Movie Recommender Report

An investigation into different machine learning methodologies to create a recommendation system

Alastair Humphrey

Jan, 2020

Introduction

Understanding and utilising machine learning has become of the utmost importance in the business world. Early adopters have been able to signifcantly disrupt established markets and completely redefined consumption. Prehaps the most prevelant examples of this are the recommender models. Amazon have revelotunionised the way we purchase books (and by extension many more items). Where previously an employee was required to offer recommendations for books or suggested titles, now a recommender system can automatically promote targeted litrature to every user of the platform. The other market disruptor harnessing the power of recommender systems is Netflix, the video streaming company. Their business model of offering content available for streaming together with recomendations on what to watch has created a paradigm shift in the entire film and television market.

Netflix have not settled for just disrupting an industry, but they have also taken it upon themselves to promote machine learning improvement within the field of computer science. They began by opening a competition in October 2006 for any indavidual or group who could improve their recommendation system. By the end of the competition in 2009, they had awarded a million dollar prize to a development team who improved their recommender model by 10%. The Netflix Prize also generated increased interest in reccomender models and machine learning, and many of the now standard techniques were popularised during this contest, such as latent factor models, which also made other contributions to the computer science field.

Recommender models can be categorised in various ways, but the two most prominent models are the 'prediction versions' and the 'ranking versions'.(Aggarwal,2016)[1] The predicition outputs of the model takes the form of a prediction of a rating for a user-component pairing. In this example, we will be looking at predicting ratings for a specific user and movie combinantions. This recommender problem variant is alternitvely known as the 'matrix completion problem'. The second prominent variant is the ranking version, in this form, the model will create a top-k recommender items for a user, or conversely, top-k users for an item. In the context of this report, the recommender model considers offering a user the top 10 movies to watch next on their streaming account.

This report is an exercise in emulating the Netflix Prize, using a proxy dataset provided by the 10 Million ratings MovieLens data set. The report is divided into two major sections, data exploration and reggression models. Finally the report concludes with a regularize reggession model.

Dataset exploration

[1] Aggarwal C.C. (2016), Recommender systems (pp. 1-28), Springer International Publishing

Dataset generation and partition

The dataset has been drawn from the group lens database, a Social Computing resource established by the University of Minnesota.[2] As the purpose of the exercise is to create a recommendation system, the data is partioned into two parts. The first is a training data set, titled 'edx'. The second is a the validation dataset. The validation dataset consists of 10% of the total 10M Movie Lens data set. The final recommender model will be evaluated for their accuracy against this validation set. For model selection purposes, the edx dataset is partitioned again into a train and test data set.

The data set is made up of 6 variables and 9,000,055 observations. The remaining observations form the validation dataset, constituting the same 6 variables and 999,999 observations. As part of this partion, the userIds and movieIds that appear only in the Validation set are added back into the edx dataset.

Table 1: The first few rows of the edx dataset

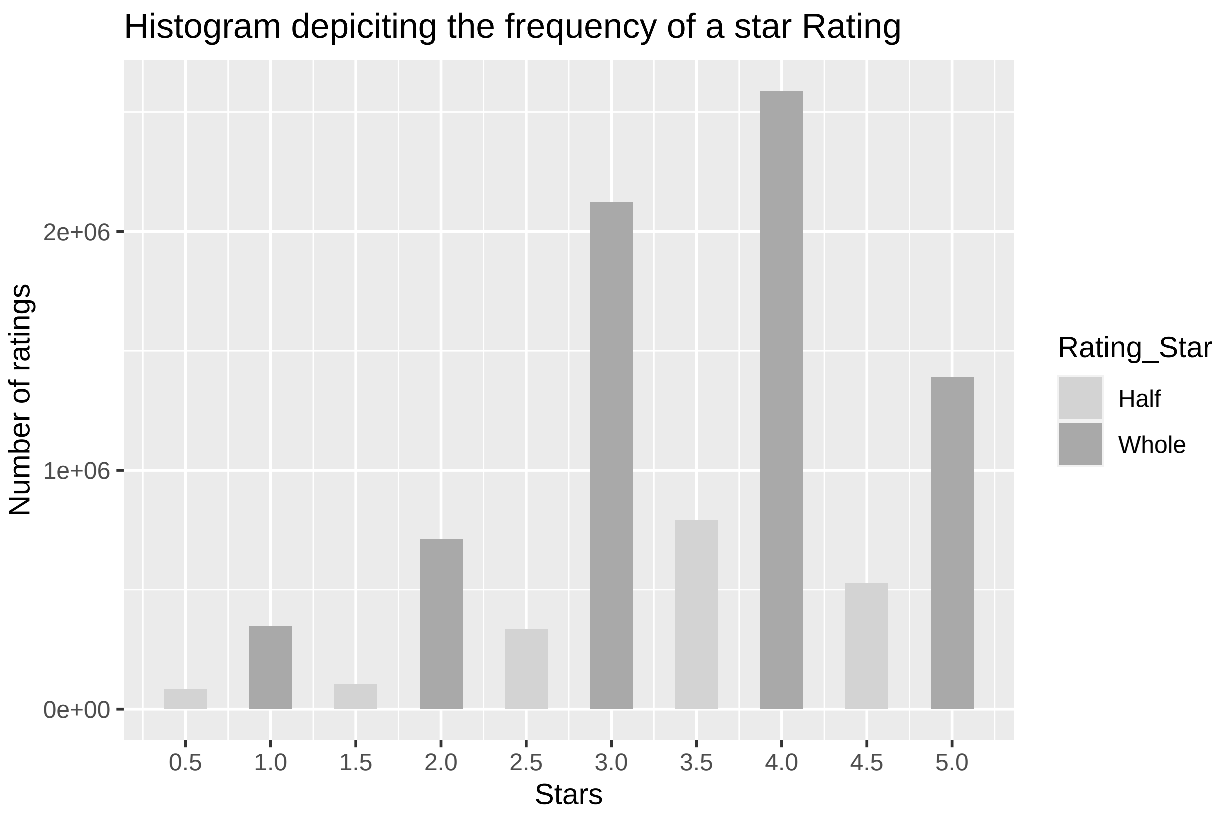
	userId	movieId	rating	timestamp	title	genres
1	1	122	5	838985046	Boomerang (1992)	Comedy|Romance
2	1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
4	1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
5	1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
6	1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
7	1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

Each of the variables has a distinct purpose. The userId is a distinct classification for each user who rated movies as part of the dataset. The movieId is a distinct identifier for each unique movie. The rating is the rating that the unique user gave the unique movie. The timestamp is the date and time of the review. Title refers to the movie title, however, these are not nessicarily distinct, hence the requirement for the unique moive identifier. The genres variable is a string of text used to broadly identify the genre or genres of the movie. The genres are deliminated by a '|' within the string.

Investigating the ratings

Each rating is the nomianl star rating from 1 to 5 at 0.5 increments for the movie. There is no spcefied criterion for this rating metric other than 5 being a high score and 1 being low. The ratings can be considered a quantitve variable. Based on this information, the ratings as a vector of data can be explored.

[2] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872


Histogram depiciting the frequency of a star Rating

Data Source: MovieLens 10M − transformed to edx

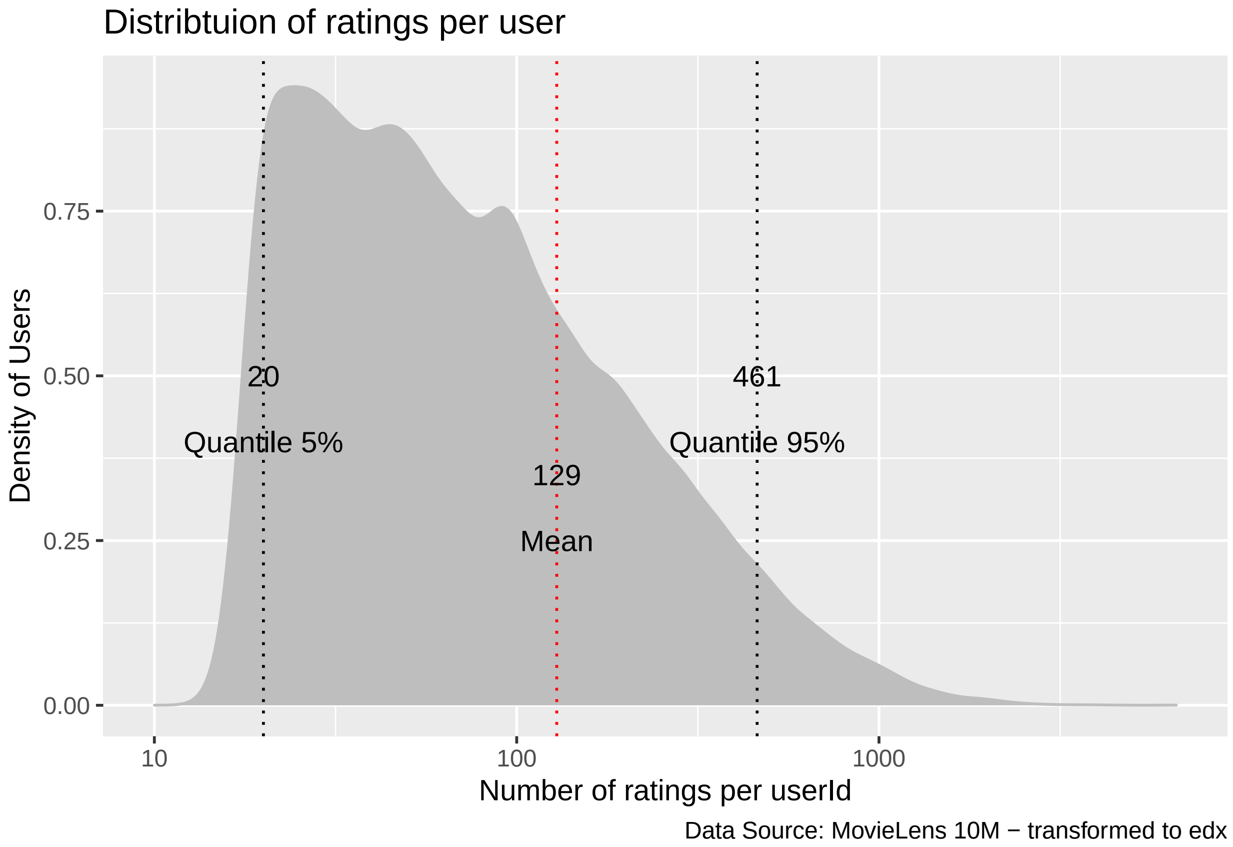
```
## [1] 3.512465
```

The histogram presented shows the number of ratings given and each star or half star awarded. The data clearly shows that most ratings were given as a whole number rather than half star. Further to this, the histogram clearly indicates that the most popular score for movies was a rating of 4 stars. The distribution itself is of interest as we can see that the rating of 3 then 5 are the second and third most popular ratings. The mean rating within the edx data set is 3.5 stars which the histogram describes with the longer left tail but higher frequency at higher scores.

Understanding the users

There 69,878 distinct users within the edx dataset. The users were seleced at random from within the movie ratings data the group lens project has collected.

```
## [1] 69878
```



On average, a user rated 129 movies. 90% of all users rated between 20 movies and 461 movies. As graphically shown, there is left skew to the distribution, with more users rating lower numbers of films. There is a longer tail to the right, with the largest number of ratings by a single user being on 6,616 movies.

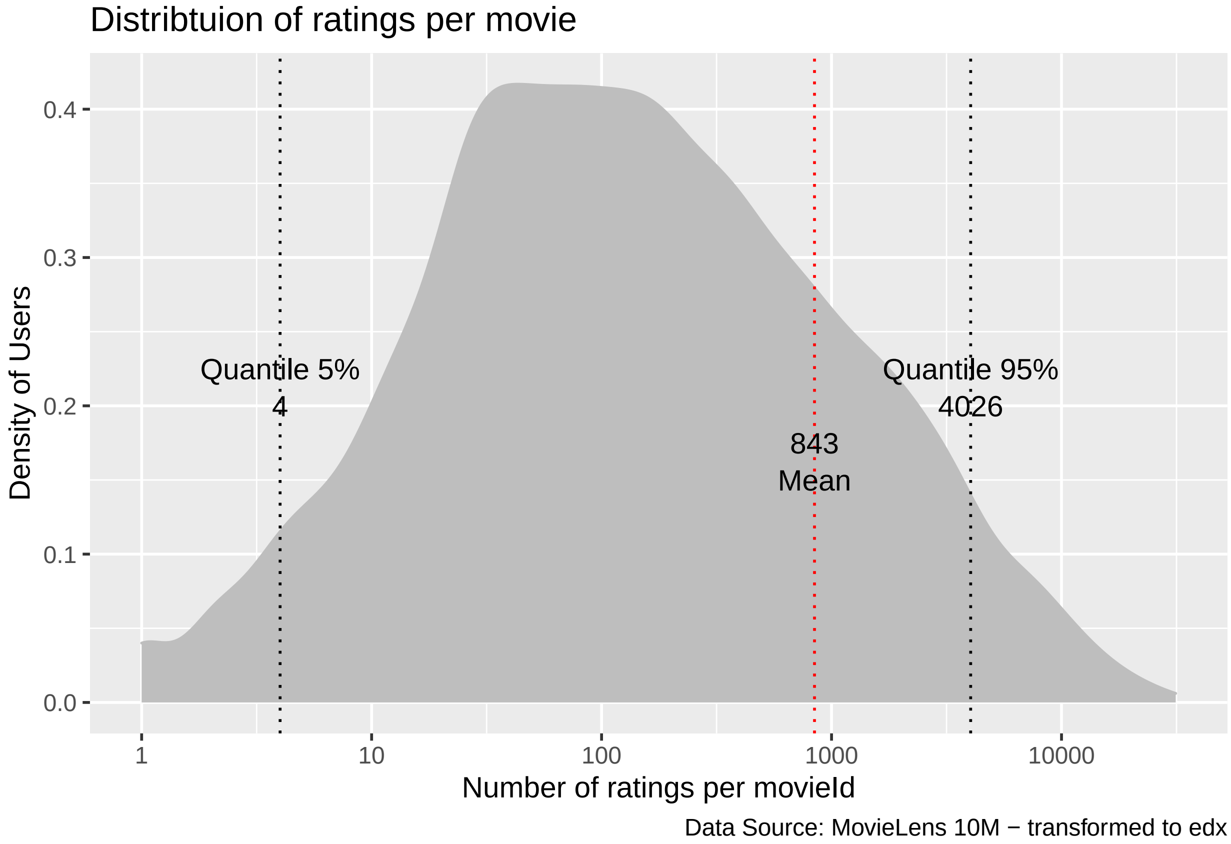
Table 2: The ten users with the largest number of movies rated

userId	n
59269	6616
67385	6360
14463	4648
68259	4036
27468	4023
19635	3771
3817	3733
63134	3371
58357	3361
27584	3142

Understanding the movies data

```
## [1] 10677
```

There are 10,677 distinct movies within the edx dataset. These movieIds are paired with a title for the movie and the year of release, as found on IMDB. Titles were entered manually by the user and therefore there could have been potential discrepancies and errors. Thus, the movieIds are used as the unique identifier.



The average number or ratings per movie is 843 ratings, with 90% of all movies recieving between 4 and 4026 ratings. Due to a few movies having very large numbers of ratings comparative to the rest of the data set, we can see that the average is significantly differenet from the median, at 122 ratings per movie.

Table 3: The ten movies with the largest number of user ratings

movieId	title	n
296	Pulp Fiction (1994)	31362
356	Forrest Gump (1994)	31079
593	Silence of the Lambs, The (1991)	30382
480	Jurassic Park (1993)	29360
318	Shawshank Redemption, The (1994)	28015
110	Braveheart (1995)	26212
457	Fugitive, The (1993)	25998
589	Terminator 2: Judgment Day (1991)	25984
260	Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672
150	Apollo 13 (1995)	24284

As is to be expected, the top ten most reated movies include huge blockbuster movies and cult classics. Pulp Fiction recieved 31,362 ratings by indavidual users. Conversely, the ten movies with the lowest rating count are extremely likely to be unkown to the average user, contextually, within the dataset there were 126 instances of a movie only being rated a single time.

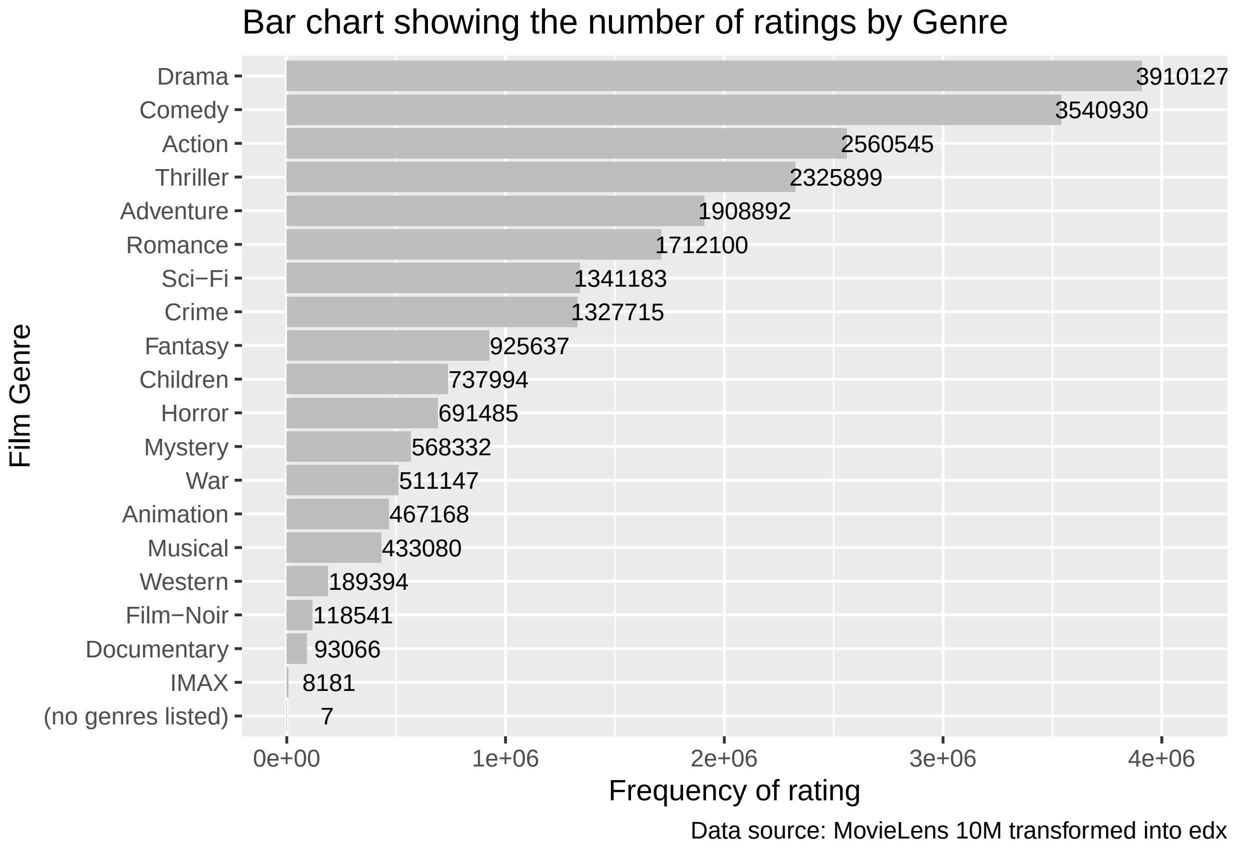
Table 4: The ten movies with the lowest number of user ratings

movieId	title	n
64903	Nazis Strike, The (Why We Fight, 2) (1943)	1
64918	Small Cuts (Petites coupures) (2003)	1
64926	Battle of Russia, The (Why We Fight, 5) (1943)	1
64944	Face of a Fugitive (1959)	1
64953	Dirty Dozen, The: The Fatal Mission (1988)	1
64976	Hexed (1993)	1
65006	Impulse (2008)	1
65011	Zona Zamfirova (2002)	1
65025	Double Dynamite (1951)	1
65027	Death Kiss, The (1933)	1

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   126
```

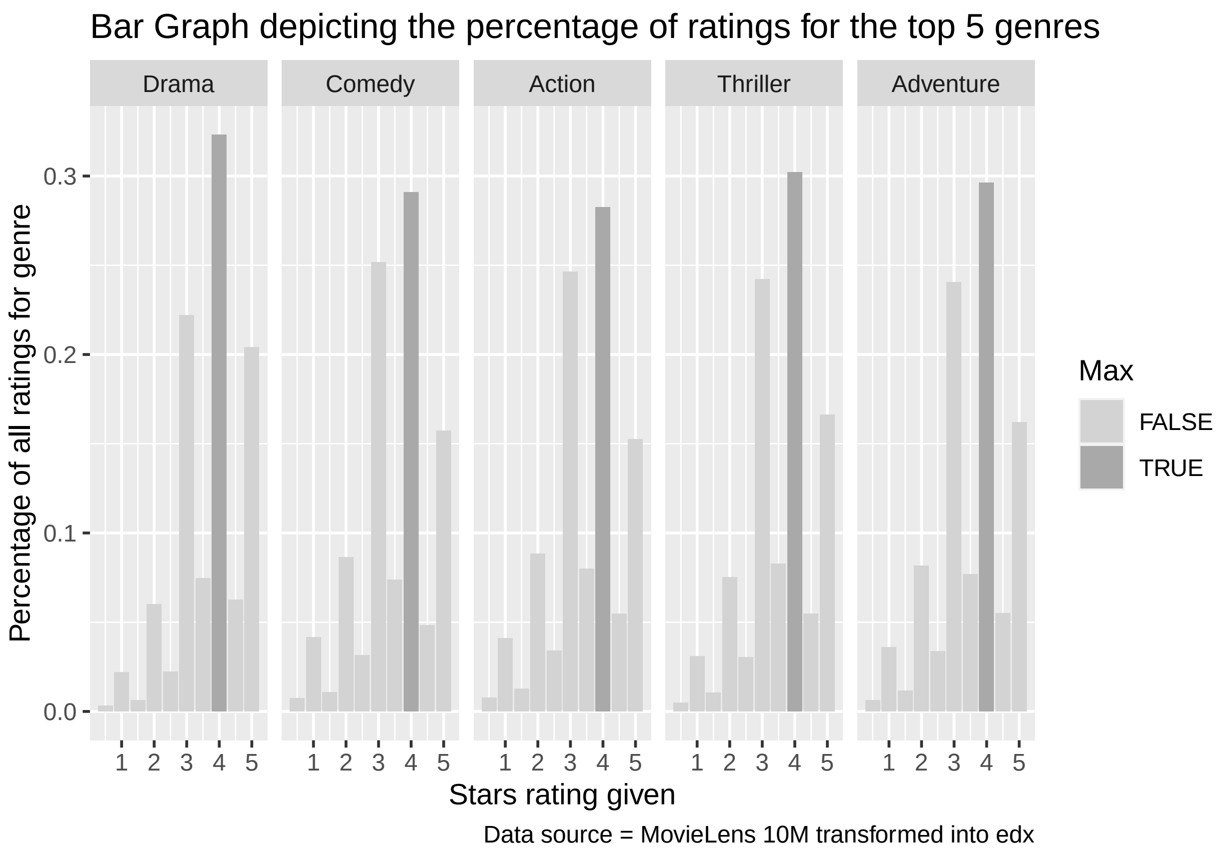
Exploring the genres variable

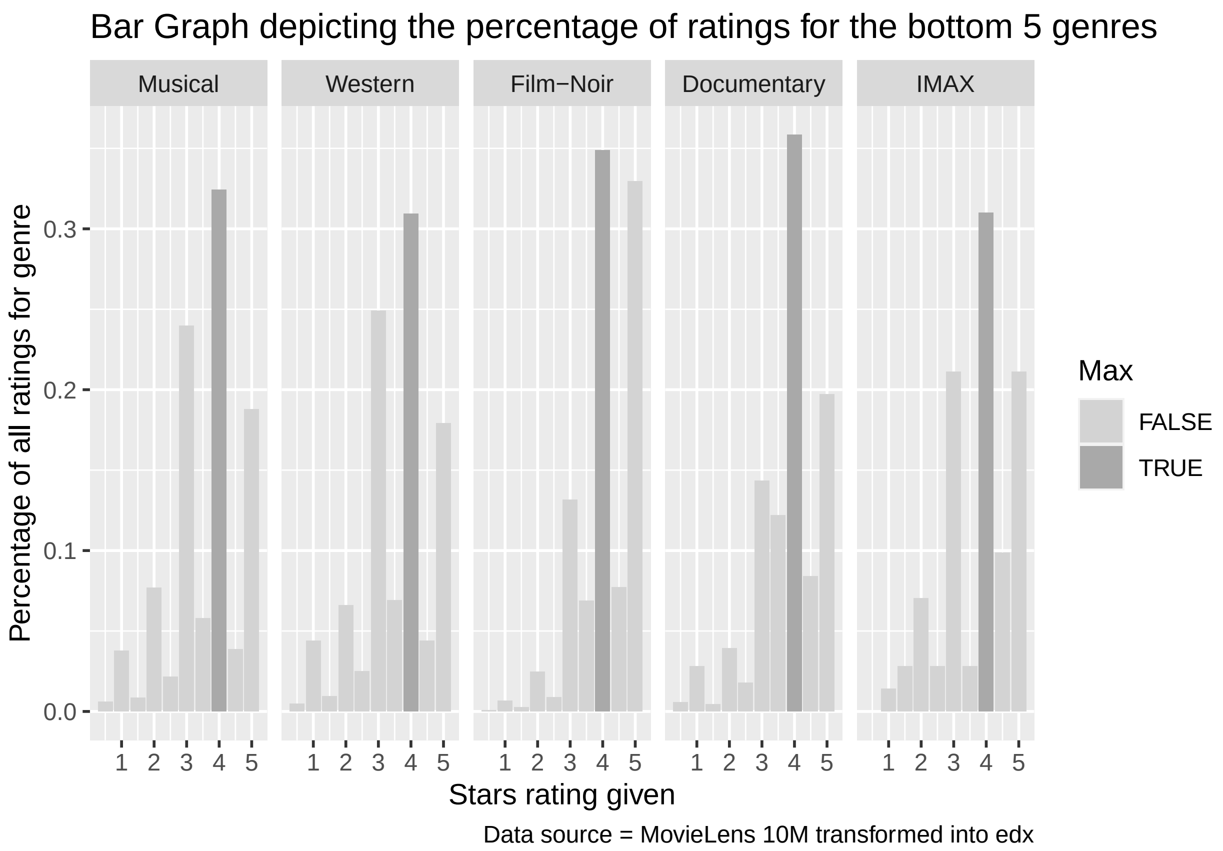
The genres within the dataset are user generated 'tags' about the movies. The first piece of exploration is to split the strings into the distinct tags and explore the quantity of each that have been rated within the edx dataset. The genres field can be considered a qualatitive variable within the dataset.



As the bar chart shows, Drama and Comedy received the largest number of ratings, whereas IMAX, Documentary and Film-Noir recieved the least ratings. The gap between the most popularly rated, drama, at 3,910,127 unique user+movie combinations vs IMAX at 8,181 is extremely large. For reference, 478 ratings were given from drama films vs 1 rating for an IMAX movie.

The genres are obviously important, but the fact that within the dataset there are only 19 tags to choose from (with various combinations) means that the genres categories are extremely broad. So a natural question that can be asked of this is do users rate different genres differently, or to put it another way, is there a 'general genre effect that can be can measured'?


Bar Graph depicting the percentage of ratings for the top 5 genres
Drama
Comedy
Action
Thriller
Adventure
Percentage of all ratings for genre
0.0
0.1
0.2
0.3
1
2
3
4
5
Stars rating given
Max
FALSE
TRUE
Data source = MovieLens 10M transformed into edx



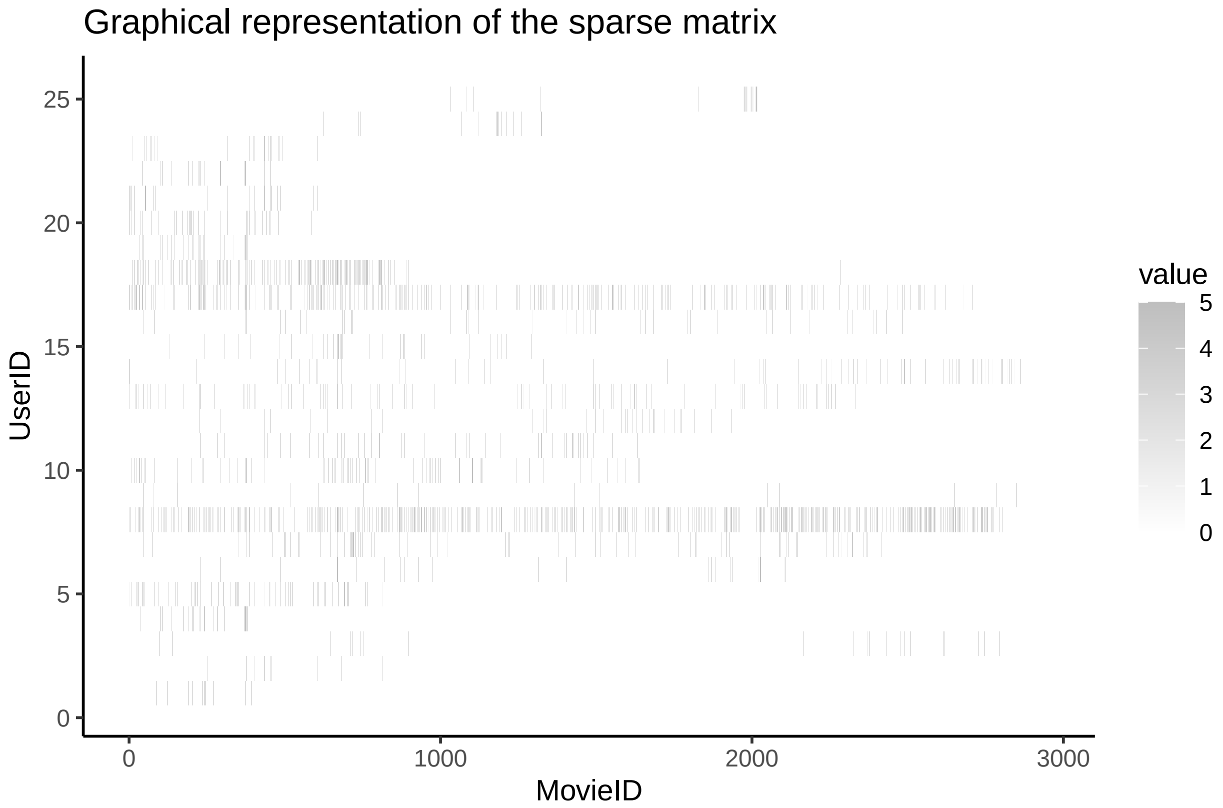
Based on the two plots, it is clear that in general, the distribution of ratings is broadly similar irrespective of the genere. Based on this cursory investigation, it would appear that any genre effect is limited at the single tag level.

Transforming the data into a matrix

Due to what has been identified thus far, the size and quantity of the data is extremely large, 69,878 distinct users rating between 1 and 6,616 distinct movies. In total, the data contains 9,000,055 ratings. To understand this data more clearly, a matrix can be created. A matrix is a re-organisation of the data from 6 columns to the creation of a data structure where along the columns we have the movieIds and along the rows we have the userIds.

What this creates is a table with the intersecting colum/row being the rating for a given movieId by a given user. As the investigation into the userIds discovered, not every user rates every movie, in fact, the user rating the highest number of movies rated less than 10% of the total movies rated within the dataset. Due to this, the matrix will have a lot of empty cells where no data is available.

To aide in the conceptual understanding, the data set has been converted into a sparse matrix and then graphically represented.



Data source: MovieLens 10M transformed into sparse matrix, first 25 users

The graphical representation of the matrix shows the first 25 userIds on the y axis, and the movieIds on the x axis. The grey squares are the ratings given by the user for the corresponding movie, with intensity of the colour denoting the rating given. The white space illustrates the movieIds with which the user has not rated the movie. As the graphic shows, the majority of the generated matrix is without any data points.

Bringing it together

The data analysis and transformation has enabled an overview of the edx data set. With such a large dataset, some interesting patterns can be discerned about the number of users, quantity of ratings they give, movie quantities, and genres. The concept of the matrix and some basic investigation of the potential bias in the data around genres have been explored.

Before creating the recommender models, there is a critical concept to cover. Within any modellig process there is a requirement for a training data set and a validation data set. In the instance of this report, this is the edx and validation data partition originally formed. To ensure that the models are generated independant of the validation data set, the edx data will be partitioned again into a 90% train and 10% test set. Using these two data sets, all models will be produced.

The reason for not using the validation set is that it will result in overfitting of the model, leading to the model performing under expectation when it is fed new data.

Based on the results of the different methods in the model development, the most efficent regression model will be selected to run against the actual validation dataset. Up to the point of running the recommender model on the validation data set, the models will use the partioned eddx dataset to test and train as well as cross validation of any tuning parameters.

Recommendation Models, Calculating the accuracy

To calculate the accuracy of the different models being produced, a single value of accuracy will be used as the determinant of the models recomendation accuracy. The measure in the context of this report will be the Root Mean Square Error (RMSE).

RMSE is constructed from the residuals between the prediction, denoted by $\hat{Y}_{u,i}$ for the ith observation and $Y_{u,i}$ the actual ith observation; thus the residual is calculated as:

$$\hat{Y}_{u,i} - Y_{u,i}$$

This residual can be a positive or negative. For examply prediction $\hat{Y}_{u,i} = 5$ and the actual value $Y_{u,i} = 4.5$, the resdiaul would be $5 - 4.5 = 0.5$. As there are 999,999 ratings to be predicted, the sum of all these residuals would not be useful as positive and negative values would cancel one another out. Thus, the RMSE takes the square of the residual to create an absolute psotivie error, take the average absolute error and square root it.

Interpreting the RMSE is similar to using a standard deviation on a population; the value can be used to understand the standard error seen within the predictions from the actual value. Anything over 1 would be a prediction an entire star rating away from the users actual rating.

The formula for RMSE includes N, the number of observations and is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{Y}_{u,i} - Y_{u,i})^2}$$

Regression Models

Regression models form part of the toolkit in the machine learning world. Regression was the first step outlined in the winning team of the Netflix prize algorithim. The following models are outlined in Irazarry.R, 2018 Ebook, Introduction to data science. This report will follow these for these regression models.[3]

The first model is a simplistic approach. Considering the sparse matrix the data is formatted too, the simplest method for approaching a score for every movie in the data set is to fill every recommendation with the average rating given to every movie. In simple terms, using the actual average rating. This would be a model that assumes that all varience form the average is random variation. This model would be denoted as:

$$Y_{u,i} = \mu + \epsilon_{u,i}$$

Where $\epsilon_{u,i}$ denotes the independant errors sampled within the same sample dataset and μ denoting the 'true' rating. We know that the μ least sqaures estimate minimises the RMSE, which in this instance is the average rating.

Based on this model:

$$\hat{Y}_{u,i} = \hat{\mu}$$

The average rating for all movies is:

[3] Irazarry.R, 2018 https://rafalab.github.io/dsbook/large-datasets.html#recommendation-systems-as-a-machine-learning-challenge accessed on 15.12.2019

```
## [1] 3.512456
```

Thus we can calculate the first models RMSE as:

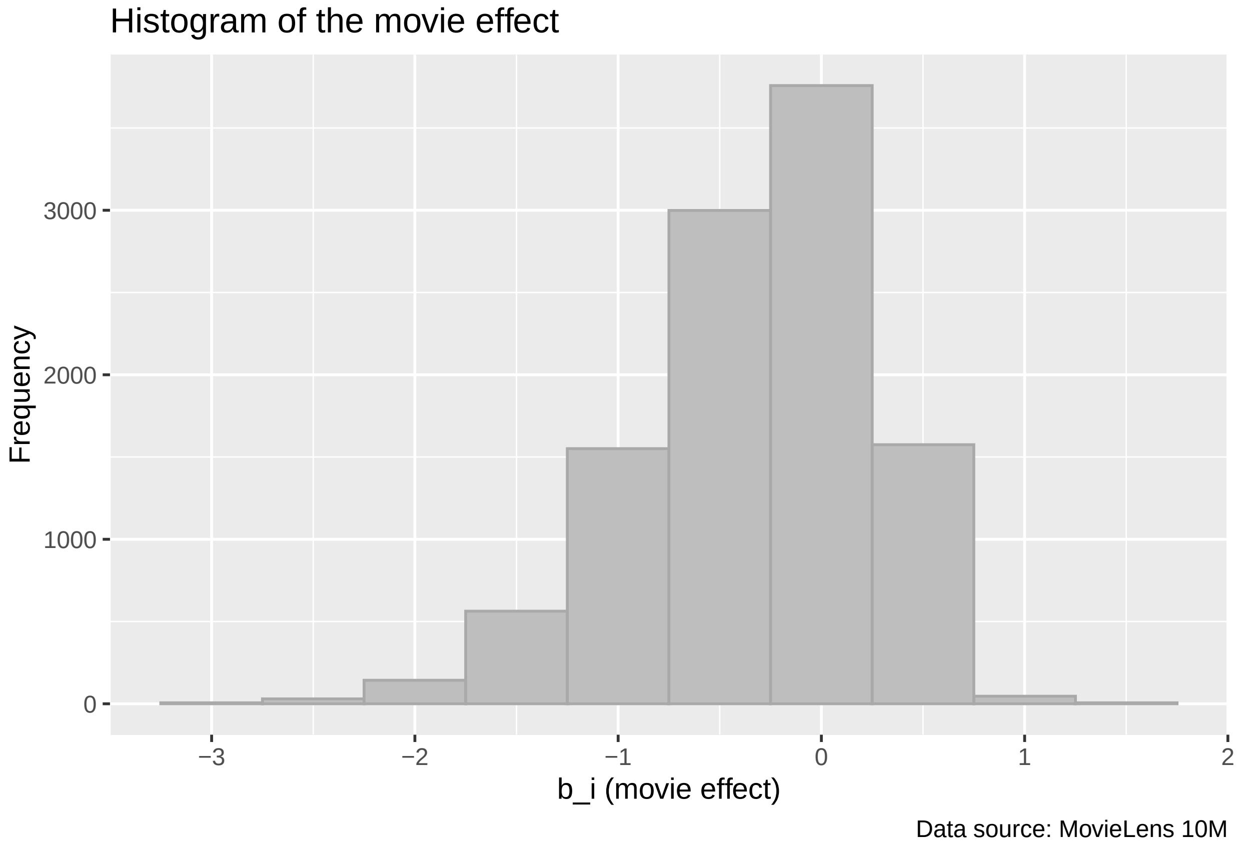
```
## [1] 1.060054
```

A note on this, if any other single number is plugged into every cell and used as the prediction, the RMSE will be higher. This is a promising start, for just the average the average error is only 1 star away from the actual users predictions.

That said, as all of the investigation showed, there are other factors that can be used to improve the model. A good place to start is the movies. Intution and the data tell us that different movies are rated differently. What can thus be done is to calculate the movie effect, b_i. In text-books this is often termed as an effect, in the Netflix Prize it was often referred to as a bias. For the purpose of this report, it will be classified as an effect but can be used interchangebly.

$$Y_{u,i} = \mu + \epsilon_{u,i} + b_i$$

It is known that $\hat{b}_i$ is the average of $Y_{u,i} - \hat{\mu}$ for each movie i.



This graph can be interpreted as the average rating ($\hat{\mu}$) of 3.5 + or - the value of b_i; therefore a +1.5 would be a perfect 5/5 average rating.

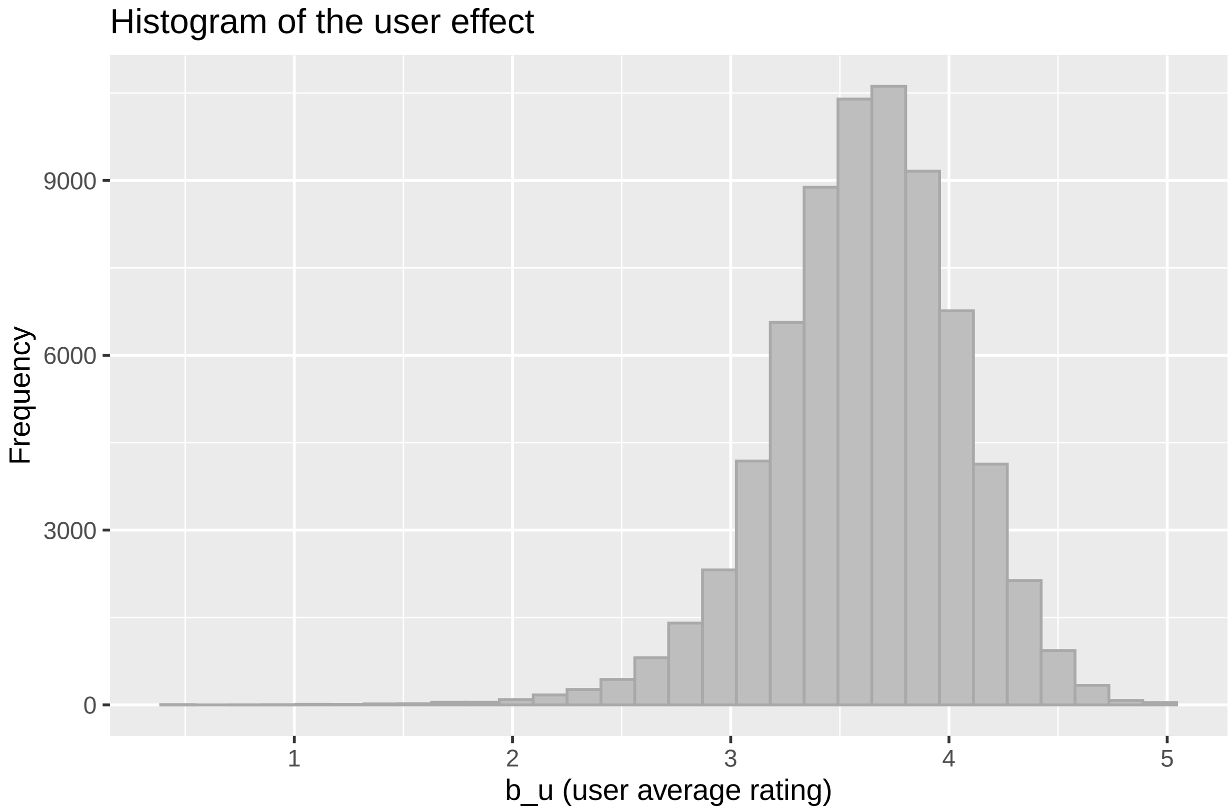
Considering the movie effect, the equation the prediction algorithim can be updated to include the movie related adjustment, thus creating the model:

$$\hat{Y}_{u,i} = \hat{\mu} + \hat{b}_i$$

```
## [1] 0.9429615
```

Based on this model, the RMSE improves just by factoring in the average rating for a movie.

Considering the success of the moive effect, it is worth considering the second biggest factor in the dataset, the users. Speculatively, it can be assumed that some users are naturally more likely to rate all movies above the average rating, and conversely others will always be harsher critics.



Data source: MovieLens 10M for all users with >100 ratings

The historgram clearly shows that some users are giving all very high ratings, and others are very low.

Based on this pattern, another adjustment to the model can be undertaken by adding the user effect, thus making the model:

$$Y_{u,i} = \mu + \epsilon_{u,i} + b_i + b_u$$

Where b_u is the spcific user effect. Thus the model is taking the average movie rating, the average rating by user and adjusting each pairing of user and movies to adjust towards the respective point away from the average of all movies scores.

Thus, $\hat{b}_u$ is calculated as:

$$Y_{u,i} - \hat{\mu} - \hat{b}_i$$

Thus our model prediction is:

$$\hat{Y}_{u,i} = \hat{\mu} + \hat{b}_i + \hat{b}_u$$

```
## [1] 0.8646843
```

Based on this adjustment, the model again impoves by taking the predictions to within 0.9 stars of the actual ratings. Within the Netflix Prize winners outputs, they termed this the Anna effect.

In terms of conceptualising the model, the matrix now has a unique prediction for every user movie combo. In effect, this can be visualised by thinking that for every column there is an adjustment given of $\hat{b}_i$ based on the average rating for that movie. For every row in our matrix, there is an adjustment by $\hat{b}_u$ based on the average rating from that user. Based on these two factors and taking the average rating for all movies, the user is given a prediction for that movie.

Regularization

The regression models created thus far have seen a significant improvement on just plugging the average movie into every prediction. However, the model is not perfect and is only as good as the data it is using as a predictor.

Based on the current model, the best movies within the entire data set are as follows:

title	b_i	n
Hellhounds on My Trail (1999)	1.487544	1
Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)	1.487544	1
Shadows of Forgotten Ancestors (1964)	1.487544	1
Fighting Elegy (Kenka erejii) (1966)	1.487544	1
Sun Alley (Sonnenallee) (1999)	1.487544	1
Blue Light, The (Das Blaue Licht) (1932)	1.487544	1
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)	1.237544	4
Life of Oharu, The (Saikaku ichidai onna) (1952)	1.237544	2
Human Condition II, The (Ningen no joken II) (1959)	1.237544	4
Human Condition III, The (Ningen no joken III) (1961)	1.237544	4

All of the listed movies are relatively obscure movies and yet they all recieve the highest ratings within the current model.

Coversely, if the model is to select the worst rated 10 movies based on the movie efftect b_i, then the same pattern emerges, a list of obscure movies:

title	b_i	n
Besotted (2001)	-3.012456	1
Hi-Line, The (1999)	-3.012456	1
Accused (Anklaget) (2005)	-3.012456	1
Confessions of a Superhero (2007)	-3.012456	1
War of the Worlds 2: The Next Wave (2008)	-3.012456	2
SuperBabies: Baby Geniuses 2 (2004)	-2.767775	47
Disaster Movie (2008)	-2.745789	30
From Justin to Kelly (2003)	-2.638139	183
Hip Hop Witch, Da (2000)	-2.603365	11
Criminals (1996)	-2.512456	1

So what is occuring here?

This is an issue of a low number of ratings for a movie within the data frame. One high or low rating dispraportionately will impact the data set and cause the movie to appear dispraportionately better or worse based solely on the low number of ratings.

For the most part, it can also be seen that the number of ratings is reasonably low. For the top 10 movies, all of the movies have <5 ratings. Within the worst movies predicted by the model, the first 5 have 2 or less ratings. *From Justin to Kelly* looks likely to be a low rated movie.

The issue then is that a low number of ratings can have a dispraportionate effect on the model. Given that the data for validating the model is also unlikely to feature this movie, it will likely become an irreguraly tested hypothesis. The outcome of this could be that Netflix displays an awful movie to a large number of people inappropriately.

Thus, the use of regurlarization allows the model to penalise small sample sizes. The concept behind regurlarization is to constrain a single rating from becoming over-represented within the model. In the instance of a low number of ratings, in all likely hood the average for the entire dataset μ may be a better prediction than the model:

$$\hat{b}_i = Y_{u,i} - \hat{\mu}$$

With this in mind, an equation containning a penalty is added.

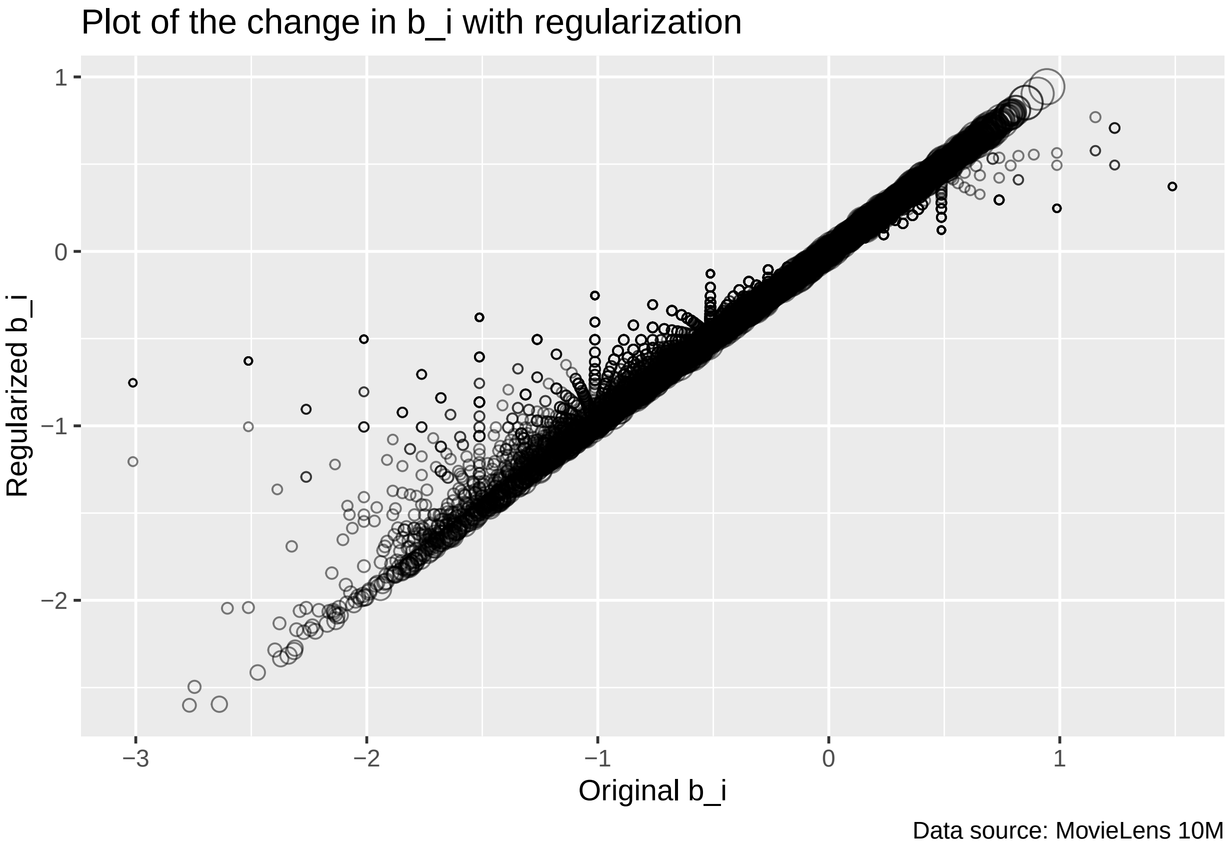
$$\frac{1}{N}\sum_{u,i}(Y_{u,i} - \mu - b_i)^2 + \lambda\sum_{i} b_i^2$$

The equation has two constituent parts, the first just being the least squares, the second is a penalty that causes any big b_i movie adjustment factors to be penalised by the the constant λ. Using calculus it can be shown that large sample szes make this penalty clause redundant:

$$\hat{b}_i(\lambda) = \frac{1}{\lambda + n_i}\sum_{u=1}^{n_i}(Y_{u,i} - \hat{\mu})$$

Where n_i is the number of ratings in the dataset for that moive i, the impact of the penalty λ thus becomes negated as when n_i is large then the equation becomes $n_i + \lambda \approx n_i$. However, if n_i is small, then the estimate of $\hat{b}_i(\lambda)$ is shruken towards 0, removing the impact of small sample sizes from the model. Based on this penalty, λ becomes an important number as the larger the value, the greater the penalty for small sample sizes is.

Based on this new penalty clause, the impact of the regularisation can be shown via the following graphic. The $\lambda = 3$ in this model for simplicity purposes.



The graph can be interpreted by finding a point, the size of the point is based on the size of the sample n_i. For larger points, the regurlarisation equation will have little effect as the calculus showed. Thus if the original adjustment was $b_i = 1$, then the regularised value should $\hat{b}_i(\lambda) \approx 1$. However, for the small sample sizes n_i where there has been less ratings for a movie, then the point is smaller. The original was $b_i = 1$, however, after regularisation due to the small sample size the value is $\hat{b}_i(\lambda) \approx 0.25$.

Based on the regularised results, using $\lambda = 3$ again, the top 10 movies are now:

title	b_i	n
Shawshank Redemption, The (1994)	0.9439985	25188
Godfather, The (1972)	0.9040256	15975
Usual Suspects, The (1995)	0.8539646	19457
Schindler's List (1993)	0.8515069	20877
Rear Window (1954)	0.8121492	7115
Casablanca (1942)	0.8069487	10141
Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)	0.8026192	2633
Double Indemnity (1944)	0.7937591	1950
Seven Samurai (Shichinin no samurai) (1954)	0.7927556	4648
Paths of Glory (1957)	0.7918879	1410

And the lowest rated moveis after regularization are:

title	b_i	n
SuperBabies: Baby Geniuses 2 (2004)	-2.601708	47
From Justin to Kelly (2003)	-2.595588	183

title	b_i	n
Disaster Movie (2008)	-2.496172	30
PokÃ©mon Heroes (2003)	-2.413736	124
Barney's Great Adventure (1998)	-2.335009	186
Glitter (2001)	-2.316477	311
Gigli (2003)	-2.290493	281
Carnosaur 3: Primal Species (1996)	-2.285309	61
Pokemon 4 Ever (a.k.a. PokÃ©mon 4: The Movie) (2002)	-2.274040	188
Faces of Death 6 (1996)	-2.183017	73

In both instances, there are very few suprises. The top 10 movies are all but one famous movies with a large number of ratings. The lowest rated movies are all obscure titles with a reasonable number of ratings.

Despite this penalty factor being applied, there is still the movie More (1998) with only 7 ratings being displayed in the top 10. This may be a function of $\lambda = 3$ and will be explored in more detail later.

```
## [1] 0.9429453
```

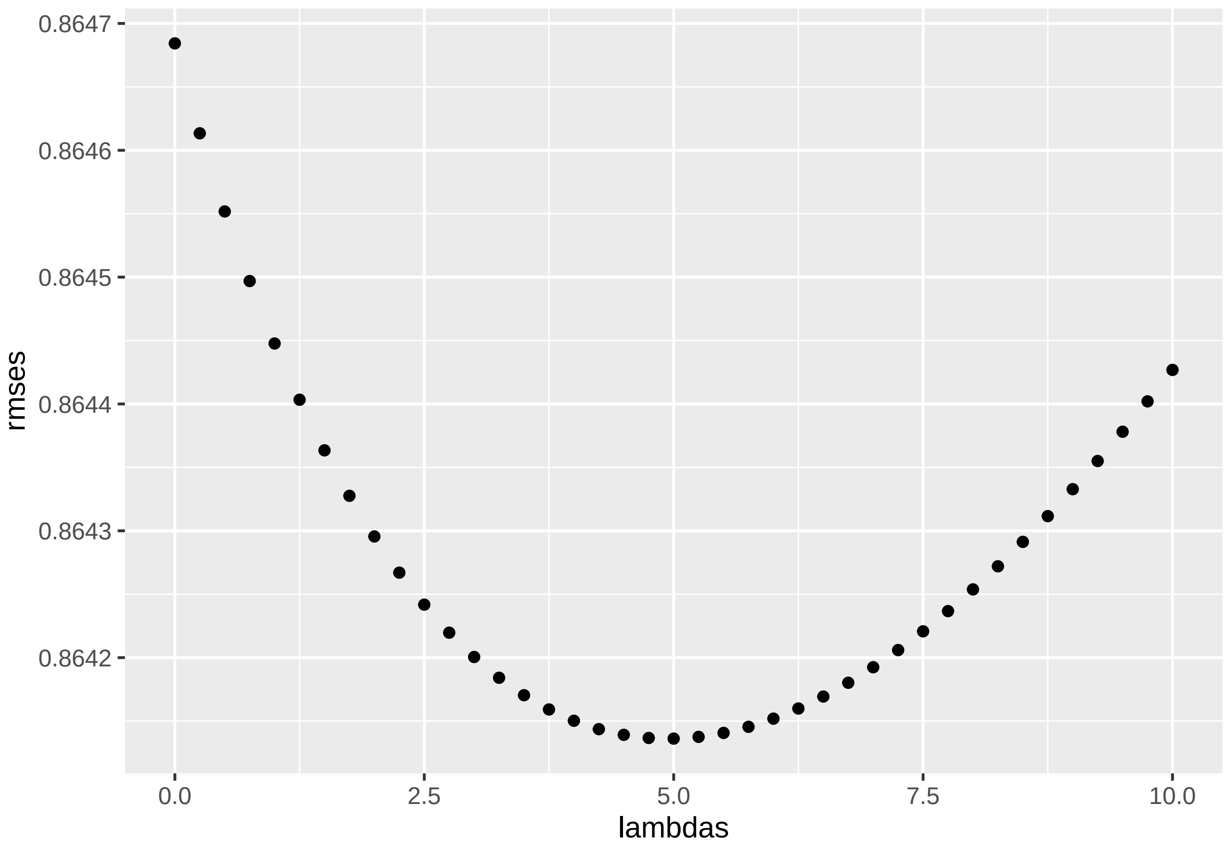
Based on the the new regularised model for the movies, there is improvement compared to the original Movie Effect Model. However, it is not better than the Movie + User Effect Model. Thus, it makes sense to also regularize the User Effect for those users with a low number of ratings, in the same way as before, this is to reduce the chances of users with two ratings looking overly positive or negative.

```
## [1] 0.8642005
```

Based on the regularized Movies and User Effect model with a $\lambda = 3$, the overall model is again improved by a marginal increment.

To create a further improvement in the model, the appearance of More (1998) in the top 10 movies needs to be explored. This movie scored the fourth highest movie effect out of all movies within the dataset, despite only being given $n_i = 7$ ratings.

A hypothesis for this is that $\lambda = 3$ is too small to effectively penalise the quantity of ratings. Based on this, λ can be changed to test for the best value to act as a penalty. Rather than just randomly plugging in numbers and re-running the model, it is possible to use an algorithim to re-run the same process repeatedly for as many iterations as is requested to then ascertain the best value for λ. This is known in data science as a *tuning paramter* tuned using cross-validation. The algorithim will go through the iterations, test the results and then select the λ which results in the best output parameter for the model using the train and test set.



```
## [1] 5
```

The graphical plot of lambdas quickly shows which lambda is the most efficent as well as using the data to select the correct lambda to use in the model.

In this model, there is also the addition of regularization for the user effect. This requires the model to be updated to:

$$\frac{1}{N}\sum_{u,i}(Y_{u,i}-\mu-b_i-b_u)^2+\lambda(\sum_i b_i^2+\sum_i b_u^2)$$

Based on this $\lambda = 5$, the model performs better than the nominally chosen 3.

```
## [1] 0.8641362
```

Table 9: Results from the recommender models based on the training dataset

Method	RMSE
Just the average	1.0600537
Movie Effect Model	0.9429615
Movie + User Effects Model	0.8646843
Regularized Movie Effect Model	0.9429453
Regularized Movie & User Effect Model Lambda =3	0.8642005
Regularized Movie & User Effect Model ~ Lambda	0.8641362

As the table shows, the models have improved with the different additions added to them. Based on the new lambda figure, the hypothesis regarding $\lambda = 3$ being too small has been shown to be correct. Based on the newly regularised data with $\lambda = 5$, the movie More (1998) no longer appears in the top ten films. Thus confirming niavely that the regularization has been improved.

title	b_i	n
Shawshank Redemption, The (1994)	0.9424978	28015
Godfather, The (1972)	0.9026465	17747
Usual Suspects, The (1995)	0.8531914	21648
Schindler's List (1993)	0.8508447	23193
Casablanca (1942)	0.8075991	11232
Rear Window (1954)	0.8056787	7935
Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)	0.8020419	2922
Third Man, The (1949)	0.7976163	2967
Double Indemnity (1944)	0.7965030	2154
Paths of Glory (1957)	0.7937292	1571

The Regularized Movie & User Effect Model with the cross validated λ tuning parameter performed the best out of all of the regression models and will thus be run against the validation data set using the full edx data set to train the model with the $\lambda = 5$ for the tuning parameter.

Based on using this model, the final RMSE for the Regulized Movie effect and user effect model is:

Method	RMSE
Regularized Movie & User Effect Model with cross validated Lambda	0.8648177

```
## [1] 0.8648177
```

Conclusion & Learning Outcomes

In conclusion using a regression model with a Movie Effect and User Effect adjustment that is regularised based on a cross-validated lambda can lead to a prediction model achieving an RMSE of 0.8648177. This figure can be improved through the use of Matrix Factorisation among certain other techniques.

The applicable learning from this work is to be able to use regularised regression models in other contexts. Regularisation can be applied to any complex predicition model, such as forecasting demand for a range of products with varying requirements by product. The learning around data wrangling, data analysis, data partitioning, model selection, fitting, cross-validating and then final model presentation can be applied to a wide range of technical problems, from forecasting, natural language processing, prediction models, recommender models among many other applications. The skills learned using RMarkdown can be applied for creating a number of reports which format in a highly proffessional manner quicly and simply in a range of formats.