EDX Data Science Capstone - MovieLens Project

Yin Thu Win

6/27/2025

1.Introduction:

The MovieLens project focuses on building machine learning models to predict user movie ratings and enhance recommendation systems. Using a dataset from GroupLens Research at the University of Minnesota, which includes millions of user-generated ratings, the project involves exploratory data analysis (EDA) and model development. Key predictors include user preferences, movie characteristics, and film age. Model performance is evaluated using metrics like RMSE, with regularization techniques applied to improve accuracy. As online behavior shifted during COVID-19, recommendation systems became vital in platforms like Netflix and Amazon. These systems rely on machine learning to adapt to big data, evolving preferences, and overfitting challenges.

My Github Repo

2. Methods and Analysis:

We will start by preparing the data and loading it from the GroupLens Website. The data will be divided into two sets, “edx” and “validation,” before being further split into training and test sets. A preliminary exploratory analysis will be performed to review the dataset’s features and identify any potential biases that could impact and reduce the predictive accuracy of our models.

The dataset will be cleaned to remove missing values and organized into a tidy format. Exploratory data analysis (EDA), including visualizations like histograms and scatter plots, will guide model development. The MovieLens 10M dataset is split into training and validation sets, and predictors are analyzed using the group_by function to assess their effects on ratings. Simple linear regression models are created for individual predictors, and combinations are tested for improved performance. A correlation matrix identifies key variables. Regularization is applied to optimize tuning parameters, and model performance is evaluated using RMSE. The final model aims for an RMSE below 0.8649.

2.1 Data Preparation and Required Packages

2.1.1 Exploratory Data Analysis

After deploying codes provided for this project, it is shaped into the edx training dataset consist of 90% of the data and the validation testing dataset consist of 10% of data. It is found that no missing values in any column.

	x
userId	FALSE
movieId	FALSE
rating	FALSE
timestamp	FALSE
title	FALSE
genres	FALSE

Data set dimension of movieLens as shown below.

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

Confirm the data is tidy:

```
## # A tibble: 9,000,055 × 6
##    userId movieId rating timestamp title
genres
##     <int>   <dbl>  <dbl>     <int> <chr>
<chr>
##  1      1     122      5 838985046 Boomerang (1992)
Comed…
##  2      1     185      5 838983525 Net, The (1995)
Actio…
##  3      1     292      5 838983421 Outbreak (1995)
Actio…
##  4      1     316      5 838983392 Stargate (1994)
Actio…
##  5      1     329      5 838983392 Star Trek: Generations (1994)
Actio…
##  6      1     355      5 838984474 Flintstones, The (1994)
Child…
##  7      1     356      5 838983653 Forrest Gump (1994)
Comed…
##  8      1     362      5 838984885 Jungle Book, The (1994)
Adven…
##  9      1     364      5 838983707 Lion King, The (1994)
Adven…
## 10      1     370      5 838984596 Naked Gun 33 1/3: The Final Insult (1…
```

```
Actio…
## # ℹ 9,000,045 more rows
```

Explore the features and classes of edx while also confirming its observations

```
## Rows: 9,000,055
## Columns: 6
## $ userId    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, …
## $ movieId   <dbl> 122, 185, 292, 316, 329, 355, 356, 362, 364, 370, 377,
420, …
## $ rating    <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, …
## $ timestamp <int> 838985046, 838983525, 838983421, 838983392, 838983392,
83898…
## $ title     <chr> "Boomerang (1992)", "Net, The (1995)", "Outbreak
(1995)", "S…
## $ genres    <chr> "Comedy|Romance", "Action|Crime|Thriller",
"Action|Drama|Sci…
```

Deployment for the unique number of userIds, movieIds, and genres

```
##   unique_users unique_movies unique_genres
## 1        69878         10677           797
```

2.2 Ratings:

Deployment of 10 different ratings creation to recommend the movies:

```
## [1] 10

## [1] 0.8242332
```

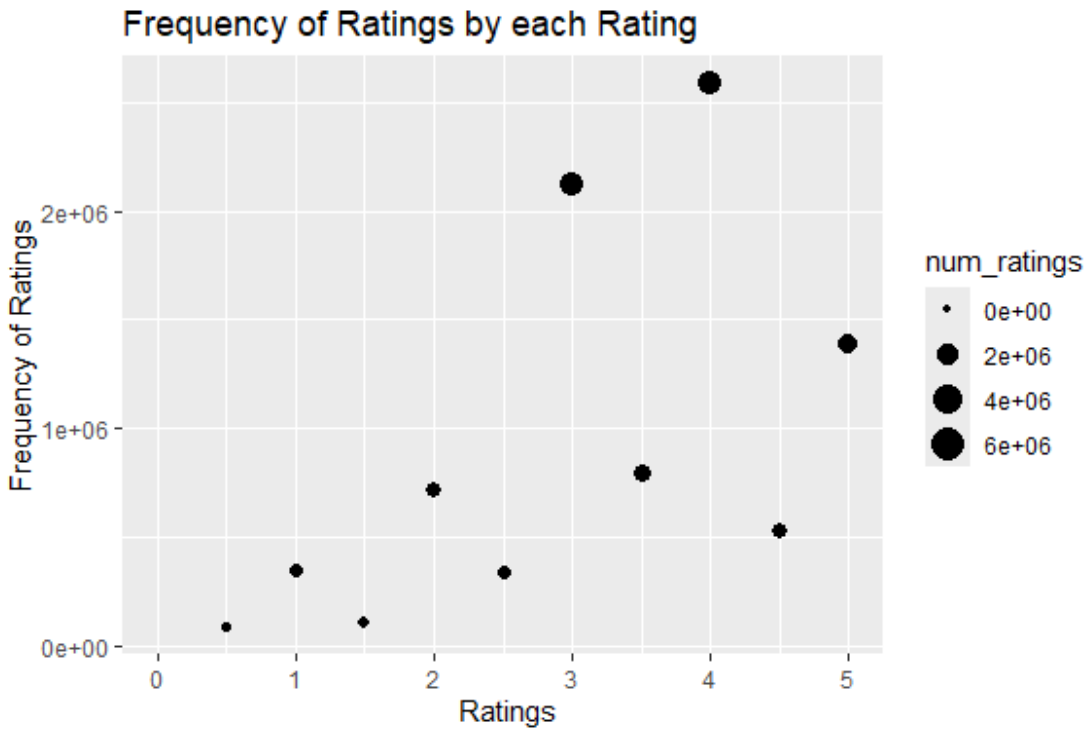
Overall average rating for edx data set is 3.5126, Whole Median rating for edx data set is 4.

Quantitative features userId: discrete, Unique ID for the user movieId: discrete, Unique ID for the movie timestamp: discrete, Date and time the rating was given Qualitative features title: nominal, movie title (not unique) genres: nominal, genres associated with the movie Outcome, rating: continuous, a rating between 0 and 5 with 0.5 increment for the movie

The rating data given by users to the particular movies is analysis by group of rating. Rating included whole star 1 to 5 scale and half star rating from 0.5 to 4.5. Frequency of each rating are summarized with their whole and half star rating groups. According to the data visualization, the user more likely to give whole star rating than half star rating.

Users give highest number rating frequency with 4 stars rating and 3, 5, 3.5, 2 stars rating accordingly. The majority of user prefer to give higher rating on the movie they reviewed.

```
## # A tibble: 10 × 2
##    rating num_ratings
##     <dbl>       <int>
##  1    4       2588430
##  2    3       2121240
##  3    5       1390114
##  4    3.5      791624
##  5    2        711422
##  6    4.5      526736
##  7    1        345679
##  8    2.5      333010
##  9    1.5      106426
## 10    0.5       85374
```



The distribution of ratings in the dataset reveals interesting trends regarding user preferences. The most popular rating is 4.0, with 2,588,430 occurrences, followed by 3.0 with 2,121,240 ratings. These indicate that users tend to rate films more positively, with a higher concentration of ratings clustered around the middle to upper range.

Ratings of 5.0 are also common, with 1,390,114 occurrences, reflecting the presence of highly rated films. On the other hand, ratings on the lower end of the scale, such as 1.0, 2.0, and 0.5, are awarded significantly less frequently, with 345,679, 711,422, and

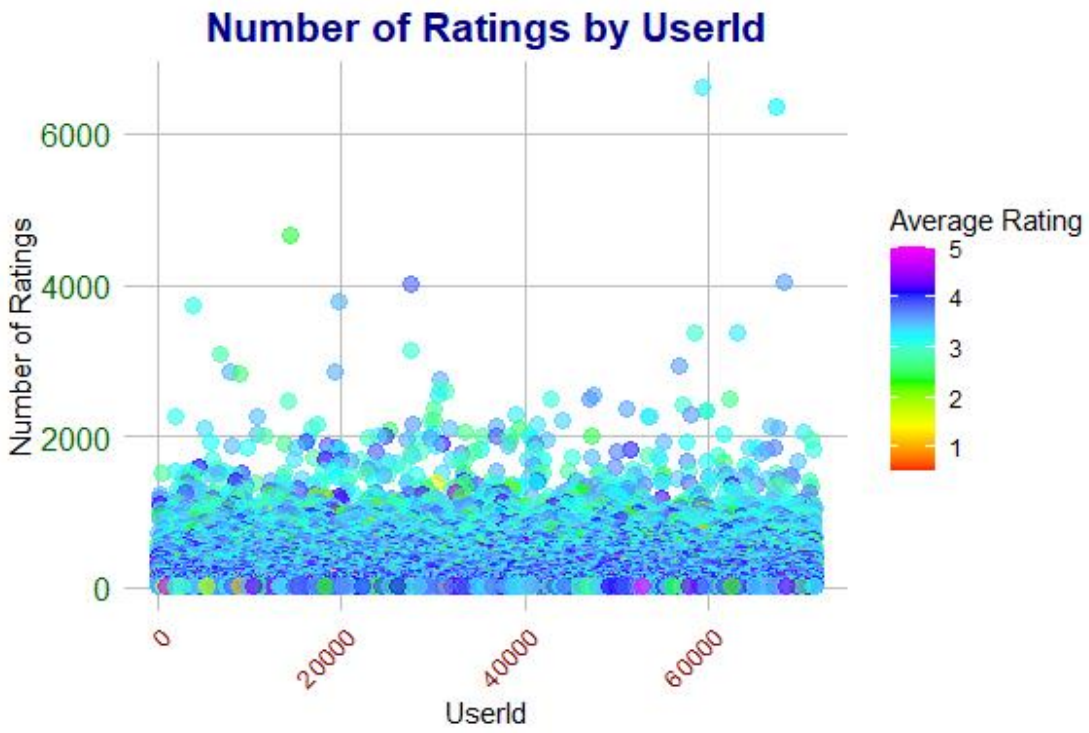
85,374 occurrences, respectively. The 0.5 rating, in particular, is the least frequent, suggesting that users rarely give the lowest possible rating.

This distribution indicates a general tendency for users to avoid giving extremely low ratings, possibly reflecting a bias towards more moderate or positive feedback. The presence of a relatively high number of 3.5 and 4.5 ratings further supports this notion, as users appear more inclined to rate films in the upper-middle range rather than at the extremes.

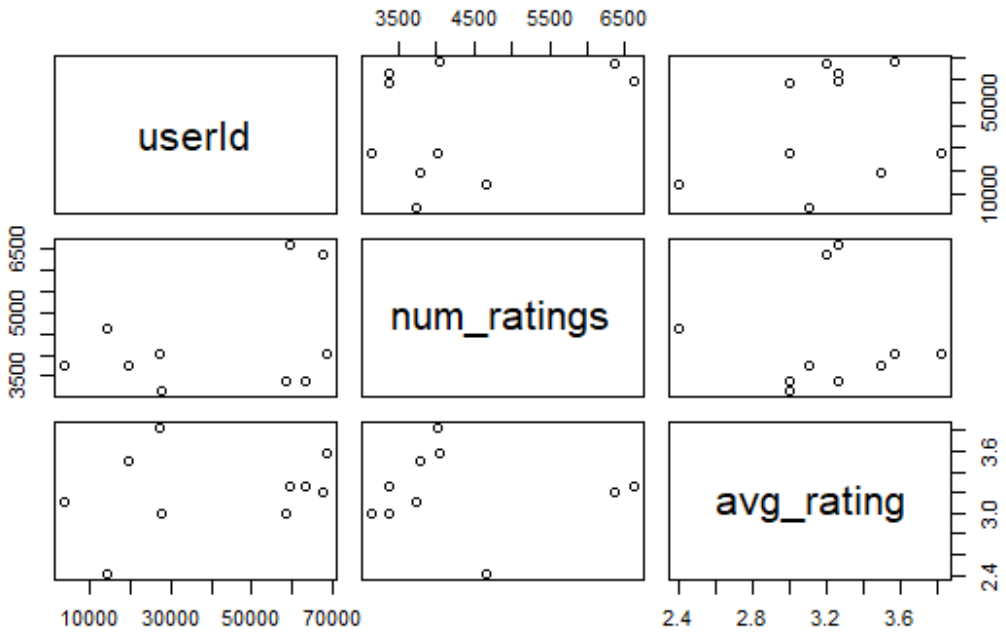
2.2.1 User

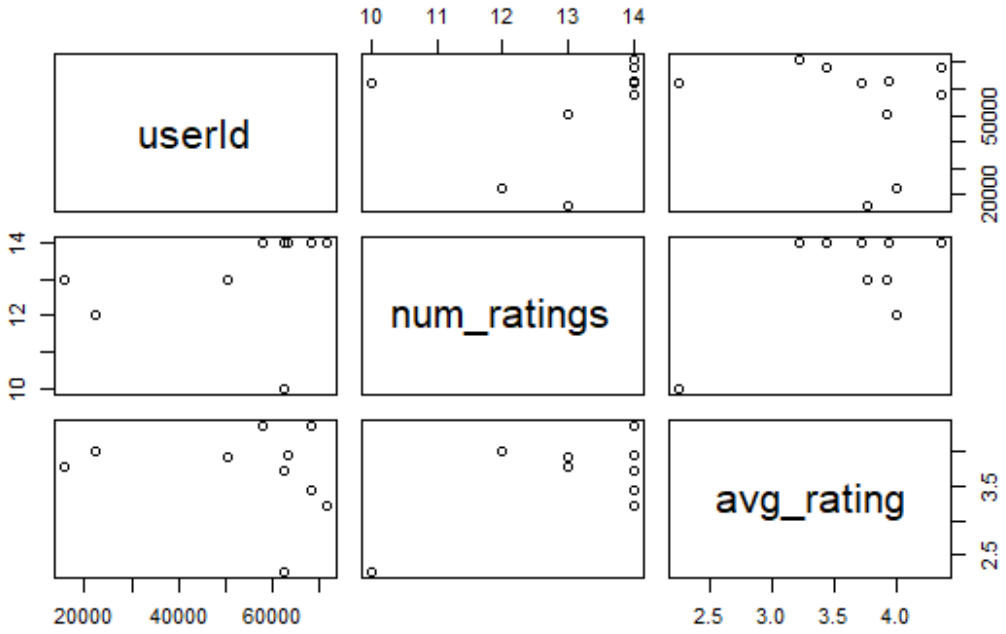
User is a critical feature in providing ratings for movies at specific times. The edx dataset contains 69,878 unique users. We analyze the user data by grouping it by userId and summarizing the frequency of their ratings in descending order. A visual exploration of the number of ratings per userId and the average rating reveals the following relationships.

```
##   num_users
## 1     69878
```



Some users are more active than others when it comes to reviewing and rating movies. Specifically, the following provides a detailed view of the rating behavior of the top and bottom 10 users, highlighting their rating frequency.



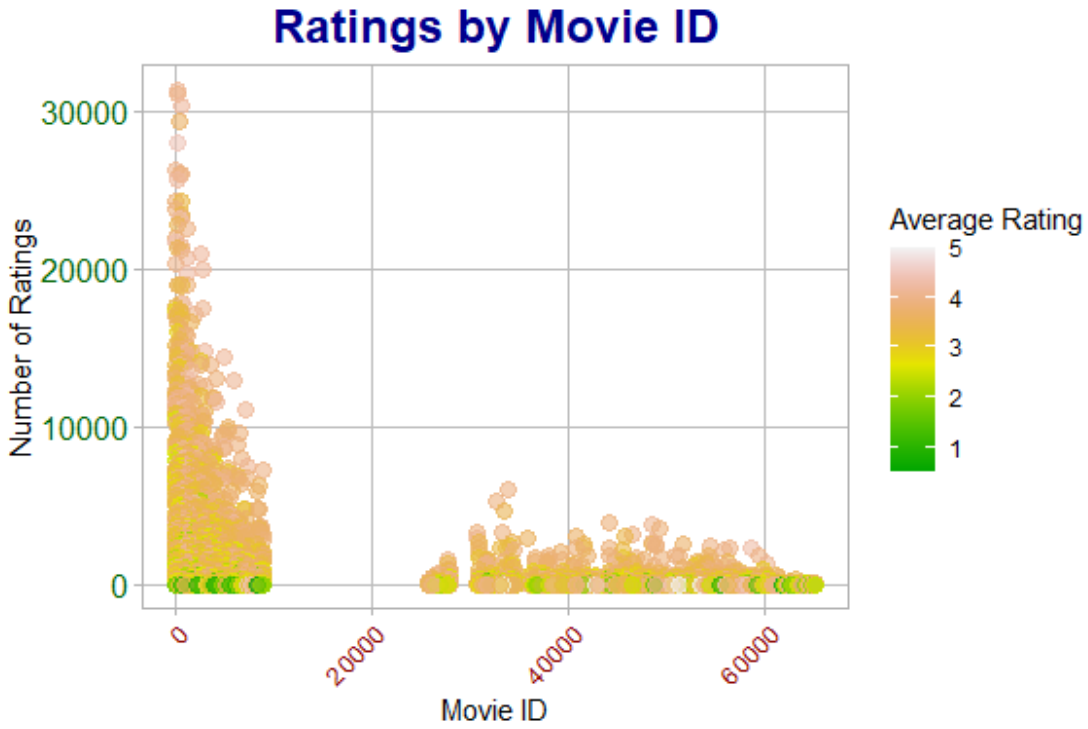


According visualize data, top 10 user rate around 2500 to 6600 movies and their average rating range from 2.4 to 3.6. However, bottom 10 user rate 10 to 14 movies only and their average rating rage from 2.5 to 4.5. This observation proved that there has variation in rating of the movies by different users.

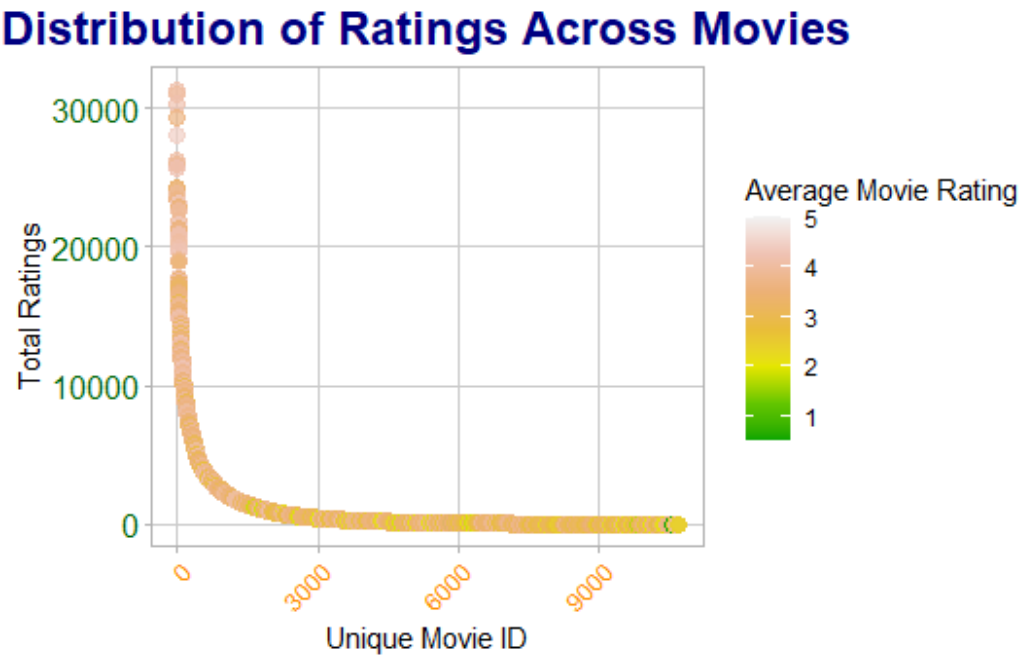
2.2.2 Movie

Movies are a key feature in determining ratings from different users for a particular movieId. Each year, new movies are released, and ratings for these movies are accumulated over the following years, depending on the users. The given edx dataset contains 10,677 movies. To further analyze the movie data, we group the movies by their movieId and summarize the frequency of their ratings in descending order.

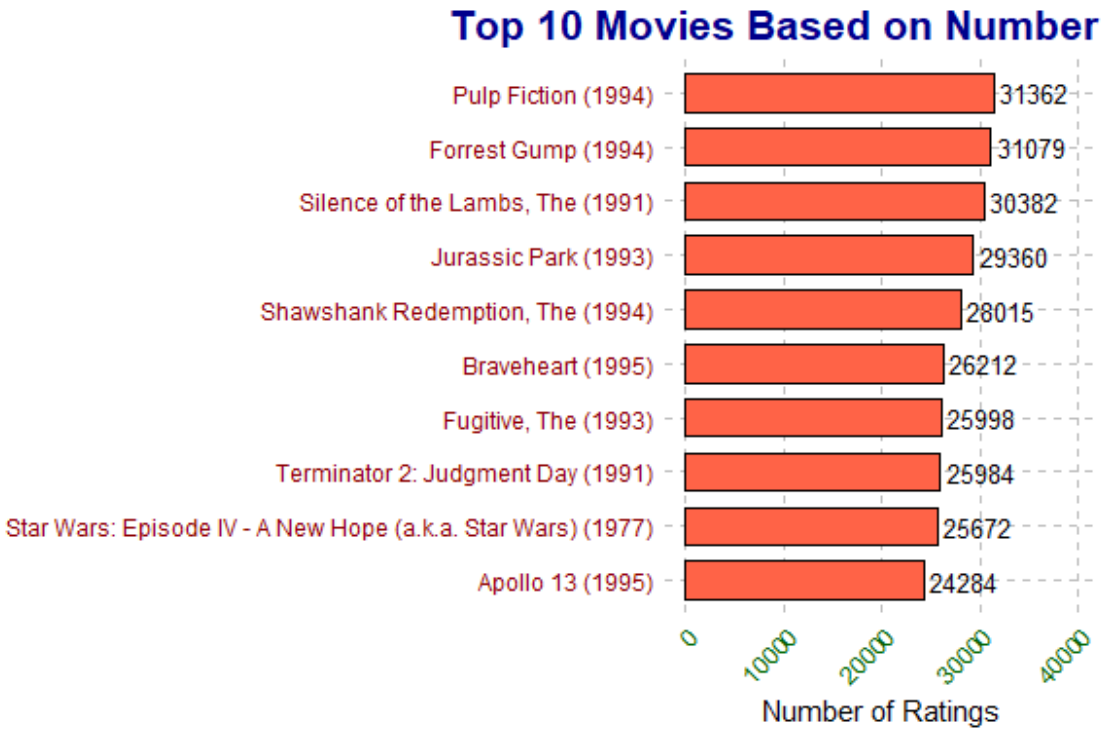
```
##   n_movies
## 1    10677
```



The relationship between movieId and number of ratings with average rating show that some popular movies get huge rating than other ordinary or infamous movie. Additionally, there are some movieId not get rating or not in the data set can be found. Hence, we make another plot of number rating with vs unique movie and it can see clearer picture of different rating between movies.



Top 10 number of rated movies can be discovered in the following.

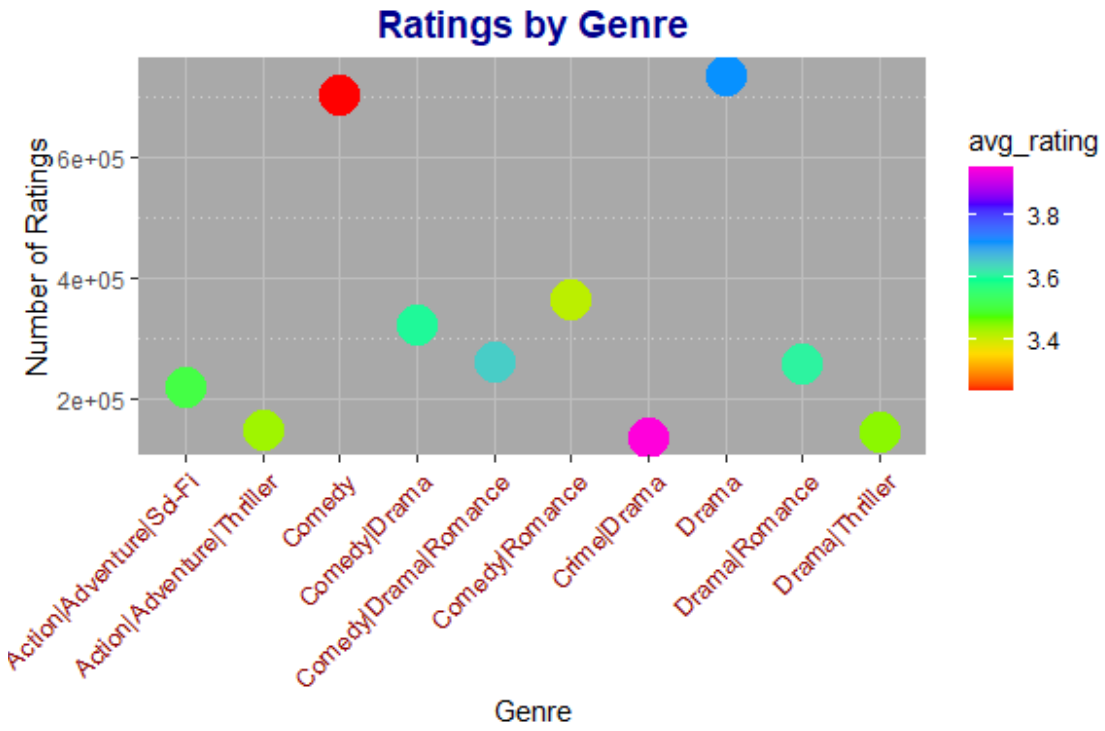


2.2.3 Genres

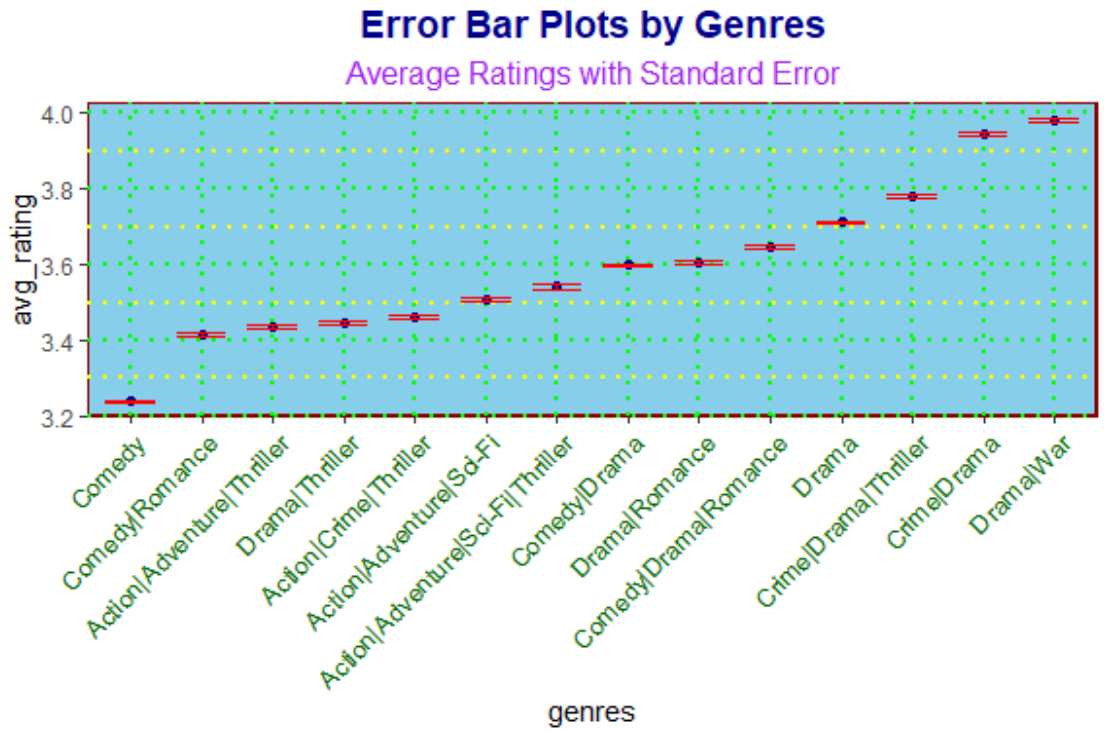
A genre refers to a category of creative work, including literature, art, entertainment, and movies. In the edx dataset, there are 20 unique genres, and some movies belong to multiple genres. After analyzing the data using the group_by function, we identified 797 unique genre combinations. If we were to split every unique genre combination, it would result in millions of additional rows, which could cause the computer to crash during processing. Therefore, for this project, we will only use the default genres provided in the dataset. Below, you will find the top 10 and bottom 10 genres based on the analysis.

```
## # A tibble: 10 × 3
##    genres                    num_ratings avg_rating
##    <chr>                           <int>      <dbl>
##  1 Drama                          733296       3.71
##  2 Comedy                         700889       3.24
##  3 Comedy|Romance                 365468       3.41
##  4 Comedy|Drama                   323637       3.60
##  5 Comedy|Drama|Romance           261425       3.65
##  6 Drama|Romance                  259355       3.61
##  7 Action|Adventure|Sci-Fi        219938       3.51
##  8 Action|Adventure|Thriller      149091       3.43
##  9 Drama|Thriller                 145373       3.45
## 10 Crime|Drama                    137387       3.95

## # A tibble: 10 × 3
##    genres                                       num_ratings avg_rating
##    <chr>                                              <int>      <dbl>
##  1 Action|Adventure|Animation|Comedy|Sci-Fi                3       4
##  2 Horror|War|Western                                     3       3.33
##  3 Action|Animation|Comedy|Horror                         2       1.5
##  4 Action|War|Western                                     2       3.75
##  5 Adventure|Fantasy|Film-Noir|Mystery|Sci-Fi             2       4
##  6 Adventure|Mystery                                      2       3.25
##  7 Crime|Drama|Horror|Sci-Fi                              2       3.25
##  8 Documentary|Romance                                    2       3.75
##  9 Drama|Horror|Mystery|Sci-Fi|Thriller                   2       3.5
## 10 Fantasy|Mystery|Sci-Fi|War                             2       2.5
```

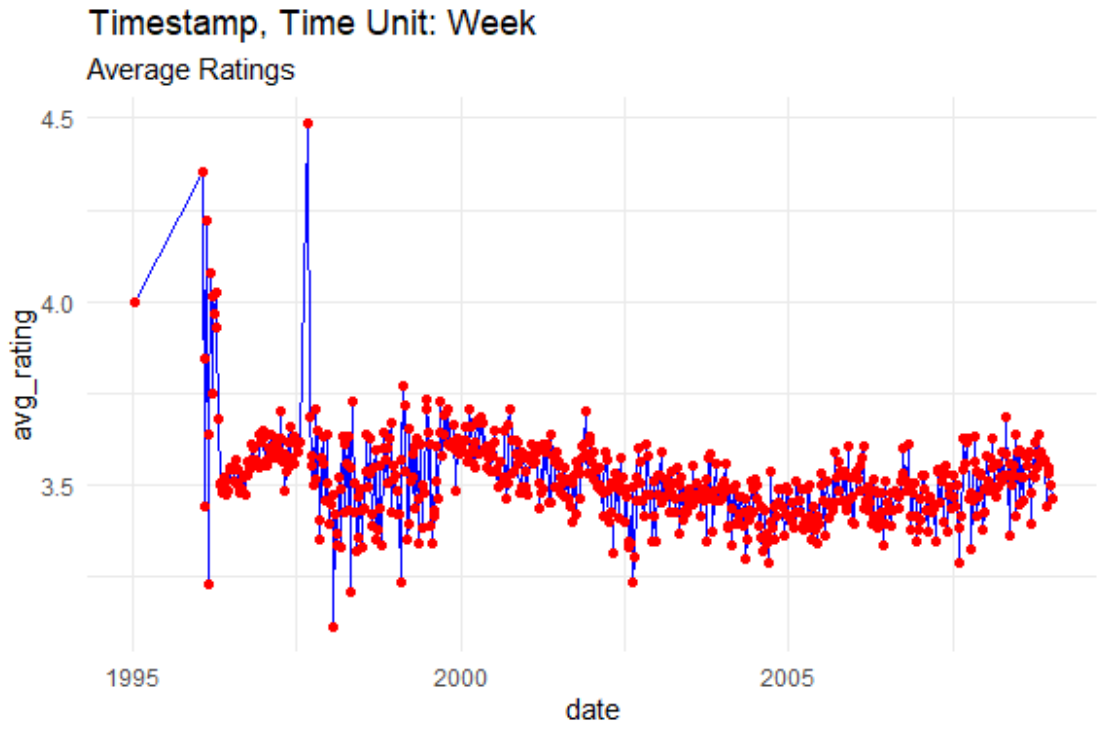


We will focus on the top genres that have received more than 100,000 ratings. For each of these genres, we calculate the average rating and standard error, then plot these values as error bar plots. The resulting plot highlights the variation in ratings across genres, indicating genre-specific differences in the number of ratings.



2.2.4 Time

In the edx dataset, the timestamp represents the time at which users rated different movies, with the units measured in seconds since January 1, 1970. To enhance our data analysis, we create a new column called “date” using the round_date function and add it to both the edx and validation datasets. We then compute the average rating for each week and visualize this average rating over time using a plot.



Analysing the trend of the average ratings versus the timestamp, there is significant effect in the early two year of the plot and then after 1996, average ratings were populated widely. After 1998, surfing internet started popular and movie average ratings have much improvement noticeable. It can be seen that, after 2000 year the most of the average rating occurred between 3 and 3.5 respectively.

2.3 Developing the Linear Regression Models

In this machine learning project, each outcome Y is associated with a different set of predictors. Specifically, user u provides a rating for movie i at time stamp t, and the popularity of the movie, categorized by genres g, is also considered as a predictor for the outcome Y.

2.3.1 Simple linear regression model (Average)

We begin with a simple approach to build our linear regression models, where the same rating is predicted for all movies, regardless of the users or other related independent variables. This represents the most basic form of a recommendation system.

$$Y_{u,i} = \mu + \varepsilon_{u,i} - -(1)$$

Where, $\varepsilon_{i,u}$ independent errors sampled from the same distribution centred at 0 μ the "true" rating for all movies (the average of all ratings)

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

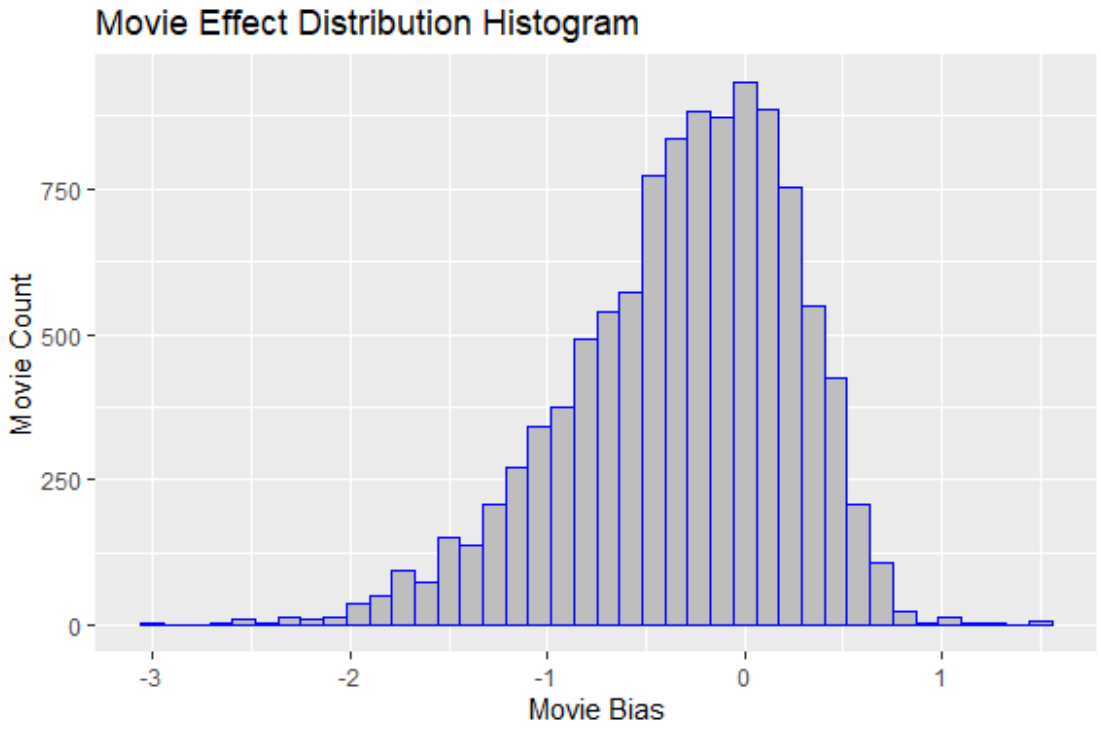
Considering individual effect of predictors to the first model

2.3.2 Movie Effect Model

It is well-known that different movies receive varying ratings, with a clear variation between them. In this context, we use the average rating of a movie effect denote as e_i in simple model. We can add augment our first model by adding the term e_i to represent average ranking for movie i.

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

Where, μ the "true" rating for all movies e_i is effects or bias, movie-specific effect $\varepsilon_{u,i}$ independent errors sampled from the same distribution centered at 0



The left screw distribution illustrates the movie effect/bias.

2.3.3 User Effect Model

We also observed that some users are more active in providing ratings, while others are less active and rate only a few movies. This results in a clear variation in the level of engagement between users. Here, we use the average ranking of user effect denote as

e_u. We can add augment our simple average model by adding the term e_u to represent average ranking for user u.

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

where , e_u is effects or bias, user-specific effect μ and $\varepsilon_{u,i}$ are defined as in (1)

The right screw distribution presents the user effect/bias.

2.3.4 Genres Effect Model

It is obvious that certain movie genres receive significantly higher ratings than others, due to the varying popularity of movies within those genres among reviewers. Here, we use the average ranking of genres effect denote as e_g. We can add augment our simple average model by adding the term e_g to represent average ranking for genres g.

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

where: e_g is effects or bias, Genres-specific effect μ and $\varepsilon_{u,i}$ are defined as in (1)

The left screw distribution presents the genres effect/bias.

2.3.5 Time Effect Model

We also observed significant variation in ratings from certain earlier years, indicating the presence of temporal trends in rating behavior over specific time periods. Here, we use the average ranking of time effect denote as e_g. We can add augment our simple average model by adding the term e_g to represent average ranking for time t.

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

Where, $e_t$is effects or bias, Time-specific effect μ and $\varepsilon_{u,i}$ is defined as in (1)

The time bias /effect not strongly present in distribution. Still have some effect.

Combining multiple predictors effect to the simple model

2.3.6 Combine Movie and User Effects Model

Due to left and right screw variation in both user and movie bias specific effect, following combination may implies that an additional improvement to our model.

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

Where, e_u is a user-specific effect and e_i is a movie-specific effect μ and $\varepsilon_{u,i}$ is defined as in (1)

2.3.7 Combine Movie + User + Genres Effects Model

The visualize data shows some evidence of a genre effect. If we define $g_{u,i}$ as the genre for user's u rating of movie i, the new model will be like following,

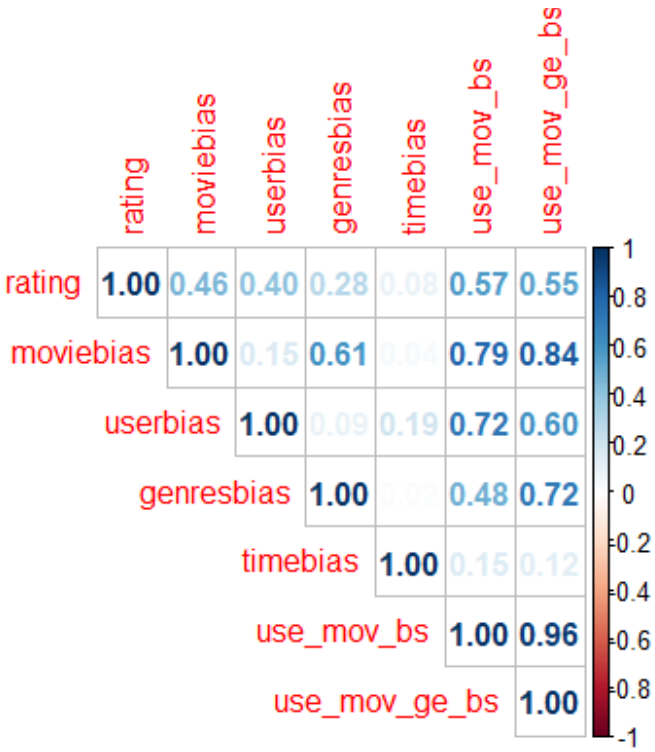
$$Y_{u,i} = \mu + e_i + e_u + \sum_{k=1}^{K} x_{u,i}\,\beta_k + \varepsilon_{u,i}$$

Where $ g_u,i$ is defined as the genre for user's u and $x_{u,i}^{k} = 1$ if $g_{u,i}$ is genre k. μ is defined as in (1) We try to present the model here for explanation, and we will select predictors correlation matrix in following section for model improvement.

2.4 Correlation between dependent and independent variables

Before applying machine learning, low-value predictors were removed to enhance model performance. A correlation matrix identified key variables influencing movie ratings. Strong correlations with the target variable, especially from the user-movie bias/effect model, guided predictor selection. Multicollinearity was avoided by excluding highly correlated predictors, ensuring model stability and interpretability.

```
##    userId rating         e_i      e_u          e_g        e_t
##     <int>  <num>       <num>    <num>        <num>      <num>
## 1:      1      5 -0.65387934 1.487535 -0.097978844 0.02633554
## 2:      1      5 -0.38313118 1.487535 -0.051175731 0.02633554
## 3:      1      5 -0.09445454 1.487535 -0.165794197 0.02633554
## 4:      1      5 -0.16278816 1.487535 -0.005058569 0.02633554
## 5:      1      5 -0.17500816 1.487535 -0.358065712 0.02633554
## 6:      1      5 -1.02467799 1.487535 -0.637411965 0.02633554
```




```
## [1] 0.5690665

## [1] 0.5453741
```

```
## [1] 0.955177
```

2.5 Evaluation Method of the models

In this project, RMSE (Root Mean Square Error) was used to evaluate and determine the best model. RMSE is interpreted similarly to a standard deviation and represents the typical error made when predicting a movie rating. A lower RMSE value indicates a better-performing machine learning model. The RMSE is calculated as follows:

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

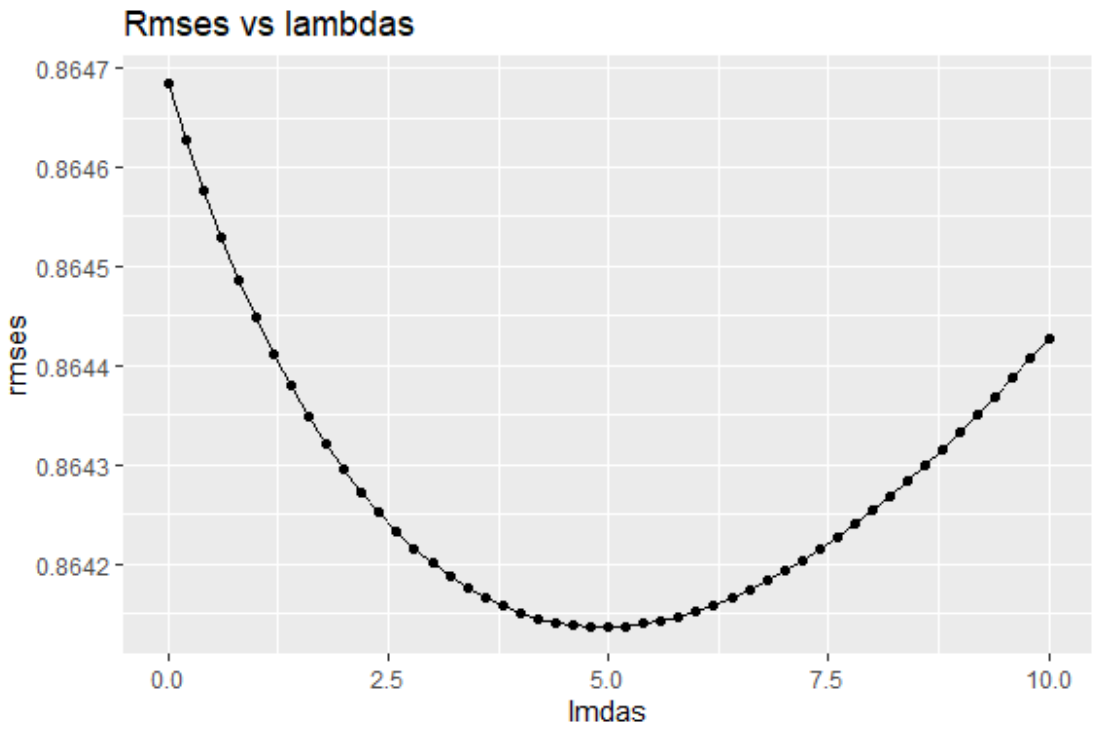
Where, $y_{u,i}$ is the rating for movie i and user u $\hat{y}_{u,i}$ is our prediction N is the number of user/movie combinations sum is over all above combinations

A function that computes the RMSE for vectors of ratings and their corresponding predictors can be written as following code.

2.6 Regularization

A major challenge in machine learning is overfitting, where models capture noise rather than meaningful patterns, reducing accuracy on new data. Cross-validation helps combat this by estimating performance on unseen data and guiding predictor selection. Regularization further controls overfitting by adding a penalty to the loss function, discouraging complexity and improving generalization, especially with correlated predictors. The regularization strength is controlled by lambda; higher values shrink coefficients more and reduce variance. In this project, the edX dataset was split 90/10 into training and validation sets using the caret package in R, enabling efficient cross-validation and model optimization.

To identify the optimal lambda value, we evaluated the updated training and validation datasets provided by edX. As illustrated in the figure below, a lambda value of 5 resulted in the lowest RMSE for the regularized model. Consequently, this value was selected for the final model evaluation.




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

3. Results and Discussions

After developing the models, a final validation test was performed using the validation dataset. Prior analysis showed that user and movie-related biases had the greatest impact among all predictors. The correlation matrix also identified the most influential variables for model selection during preprocessing. Based on these findings, we hypothesized that combining user and movie effects would produce the lowest RMSE. The RMSE results for all tested models are summarized in the table below.

Method	RMSE
The Average	1.0612018
Time Effect Model	1.0575018
Genres Effect Model	1.0184056
User Effect Model	0.9783360
Movie Effect Model	0.9439087
Combine Movie and User Effects Model	0.8850398
Regularized combine Movie and User Effects Model	0.8648177

The results of the various models built to predict movie ratings, measured by Root Mean Squared Error (RMSE), demonstrate a clear trend of improvement as more complex effects are incorporated into the models. Here’s a breakdown of the performance for each model:

1.Average Model (RMSE = 1.0612018): This is the baseline model that predicts ratings by simply using the average rating across all movies. Its RMSE value is the highest, indicating that predicting the mean rating for all users does not provide a good fit for the data.

2.Time Effect Model (RMSE = 1.0575018): By incorporating the time effect, this model slightly improves upon the average model. The time effect takes into account the trend in ratings over time, which may reflect shifting user preferences, but the improvement is marginal.

3.Genres Effect Model (RMSE = 1.0184056): This model introduces the genre of the movie as a predictor, showing a noticeable improvement in RMSE. The fact that genre influences ratings suggests that users have preferences for specific genres, and this factor helps refine predictions.

4.User Effect Model (RMSE = 0.9783360): Incorporating user-specific effects, this model improves further. This indicates that individual user preferences play a significant role in predicting ratings, as users tend to rate movies based on their unique tastes.

5.Movie Effect Model (RMSE = 0.9439087): Adding movie-specific effects reduces the RMSE even more, indicating that the popularity or inherent characteristics of the movie also impact how users rate them. This model takes into account the biases related to individual movies.

6.Combine Movie and User Effects Model (RMSE = 0.8850398): When combining both movie and user effects, the model performs significantly better than its predecessors. This indicates that both the user's preferences and the characteristics of the movie together provide more accurate predictions.

7.Regularized Combine Movie and User Effects Model (RMSE = 0.8648177): The final model, which includes both regularization and the combined movie and user effects, yields the best performance with the lowest RMSE. Regularization helps reduce overfitting by penalizing overly complex models, leading to better generalization on unseen data.

In summary, as the model incorporates more complex features—such as user preferences, movie attributes, and their interactions—the RMSE consistently decreases, indicating improved predictive accuracy. The final regularized model, which combines both movie and user effects, achieves the lowest RMSE, making it the most accurate among all models evaluated.

4. Conclusion and recommendation

This project was successfully implemented and surpassed the initial goal of predicting movie ratings using the 10 million MovieLens dataset. After splitting the data into the edx and validation sets for training and testing purposes, we began with thorough data exploration. The dataset was then cleaned, wrangled, and visualized to extract meaningful insights. Understanding data correlations played a crucial role in enhancing

the predictive performance of the model. To further improve accuracy, regularization was applied with a lambda value of 5, resulting in a more robust prediction model. As a result, the final model was able to achieve the target RMSE, fulfilling the objectives of the Capstone project.

Potential Impact

Today, recommendation engines and machine learning are widely deployed across platforms such as YouTube, Facebook, LinkedIn, Netflix, Alibaba, as well as various web, mobile, and mini applications. These technologies play a significant role in driving digital transformation and accelerating the growth of e-commerce.

Limitation

From this Capstone project, we have gained valuable insights along with a better understanding of its limitations. One key takeaway is the importance of internet speed and having a fast, high-memory laptop to ensure smooth execution and efficient data analysis. Additionally, we recognized that the project has limitations in terms of the variables considered. For a more realistic and personalized recommendation system—whether for movies or businesses—factors such as user behavior, movie language, user age group (child or adult), celebrity popularity, current trends, geographical location, and socio-economic status (to estimate spending habits) are critical. Moreover, the content and context of the movies themselves play an essential role in tailoring recommendations to individual users. Incorporating these aspects would lead to deeper insights and a more customized user experience.

Future work

Finally, the capstone project was successfully deployed to achieved the desired target, It is likely that more efficient Neural Network and Machining Learning models are recommended for future iterations of models involving K-Nearest-Neighbors and Collaborative Filtering with content base and context to improve the overall user experience for streamers everywhere.

5.References

1.Irizarry, R., 2019. Introduction To Data Science. [online] Rafalab.github.io. Available at: rafalab.github.io/dsbook.

2.“MovieLens 10M Dataset.” GroupLens, 2019, Available at: grouplens.org/datasets/movielens/10m/.

3. Jannach, S., and Adomavicius, G., “Recommender Systems: Challenges and Research Opportunities,” Springer, 2016.
4. Karypis, G., and Rajan, Y., “Collaborative Filtering Recommender Systems,” Springer, 2011.
5. Shani, B. S., and Gunawardana, A., “Evaluating recommendation systems,” in Recommender Systems Handbook, Springer, pp. 257-297, 2011.
6. Ricci, M., Rokach, L., and Shapira, B., “Introduction to Recommender Systems Handbook,” Springer,
7. Koren, Y., Bell, R., and Volinsky, C., “Matrix Factorization Techniques for Recommender Systems,” IEEE Computer, vol. 42, no. 8, pp. 30-37, 2009.
8. Chatgpt for code optimization and machine learning and reporting