MovieLens Final Project

Harvard EdX Professional Data Science Certificate

Samara Angel

Spring 2021

Introduction

Recommendation systems are made using machine learning algorithms. In the case of movie recommendations, they are made with the goal of predicting the rating a user would make of a movie in order to recommend movies to that user. In October of 2006, Netflix set a challenge to data scientists to improve the recommendation system of their in-house software, Cinematch. The winning team who could improve the algorithm by 10% would receive one million dollars. The winner of the challenge was announced in September of 2009 and was a group called BellKor's Pragmatic Chaos, a 7-person team of statisticians, computer-engineers, and machine learning experts. While the Netflix data is not available to the public, a similar database generated by the GroupLens research lab is available. The database contains over 20 million ratings, over 27,000 movies, and over 138,000 users. In this project, I will use a subset of the GroupLens data, available in the dslabs package. Each row of the data is representative of one user's rating of one specific movie, but it is important to note that not every user rated each movie. A recommendation system essentially fills in predicted ratings for those that do not yet exist. I will use Root Mean Square Error (RMSE) as the evaluation for the models I develop in this project, which is the same metric used to evaluate the Netflix Challenge. The RMSE is the square root of the averaged squared difference between the target value and the value predicted by the model. My goal in this project is to minimize RMSE, specifically to get a final RMSE < 0.86490. To do this, I will pre-process the data to generate a train set, "edx," (90%) and a test set, "validation," (10%). I will then split the EdX set into its own train set, "edx_train_set," (90%) and test set, "edx_test_set," (10%), and will use these data in order to train the models which I develop. Once I have trained each model using the edx_train_set and edx_test_set, I will then use the Validation set (the final hold-out set) to test my recommendation system.

The 10M version of the MovieLens dataset is available at the following link: https://grouplens.org/datasets/movielens/10m/

Methods and Analysis

Data Cleaning and Pre-Processing

I cleaned the data and partitioned them using source code from the EdX course materials that were provided. This involved installing the required packages, downloading the MovieLens data, generating column names, and partitioning the data into an EdX set (90%) and a Validation hold-out set (10%). I then did some additional data pre-processing, which involved splitting the EdX set itself into a train set (90%) and a test set (10%). I established the RMSE function, which is $\sqrt{\frac{1}{N}\sum_{u,i}\left(\hat{y}_{u,i}-y_{u,i}\right)^2}$ and which will be used to evaluate the models.

Data Exploration and Visualization

In this section, I will explore the data set and visualize the data. First, I show the head of the datasets generated. Here is the head of the EdX set.

Table 1: Head of EdX Data Set

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	231	5	838983392	Dumb & Dumber (1994)	Comedy
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
1	356	5	838983653	Forrest Gump (1994)	Comedy|Drama|Romance|War
1	362	5	838984885	Jungle Book, The (1994)	Adventure|Children|Romance
1	364	5	838983707	Lion King, The (1994)	Adventure|Animation|Children|Drama|Musical

And here is the head of the Validation set, which will not be used in the model generation. It will only be used at the end to test the efficacy of the model.

Table 2: Head of Validation Data Set

userId	movieId	rating	timestamp	title	genres
1	588	5.0	838983339	Aladdin (1992)	Adventure|Animation|C
2	1210	4.0	868245644	Star Wars: Episode VI - Return of the Jedi (1983)	Action|Adventure|Sci-Fi
2	1544	3.0	868245920	Lost World: Jurassic Park, The (Jurassic Park 2) (1997)	Action|Adventure|Horro
3	151	4.5	1133571026	Rob Roy (1995)	Action|Drama|Romance
3	1288	3.0	1133571035	This Is Spinal Tap (1984)	Comedy|Musical
3	5299	3.0	1164885617	My Big Fat Greek Wedding (2002)	Comedy|Romance
4	380	3.0	844416656	True Lies (1994)	Action|Adventure|Come
4	435	3.0	844417070	Coneheads (1993)	Comedy|Sci-Fi
4	480	5.0	844416834	Jurassic Park (1993)	Action|Adventure|Sci-Fi
5	477	3.0	857912840	What's Love Got to Do with It? (1993)	Drama|Musical

And here is the head of both the EdX train and EdX test sets.

Table 3: Head of EdX Train Data Set

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	231	5	838983392	Dumb & Dumber (1994)	Comedy
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
1	356	5	838983653	Forrest Gump (1994)	Comedy|Drama|Romance|War
1	362	5	838984885	Jungle Book, The (1994)	Adventure|Children|Romance
1	364	5	838983707	Lion King, The (1994)	Adventure|Animation|Children|Drama|Musical

Table 4: Head of EdX Test Data Set

userId	movieId	rating	timestamp	title	genres
1	589	5.0	838983778	Terminator 2: Judgment Day (1991)	Action|Sci-Fi
3	590	3.5	1136075494	Dances with Wolves (1990)	Adventure|Drama|Western
3	1552	2.0	1133571139	Con Air (1997)	Action|Adventure|Thriller
3	1564	4.5	1136418605	For Roseanna (Roseanna's Grave) (1997)	Comedy|Drama|Romance
3	5505	2.0	1136075848	Good Girl, The (2002)	Comedy|Drama
4	21	3.0	844416980	Get Shorty (1995)	Action|Comedy|Drama
4	440	3.0	844417037	Dave (1993)	Comedy|Romance
4	587	5.0	844416980	Ghost (1990)	Comedy|Drama|Fantasy|Romance|Thril
5	25	3.0	857911265	Leaving Las Vegas (1995)	Drama|Romance
5	736	1.0	857911264	Twister (1996)	Action|Adventure|Romance|Thriller

I then determine the number of rows and columns in the EdX data. Our data set has about 9 million rows, where each row represents one user's rating of one specific movie

Table 5: Number of Rows and Columns in the EdX Data Set

Number of Rows	Number of Columns
9000061	6

Here, I show a plot of the count vs. ratings given. From this graph, it is clear that whole number ratings (1,2,3,4,5) are more common than half-ratings (0.5, 1.5, 2.5, 3.5, 4.5).

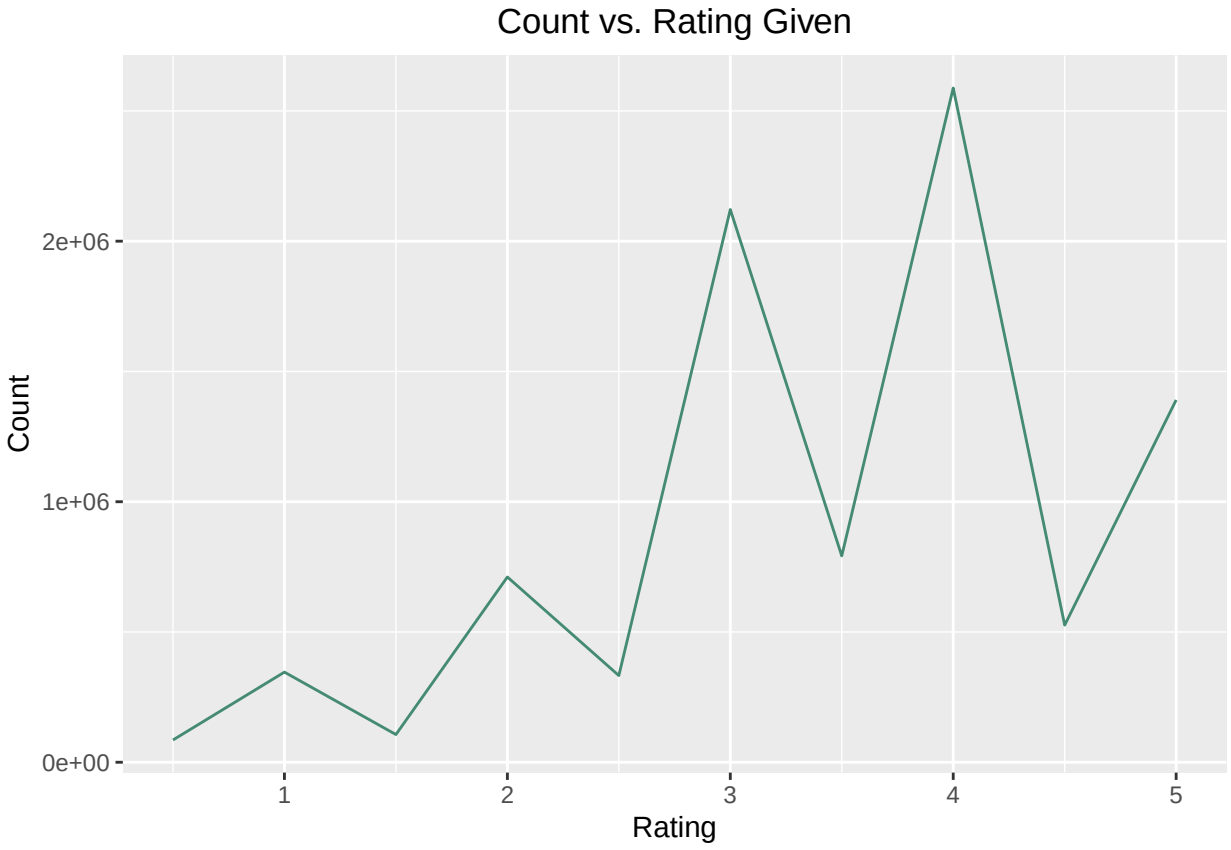

Figure 1: Plot of count of ratings by rating given.

Here, I show the number of distinct MovieIds and UserIds in the EdX dataset. If those numbers were multiplied, we would have over 745 million entries, but our data set has only around 9 million entries. Therefore, not every user rated every movie.

Table 6: Number of Distinct movieIds and userIds in the EdX Data Set

Unique MovieId	10677
Unique UserId	69878

The above concept can be visualized. Here, I show a plot of a sample of 100 ratings, with orange representing movies a user has rated and white representing movies that haven't been rated. The spaces that aren't filled in with orange have not been rated. We can think of this recommendation system as filling in those spaces.

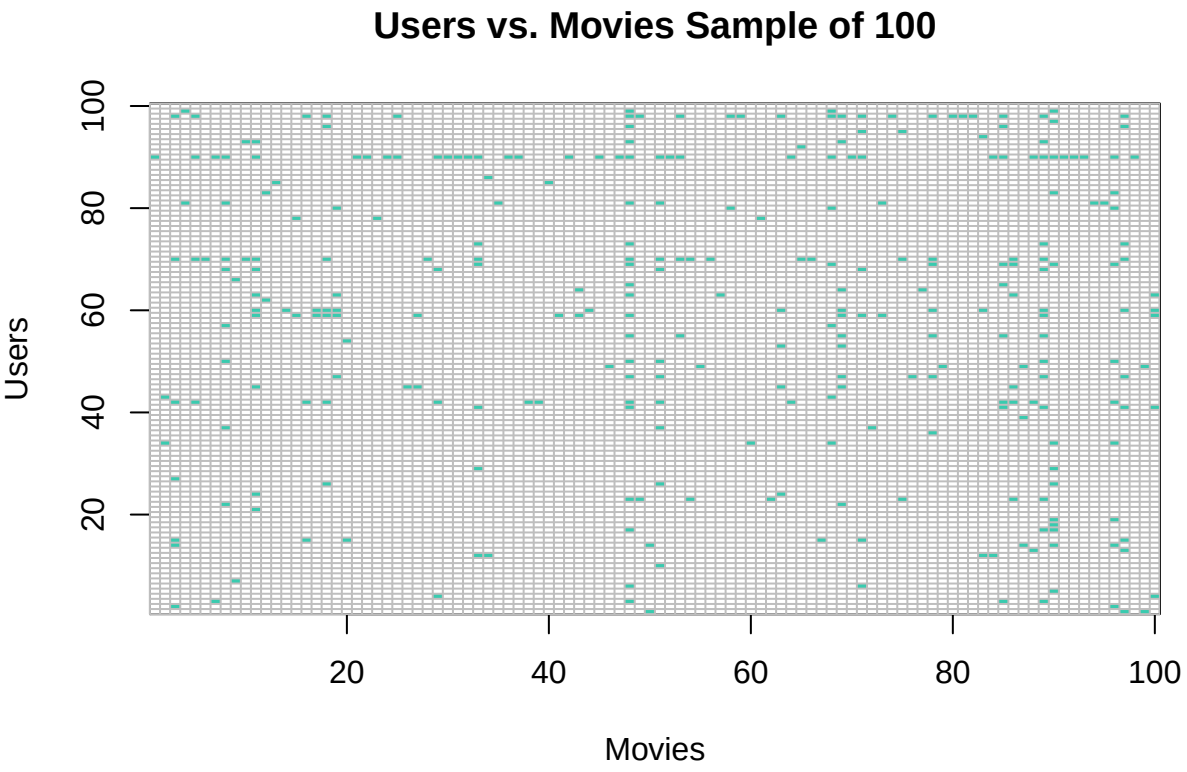

Figure 2: Plot of users vs. movies for a sample of 100 random ratings.

Non-Regularization Bias Models

In this section, I show the models I developed for the recommendation system. I begin by developing the simplest possible recommendation system, which predicts the same rating for all movies regardless of user. I call this model naive_rmse. This model is based on the following function, $Y_{u,i} = \mu + \varepsilon_{u,i}$, where μ is the "true" rating for the movies and $\varepsilon_{i,u}$ are independent errors are sampled from the same distribution centered at 0.

Table 7: Initial RMSE Results - Average Naive Model

method	RMSE
Avg Naive Model	1.059

The naive model has an RMSE of 1.059. That is our baseline on which to improve. Next, I create a movie rating system using the movieID, with the idea that some movies are generally rated higher than others. The model follows

$Y_{u,i} = \mu + b_i + \varepsilon_{u,i}$, where b_i is the bias by movie. Because of the size of the data set, I will estimate b_i using the least squares estimate $\hat{b}_i$. Here, one can see the RMSE for the average movie rating model.

Table 8: RMSE Results - Including Average Movie Rating Model

method	RMSE
Avg Naive Model	1.0590002
Avg Movie Rating Model	0.9426564

The movie rating model has an RMSE of 0.9427. This is a significant improvement. Next, I will factor in users to attempt to further improve RMSE. Some users may generally rate movies lower, while some may rate them higher – even in the case that a movie is generally highly rated. This complicates the movieID recommendation system by coupling it with information on a given user. I begin by computing average ratings for user u and visualizing that graphically. This is modeled by $Y_{u,i} = \mu + b_i + b_u + \varepsilon_{u,i}$, where b_u is the bias by user. Here, again, Because of the size of the data set, I will estimate b_u using the least squares estimate $\hat{b}_u$. I calculate the RMSE for the average movie and user rating model, comparing it to the average movie rating model.

Table 9: RMSE Results - Including Average Movie and User Rating Model

method	RMSE
Avg Naive Model	1.0590002
Avg Movie Rating Model	0.9426564
Avg Movie + User Rating Model	0.8646047

I then show a plot of the variability in estimates that come from b_u_hat, which shows that user ratings vary by user – some generally rate movies highly, others do not.

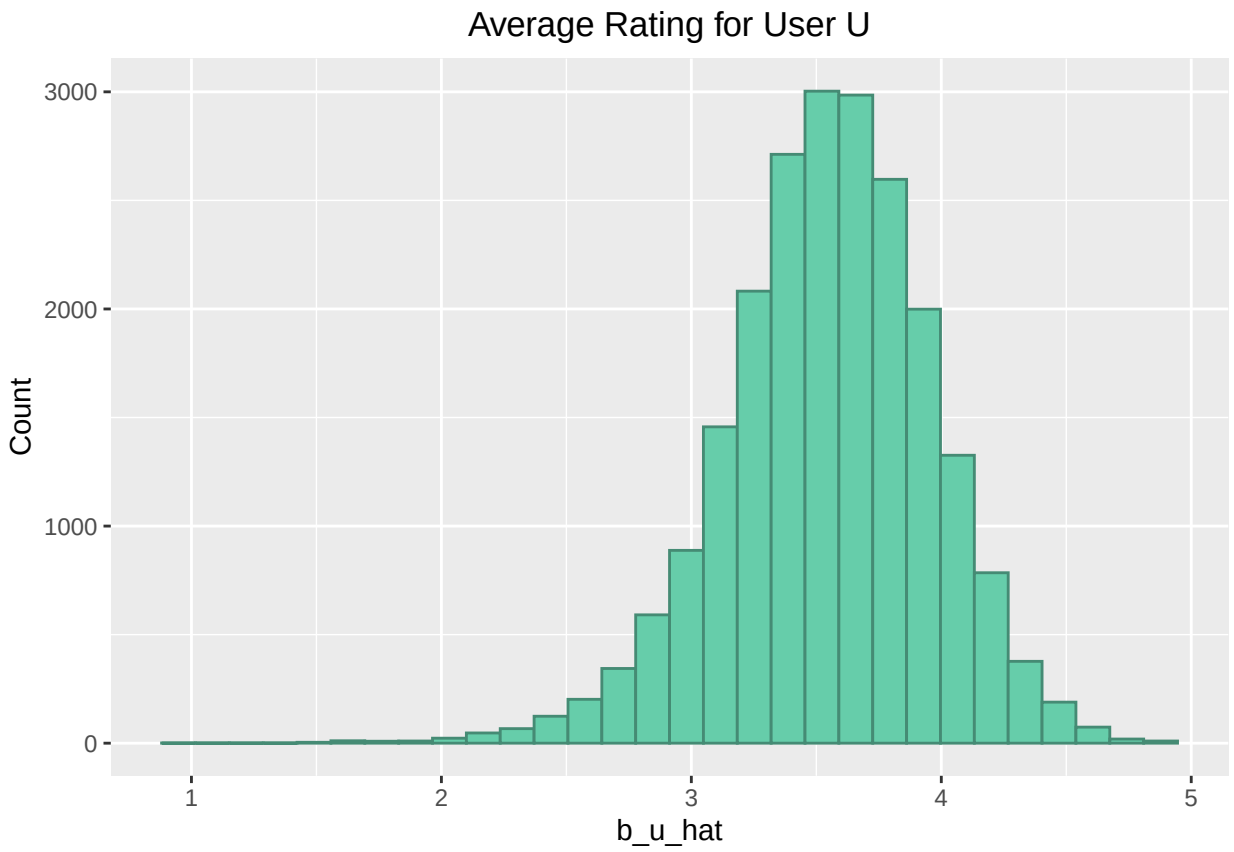

Figure 3: Plot of the average rating for user u.

Incorporating user, we once again see a large improvement with a RMSE of 0.8646. This is already below the project threshold value of 0.8649, but let's see if we can improve upon it. I will now incorporate biases other than movieID and user, namely the impact that genre has on the recommendation system. This is modeled by $Y_{u,i} = \mu + b_i + b_u + \sum_{k=1}^{K} x_{u,i}\beta_k + \varepsilon_{u,i}$, with $x_{u,i}^k = 1$ if $g_{u,i}$ is genre k. b_g is the bias by cross-listed genre and k is each given genre. Here, I use cross-listed genres as they appear in the data set. For example, Action|Adventure is treated as one genre. Because of the size of the data set, I will estimate b_g using the least squares estimate $\hat{b}_g$. I start by visualizing the average rating for each cross-listed genre g graphically. Like with users, this graph shows us that different genres generally receive different ratings.

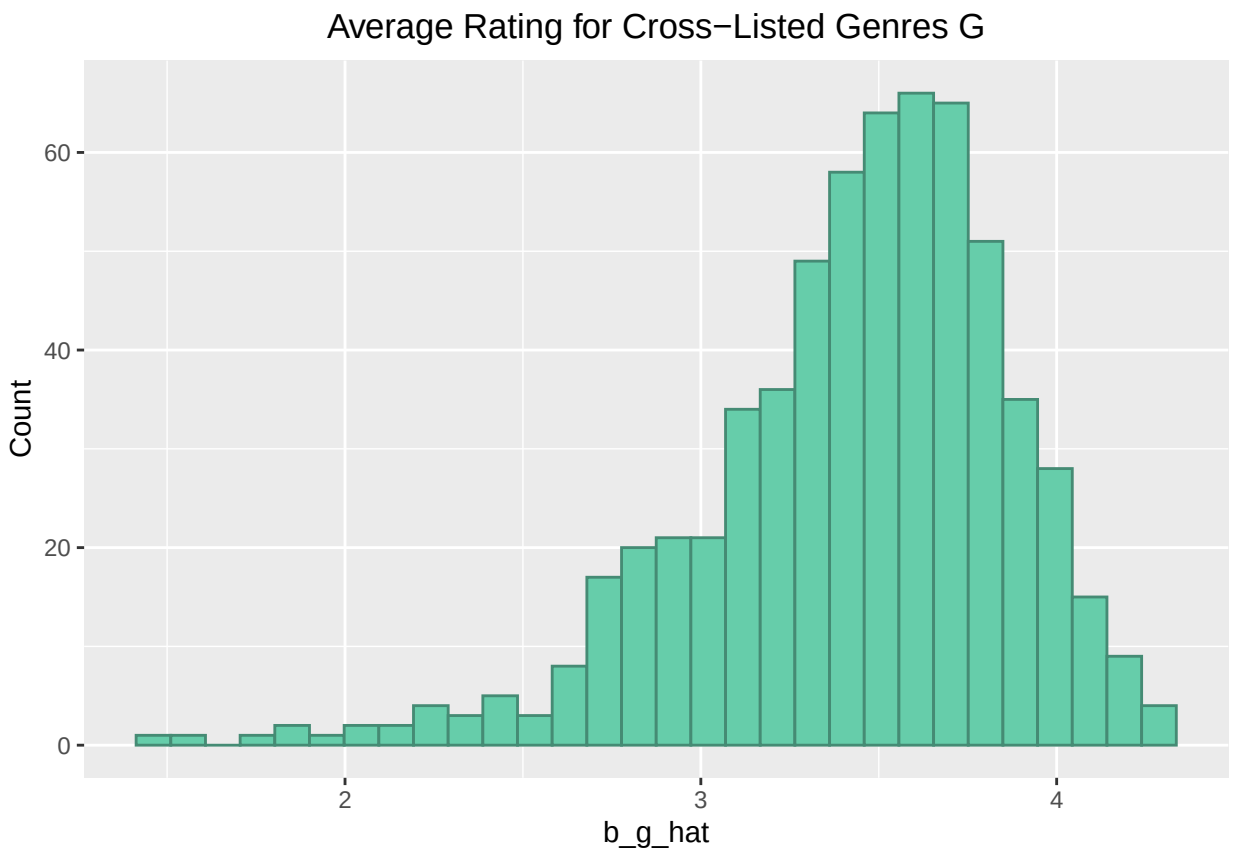

Figure 4: Plot of the average rating for cross-listed genres g.

I calculate the RMSE for the average movie and user and cross-listed genre rating model, comparing it to the prior two models.

Table 10: RMSE Results - Including Average Movie and User and Cross-Listed Genres Rating Model

method	RMSE
Avg Naive Model	1.0590002
Avg Movie Rating Model	0.9426564
Avg Movie + User Rating Model	0.8646047
Avg Movie + User + Genres Rating Model	0.8642542

With the cross-listed genres added to the model, we get an RMSE of 0.8643, which is a small improvement from the movie and user model. Now I consider genres separated out from their cross-listed format. For example, Action|Adventure is separated into Action and Adventure. This is modeled by the same equation: $Y_{u,i} = \mu + b_i + b_u + \sum_{k=1}^{K} x_{u,i}\beta_k + \varepsilon_{u,i}$, with $x_{u,i}^k = 1$ if $g_{u,i}$ is genre k. b_sg is the bias by separate genre. Because of the size of the data set, I will estimate b_sg using the least squares estimate $\hat{b}_sg$. I start by separating the data set by separate, non-cross-listed, genres. Once I do so, the head of the data set looks like the following.

Table 11: Head of Separated Genres

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy
1	122	5	838985046	Boomerang (1992)	Romance
1	185	5	838983525	Net, The (1995)	Action
1	185	5	838983525	Net, The (1995)	Crime
1	185	5	838983525	Net, The (1995)	Thriller
1	231	5	838983392	Dumb & Dumber (1994)	Comedy

Here, you can see the count of ratings for each separated genre in the train set.

Table 12: Separated Genres Count of Ratings – Train Set

genres	count
Drama	3518358
Comedy	3187341
Action	2304861
Thriller	2092662
Adventure	1717919
Romance	1540826
Sci-Fi	1207767
Crime	1194643
Fantasy	833604
Children	663980
Horror	622074
Mystery	511281
War	460251
Animation	420491
Musical	389710
Western	170534
Film-Noir	106783
Documentary	83855
IMAX	7385
(no genres listed)	5

And here is the count of ratings for each separated genre in the test set.

Table 13: Separated Genres Count of Ratings – Test Set

genres	count
Drama	391043
Comedy	353943
Action	255788
Thriller	232687
Adventure	190773
Romance	171406
Sci-Fi	133983
Crime	132274
Fantasy	92020
Children	73871
Horror	69333
Mystery	56584
War	51079
Animation	46729
Musical	43250
Western	18700
Film-Noir	11611
Documentary	9397
IMAX	805
(no genres listed)	1

I then visualize the average rating for each separate genre graphically, which shows us that different separated genres also generally receive different ratings.

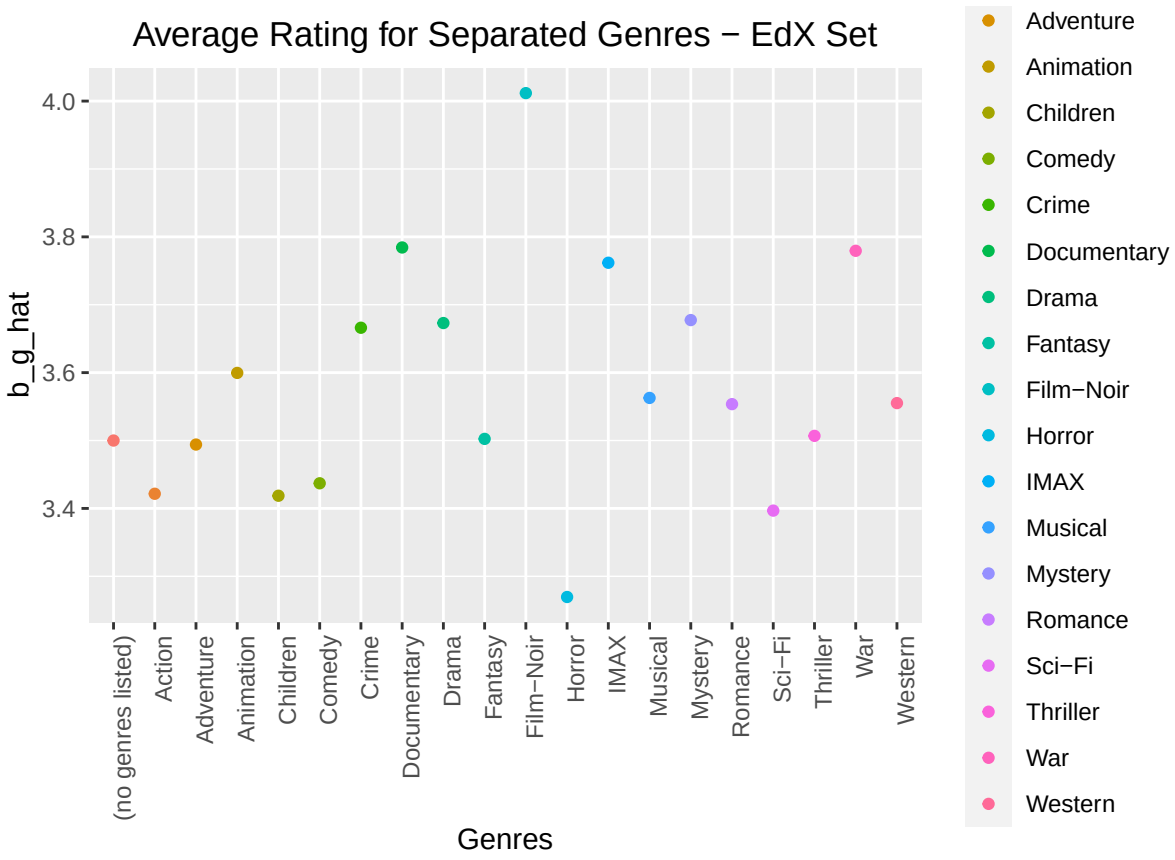

Figure 5: Plot of the average rating for separate genres.

Here, I calculate the RMSE for the average movie and user and separated genre rating model, comparing it to the prior two models.

Table 14: RMSE Results - Including Average Movie and User and Separated Genres Rating Model

method	RMSE
Avg Naive Model	1.0590002
Avg Movie Rating Model	0.9426564
Avg Movie + User Rating Model	0.8646047
Avg Movie + User + Genres Rating Model	0.8642542
Avg Movie + User + Separated Genres Rating Model	0.8626314

The RMSE for the movie, user, and separated genres model is 0.8626, which is the lowest RMSE obtained thus far. I will now see if improvement upon this model can be made using regularization.

Regularized Bias Models

To understand the reasoning behind the regularization process, I first show the 10 best movies without regularization. These films are all fairly obscure, which doesn't make much sense.

Table 15: 10 Best Movies Without Regularization

x
Hellhounds on My Trail (1999)
Satan's Tango (Sátántangó) (1994)
Shadows of Forgotten Ancestors (1964)
Fighting Elegy (Kenka erejii) (1966)
Sun Alley (Sonnenallee) (1999)
Blue Light, The (Das Blaue Licht) (1932)
Hospital (1970)
Constantine's Sword (2007)
Human Condition II, The (Ningen no joken II) (1959)
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)

Here are the 10 worst movies without regularization, which are also quite obscure.

Table 16: 10 Worst Movies Without Regularization

x
Besotted (2001)
Hi-Line, The (1999)
Grief (1993)
Accused (Anklaget) (2005)
Hip Hop Witch, Da (2000)
SuperBabies: Baby Geniuses 2 (2004)
From Justin to Kelly (2003)
Pokémon Heroes (2003)
Stacy's Knights (1982)
Dog Run (1996)

However, this obscurity can be explained when we look at the number of times these movies were rated. We can see that the top 10 best movies and the top 10 worst movies are generally rated very few times, because the low number of ratings warps their ranking among movies.

Table 17: Number of Ratings for the 10 Best Movies

title	n
Hellhounds on My Trail (1999)	1
Satan's Tango (Sátántangó) (1994)	2
Shadows of Forgotten Ancestors (1964)	1
Fighting Elegy (Kenka erejii) (1966)	1
Sun Alley (Sonnenallee) (1999)	1
Blue Light, The (Das Blaue Licht) (1932)	1
Hospital (1970)	1
Constantine's Sword (2007)	1
Human Condition II, The (Ningen no joken II) (1959)	3
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)	4

Table 18: Number of Ratings for the 10 Worst Movies

title	n
Besotted (2001)	2
Hi-Line, The (1999)	1
Grief (1993)	1
Accused (Anklaget) (2005)	1
Hip Hop Witch, Da (2000)	9
SuperBabies: Baby Geniuses 2 (2004)	56
From Justin to Kelly (2003)	177
Pokémon Heroes (2003)	127
Stacy's Knights (1982)	1
Dog Run (1996)	1

As you can see, the 10 best and worst movies without regularization are commonly rated by only one user. Because the number of users rating a given movie is so low in these cases, a given rating can elevate or tank that movie's rating. Regularization seeks to remedy that problem. I first begin by regularizing the movie rating model. Using cross-validation, I pick the optimal lambda which is, as you can see graphically, 1.5.

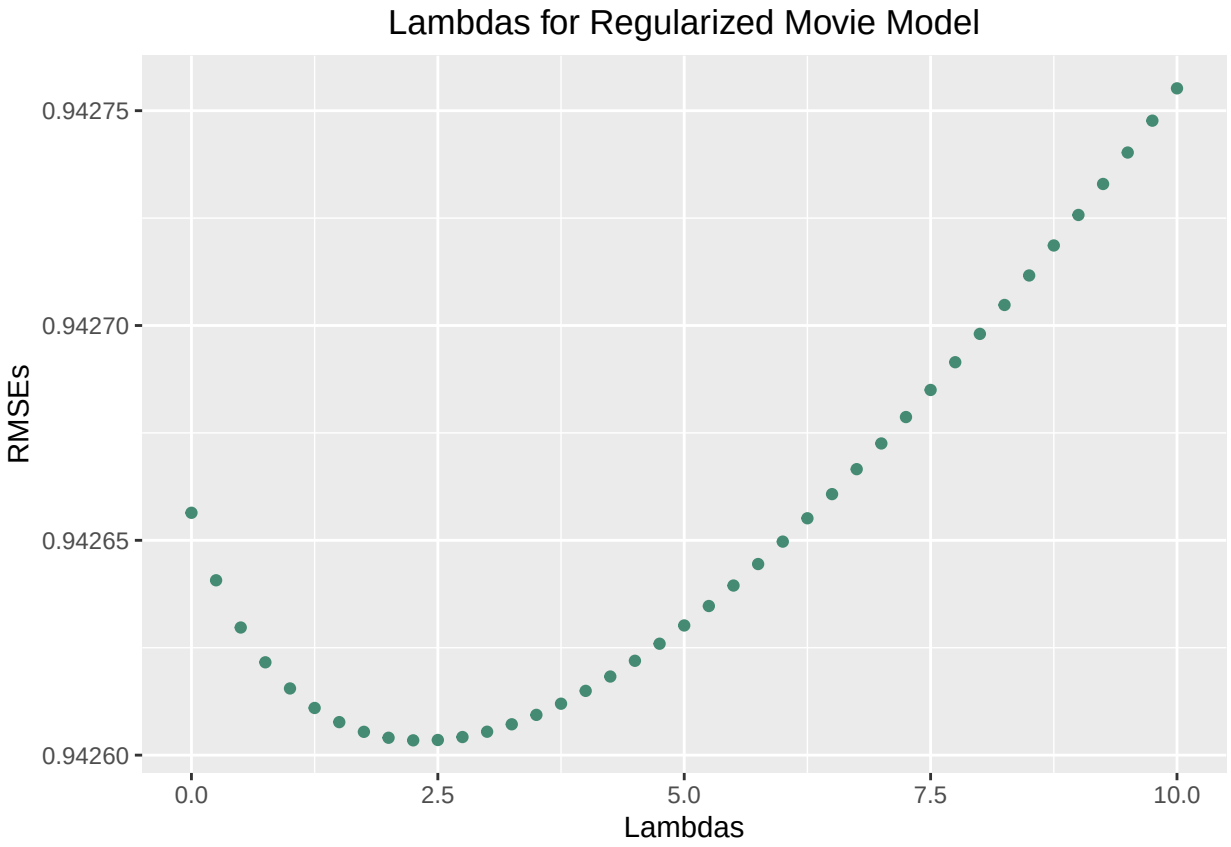

Figure 6: Plot of lambdas for the regularized movie method.

Given the optimal lambda, I then input that value to calculate the regularized movie rating RMSE.

Table 19: RMSE Results - Including Regularized Movie Rating Model

method	RMSE
Avg Naive Model	1.0590002
Avg Movie Rating Model	0.9426564
Avg Movie + User Rating Model	0.8646047
Avg Movie + User + Genres Rating Model	0.8642542
Avg Movie + User + Separated Genres Rating Model	0.8626314
Regularized Movie Rating Model	0.9426034

The RMSE of the regularized movie rating model is 0.9426, which is almost identical to the un-regularized model. The two differ only at the fifth significant figure. The contrast between the regularized method is visible by looking again at the top 10 movie estimates that we previously examined, but this time using regularization.

Table 20: 10 Best Movies With Regularization

x
Shawshank Redemption, The (1994)
Godfather, The (1972)
Usual Suspects, The (1995)
Schindler's List (1993)
More (1998)
Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)
Double Indemnity (1944)
Casablanca (1942)
Seven Samurai (Shichinin no samurai) (1954)
Rear Window (1954)

And here are the 10 worst movies with regularization.

Table 21: 10 Worst Movies With Regularization

x
SuperBabies: Baby Geniuses 2 (2004)
From Justin to Kelly (2003)
Pokémon Heroes (2003)
Hip Hop Witch, Da (2000)
Glitter (2001)
Disaster Movie (2008)
Pokemon 4 Ever (a.k.a. Pokémon 4: The Movie) (2002)
Gigli (2003)
Carnosaur 3: Primal Species (1996)
Barney's Great Adventure (1998)

These films make much more sense as the 10 best and worst rated movies. Here, I plot of the regularized estimates versus the least squares estimates to see how the estimates change with regularization.

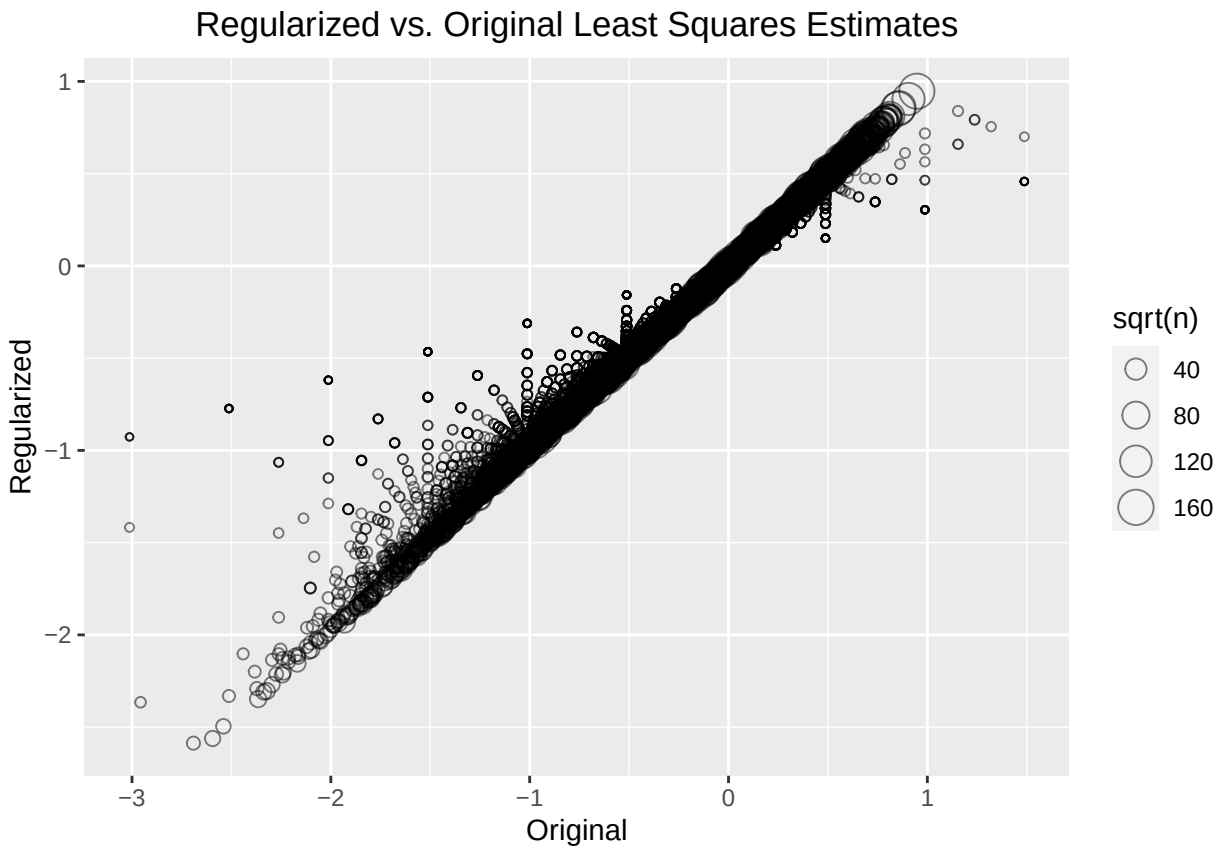

Figure 7: Regularized vs. Original plot using square root of n

The same process is then repeated for the movie and user rating model to choose the cross-validated, optimal lambda, which is 5. I then input the minimum lambda to calculate the regularized RMSE.

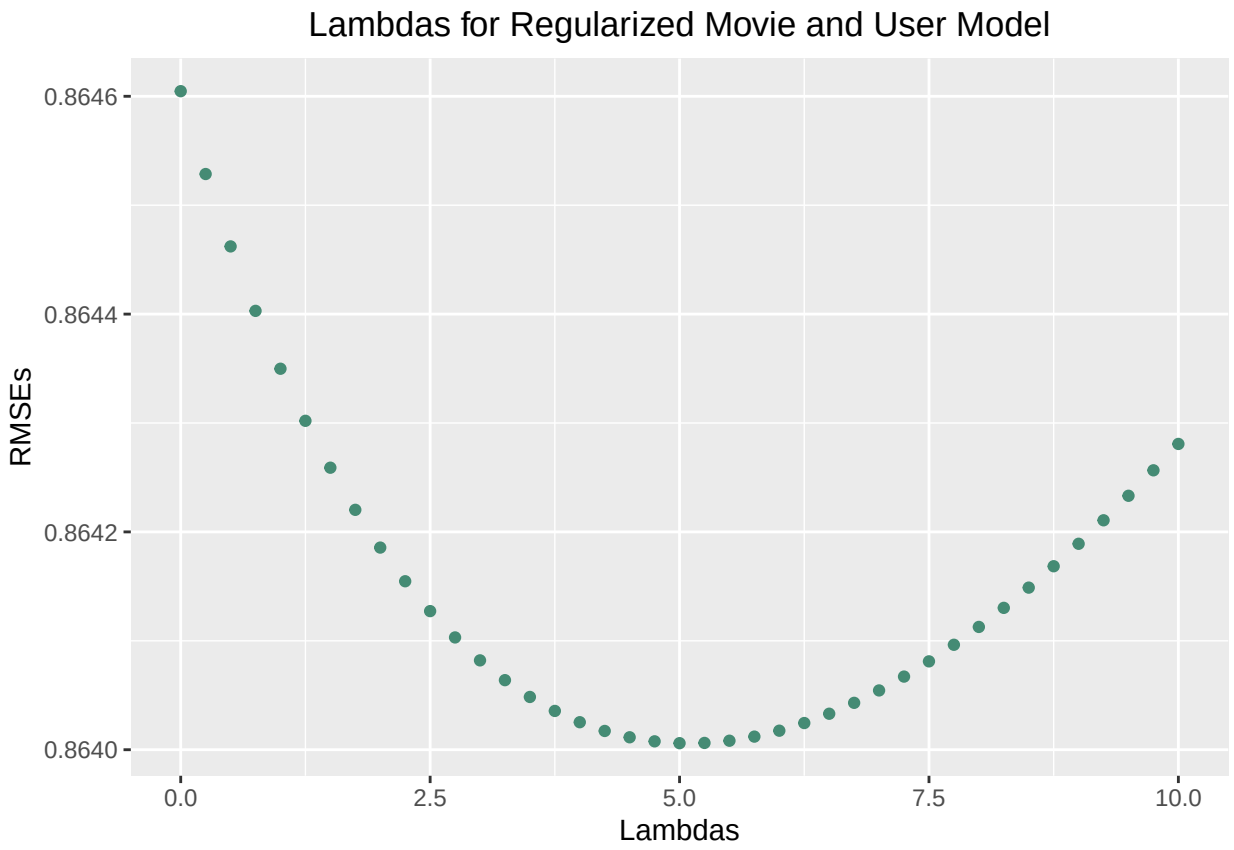

Figure 8: Plot of lambdas for the regularized movie and user method.

Table 22: RMSE Results - Including Regularized Movie and User Rating Model

method	RMSE
Avg Naive Model	1.0590002
Avg Movie Rating Model	0.9426564
Avg Movie + User Rating Model	0.8646047
Avg Movie + User + Genres Rating Model	0.8642542
Avg Movie + User + Separated Genres Rating Model	0.8626314
Regularized Movie Rating Model	0.9426034
Regularized Movie + User Rating Model	0.8640060

The RMSE of the regularized movie and user model is 0.8640, which is a slight improvement upon the movie and user rating model, but still performs worse than both models that include genre as an additional bias.

Results

The results of this analysis show that the most effective model is the average movie, user, and separated genres rating model. The results further imply that the more biases included, the more effective the model is. I will now proceed to test the most successful model with the original EdX and Validation sets to see if the RMSE remains below the project threshold RMSE of 0.8649. Here, as with before, I will separate genre instead of using the combination of the cross-listed genres, this time with the EdX and Validation sets. For example, Action|Adventure is separated into Action and Adventure. I then look at Action as a separate category from Adventure.

Here, I show the ratings, grouped by separated genre, for the EdX set and the Validation set.

Table 23: Ratings for EdX Set Grouped by Genre

genres	count
Drama	3909401
Comedy	3541284
Action	2560649
Thriller	2325349
Adventure	1908692
Romance	1712232
Sci-Fi	1341750
Crime	1326917
Fantasy	925624
Children	737851
Horror	691407
Mystery	567865
War	511330
Animation	467220
Musical	432960
Western	189234
Film-Noir	118394
Documentary	93252
IMAX	8190
(no genres listed)	6

Table 24: Ratings for Validation Set Grouped by Genre

genres	count
Drama	434797
Comedy	392784
Action	284700
Thriller	259086
Adventure	212382
Romance	189651
Sci-Fi	148739
Crime	148040
Fantasy	102858
Children	82298
Horror	76818
Mystery	63079
War	56733
Animation	51892
Musical	48214
Western	21225
Film-Noir	13198
Documentary	10202
IMAX	890
(no genres listed)	1

I start by computing average ratings for each separated genres for the Validation set and visualizing them graphically. We can again see that different separated genres generally receive different ratings. Next, I will calculate the average for each individual genre, k. Each genre combo has a different avg rating, b_g, the bias by genre. Because of the size of the data set. We will estimate b_g using the least squares estimate b_g_hat.

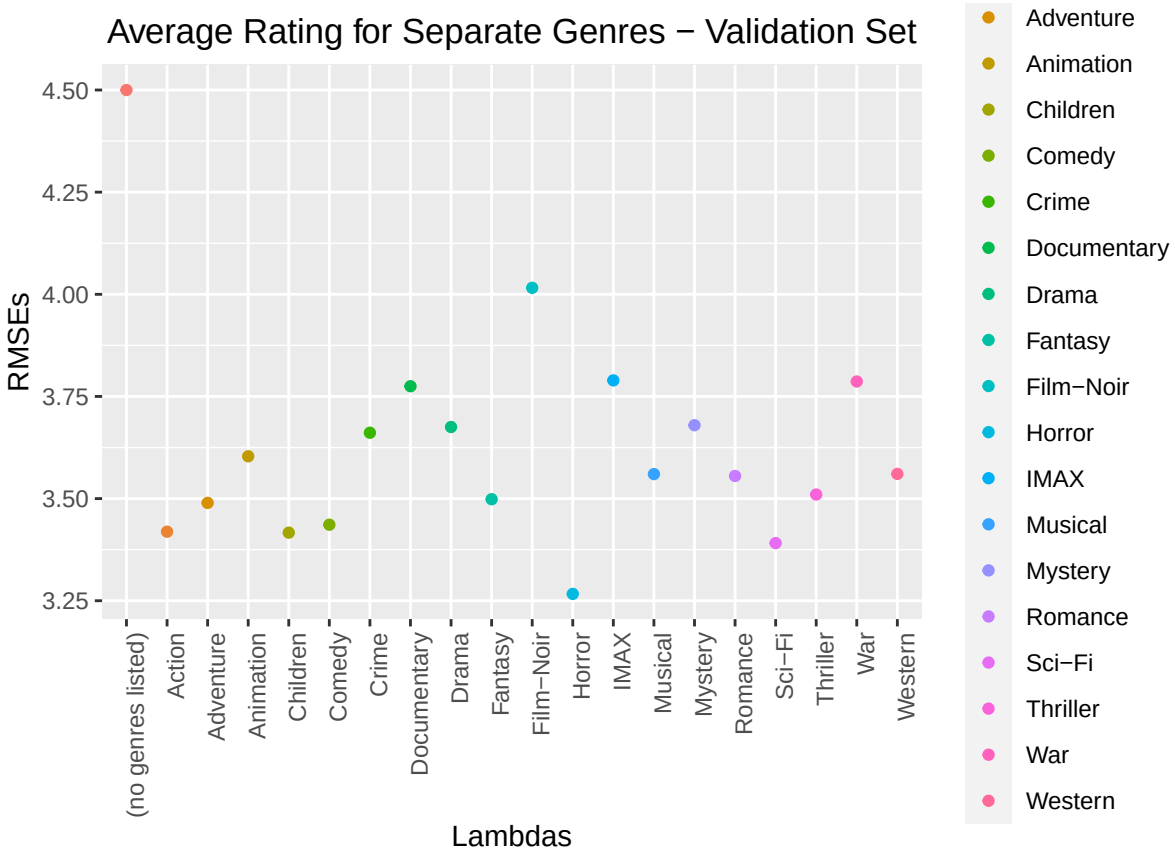

Figure 9: Plot of the average rating for separate genres for the validation set.

I then calculate the final RMSE using the Validation set for the movie, user, and separated genre model.

Table 25: RMSE Results - Final Validation Movie and User and Separated Genres Rating Model

method	RMSE
Avg Naive Model	1.0590002
Avg Movie Rating Model	0.9426564
Avg Movie + User Rating Model	0.8646047
Avg Movie + User + Genres Rating Model	0.8642542
Avg Movie + User + Separated Genres Rating Model	0.8626314
Regularized Movie Rating Model	0.9426034
Regularized Movie + User Rating Model	0.8640060
Final Validation Avg Movie + User + Separated Genres Rating Model	0.8638897

As is visible from the above table, the RMSE of the final Validation and EdX set is 0.8639. This value is higher than that obtained using the EdX train and test set, 0.8626, but is still well below the threshold RMSE of 0.8649.

Conclusion

As part of the final course for the Harvard EdX professional certificate in data science, I set out to develop a model to make movie recommendations that would have an RMSE below 0.8649. I began by pre-processing and cleaning the data, making sure to split the EdX set into a train and test set, while saving the Validation set as the final hold-out set. I then visualized the data, before beginning to test a series of models. I found that the more biases applied to a model, the better the model performs. For instance, the movies model performed worse than the movies and users model, which performed worse than the movies, users, and genres model. While regularized models performed very slightly better than their non-regularized counterparts, the difference between each corresponding pair was extremely small (variation began only at the fifth significant figure). The most successful model I tested was the movies, users, and separated genres model, which had a final RMSE of 0.8639 using the final hold-out Validation set. This is below the target RMSE of 0.8649 and is therefore a successful model that achieves the goal of this project.

While this model performed well, there is much future work that could be done to expand and improve this report. Further biases, such as time of rating, could be used to develop models with more biases. Given my results, models with more biases included would likely have lower RMSE values. In addition, more models could be tested with regularization. Because of the processing time regularization requires, I was only able to regularize the movie model and the movie and user model. In the future, I would be interested to test regularization on each model. Use of other techniques, such as matrix factorization, RandomForest, and SVD could also be interesting expansions of this study.

Sources:

- Chen, E. (2011, October 24). Winning the Netflix Prize: A Summary. http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/.
- Irizarry, R. A. (2020, March 2). Introduction to Data Science. Retrieved from https://rafalab.github.io/dsbook/introduction-to-machine-learning.html#notation-1.
- Koren, Y. (2009). The bellkor solution to the netflix grand prize. Netflix prize documentation, 81, 1-10. https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf.
- Lohr, S. (2009, September 21). Netflix Awards $1 Million Prize and Starts a New Contest. https://bits.blogs.nytimes.com/2009/09/21/netflix-awards-1-million-prize-and-starts-a-new-contest/.