MovieLens Capstone Project

Meghan Patterson

6/23/2020

Abstract

Multiple machine learning techniques were implemented in this project in order to construct an algorithm intended to predict the rating of a movie in the MovieLens dataset. Many models were created in order to find the model with the lowest root mean squared estimate (RMSE) value. Some of these models include bias terms and regularization terms of the movie, user and/or genre predictors. The model with the lowest RMSE was the regularized movie, user and genre model. When this model was tested on the validation dataset with the entire edx dataset, the final RMSE was found to be 0.8630 - making 0.8630 the RMSE of the project.

Contents

1 Introduction

Recommendation systems are common and useful applications in machine learning that allow for users to receive relevant and pertinent suggestions based off of inputs specific to that user. These recommender systems can be found within many platforms, such as on YouTube and the Instagram

Explore page. When a user utilizes the recommender system, the company will receive new data and able to further improve the algorithm.

In this project, the MovieLens dataset that was provided by the instructor is explored in order to construct a recommender system with an algorithm that can easily predict ratings based off of a selection of inputs. The 10M MovieLens dataset is comprised of 9000055 rows, 6 columns, 10677 movies, and 69878 users. The most commons ratings in this dataset are: 4, 3, 5, 3.5, 2 (in order from most to least), and the average rating is 3.52.

The edx dataset (as well as the validation dataset) was recreated with separate rows in order to easily incorporate the genre term into the algorithm. In order to build a successful algorithm, many machine learning techniques were utilized. Some of these techniques include linear regression, bias, and regularization. Linear regression is a form of supervised machine learning where the model assumes a linear relationship between the input variable(s) and the single output variable. Bias terms help account for potential oversimplification of the model and reduce the potential of predictions becoming too inaccurate. Regularization terms help regularize the coefficient estimates towards zero in order to help prevent the risk of overfitting. These three techniques are incorporated in the one or more models. Because the dataset was so large and unable to perform with the machine learning functions in RStudio, the calculations were performed manually.

The best model will be chosen based off of the lowest root mean squared estimate (RMSE). The RMSE is the standard deviation of the residuals and measures how spread out the residuals are. A lower RMSE value indicates that the model can relatively predict data accurately, while a high RMSE value indicates the opposite. The various machine learning techniques are incorporated into the ideal model for this project in order to find the lowest RMSE.

2 Methods

2.1 Exploratory Data Analysis

With exploratory data analysis techniques, new insights were gained regarding the dataset at hand. The tibble of the edx dataset shown below displays a quick glimpse of the dataset that is being used.

```
edx %>% as_tibble()
```

```
## # A tibble: 23,369,607 x 6
##    userId movieId rating timestamp title                genres
##     <int>   <dbl>  <dbl>     <int> <chr>                <chr>
##  1      1     122      5 838985046 Boomerang (1992)     Comedy
##  2      1     122      5 838985046 Boomerang (1992)     Romance
##  3      1     185      5 838983525 Net, The (1995)      Action
##  4      1     185      5 838983525 Net, The (1995)      Crime
##  5      1     185      5 838983525 Net, The (1995)      Thriller
##  6      1     231      5 838983392 Dumb & Dumber (1994) Comedy
##  7      1     292      5 838983421 Outbreak (1995)      Action
##  8      1     292      5 838983421 Outbreak (1995)      Drama
##  9      1     292      5 838983421 Outbreak (1995)      Sci-Fi
## 10      1     292      5 838983421 Outbreak (1995)      Thriller
## # ... with 23,369,597 more rows
```

The tibble above indicates that there can be multiple ratings per movieId. Figure 1 below showcases the distribution of the number of ratings per movieId, while Figure 2 showcases the distribution of the number of ratings per userId. In Figure 1, it is illustrated that some movies have a higher count of ratings compared to others. Some movies have only a handful of ratings, while others have a significantly large number of ratings. This disparity will be taken into account when creating the model. In Figure 2, it is shown that there is a significant difference between the number of ratings inputted per user. The plot shows that there are some users who rate often, while some users only rate a handful of times. This disparity will also be taken into account when creating the model.

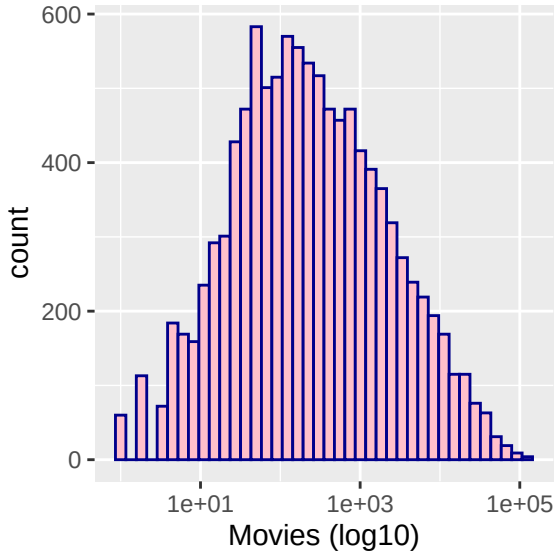

Figure 1: Distribution of the Number of Ratings per MovieId

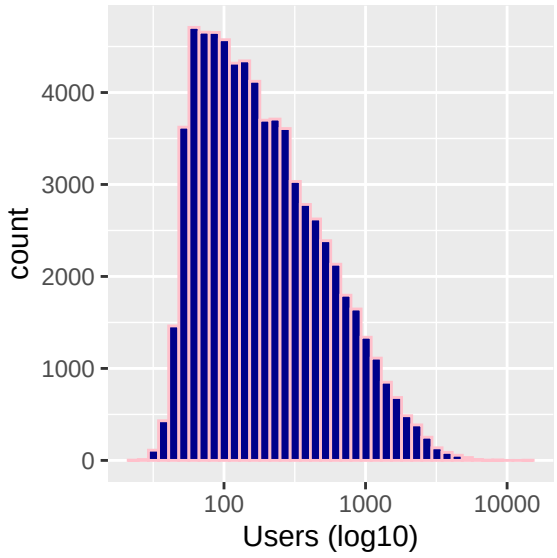

Figure 2: Distribution of the Number of Ratings per UserId

The skewed histogram in Figure 1 demonstrates that some movies have higher rating counts than others. This likely affects the average rating per all of the movies. Figure 3 below showcases the average rating given per movie. With that, it can be seen that the average rating is approximately

3.52, and additionally, the most movie ratings fall between the ratings 3.0-4.0. Figure 4 illustrates the average rating inputted by the user.

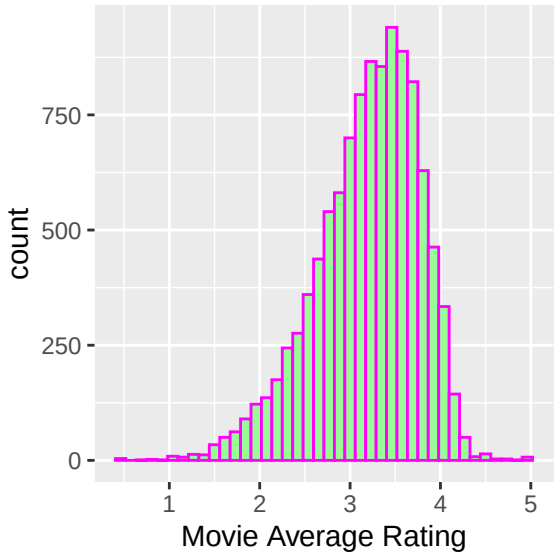

Figure 3: Distribution of the Average Ratings per Movies

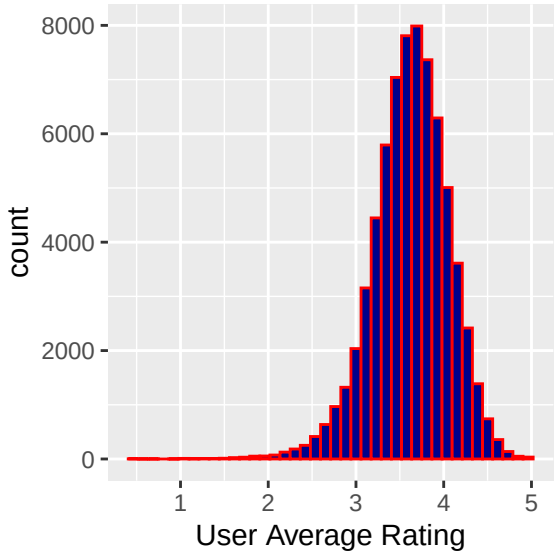

Figure 4: Distribution of the Average Ratings per User

The tibble below shows a glimpse of the number of ratings in each genre. The values in this tibble indicate that some genres have significantly more ratings than others. With that, it can be suspected that this predictor is skewed in the dataset. This will be taken into account when constructing the model.

```
## # A tibble: 20 x 2
##    genres                   n
##    <chr>                <int>
##  1 (no genres listed)       6
```

```
##  2 Action             2560649
##  3 Adventure          1908692
##  4 Animation           467220
##  5 Children            737851
##  6 Comedy             3541284
##  7 Crime              1326917
##  8 Documentary          93252
##  9 Drama              3909401
## 10 Fantasy             925624
## 11 Film-Noir           118394
## 12 Horror              691407
## 13 IMAX                  8190
## 14 Musical             432960
## 15 Mystery             567865
## 16 Romance            1712232
## 17 Sci-Fi             1341750
## 18 Thriller           2325349
## 19 War                 511330
## 20 Western             189234
```

2.2 Model Preparation

2.2.1 RMSE Function

Because the machine learning functions from various packages were inoperable on the provided dataset in RStudio, the RMSE function was crafted manually in order to account for this issue. The RMSE function used for this project is as followed:

```
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

2.3 Models

2.3.1 Average Model

The simplest model possible is the average model - this model assumes that the rating is the same for all movies, regardless of any other factors (such as the user). This model takes the form of: $Y_{u,i} = \mu + \epsilon_{u,i}$ where $\epsilon_{u,i}$ is the independent error and μ is noted as the "true" rating. The model in RStudio is created as followed:

```
mu <- mean(training_set$rating)
mu
```

```
## [1] 3.52702
```

The RMSE for the model can be seen in Table 1 below.

Table 1: Average Model and Corresponding RMSE

Model	RMSE
Average	1.052611

The RMSE value being slightly over 1 for this model implies that the overall fit between the response and the predictors is not ideal. It can be concluded that there is room for improvement in the model, which can include bias terms implemented into the potential model.

2.3.2 Movie Bias Model

From the skewed histogram in Figure 1, it was shown that the number of ratings per movie can vary greatly. With that, it can be suspected that there is some bias with regards to this skewed distribution. When adding this bias to the model, the model takes the general form of: $Y_{u,i} = \mu + b_i + \epsilon_{u,i}$ where b_i is now the movie bias term. Applying this model to the dataset has the form of

```
movie_bias <- training_set %>% group_by(movieId) %>% summarize(b_i = mean(rating - mu))
pred_ratings_movie <- mu + test_set %>% left_join(movie_bias, by = 'movieId') %>% pull(b_i)
```

Additionally, Figure 5 belows the distribution of the movie bias within the dataset.

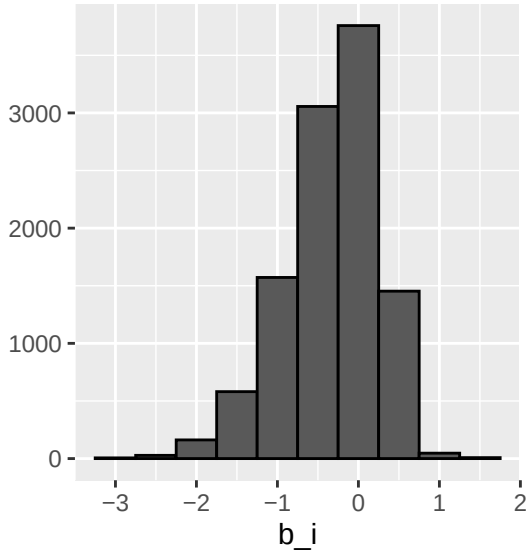

Figure 5: Distribution of Movie Bias within the dataset

The RMSE value for this model can be seen in Table 2 below:

Table 2: Movie Bias Model and Corresponding RMSE

Model	RMSE
Average	1.0526109
Movie Bias	0.9414901

The RMSE value of 0.9414 shows a significant improvement between the first model and this new model with the added movie bias. This can lead to the conclusion that the movie bias is prominent in the dataset and is important to the fit of the model.

2.3.3 User Bias Model

Figure 2 shows that there is some variability with regards to the ratings inputted by the user. Some users could possibly have given a rating when they were very unhappy or extremely happy, leading to some bias towards the rating that was given for that particular movie. Therefore, this bias must be tested in the model. When this bias is added to the model, the model takes the general form of $Y_{u,i} = \mu + b_i + b_u + \epsilon_{u,i}$ where b_u is the user bias term. Applying this model to the dataset in RStudio is as follows:

```
user_bias <- training_set %>%
  left_join(movie_bias, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu - b_i))

predicted_ratings_user <- test_set %>% left_join(movie_bias, by = 'movieId') %>% left_join(user_
```

Additionally, Figure 6 showcases the distribution of the user bias within the dataset.

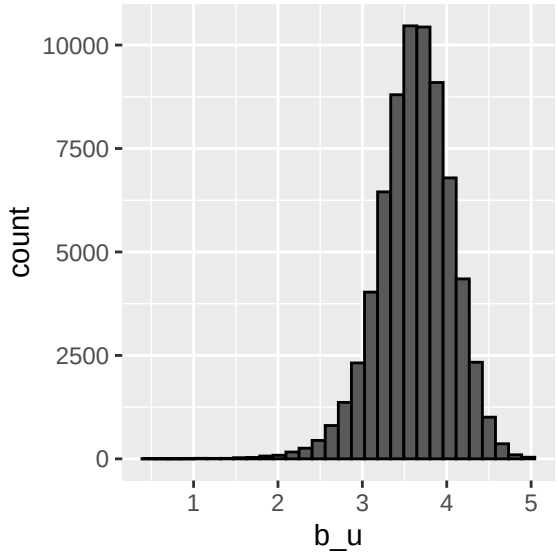

Figure 6: Distribution of User Bias

The RMSE value for this model can be seen in Table 3 below:

Table 3: Movie + User Bias Model and Corresponding RMSE

Model	RMSE
Average	1.0526109
Movie Bias	0.9414901
User Bias	0.8573267

The RMSE value of 0.8573 for this model shows an improvement from the previous model where only the average and movie bias were included. This improvement implies that the user bias does have a significant impact on the dataset, and thus, must be included in the model.

2.3.4 Genre Bias Model

The data shows that some genres have significantly more ratings than others. Because of this skewed distribution, it is possible that there is genre bias present in the data. With that, it is necessary to include and test this bias in the model. When this bias is added to the model, the model takes the general form of $Y_{u,i} = \mu + b_i + b_u + b_g + \epsilon_{u,i}$ where b_g is the genre bias. Applying this model to the dataset is as follows:

```
genre_bias <- training_set %>% left_join(movie_bias, by='movieId') %>%
  left_join(user_bias, by='userId') %>%
  group_by(genres) %>% summarize(b_g = mean(rating - mu - b_i - b_u))

predict_test_with_genre <- test_set %>% left_join(movie_bias, by='movieId') %>%
  left_join(user_bias, by='userId') %>%
  left_join(genre_bias, by='genres') %>%
  mutate(pred = mu + b_i + b_u + b_g) %>%
  pull(pred)
```

The RMSE value for this model can be seen in Table 4 below:

Table 4: Movie + User + Genre Bias Model and Corresponding RMSE

Model	RMSE
Average	1.0526109
Movie Bias	0.9414901
User Bias	0.8573267
Genre Bias	0.8572360

The RMSE value of 0.8572 for this model is a slight decrease compared to the previous model with the user and genre bias. This indicates that the genre bias is slightly significant in the model. The RMSE value would be significantly lower than the previous model if this genre bias was more significant in the model.

2.3.5 Regularized Movie Bias Model

The regularization term will added to each model with bias(es) in order to understand the trend and to analyze the RMSE for each. The movie bias only model will be the first model to be analyzed with the regularization term. When the regularization term is added, the model takes the general form of:$Y_{u,i} = \mu + b_i + \epsilon_{u,i} + \lambda(b_i)$ where $\lambda(b_i)$ is now the regularization term. The model in RStudio takes the form as follows:

```
lambdas <- seq(0, 10, 0.25)
rmses <- sapply(lambdas, function(l){
```

```
  mu <- mean(training_set$rating)

  b_i <- training_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+l))

  reg_movie_model_predict <- test_set %>%
    left_join(b_i, by = 'movieId') %>%
    mutate(pred = mu + b_i) %>%
    pull(pred)

  return(RMSE(reg_movie_model_predict, test_set$rating))
})
```

Figure 7 below shows a plot of the lambdas versus the RMSEs for the model:

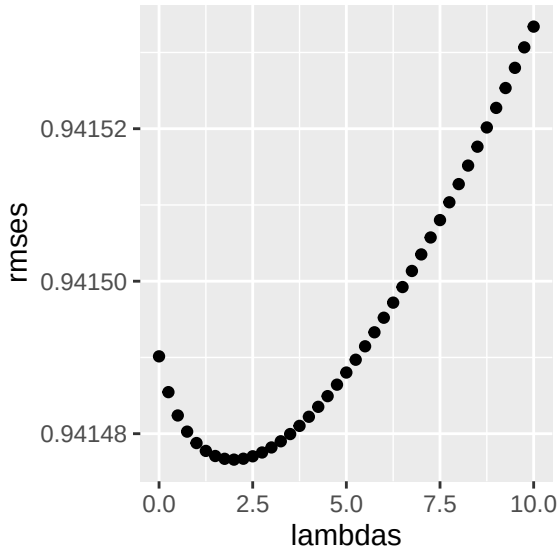

Figure 7: Plot of lambdas versus RMSEs for the Regularized Movie Bias Model

This plot shows that the RMSE value decreases as lambdas approaches 1.75 and increases as lambdas is further from 1.75. The minimum lambdas for this model is 1.75 with a corresponding RMSE value of 0.9415. The RMSE result can be found in Table 5 below.

Table 5: Regularized Movie Bias Model and Corresponding RMSE

Model	RMSE
Average	1.0526109
Movie Bias	0.9414901
User Bias	0.8573267
Genre Bias	0.8572360
Regularized Movie Bias	0.9414766

2.3.6 Regularized Movie + User Bias Model

The next model that will receive the regularization term is the movie and user bias model. When this regularization term is added to the model, the model takes the general form of $Y_{u,i} = \mu + b_i + b_u + \epsilon_{u,i} + \lambda(b_i + b_u)$ where $\lambda(b_i + b_u)$ is now the regularization term. The model in RStudio takes the form as follows:

```
lambdas <- seq(0, 10, 0.25)
rmses <- sapply(lambdas, function(l){
  mu <- mean(training_set$rating)

  b_i <- training_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+l))

  b_u <- training_set %>%
    left_join(b_i, by='movieId') %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_i - mu)/(n()+l))

  reg_user_model_predict <- test_set %>%
    left_join(b_i, by = 'movieId') %>%
    left_join(b_u, by='userId') %>%
    mutate(pred = mu + b_i + b_u) %>%
    pull(pred)

  return(RMSE(reg_user_model_predict, test_set$rating))
})
```

Figure 8 below shows a plot of the lambdas versus the RMSEs for the model:

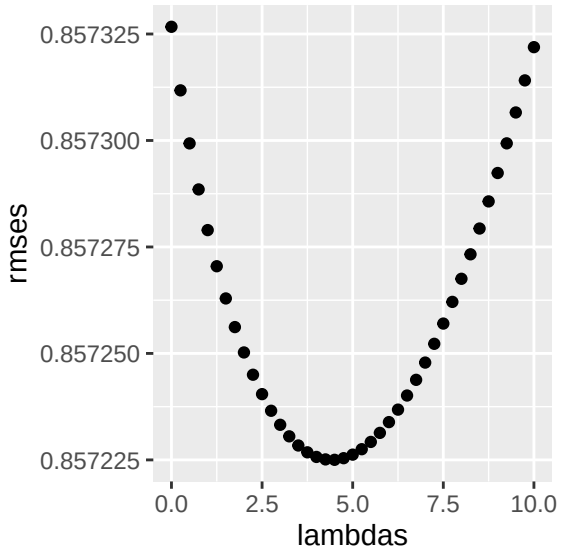

Figure 8: Plot of lambdas versus RMSEs for the Regularized Movie + User Bias Model

This plot shows that the RMSE value decreases as lambdas approaches 4.75 and increases as lambdas is further from 4.75. The minimum lambdas for this model is 4.75 with a corresponding RMSE value of 0.8572. The RMSE result can be found in Table 6 below.

Table 6: Regularized Movie + User Bias Model and Corresponding RMSE

Model	RMSE
Average	1.0526109
Movie Bias	0.9414901
User Bias	0.8573267
Genre Bias	0.8572360
Regularized Movie Bias	0.9414766
Regularized Movie + User Bias	0.8572250

2.3.7 Regularized Movie + User + Genre Model

The final model that will be tested with the regularization term is the movie, user and genre model. When the regularization term is added, the model takes the general form of $Y_{u,i} = \mu + b_i + b_u + b_g + \epsilon_{u,i} + \lambda(b_i + b_u + b_g)$ where $\lambda(b_i + b_u + b_g)$ is the regularization term. The model in RStudio takes the form as follows:

```
lambdas <- seq(0, 20, 0.25)
rmses <- sapply(lambdas, function(l){
  mu <- mean(training_set$rating)

  b_i <- training_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+l))

  b_u <- training_set %>%
    left_join(b_i, by='movieId') %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_i - mu)/(n()+l))

  b_g <- training_set %>%
    left_join(b_i, by='movieId') %>%
    left_join(b_u, by='userId') %>%
    group_by(genres) %>%
    summarize(b_g = sum(rating - b_i - mu - b_u)/(n() + l))

  reg_genre_model_predict <- test_set %>%
     left_join(b_i, by = 'movieId') %>%
     left_join(b_u, by='userId') %>%
     left_join(b_g, by='genres') %>%
     mutate(pred = mu + b_i + b_u + b_g) %>%
     pull(pred)
```

```
    return(RMSE(reg_genre_model_predict, test_set$rating))
})
```

Figure 9 below shows a plot of the lambdas versus the RMSEs for the model:

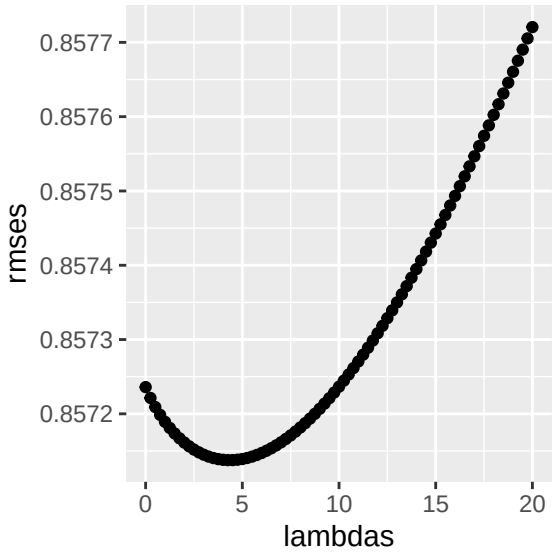

Figure 9: Plot of lambdas versus RMSEs for the Regularized Movie + User + Genre Bias Model

This plot shows that the RMSE value decreases as lambdas approaches 4.5 and increases as lambdas is further from 4.5. The minimum lambdas for this model is 4.5 with a corresponding RMSE value of 0.8571. The RMSE result can be found in Table 7 below.

Table 7: Regularized Movie + User + Genre Bias Model and Corresponding RMSE

Model	RMSE
Average	1.0526109
Movie Bias	0.9414901
User Bias	0.8573267
Genre Bias	0.8572360
Regularized Movie Bias	0.9414766
Regularized Movie + User Bias	0.8572250
Regularized Movie + User + Genre Bias	0.8571377

Table 7 indicates that the smallest RMSE value out of all the models is the value of 0.8571 that is associated with the regularized movie, user, and genre model. Thus, this model is the best model to use on the validation dataset and the lambdas that will be used is 4.5.

2.3.8 Validation Data - Regularized Movie + User + Genre Model

It was confirmed that the best model is the regularized movie, user and genre model. With that, this model will be applied to the validation dataset in order to acquire the final RMSE value of the

dataset. The edx dataset will be used in this final test in order to utilize as much data as possible for the final prediction. The value of lambdas that is being used is the lambdas from the smallest RMSE, which in this case is the RMSE associated with the regularized movie, user, and genre bias model. This lambdas has a value of 4.5. In RStudio, the prediction takes the form as follows:

```
lambda <- min_genre_lambda

mu_edx <- mean(edx$rating)

movie_avgs_reg <- edx %>% group_by(movieId) %>%
  summarize(b_i = sum(rating - mu_edx) / (n() + lambda))


user_avg_reg <- edx %>% left_join(movie_avgs_reg, by = "movieId") %>% group_by(userId) %>%
  summarize(b_u = sum(rating - b_i - mu_edx) / (n() + lambda))


genre_avg_reg <- edx %>% group_by(genres) %>%
  left_join(movie_avgs_reg, by = "movieId") %>%
  left_join(user_avg_reg, by = "userId") %>%
  summarize(b_g = sum(rating - b_i - b_u - mu_edx) / (n() + lambda))


predict_valid <- validation %>% left_join(movie_avgs_reg, by = "movieId") %>%
  left_join(user_avg_reg, by = "userId") %>% left_join(genre_avg_reg, by = "genres") %>%
  mutate(pred = mu_edx + b_i + b_u + b_g) %>% pull(pred)
```

The RMSE result can be found in Table 8 below.

Table 8: Regularized Movie + User + Genre Bias Model and Corresponding RMSE - Validation Set

Model	RMSE
Average	1.0526109
Movie Bias	0.9414901
User Bias	0.8573267
Genre Bias	0.8572360
Regularized Movie Bias	0.9414766
Regularized Movie + User Bias	0.8572250
Regularized Movie + User + Genre Bias	0.8571377
Regularized Movie + User + Genre Bias - Validation	0.8631535

3 Results

Table 9 below displays the RMSE values collected for all of the models. This completed table demonstrates the trend that is seen with the RMSE values. The RMSE values decrease for each bias added, and when a regularization term is added, the RMSE term decreases for the respective

model that the regularization term was added to. The model with the lowest RMSE value on the test set was the Regularized Movie + User + Genre Bias Model. The RMSE for this model when tested on the test set had a value of 0.8571, making it the lowest in the table. With that, this model was deemed the best model for the dataset. When the validation set was tested with this model, the RMSE value was 0.8630. Thus, the final RMSE value was 0.8630.

Table 9: Complete Table of Models and Corresponding RMSE Values

Model	RMSE
Average	1.0526109
Movie Bias	0.9414901
User Bias	0.8573267
Genre Bias	0.8572360
Regularized Movie Bias	0.9414766
Regularized Movie + User Bias	0.8572250
Regularized Movie + User + Genre Bias	0.8571377
Regularized Movie + User + Genre - Validation	0.8630539

4 Conclusion

Machine learning techniques were successfully used on the MovieLens dataset in order to construct an algorithm that can predict the rating of a movie based off of a series of inputs. With the use of linear regression, bias terms, and regularization, the final model has an RMSE value of 0.8630. Future projects with this same dataset can involve subsetting the dataset into one of smaller size, allowing for the unique packages and functions in RStudio to be performed on the dataset. The large size of the dataset posed an issue when trying to incorporate these functions, and resulted in the functions to be performed manually. A subset of the data can lead to further analysis of the dataset because of the size not restricting the functions that can be used.