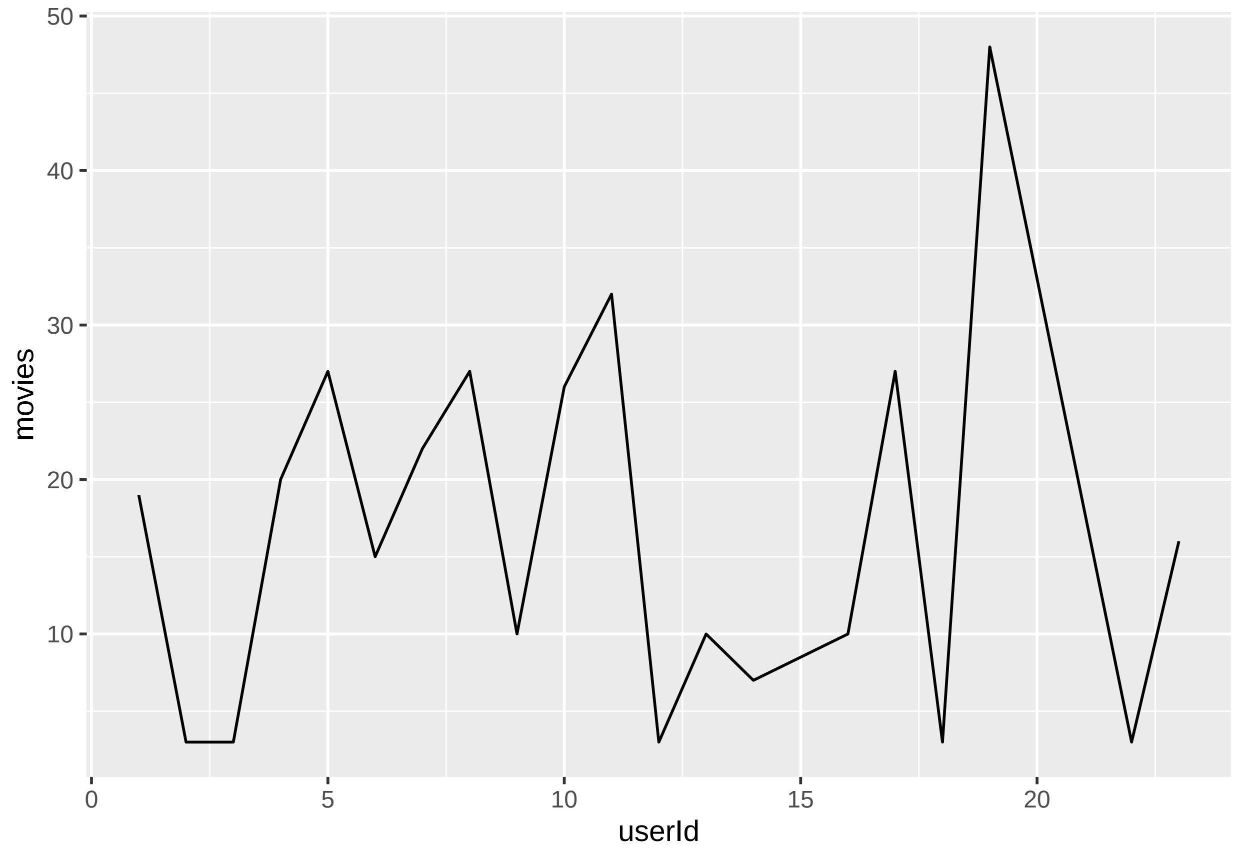
```
##             Comedy            Romance             Action              Crime
##            3540930            1712100            2560545            1327715
##           Thriller              Drama             Sci-Fi          Adventure
##            2325899            3910127            1341183            1908892
##           Children            Fantasy                War          Animation
##             737994             925637             511147             467168
##            Musical            Western            Mystery          Film-Noir
##             433080             189394             568332             118541
##             Horror        Documentary               IMAX (no genres listed)
##             691485              93066               8181                  7
```

As it can be seen, “Drama” and “Comedy” have been rated the most, whereas “IMAX” has been rated the least, along with a few movies for which no genre has been listed.

Next we would like to know a bit about the users. There are totally 69,878 users who have rated movies in the edx dataset. Out of this, we can find out how many movies each user has given say, 5 stars to:

```
## # A tibble: 20 x 2
##    userId movies
##     <int>  <int>
##  1      1     19
##  2      2      3
##  3      3      3
##  4      4     20
##  5      5     27
##  6      6     15
##  7      7     22
##  8      8     27
##  9      9     10
## 10     10     26
## 11     11     32
## 12     12      3
## 13     13     10
## 14     14      7
## 15     16     10
## 16     17     27
## 17     18      3
## 18     19     48
## 19     22      3
## 20     23     16
```

We see from an example of the first 20 users, that each user's perspective of rating 5 stars to movies is different. To visualize this better:



As we can see, people's opinions on what movie deserves a perfect rating, is very diverse. Hence our algorithm must be structured in such a way that it takes every user's choices into account.

5 Building the Algorithm

In order to build the algorithm, the first step is to split our existing data into 2 subsets; the edx *train* set and the edx *test* set. We will train our algorithm on the *train* set and test its accuracy on the *test* set.

To start with, we create a baseline model; a simple model that predicts same rating for all movies regardless of user. Model is given in the form of a regression equation, taking Y as dependant variable, X as independent variable, β (beta) as parameter and ϵ (epsilon) as error:
Y = f(X,β) + ϵ
We could further simplify this equation into:
Y = μ + ϵ
where Y is predicted rating and μ is actual rating, given by average of all ratings in edx dataset. We get μ to be 3.512 approximately.

We can now evaluate this model, by finding its RMSE after calling upon the *caret* package. This is the result we get:

Model	RMSE
Average rating only	1.060247

This is just a start. As we further build the model, we will try to implement parameters through trial-and-error method to lower this RMSE value.

We will now take the more obvious parameter from the edx dataset; the *movieId*, and see how including it, will affect the accuracy of this model:

Model	RMSE
Average rating only	1.0602474
Movie model	0.9441335

As we can see, the RMSE has tremendously lowered, implying a positive effect of the movieId on the model. This means that a user's rating of a movie depends on what movie it is; such as its title.

We can further improve the model by bringing another parameter, such as the *userId*:

Model	RMSE
Average rating only	1.0602474
Movie model	0.9441335
Movie + User	0.8696665

So what does this new lowered value of RMSE tell us ? It shows us that the *userId* has also had a positive effect on the model. The rating predicted by this model depends on which user has given what kind of ratings. But it can also be observed from the difference in RMSE values of the first and second models (0.116), and second and third respectively (0.07), that the *movieID* has a greater positive effect, than does the *userId*.

We would now like to see if the type of genre, has an effect on the rating predicted by our model. For this, we include the *Genres* effect into the model and calculate the its RMSE accordingly.

Model	RMSE
Average rating only	1.0602474
Movie model	0.9441335
Movie + User	0.8696665
Movie + User + Genres	0.8693186

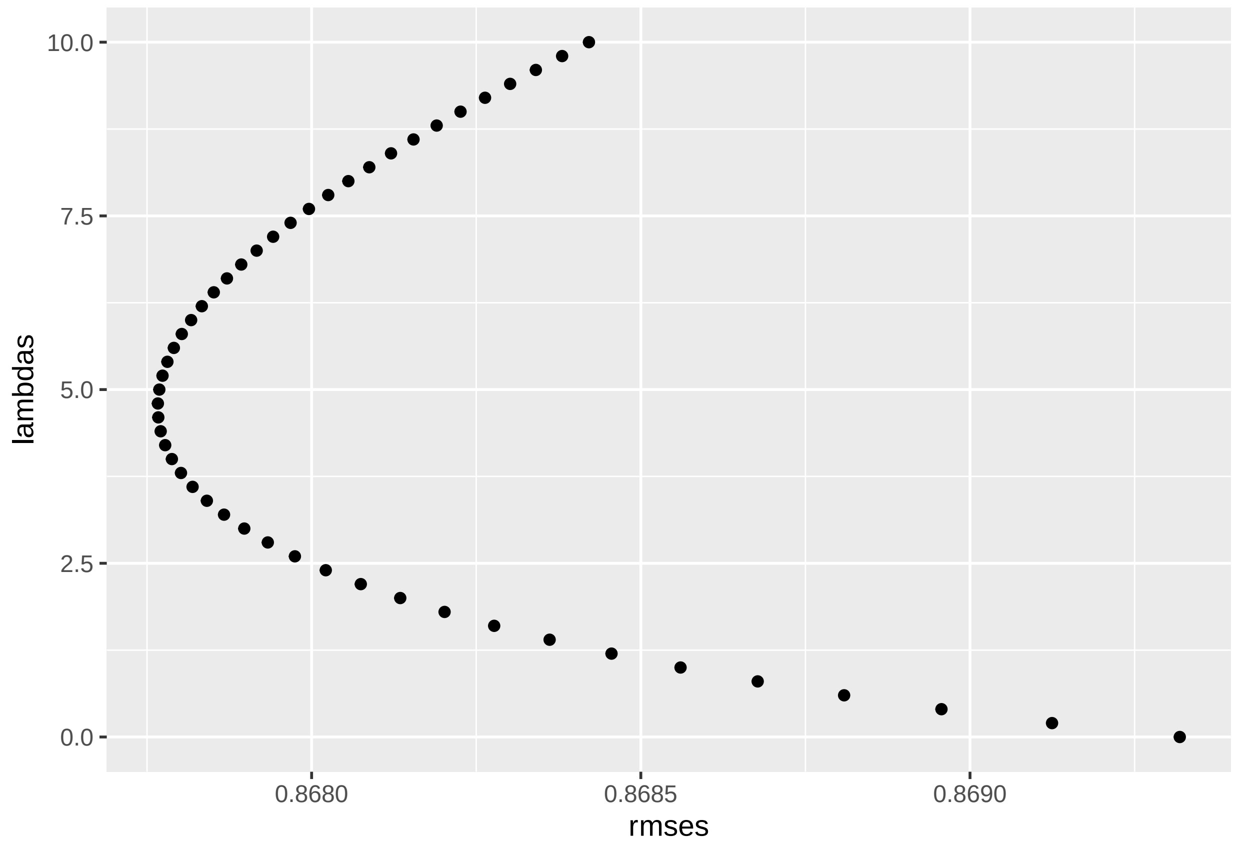
Not surprisingly, the *Genres* parameter also has a direct positive effect on our model, although it it worthwhile to note that, the reduction in RMSE value is significantly lower for the 4th model, as compared to the 3rd or second. This could only imply that while including the *Genres* parameter may have reduced our RMSE thereby improving the model, the improvement is marginally less. Hence to further move towards our targetted RMSE (0.86490), we will use a bit of regularization using the existing tested parameters (*movieId*, *userId*, *Genres*) and find the RMSE of the regularized model.

In order to create a regularized model, we first need a tuning parameter. Tuning is the process of maximizing a model's performance without overfitting or creating too high of a variance. Therefore this tuning parameter, represented by λ, will shrink data points towards the mean, thus reducing penalty effects on our model. We will take a random sample of numbers from 0 to 10, spaced out by intervals of 0.2, and find our optimized λ which we can use for regularization.

Model	RMSE
Average rating only	1.0602474
Movie model	0.9441335
Movie + User	0.8696665
Movie + User + Genres	0.8693186
Regularized model	0.8677665

Model	RMSE
Average rating only	1.0602474
Movie model	0.9441335
Movie + User	0.8696665
Movie + User + Genres	0.8693186
Regularized model	0.8677665

This is how the RMSE value changes with respect to our lambda values:



From regularization, we obtained the optimal lambda corresponding to the smallest RMSE value (0.86809) and found it to be 4.75. This value is our tuning parameter, which we will use for further improving our model.

So far we had built our model on just a portion of the edx dataset; the *train* set. We will now train our regularized model on the entire edx set, to see how it performs.

We obtain **edx** specific parameters; our optimized parameters, and use it to test the final accuracy of our model, using the final hold-out set, also called the **validation** set, which comprises 10% of the Movielens dataset.

6 Final Results

Model	RMSE
Average rating only	1.0602474
Movie model	0.9441335
Movie + User	0.8696665
Movie + User + Genres	0.8693186
Regularized model	0.8677665
Final Model	0.8644510

Our final RMSE comes to 0.8644510, which is a little less than the targetted RMSE (0.86490), thus concluding our model's targeted accuracy.

7 Conclusions

The built movie recommendation system used 3 parameters directly from the **edx** dataset, to build the algorithm; namely, *movieId*, *userId* and *Genres*. An additional parameter called the *timestamp*, which gives information about the date of release of the movie, was also initially fit into the model, but it increased the RMSE to over 1.06, thus showing a negative impact on our model. Hence this parameter was removed. It can be observed from the final results table that the *movieId* individually seems to have the highest impact on our model, followed by *userId*, followed by *Genres*. While initially building the algorithm, it was observed that taking *movieId* as the first parameter and later adding *userId* and *Genres* seemed to have the highest positive impact on the model, as other combinations of the same parameters did not show significant reduction in RMSE. Regularization only marginally improved our model. The targetted RMSE could be achieved only when the regularized model was fitted on the entire **edx** dataset. This model can further be improved by fitting different algorithms and trying out other various machine learning methods and techniques; but the author would like to conclude this report with the obtained final model to stay on par with the requirements of this project.

8 References

[1] https://www.edx.org/course/data-science-machine-learning

[2] http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/

[3] https://www.analyticsvidhya.com/blog/2020/11/create-your-own-movie-movie-recommendation-system/#h2_4

[4] https://www.geeksforgeeks.org/root-mean-square-error-in-r-programming/

[5] https://rpruim.github.io/s341/S19/from-class/MathinRmd.html