Movie Recommendation Model

Chandra Sekhar Polisetti

14 December, 2021

Contents

1 Introduction

Movie streaming applications like Netflix , Amazon prime, Disney etc request the users to rate movies that they watch, and eventually the rating collected from several users would be used to predict the rating that a potential user would give to a movie, and furthermore based on the predicted ratings movie recommendations would be suggested to the users.

Most of these movie steaming applications use machine learning based Movie Recommendation Model to predict how a given user would rate a specific movie, and this report presents you a Movie Recommendation Model which predicts the user ratings for the movies.

2 Overview

A large Movielense dataset, which has around 10 million ratings given to 10677 movies by 69878 distinct users, is provided by the capstone project, and is used to build the Movie Recommendation Model.

This report first presents the analysis of the given data to gain insights into the data. Based on the insights gained through analysis, a machine learning model would be built step by step from the scratch.

In each step the model would be evaluated to assess its performance. Root Mean Square Error (RMSE) would be used as a measure to evaluate the performance of the model and the model with least RMSE would be chosen.

The objective is to build an model with a target RMSE < 0.86490.

Here is the High level process followed to build the machine learning model, and is also the outline of this report, each step will be dealt with great detail in the subsequent sections

i. `Data Preparation - Download movielense dataset, split the data for training and testing.`

ii. `Analysis - Analyze the data to gain insights into the data.Various data visualization techniques are used to understand the impact of the Movie, User , Genre , Movie Release Year , Movie Rated Date.`

iii. Methods - Insights gained in the analysis step would be used to build a Movie Recommendation Model that with a target RMSE < 0.86490.

iv. `Results - Movie Recommendation Model prediction results and how much of the result was contributed by each of the effects used in the model are summarized .`

v. `Conclusion - Conclude the report with limitations and future work`

3 Data Preparation

Data Preparation downloads and wrangles the data to facilitate the data analysis. Primary focus would be to download, clean the data, and partition the data for validation and training.

3.1 Data Cleanup

The following are the sequence of steps that are performed for the data cleanup.

1) Download the dataset from http://files.grouplens.org/datasets/movielens/ml-10m.zip
2) Read the ratings data from "ml-10M100K/ratings.dat" file and name the columns of data frame as "userId", "movieId", "rating", "timestamp"
3) Read the movies data from "ml-10M100K/movies.dat" file and name the columns of data frame as "movieId", "title", "genres". Convert the movieId to numeric, convert the title and genres to character data types.
4) Join the movies and ratings data frames by movieId and call it as movielense.

3.2 Data Partition for Validation and Training

Once clean dataset is obtained in the above step, data in the movielense dataset is split into two parts, one for the training and optimizing the model, and the other for validating the model.

Following are the steps that are performed for data partitioning

1) 10% of the data rows from the movielense dataset are randomly selected and placed in *validation* dataset, and this dataset will be kept aside for performing the validation of the final Movie Recommentation Model. This dataset will not be used for training and optimizing the model. This dataset is not used for training mainly to avoid overfitting the data.
2) The remaining 90% of the data rows from the movielense dataset are brought into the edx dataset.This dataset is mainly used for training and optimizing the algorithms.The edx dataset would be further partitioned into training and test datasets down the line in the methods section to facilitate the building and optimizing the Movie Recommendation Model.
3) It was made sure that all the movieIds and userIds in validation set are present in the edx dataset. The movieIds and userIds that are not present in the edx are added back to the edx dataset.This step is performed to make sure that the machine learning algorithms are using the same movies and users used during training to make the predictions.

After all the above steps the following datasets would be obtained

1) edx - This dataset has 9000055 records and this will be primarily used for training the machine learning algorithm
2) validation - This dataset has 999999 and this will only be used to get the final predictions.

4 Analysis

Once clean data is available and is partitioned into validation and edx. We keep the validation dataset aside for the final validation of the Movie Recommendation Model, and this will not be used for analysis, training and optimizing the algorithms.

The edx dataset is used for analysis, training and optimizing the algorithms.

This section would mainly analyze data in the edx dataset by starting with the preliminary understanding of the overall data content, and then move on to a deeper understanding of the individual predictors movieId, userId, timestamp and genre, of the dataset. Data visualizations techniques will be used wherever necessary to backup the understanding of the data.

4.1 Training data set fetures

Here is a quick look at the movielense data properties and its sample data

Dimensions of the dataset

```
## [1] 9000055       6
```

Here is the structure of the dataset

```
## Classes 'data.table' and 'data.frame':   9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 838
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)"
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller"
##  - attr(*, ".internal.selfref")=<externalptr>
```

Here is some sample data

Table 1: First 6 rows of data from edx dataset

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

As we can see from the above , each row is a particular user's rating given for a movie, where userId is the unique id of the user , movieId is the unique id of the movie and rating is the numeric

rating given to the movie , timestamp is when the movie is rated , title is the title of the movie, and genres are the categories of the movie.

Lets look at rating value frequencies in multiple ways

See below to look at how users gave ratings to different categories of rating

Table 2: Individual rate count and its percentage

Rating	No. of Ratings	Percentage Of Rating
4.0	2588430	0.2876016
3.0	2121240	0.2356919
5.0	1390114	0.1544562
3.5	791624	0.0879577
2.0	711422	0.0790464
4.5	526736	0.0585259
1.0	345679	0.0384085
2.5	333010	0.0370009
1.5	106426	0.0118250
0.5	85374	0.0094859

Here is a barplot of how the rating values are distributed

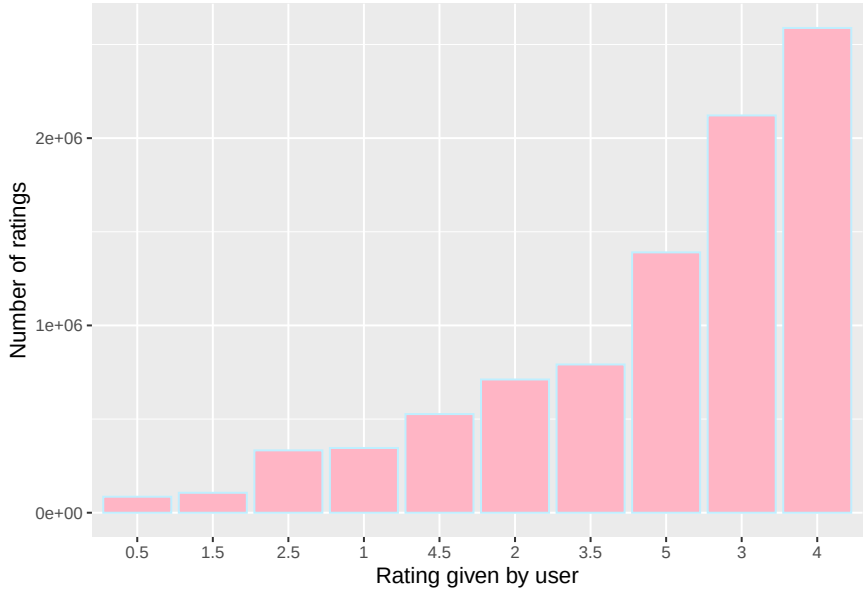

Figure 1: Rating Frequency Distribution

Lets look at the boxplot of rating values to look at the 5 point summary of the rating values

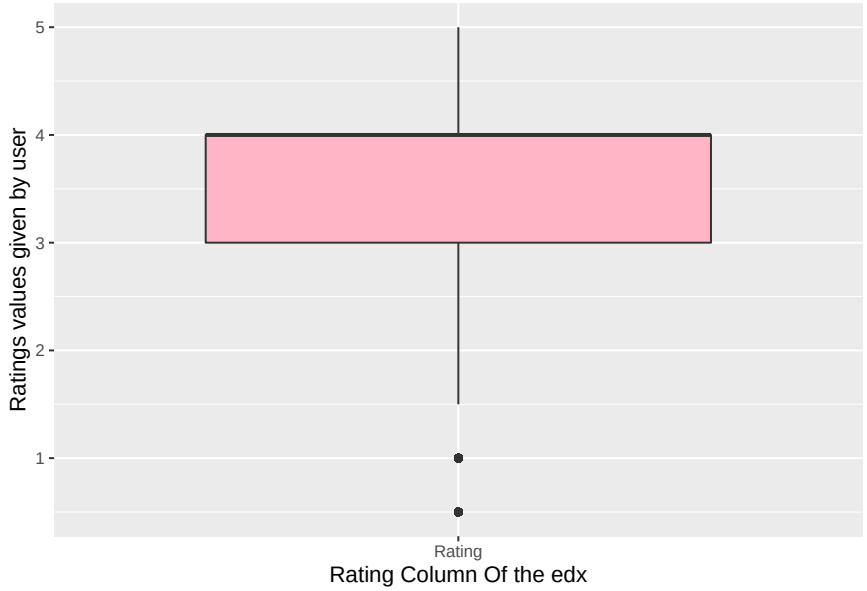

Figure 2: Rating values 5 point summary

We can infer the below points by analyzing the above graphs

1) Rating values are not uniformly distributed
2) Users have around 85% of the times have given 4 , 3 , 5 , 3.5 and 2, and around 15% of the time gave other ratings.
3) Users have given 4 and 3 50% of the times and gave 0.5 and 1.5 around 2 % of the time.
4) Users have more whole number rating values than decimal value ratings.

4.2 Movie effect on rating

Lets look at data to understand the impact of a movie on the rating.

Here are the number of distinct movies and users

Table 3: Distinct users and movies

no_of_movies	no_of_users
10677	69878

As there are more users than movies , same movie rated by more than one user.

Lets look at the average rating per movie.

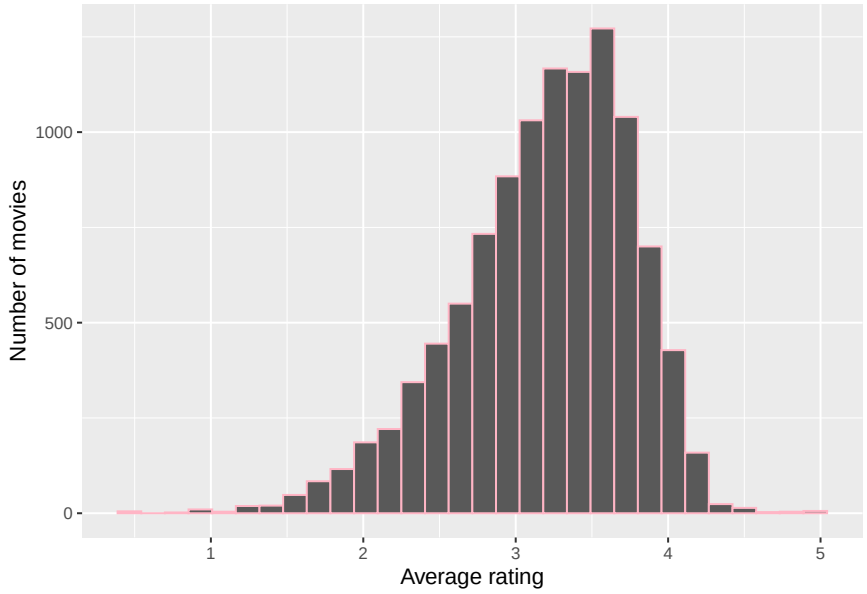

Figure 3: Average rating per Movie

As we can see from the above graph that there is a significant variation in the average rating a movie received and we can capitalize this effect in building the Movie Recommendation Model.

In order to capitalize the movie effect we still need some more research to see how number of ratings are distributed.

Let's look at number of ratings per movie

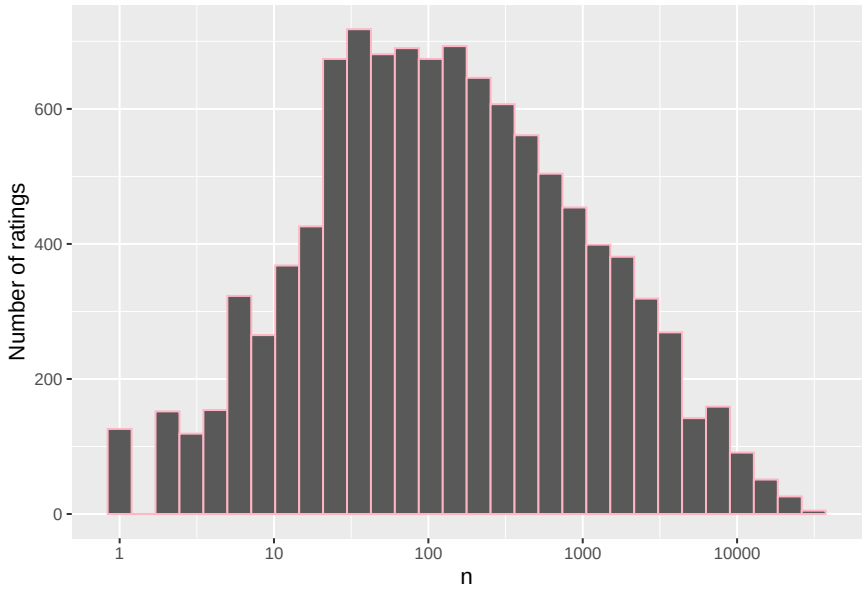

Figure 4: Number of ratings per movie

As we can see from the above picture some movies got more ratings while others have less ratings.

We know from intuition that the famous movies get more ratings while independent movies get less ratings as fewer people watch them.

Lets confirm our intuition by looking at the top 5 most rated movies and top 5 least rated movies.

Table 4: Most Rated Movies

title	n
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Jurassic Park (1993)	29360
Shawshank Redemption, The (1994)	28015

Table 5: Least Rated Movies

title	n
4 (2005)	1
Delgo (2008)	1
Hexed (1993)	1
Love (2005)	1
Vinci (2004)	1

From all of the above we can summarize the findings as below,

1) There is a significant variation in average rating per movie and we can capitalize this in building the Movie Recommendation Model.
2) There is a lot of variance in the number of ratings per movie, and there are many movies with low number of ratings, we need to regularize the movie effect while building the Movie Recommendation Model.

4.3 User effect on rating

As we know from intuition that different users give different ratings to a Movie, and while some users provide rating and others do not care to rate movies.

We can expect a variation in average user ratings similar to that of a movie effect.

Lets confirm our intuition with data.

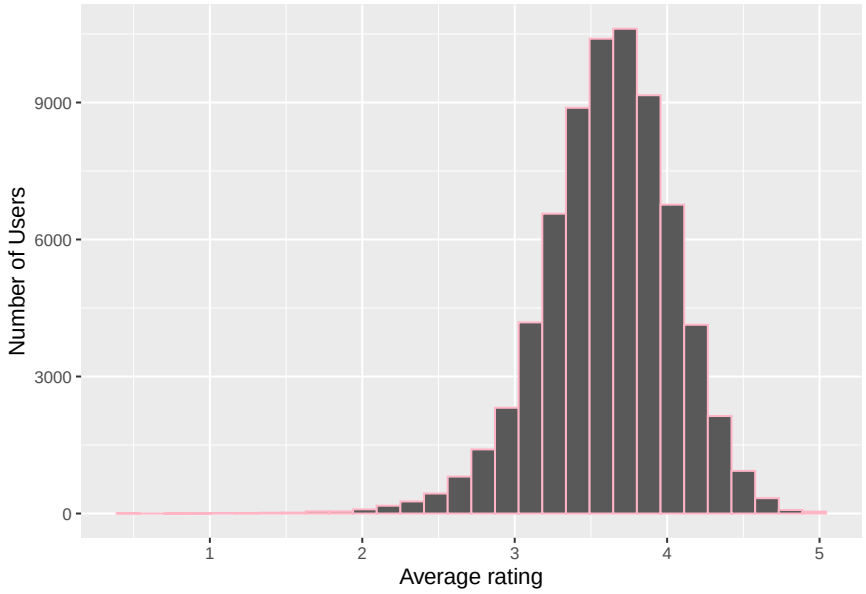

Figure 5: Average rating per User

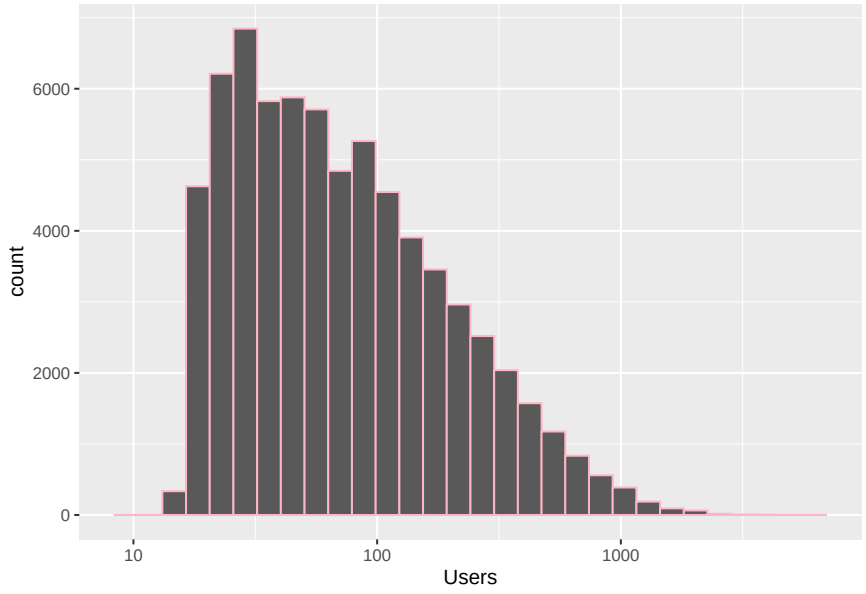

Figure 6: Number of ratings per User

```
## NULL
```

Here is the summary of the user impact based on rating

1) There is a significant variation in the user average rating that we can capitalize while building the Movie Recommendation Model
2) As there is also a lot of variance in the number of ratings given by users we need to regularize the ratings to penalize the ratings obtained from smaller numbers of ratings

4.4 Temporal Effect on Ratings

The following temporal effects are going to be discussed in this section

1) Year on which a movie is released.
2) Date on which a movie is rated

4.4.1 Impact of Movie Release Year

Lets analyze the impact of the movie release year on the rating. Movie release year is embedded in the title column so we need to extract the year in order to analyze it.

As we can see from the movie analysis reports, see Table -4 above, year is part of the title string, and it is always in the end of the title. It always start 5 characters from the end of the string and ends 1 character from the end of the string. Here is an example of the title *Delgo*(2005). We can make use of these facts and exact the year from title.

Lets look at the average rating against the release year of the movie, and also the number of ratings per year.

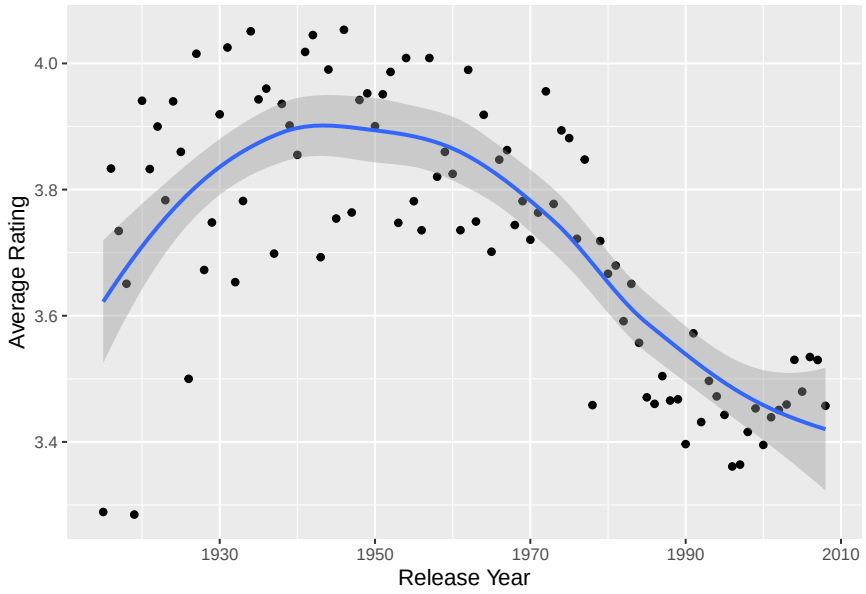

Figure 7: Average Rating vs Movie Release Year

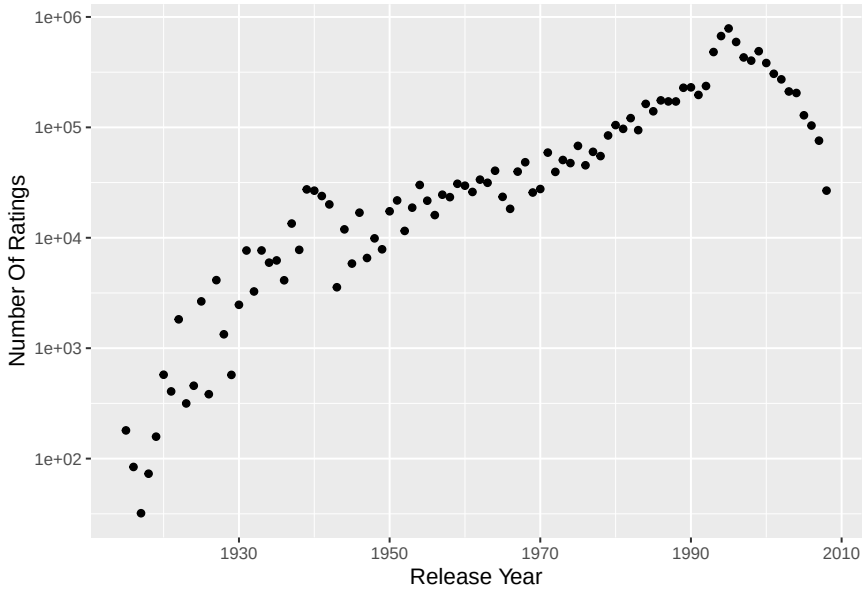

Figure 8: Number Of Ratings vs Movie Release Year

Here are the take away points from the analysis

1) There is a slight variance in the average rating by movie release year, so we can consider year while building the Movie Recommendation Model
2) As there is significant variation in the number of ratings per movie release year , we need to consider regularizing the average ratings per year to penalize the years which have smaller number of ratings

4.4.2 Impact of Movie Rated Date

Lets look at the impact of the date on which a movie is rated. First the movie rated date has been extracted from timestamp field in the dataset and has been rounded to week to smooth the data.

Here are the average rating per week and the number of ratings per week graphs.

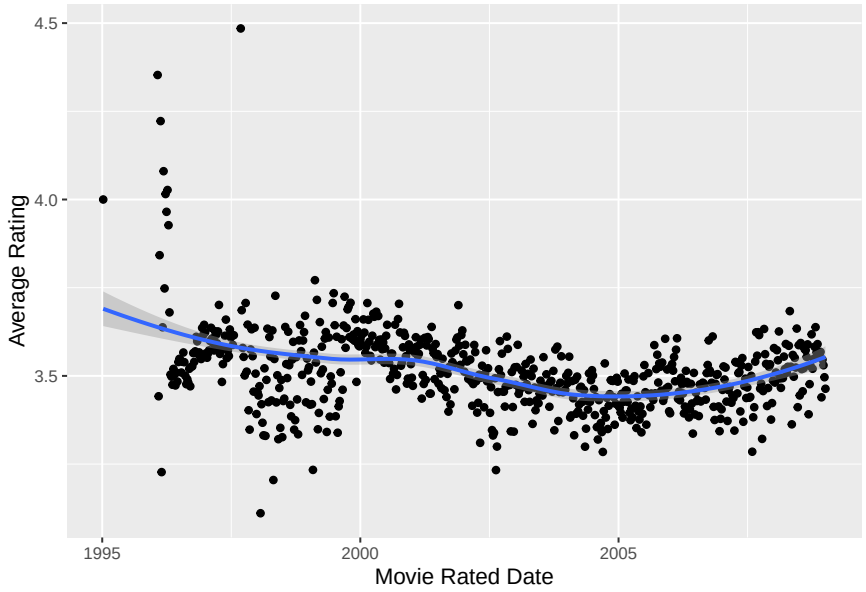

Figure 9: Average Rating vs Movie Rated Date

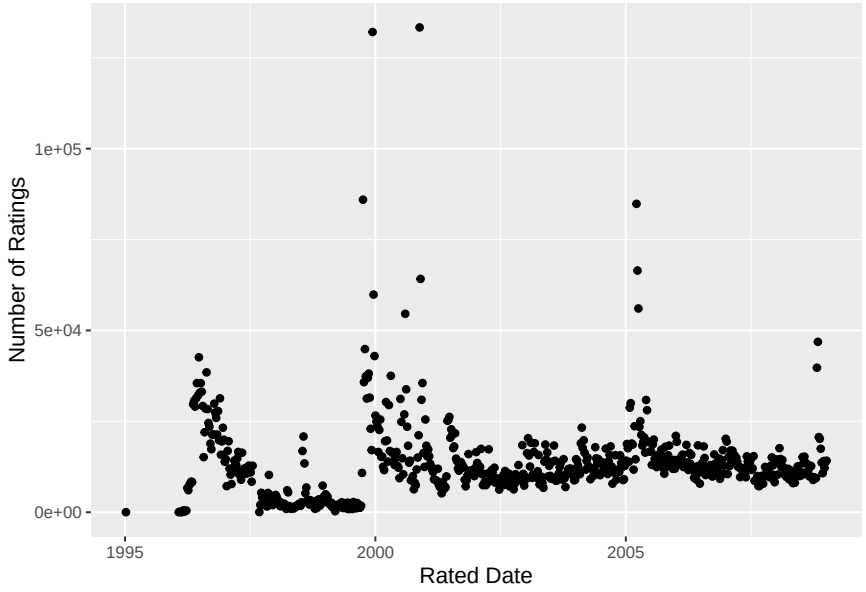

Figure 10: Number Of Ratings vs Movie Rated Date

Here is the summary of the impact of Movie Rated Date on the rating

1) There is a some variance in the average ratings per week. Even though the variance is not significant we could still incorporate this factor and see whether this explains some of the the unknown variance.
2) As the number of ratings varies across the Movie Rated Dates we need to regularize the average ratings while incorporating the Movie Rated Date into the Movie Recommendation Model

4.5 Genre Effect on Rating

As we know from intuition that peoples interest varies across genre's. To see how significant the impact, we need to look at the below graphs.

Average rating per movie genre would help us understand the impact. see below for these figs

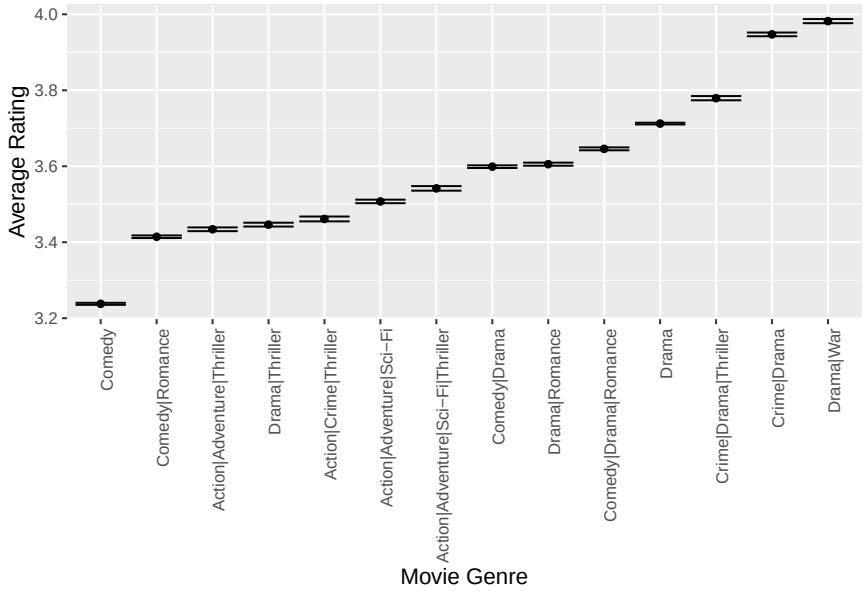

Figure 11: Average Rating vs Movie Genre

Table 6: Genre's with most ratings

genres	n
Drama	733296
Comedy	700889
Comedy|Romance	365468
Comedy|Drama	323637
Comedy|Drama|Romance	261425
Drama|Romance	259355
Action|Adventure|Sci-Fi	219938
Action|Adventure|Thriller	149091
Drama|Thriller	145373
Crime|Drama	137387

Here is the summary of the impact of Movie Genre on the rating

1) There is a some variation in the average ratings across Movie Genre's.
2) *Comedy* has got lower average rating while $Drame|War$ has got higher ratings
3) There is a lot of variance in the number of ratings a Movie Genre received and we need to perform regularizing to penalize the ratings with low number of ratings.

5 Methods

The following is the high level outline of this section, each step would be detailed out in the subsequent sub-sections.

1) Data Wrangling
2) Partition edx dataset into train and test datasets
3) Algorithm evaluation criteria
4) Incrementally build Movie Recommendation Model with the below aspects
    a) Novice Model
    b) Movie Effect
    c) User Effect
    d) Genre Effect
    e) Movie Release Year Effect
    f) Movie Rated Date Effect
    g) Regularization
    h) Optimization
    i) Validation

5.1 Data Wrangling

We need to perform the below tasks, pre processing steps, to make the edx and validation datasets ready for training and validation repectively.

1) $MovieReleaseYear$ - We need to incorporate Movie Release Year effect to the Movie Recommendation Model,and for that,we need to add the Movie Release year to the both edx and validation datasets. Movie Release Year is embedded in the title column, and we need to pull out year from the title and save it as a separate attribute in the both edx and validation datasets.

2) $MovieRatedDate$ - We need to incorporate Movie Rated Date effect to the Movie Recommendation Model, and Movie Rated Date is not ready available to use it. But we can extract it from the timestamp field and add it to the edx and validation datasets for further processing. Firstly Movie Rated Date is obtained by converting timestamp attribute to datetime, and secondly the Movie Rated Date was rounded to the nearest *week* for smoothing the data.

5.2 Partition edx dataset into train and test datasets

edx and validation datasets were provided as part of the project. validation dataset is a holdout dataset which is primarily used for validating the final optimized Movie Recommendation Model. So edx is the only dataset that needs to be used for training and optimizing the model. As we need two independent datasets for training and testing the trained model we need to partition the edx dataset.

So edx dataset is partitioned into the following two datasets

1) train dataset, 80% of edx dataset, for training and optimizing the model

2) test dataset, 20% of edx dataset, for validating the trained model

The same method that we followed to partition the movielense into edx and validation datasets is followed to partition the edx into train and test datasets.

5.3 Algorithm evaluation criteria

Root Mean Squared(RMSE) Error is the measure used to evaluate the algorithm.

Here is the formula for calculating the RMSE.

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

In the formula shown above, $y_{u,i}$ is the actual rating provided by user i for movie u, $\hat{y}_{u,i}$ is the predicted rating for the same, and N is the total number of ratings predicted.

We train the model using the train dataset without using the test dataset. Once the model is built, we predict the ratings in the test dataset using the model, the predicted rating $\hat{y}_{u,i}$ and the actual rating in the test dataset $y_{u,i}$ will be used to calculate the RMSE using the above formula.

The objective of the project was to develop an algorithm with RMSE below 0.86490. RMSE of every improvisation to the model would be calculated and compared against the target RMSE to see whether the changes to model is taking us towards the target RMSE. Model improvisations would continue until the target RMSE is reached.

R^2 and $AdjustedR^2$ would be calculated to see what percentage of variance is explained by the model.

R^2 gives the percentage of variance in the ratings explained by the model without considering the number of parameters in the model, where as Adjusted R^2 gives the variance in the ratings explained by the model by considering the number of parameters in the model.

Here are the formulas that are used for R^2 and $AdjustedR^2$.

$$R^2 = \frac{SumOfSquares(mean) - SumOfSquares(fit)}{SumOfSquares(mean)}$$

$$AdjustedR^2 = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$$

A tracking table to log the RMSE results would be used in the subsequent sections to track the model progress.

Lets build a simple RMSE table to track the model performance

Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
Project Goal	0.86490	-	-	-

5.4 Build Movie Recommendation Model

Now that we have completed the analysis of the data and gained insights into the data, and datasets are ready for training and testing the Models, and loss function finalized in the previous section we are ready to build the machine learning model to predict the rating.

An incremental approach to build the algorithm has been taken in order to highlight the contribution of each of the effects in explaining the variance in the ratings.

This model built in this section is completely based on the insights gained in the analysis section.

Based on the analysis of the data , we have confirmed that the following factors have shown variance in the average ratings, and hence they will be incorporated into the model in the following sub-sections.

1) Movie Effect
2) User Effect
3) Genre Effect
4) Movie release year
5) Movie Rated date

5.4.1 Novice Model

Let's start by building the simplest possible recommendation Model, we predict the same rating for all movies regardless of user or movie

Here is the model.

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

Here, the actual rating for movie i by user u, $Y_{u,i}$, is the sum of this "true" rating, μ, plus $\epsilon_{u,i}$, the independent errors sampled from the same distribution centered at 0 and μ the "true" rating for all movies.

We know that the estimate that minimizes the RMSE is the least squares estimate of μ and, in this case, is the average of all ratings.

Lets calculate μ , predict the ratings, and check the performance of the model.

Table 7: Model 1 Novice Model Performance

	Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
2	Model_1	1.0607	0.1958	-	-

As we can see from the above table RMSE is more than 1 which is the least rating a user can give to a movie so this RMSE is not acceptable and far away from the target rmse so we will continue and incorporate other factors one by one.

5.4.2 Movie Effect

Let's add movie effect to our previous model by adding the term b_i to represent average ranting for movie i:

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

We can again use least squares to estimate the b_i in the following way:

$$fit = lm(rating\ as.factor(movieId), data = train\_set)$$

Because there are thousands of b_i as each movie gets one, the `lm()` function will be very slow here and because of this lm is not used it to determine the movie effect. But in this particular situation, we know that the least squares estimate $\hat{b}_i$ is just the average of $Y_{u,i} - \hat{\mu}$ for each movie i.

Here is formula used to calculate $\hat{b}_i$

$$\hat{b}_i = mean\left(y_{u,i} - \hat{\mu}\right)$$

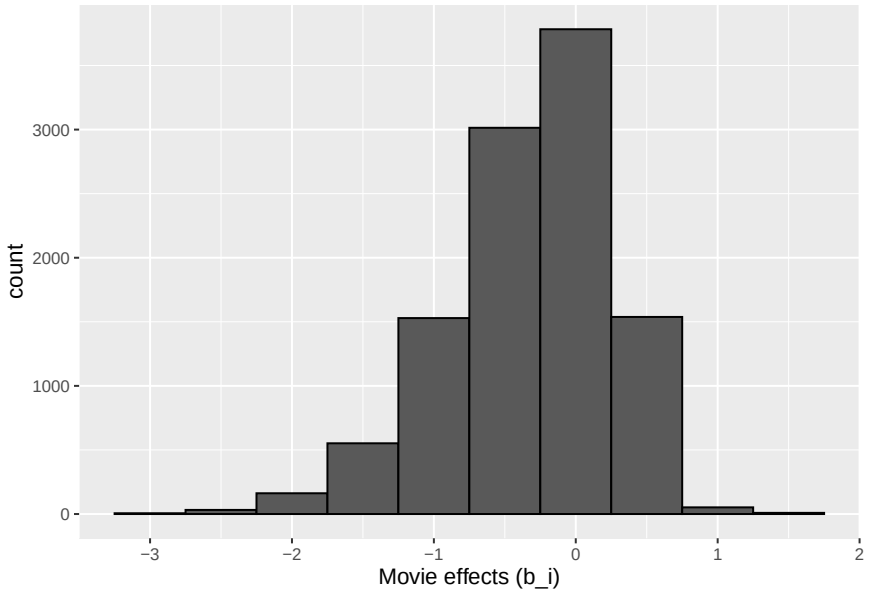

Figure 12: Distribution of movie effects

Let's see how much our prediction improves once we use $\hat{y}_{u,i} = \hat{\mu} + \hat{b}_i$:

Look at the below RMSE tracking table for model performance after adjusting the model for movie effect.

As we can see from the above that there is a significant reduction in RMSE, Adjusted R Square shows that 20.8423386112214 percent of the variance in the ratings is explained by adding the movie effect. But the RMSE of the current model is still 0.07881 above the target RMSE so the Model augmentation will continue with other factors to improve the model. Figure 12 shows that these estimates vary substantially.

Table 8: Model 2 Model with Movie Effect Performance

	Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
3	Model_2	0.94371	0.07881	20.8424265658399	20.8423386112214

5.4.3 User Effect

Data analysis has shown a significant impact of user average rating on the ratings, so lets add the user impact to the model.

Here is the model after adding the user impact

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

where b_u is a user-specific effect.

To fit this model, we could again use `lm` like this:

$$lm(rating\ as.factor(movieId) + as.factor(userId))$$

but, for the reasons described earlier lm is not used. Instead, computes an approximation by computing $\hat{\mu}$ and $\hat{b}_i$ and estimating $\hat{b}_u$ as the average of $y_{u,i} - \hat{\mu} - \hat{b}_i$:

Here is the formula used to calculate $\hat{b}_u$

$$\hat{b}_u = mean\left(\hat{y}_{u,i} - \hat{\mu} - \hat{b}_i\right)$$

Look at the below RMSE tracking table for model performance after adjusting the model for user effect.

Table 9: Model 3 Model with Movie + User Effect Performance

	Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
4	Model_3	0.86616	0.00126	33.3177812969318	33.3176701576161

As we can see from the above that there is a significant reduction in RMSE, Adjusted R Square shows that 12.4753315 percent of the variance in the ratings is explained by adding the user effect alone and the overall 33.3176701576161 percent of the variance in the ratings is explained with this new augmented model. We are pretty close to our target, but RMSE of the current model is still 0.00126 above the target model so the model improvisation will continue by adding other factors.

5.4.4 Genre Effect

As we have seen from the analysis section that there is significant variation in the average ratings across different genre's, let's add genre effect,b_g, to the current model.

Here is the updated model after adding genre effect.

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

least squares estimate of the genre effect, $\hat{b}_g$ is calculated using below formula.

$$\hat{b}_g = mean\left(y_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u\right)$$

Look at the below RMSE tracking table for model performance after adjusting the model for Genre effect.

Table 10: Model 4 Model with Movie + User + Genre Effect Performance

	Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
5	Model_4	0.86582	0.00092	33.3698430801543	33.369695010013

As we can see from the above that there is a slight improvement to the model. RMSE of the current model is still 9.2×10^{-4} above the target model so the model improvisation will continue by adding other factors.

5.4.5 Movie Release Year Effect

As we have seen from the analysis section that there is some variation in the average ratings across different movie years, let's add movie release year effect,b_y, to the current model.

Here is the updated model after adding movie release year effect to the model.

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

least squares estimate of the movie release year effect, $\hat{b}_g$ is calculated using below formula.

$$\hat{b}_y = mean\left(y_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u - \hat{b}_g\right)$$

Look at the below RMSE tracking table for model performance after adjusting the model for Movie Release Year effect.

Table 11: Model 5 Model with Movie + User + Genre + Movie Release Year Effect Performance

	Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
6	Model_5	0.86565	0.00075	33.3971419428581	33.3969569309105

As we can see from the above that there is a slight improvement to the model. RMSE of the current model is still 7.5×10^{-4} above the target model so the model improvisation will continue by adding other factors.

5.4.6 Movie Rated Date Effect

As we have seen from the analysis section that there is some variation in the average ratings across different Movie Rated dates, let's augment the model built in the previous section with the Movie Rated Date effect.

Ideally our model should look like the below

$$Y_{u,i} = \mu + b_i + b_u + b_g + b_y + f(du, i) + \epsilon_{u,i}$$

We can not directly take the Movie Rated date from the timestamp and apply it to the model, as it causes lot of variance to the model, and hence we need to apply a smooth function on the Movie Rated Date to smooth the Movie Review Date ,$f(du, i)$, as shown above in the formula.

As we know that the smoothing function computation takes long time, as an alternative method primarily to make the computation quicker, the Movie Rated date was rounded to the nearest week to smooth the data, essentially performing the same smoothing.

So as we took care of the smoothing through rounding, so the updated above model with the Movie Rated date effect b_r has changed to the below

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

The least squares estimate for $\hat{b}_r$ is shown below.

$$\hat{b}_r = mean\left(y_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u - \hat{b}_g - \hat{b}_y\right)$$

Look at the below RMSE tracking table for model performance after adjusting the model for Movie Rated Date effect.

Table 12: Model 6 Model with Movie + User + Genre + Movie Release Year + Movie Rated Date Effect Performance

	Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
7	Model_6	0.86546	0.00056	33.4253845217353	33.4251626014188

As we can see from the above that there is a very small effect to the model, but RMSE of the current model is still 5.6×10^{-4} above the target model but very very close to the target RMSE so will do one last improvisation in the next section.

5.4.7 Regularising the algorithm

Regularization constrains the total variability of the effect sizes by penalizing large estimates that come from small sample sizes.

In our particular scenario regularization is needed because most of the effects (Movie, User , Genre , Rated Date) have shown considerable variance in the number of ratings. Regularization would penalize the estimated average rating obtained from lower number of ratings.

As an example, some movie's were rated by thousands of people while some were rated by 1 person, so taking the movie effect from 1 person will not be useful , and could give us incorrect effect for those movies. Regularization would penalize the estimates that are coming from lower number of ratings and shirk those estimates.

Let's briefly look at what change regularization does to the expression that we minimize for movie effect and later we will apply the same to all the effects

Here is a movie effect least squares expression with out regularization.

$$\frac{1}{N}\sum_{u,i}\left(y_{u,i}-\mu-b_i\right)^2$$

Here is a movie effect least squares expression with regularization.

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

As you can notice in the above expression that a penalty term, $\lambda\sum_i b_i^2$, is added for regularization.

Here λ in the penality term, is a tuning parameter chosen using cross-validation within the edx dataset.

The equation that minimizes the above expression is given below. The below equation could be derived by differentiating the above expression and equating it to zero.

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

Here n_i is the number of ratings made for movie i.

The impact of $\frac{1}{\lambda+n_i}$ is that when the number of ratings are more, n_i is a big number and as λ is small compared to the n_i, the sum in the denominator would be almost as n_i, so the estimate for $\hat{b}i$ would be same as before and will not get impacted. Where as when the number of ratings are less, this is where the issue is, the average would not yield the correct result, so in this case the value of λ big compared to number of ratings, so the denominator would be big, so the the estimate shrinks.

Here is the final expression that takes care of regularizing the model, as you can see it has got all the penalty terms for all the effects(Movie, User , Genre , Movie Release Year , Movie Rated Date)

$$\frac{1}{N}\sum_{u,i}\left(y_{u,i}-\mu-b_i-b_u-b_g-b_y-b_r\right)^2+\lambda\left(\sum_i b_i^2+\sum_u b_u^2+\sum_g b_g^2+\sum_y b_y^2+\sum_r b_r^2\right)$$

The following formulas are used for calculating the Movie, User , Genre , Movie Release Year and Movie Rated Date effect, as these are the expressions that minimize the above expression.

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

$$\hat{b}_u(\lambda)=\frac{1}{\lambda+n_u}\sum_{u=1}^{n_i}\left(Y_{u,i}-\hat{\mu}-bi\right)$$

$$\hat{b}_g(\lambda)=\frac{1}{\lambda+n_g}\sum_{u=1}^{n_i}\left(Y_{u,i}-\hat{\mu}-bi-bu\right)$$

$$\hat{b}_y(\lambda)=\frac{1}{\lambda+n_y}\sum_{u=1}^{n_i}\left(Y_{u,i}-\hat{\mu}-bi-bu-bg\right)$$

$$\hat{b}_r(\lambda)=\frac{1}{\lambda+n_r}\sum_{u=1}^{n_i}\left(Y_{u,i}-\hat{\mu}-bi-bu-bg-by\right)$$

In order to calculate the above effects we need to find the value of lambda which produces the best RMSE and this process is called Optimization and Optimization is dealt in detail in the next section.

5.4.8 Optimization

Optimization is tuning the Model parameter values, and to find the values which performs better than others values in a set, in our particular case it is finding the lambda value with which the model performs better than any other lambda value in the input set, by producing a minimum RMSE.

We can use bootstrapping or cross validation for turning the lambda and in this section we are going to use cross validation.

5.4.8.1 Cross Validation The following cross validation process is used to find the lambda which produces the best RMSE value, that is the model with lowest RMSE.

1) 10 splits of the edx dataset were taken, where each split is a random sampling of data with 80% train data and 20% test data.
2) For each split or fold of data the following steps were repeated,

```
i)  A range of potential lambda values for turning the model were taken.
(range: 4.5-5.4, with increments of 0.1).
Note: Much large set of lambda values were taking while working on the project,
from 4 to 6 with 0.1 increment, that is 30 values. But to reduce the run time
of this report narrow set with the optimal value is given here.
ii) For each lambda value in the range, the following steps were performed
    a) Calculate the Movie, User , Genre , Year and Release Date effects
    b) Predict the test_set ratings using the effects calculated in the above step
    c) calculate RMSE for these predictions
    d) store the lambda value and it's respective RMSE
iii) By the end of the above step we get RMSE values for all lambda values in the set
iv)  Pick the lambda with minimum RMSE from the above set.
iv) The lambda that we obtained is the optimal lambda value for one particular split.
```

3) By the end of previous step(step 2) we would get a set of lambda values and their respective RMSE values for all the splits.
4) We pick the lambda that has occurred most,that is the lambda with most frequency, and this lambda value would be considered as an optimal parameter for the regularized model.

Lets look at the cross validation results.

Table 13: Cross Validation Results

Split	lambda	RMSE
1	5.1	0.8647254
2	5.1	0.8656579
3	5.0	0.8648887
4	5.0	0.8657913
5	5.1	0.8642094
6	5.1	0.8650295
7	5.2	0.8651030
8	4.9	0.8638025
9	4.9	0.8647524
10	5.1	0.8648556

The lambda values in the table are the optimal lambda values in individual splits which have produced the least RMSE.

Now we need to pick a lambda from the cross validation output and that would be explained in the next section.

5.4.8.2 Parameter Selection Now the optimal lambda for the model would be the one which has most frequency, the lambda that was voted by most of the splits, would be taken as the final lambda, and the average RMSE of the optimal lambdas would be considered as the RMSE of the model.

Now lets look at frequency of lambdas and their respective RMSE values

Table 14: lambda frequencies from cross validation results

lambda	frequency
4.9	2
5.0	2
5.1	5
5.2	1

Based on the above table we can see that the optimal lambda value is 5.1 and the RMSE value for this lambda is 0.8648956, and this value obtained by taking the average of all the RMSE's associated with the optimal lambda.

Lets finally look at the RMSE tracking table and see how far close are we to the target RMSE we want to achieve.

Table 15: Regularized Model Performance

	Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
8	Regularized Model	0.8649	0	-	-

As you can see from above that we have achieved the target RMSE and its time for validation.

5.4.9 Validation

Now that the machine learning model is built and optimized , and it's RMSE is below the target target RMSE of 0.86490, that means we have achieved the target RMSE in the training dataset, we are ready to run and see how our final regularized Movie Recommendation Model predict the ratings in the validation dataset.

Here is the process followed to perform the validation

1) Use the optimal lambda value that we got in the previous section, that is 5.1
2) Use the regularized Movie Recommendation Model on the entire edx dataset to calculate the Movie, User , Genre , Year and Movie Rated Date effects. Note that while building and optimizing the model we only used train_set, which is only 80% of the edx dataset, for calculating the effects, now as the model is finalized, we take the entire dataset, so that the model gets to use more data to calculate the effects.
3) Use the Movie, User, Genre , Year and Rated Date effects calculated in step 2 to predict the user ratings in the validation dataset.
4) Calculate the RMSE

5) Save the Model RMSE to the RMSE tracking table and print the output.

Look at the below RMSE tracking table for Movie Recommendation Model performance on the validation dataset

Table 16: Final Optimized Model Validation Results

	Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
9	Final Model	0.86408	-0.00082	33.7007995411614	33.7004017427762

As we can see from the RMSE tracking table, Movie Recommendation Model that has been built in this report has accomplished the goal of this report, and its RMSE is 0.8640763, and this validation RMSE is less than target RMSE 0.8649.

6 Results

Now that the Movie Recommendation Model algorithm is built and validated. Lets look at the results obtained in various steps from the RMSE Tracking table and summarize.

Here is the results table with all the effects

Method	RMSE	Difference	R_SQUARE	ADJUSTED_R_SQUARE
Project Goal	0.86490	-	-	-
Model_1	1.0607	0.1958	-	-
Model_2	0.94371	0.07881	20.8424265658399	20.8423386112214
Model_3	0.86616	0.00126	33.3177812969318	33.3176701576161
Model_4	0.86582	0.00092	33.3698430801543	33.369695010013
Model_5	0.86565	0.00075	33.3971419428581	33.3969569309105
Model_6	0.86546	0.00056	33.4253845217353	33.4251626014188
Regularized Model	0.8649	0	-	-
Final Model	0.86408	-0.00082	33.7007995411614	33.7004017427762

Here is the summary of results from the above table

1st Model - Novice Model: Novice Model is rudimentary and we used mean rating as to predict the ratings. Even though it is a very basic model it gives a good baseline to compare against. Its RMSE is 1.0607, and as this RMSE is higher than the lowest rating given by the users. Its RMSE is still 0.1958 higher than the target RMSE 0.86490

2nd Model with Movie Effect: We augmented 1st Model with Movie Effect to predict the ratings. Adjusted R Square shows that 20.8423386112214 percent of the variance in the ratings is explained by adding the movie effect. It's RMSE is 0.94371, and this RMSE is better than the Novice Model , and it has eliminated 11.0295088 percent of the Novice Model error and is a significant improvement to the Model.Its RMSE is still 0.07881 higher than the target RMSE 0.86490

3rd Model with User Effect: We augmented 2nd Model with User Effect to predict the ratings. Adjusted R Square shows that 12.4753315 percent of the variance in the ratings is explained by adding the user effect alone and the overall 33.3176701576161 percent of the variance in the ratings is explained with this new augmented model. Its RMSE is 0.86616, and as this RMSE is better than the 2nd Model and it has eliminated 8.2175668 percent of the 2nd Model error and is still a good improvement to the Model.Its RMSE is still 0.00126 higher than the target RMSE 0.86490

4th Model with Genre Effect: We augmented 3rd Model with Genre Effect to predict the ratings. Its RMSE is 0.86582, and as this RMSE is better than the 3rd Model and it has eliminated 0.0392537 percent of the 3rd Model error and is still a good improvement to the Model. Its RMSE is still 0.00092 higher than the target RMSE 0.86490

5th Model with Movie Release Year Effect: We augmented 4th Model with Movie Release Year Effect to predict the ratings. Its RMSE is 0.86565, and as this RMSE is better than the 4th Model and it has eliminated 0.0196346 percent of the 3rd Model error and is a improvement to the Model. Its RMSE is still 0.00075 higher than the target RMSE 0.86490

6th Model with Movie Rated Date Effect: We augmented 5th Model with Movie Rated Date Effect to predict the ratings. Its RMSE is 0.86546, and as this RMSE is better than the 5th Model and it has eliminated 0.0219488 percent of the 3rd Model error and is a improvement to the Model. Its RMSE is still 0.00056 higher than the target RMSE 0.86490

7th Model - Regularized 6th Model: We regularized 6th Model to predict the ratings. Its RMSE is 0.8649, and as this RMSE is better than the 6th Model and it has eliminated 0.0647055 percent of the 6th Model error and is a improvement to the Model. We have achieved the target RMSE.

Final Movie Recommendation Model: Final Movie Recommendation Model which was regularized in the previous step is used to predict the ratings of the validation dataset. Its RMSE is 0.86408 is better than the target RMSE 0.86490 goal of this project. With this we have accomplished the project goal.

7 Conclusion

A Movie Recommendation Model which predicts the user ratings for the movies with an RMSE <= 0.86490 has been developed, validated and the results of the output has been discussed in detail in this project.The goal of this project has been accomplished.

The following are the limitations of the model,

1) Increasing the size of the input movielense dataset increase the algorithm runtime significantly , and increasing the dataset multifold would make the program take a very long time to finish.

2) This algorithm will not predict the ratings for user's or movie's that does not exist in the movielense dataset.If you want to add a movie or a user which is not there in the movielense dataset to the validation data the algorithm will fail as Movie, User and other effects would not exists.

3) Due to the limited personal resources, personal computer, the following things which would give us a better estimate for the Movie, User , Genre , Release Year and Movie Rated Date could not be performed

    a) lm function to determine more precise effects as it need more computing resources
    b) smooth function on the Movie Rated Date could not be performed as it needs more computing resources

4) The following algorithms could not be performed as they take more computing resources for the given movielense dataset

a) KNN
b) Decision Trees and Random Forest

Following are the future considerations for the Movie Recommendation Model

1) The above mentioned 1, 3 and 4 could be addressed using a *BigDataCluster*, which would facilitate parallel processing capabilities to run the algorithms faster.
2) The following alternative approaches to develop the Movie Recommendation Model needs to be explored
    a) Neighborhood Models - The following two methods could be explored under this approach
        i) Find Movies similar to the Movie that a User has rated and calculate the average rating
        ii) Find a set of Users similar to the User we want to rate , and take the average ratings that the similar Users given to the Movie.
    b) Matrix Factorization - The following SVD inspired methods from Netflix Prize could be explored for future work
        i) Standard SVD
        ii) Asymmetric SVD
        iii) SVD++
    c) Ensemble Methods - Ensemble could be used to combine the various models and exploit the strength of the individual models.