EDX MOVIE RECOMMENDATION PROJECT

SUBMITTED BY

EBOIGBE, UKPONAYE DESMOND

IN PARTIAL FULFILMENT OF A PROFESSIONAL CERTIFICATE IN DATA SCENCE IN THE HARVARDx DATA SCIENCE PROGRAMME

NOVEMBER, 2020

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

A recommendation system is an implementation of artificial intelligence whereby the system is capable of learning patterns in a dataset and providing relevant suggestions from learning and analyzing behavioral patterns or trend in the data. A recommendation system provides its users with relevant contents and substitutes based on their preferences and likes. A recommendation system takes the information about the user as an input and analyze it with machine learning algorithms to predict the user's behaviour.

A recommendation systems is more or less an expert system that applies machine learning algorithms to make predictions about user behaviour. It is a subset of Data Science which undoubtedly is the driving force in the fourth industrial revolution where data mining technologies are dominating. A recommendation system helps to improve services to customers and aid the advancement of the Internet of Things (IoT).

The primary goal of recommendation systems is to help users find what they want based on their preferences and previous interactions, and predicting the rating for a new item. Online shops and service providers usually study users browsing history to either recommend a product or service for them. Search engines also keep track of users browsing habit to recommend website domains with potential interest for users.

Recommendation systems plays an important role in e-commerce and online streaming services, such as Netflix, YouTube and Amazon. Making the right recommendation for the next product, music or movie increases user retention and satisfaction, leading to sales and profit growth. Companies competing for

customer loyalty invest on systems that capture and analyses the user's preferences, and offer products or services with higher likelihood of purchase.

The economic impact of such company-customer relationship is clear: Amazon is the largest online retail company by sales and part of its success comes from the recommendation system and marketing based on user preferences. In this context, a movie recommendation system will be built using machine learning algorithms to predict user rating for movies. This will be achieved by studying and analyzing existing dataset of user rating for movies (edx dataset).

The edx data is a subset of the *movielen* data. It has 9000055 rows and 6 columns. The edx data contains 10677 different movies with ratings respectively. The edx data contains 69878 different users that performed the ratings. The data will be processed and partitioned to get ninety percent training dataset ($p = 0.1$) from the edx dataset. The training dataset obtained from the edx dataset will be analyzed with machine learning algorithm and tested using the test set from the edx data. The residual mean squared error (RMSE) will be evaluated against the final hold out test set (validation set).

Usually recommendation systems are based on a rating scale from 1 to 5 grades or stars, with 1 indicating lowest satisfaction and 5 is the highest satisfaction. Other indicators can also be used, such as comments posted on previously used items; video, music or link shared with friends; percentage of movie watched or music listened; web pages visited and time spent on each page; product category; and any other interaction with the company's web site or application can be used as a predictor.

CHAPTER TWO
METHODS AND ANALYSIS

2.1 PROCESS AND WORKFLOW

The major procedure in a data analysis include:

1. Data preparation: download, parse, import and prepare the data to be processed and analysed.
2. Data exploration and visualization: explore data to understand the features and the relationship between the features and predictors.
3. Data cleaning: eventually the dataset contains unnecessary information that needs to be removed.
4. Data analysis and modeling: create the model using the insights gained during exploration. Also test and validate the model.
5. Communicate: create the report and publish the results.

First is to download the dataset from MovieLens website and split into two subsets used for training and validation. The training subset is called edx and the validation subset is called validation (codes already provided). The edx set is then split again into two subsets used for training and testing. When the model reaches the optimum RMSE in the testing set, again the edx set is further trained with the model and use the validation set for final validation. It is assumed that the validation set is a new data with unknown outcomes.

The next step will be to create charts, tables and statistics summary to understand how the features can impact the outcome. The information and insights obtained during exploration will help to build the machine learning model.

Creating a recommendation system involves the identification of the most important features that helps to predict the rating any given user will give to any movie. To do this, a simple model was built to evaluate RMSE with random

dataset, upper and lower limits of the spread, the first and third quartiles as well as the mean value of rating; for which the mean value produced the best result. Thereafter a more complex linear model was built to add user and movie bias to the mean value. Finally, the user and movie effects receive regularization parameter that penalizes samples with few ratings.

The edx set is used for training and testing, and the validation set is used for final validation to simulate the new data. The edx set is split in 2 parts: the training set and the test set. The same procedure used to create edx and validation sets was also applied. The training set will be 90% of edx data and the test set will be the remaining 10%. The model building is done in the training set, and the test set is used to test the model. When the model is complete, the validation set is used to calculate the final RMSE.

2.2 EXPLORATORY DATA ANALYSIS

To build an efficient movie recommendation system, one needs to first of all study the data carefully to understand patterns in the data which includes user behaviour in terms of movie rating. Users have the option to choose a rating value from 0.5 to 5.0, totaling 10 possible values. This is an unusual scale, so most movies get a rounded value rating, as shown below:

	Rating value	No. of Ratings
1	0.5	76889
2	1	311281
3	1.5	95614
4	2	640384
5	2.5	299484
6	3	1909183
7	3.5	712042
8	4	2330009
9	4.5	473704
10	5	1251464

To further deepen knowledge regarding the task of building a recommendation system with the edx dataset let's take a look at the first few data set (head) of the training set. Ostensibly, the first few data have 5 ratings as shown below, but this is not the actual situation as there are other rating values in the dataset. Technically the median rating value is 4 as revealed in (figure 1).

	userId	movieId	rating	timestamp	title	genres
1	1	122	5	838985046	Boomerang (1992)	Comedy|Romance
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
8	1	356	5	838983653	Forrest Gump (1994)	Comedy|Drama|Romance|War

Meanwhile, user rating is influenced by several factors which cannot be holistically determined, hence, the need to study the user behaviour in terms of rating. The training dataset obtained from the edx dataset has an average rating of `3.512491` and a median rating of `4.0`. To further illustrate the rating patterns, a boxplot of rating in the training set is shown below (figure 1.0)

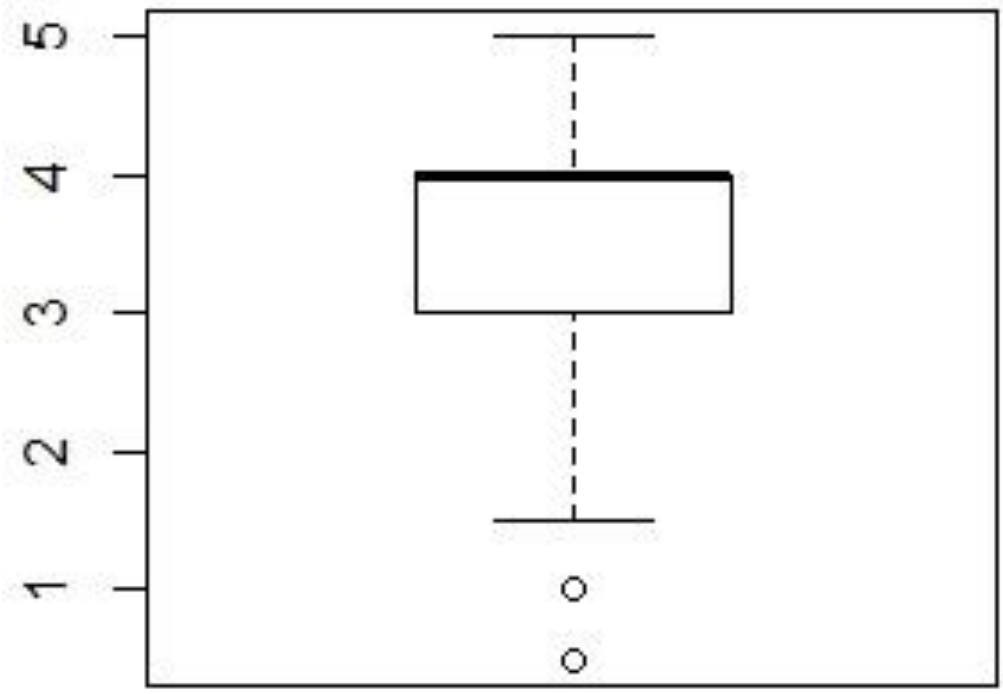


Figure 1.0: A boxplot showing user rating in the training dataset

It is evident from the plot that the median rating is `4.0` while the interquartile range is between `3.0` and `4.0` ratings. This gives an insight into building an efficient system algorithm to predict user behavior in rating movies.

Due to randomness in user's behaviour, human factor can sometimes be difficult to control, the user behavior towards movies needs to be studied as well. A scatterplot can be applied to examine user's behaviour towards movies (figure 2.0).

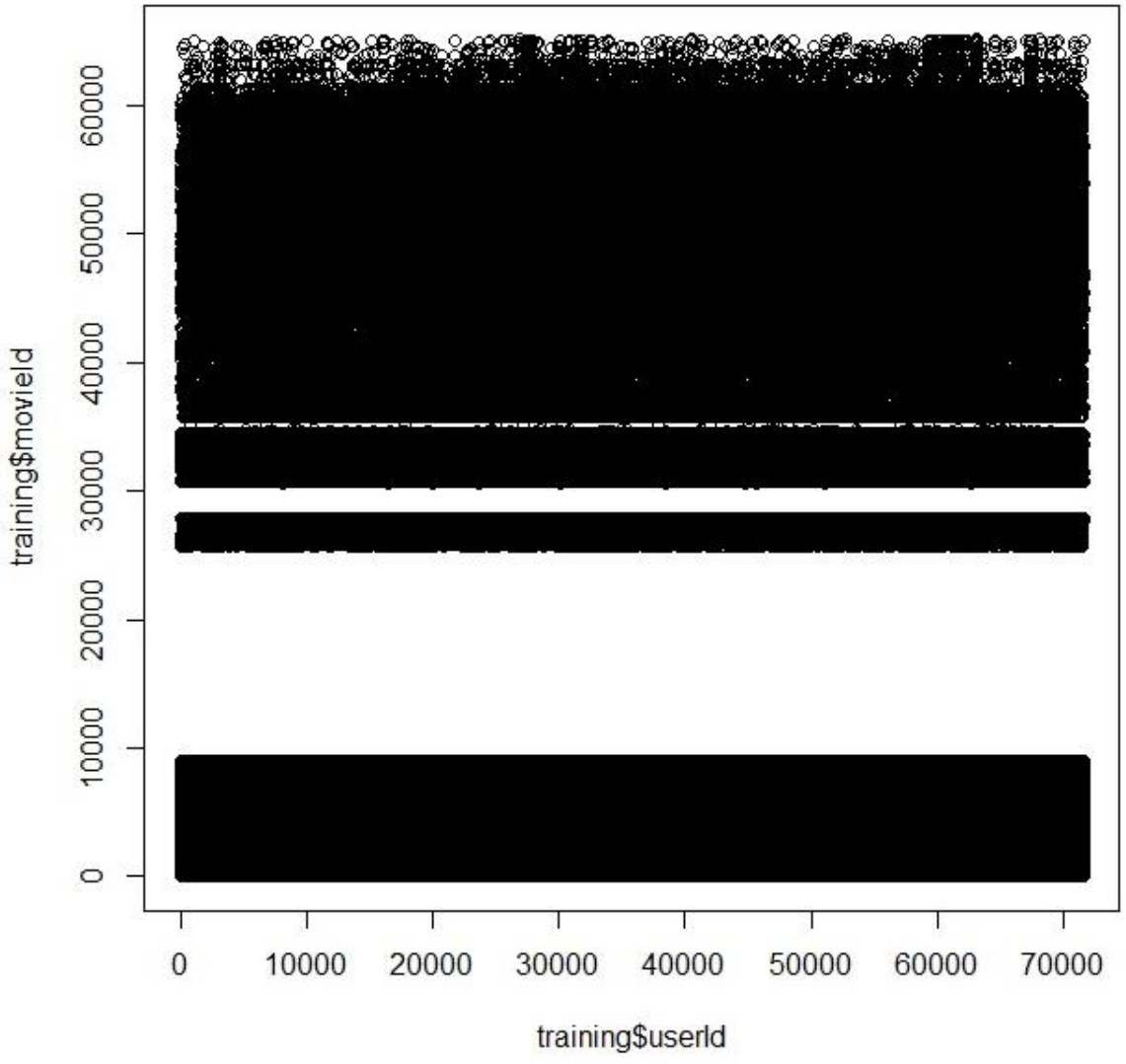


Figure 2.0: A scatterplot showing user's behaviour to movies

However, determining user behaviour towards the rated movies will be instrumental to understanding the bias associated with rating. Some movies are overrated while others are underrated due to some factors peculiar to individual users. The spread of rating (standard deviation) from the mean is `1.060035` which gives further understanding about the rating patterns. This can be better appreciated in the scatterplot (figure 2.0) showing User's behavior to Movies. The scatterplot shows the region of concentration revealing a unique pattern. The dark region at the bottom shows strong concentration which tends to fade away at the top with some vertical gaps in between.

To further explain the patter observed in the User and Movie relationship, it is necessary to study the distributions.

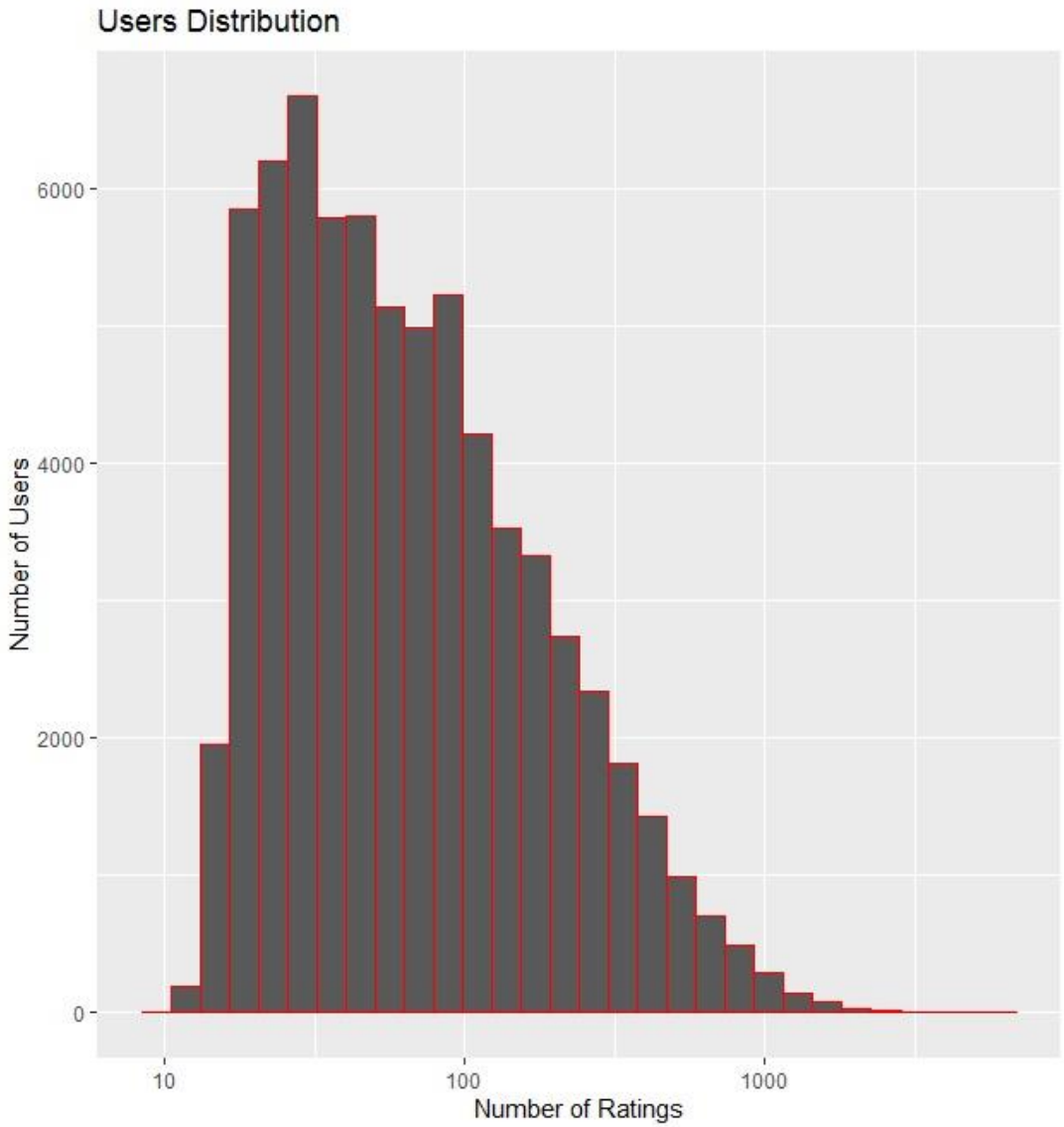


Figure 3.0: User Distribution in the Training Dataset

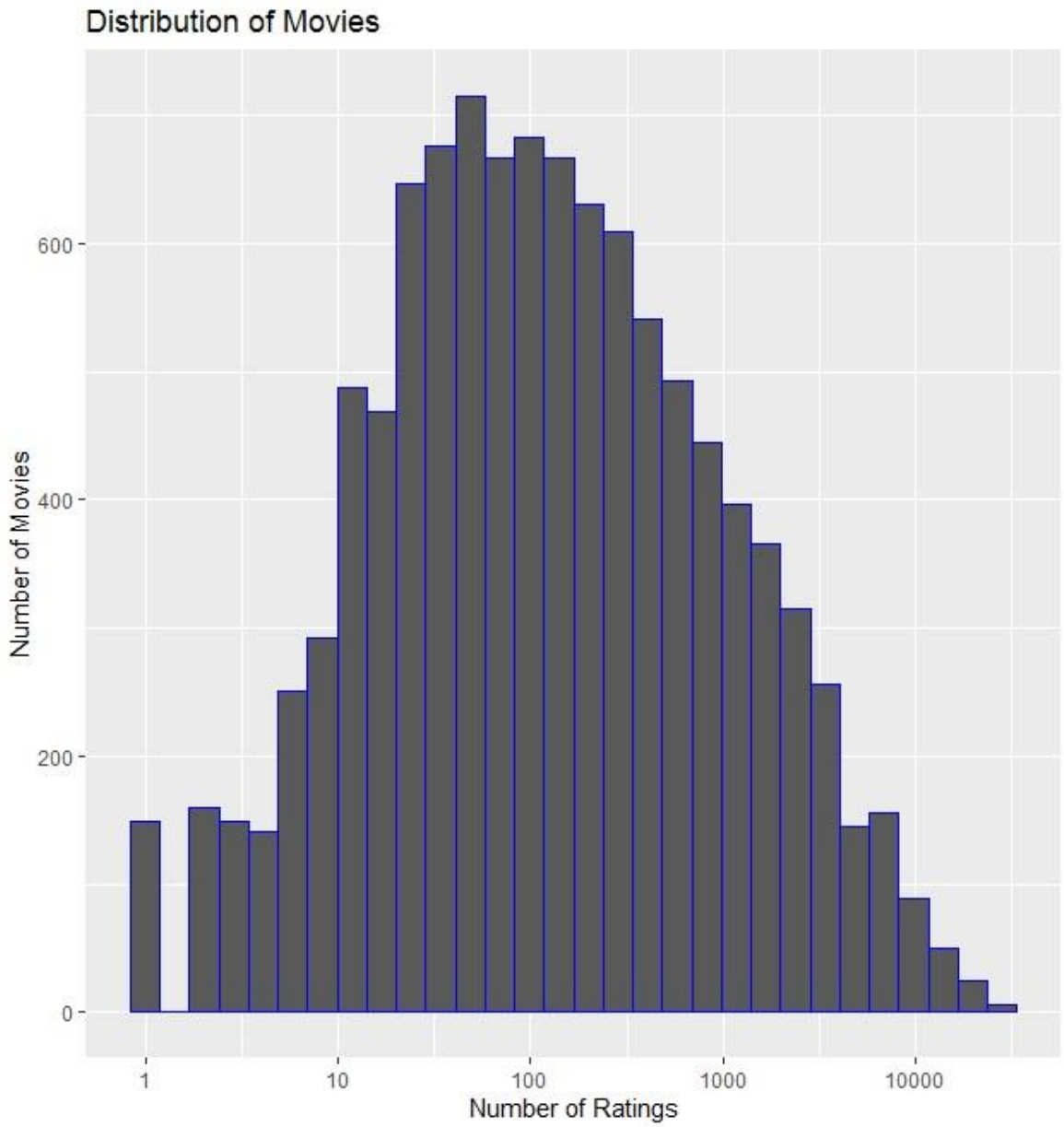


Figure 4.0: Movie Distribution in the Training Dataset

As shown in the plot, the user distribution is right skewed while the movie distribution is seemingly symmetrical. This explains the gap in (figure 2.0). However, the relationship between users and movie rating can be further tested using matrix. The sparsity of data points revealed in the resultant heat map of a sample of user behavioral pattern in movie rating further attests to the uniqueness of the pattern revealed by (figure 2.0).

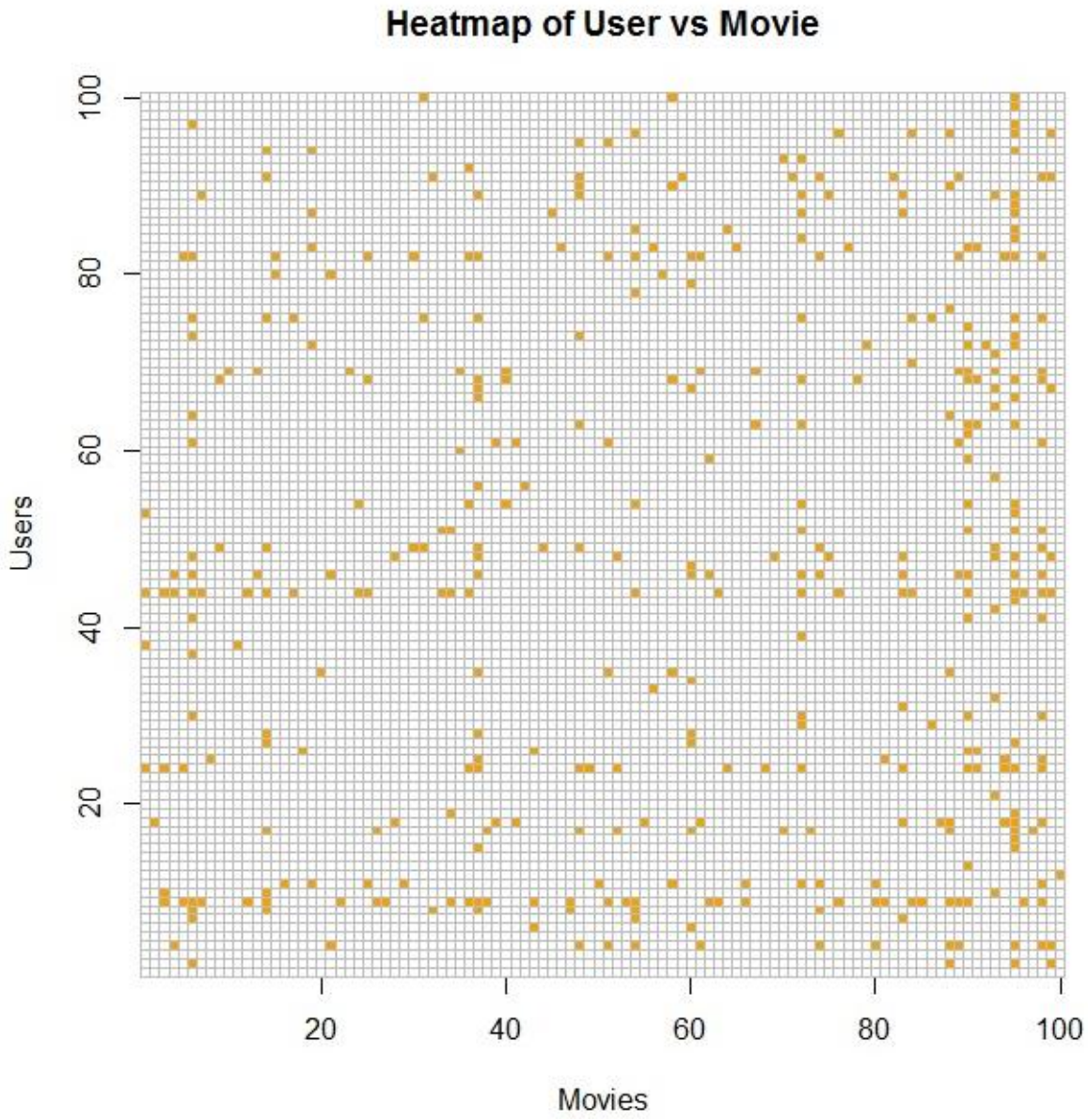


Figure 5.0: Heat of Users and Movies Distribution

In view of the insight gathered so far in the exploratory data analysis, the average rating and the spread is already determined with visualization to reveal hidden patterns in the dataset especially the peculiar user's behaviour towards the various movies (figure 2.0). Consequently, in order to effectively build a movie recommendation system in the context of this project, it is expedient to consider several prediction algorithms and examine their performance level.

2.2 MODEL EVALUATION

The goal of this project is to create a recommendation system with minimal RMSE. However, there is no specific target for the mean squared error (MSE) and mean absolute error (MAE).

The evaluation of machine learning algorithms entails comparing the predicted value with the actual outcome. The loss function measures the difference between both values. There are other metrics that go beyond the scope of this study.

The most common loss functions in machine learning are the mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE). Regardless of the loss function, when the user consistently selects the predicted movie, the error is equal to zero and the algorithm is perfect.

2.3 PREDICTION ALGORITHMS AND PERFORMANCE MEASURE

Since errors are not simply present or absent in a prediction, they come in different sizes and forms, thus, the basic principle in evaluating numerical prediction is to use an independent test set rather than the training set for performance evaluation. The holdout method, and cross-validation, apply equally well to numeric prediction but the basic quality measure offered by the error rate is no longer appropriate, (Witten *et al.* 2011). Here are some prediction performance measure which evaluates the prediction errors in machine learning algorithms.

1. Mean Squared Error - MSE

The mean squared error is the principal and most commonly used measure; sometimes the square root is taken to give it the same dimensions as the predicted value itself. Many mathematical techniques (such as linear regression) use the mean-squared error because it tends to be the easiest measure to manipulate mathematically: it is, as mathematicians say, “well behaved.” However, considering it as a performance measure: all the performance measures are easy

to calculate, so mean-squared error has no particular advantage. The question is, is it an appropriate measure for the task at hand?

2. Mean Absolute Error - MAE

This is an alternative of the mean squared error which measures the average of the magnitude of individual errors without taking account of their sign. Mean-squared error tends to exaggerate the effect of outliers, instances whose prediction error is larger than the others, but absolute error does not have this effect: all sizes of error are treated evenly according to their magnitude. Sometimes it is the relative rather than absolute error values that are of importance. For example, if a 10% error is equally important whether it is an error of 50 in a prediction of 500 or an error of 0.2 in a prediction of 2, then averages of absolute error will be meaningless: relative errors are appropriate. This effect would be taken into account by using the relative errors in the mean-squared error calculation or the mean absolute error calculation,

3. Relative Squared Error - RSE

The error is made relative to what it would have been if a simple predictor had been used. The simple predictor in question is just the average of the actual values from the training data. Thus relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the default predictor.

4. Relative Absolute Error - RAE

It is the total absolute error, with the same kind of normalization. In these three relative error measures, the errors are normalized by the error of the simple predictor that predicts average values.

5. Root Mean Squared Error - RMSE

The Root Mean Squared Error, RMSE, is the square root of the MSE. It is the typical metric to evaluate recommendation systems due to its usefulness in

penalizing large deviations from the mean and is appropriate in cases that small errors are not relevant. Contrary to the means squared error (MSE), the error has the same unit as the measurement.

The ‘squared error’ and ‘root squared error’ estimates weigh large discrepancies much more heavily than small ones, whereas the ‘absolute error estimates’ do not. Taking the square root (root mean-squared error) or residual mean squared error just reduces the figure to have the same dimensionality as the quantity being predicted. The relative error figures try to compensate for the basic predictability or unpredictability of the output variable: if it tends to lie fairly close to its average value, then you expect prediction to be good and the relative figure compensate for this. Otherwise, if the error figure in one situation is far greater than that in another situation, it may be because the quantity in the first situation is inherently more variable and therefore harder to predict, not because the predictor is any worse.

Finally, to answer the underlying question: “which of these measures or error estimates is more appropriate in any given situation?” is only a matter that can be determined by studying the application itself; what to minimize? What is the cost of different kinds of error? Often it is not easy to decide. Fortunately, it turns out that in most practical situations the best numeric prediction method is still the best no matter which error measure is used, (Witten *et al.*, 2011).

According to Witten *et al.* (2011), several alternative measures, summarized in the table below can be used to evaluate the success of numeric prediction. The predicted values on the test instances are $\boldsymbol{P_1, P_2,...,P_n}$; the actual values are $\boldsymbol{a_1, a_2,...,a_n}$. Note that $\boldsymbol{P_i}$ refers to the numeric value of the prediction for the $\boldsymbol{i^{th}}$ test instance.

Table 1.0: Performance measures for numeric predictions.

Performance	Formula
Mean-square error	$\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}$
Root mean-square error	$\sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}}$
Mean absolute error	$\frac{\lvert p_1 - a_1\rvert + \ldots + \lvert p_n - a_n\rvert}{n}$
Relative squared error	$\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \ldots + (a_n - \bar{a})^2}$
Root relative squared error	$\sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \ldots + (a_n - \bar{a})^2}}$
Relative absolute error	$\frac{\lvert p_1 - a_1\rvert + \ldots + \lvert p_n - a_n\rvert}{\lvert a_1 - \bar{a}\rvert + \ldots + \lvert a_n - \bar{a}\rvert}$

Note: *p* are the predicted values, and *a* are the actual values
Source: Witten *et al.* 2011

2.4 LEAST SQAURES ESTIMATION

In practice, prediction analysis using least squares method involves a collection of observations without known values of coefficient $\beta_0, \beta_1, \ldots \beta_k$ which need to be estimated from the data. The least squares principle provides a way of choosing the coefficients effectively by minimizing the sum of the squared errors, that is choosing the values of the unknowns ($\beta_0, \beta_1, \ldots \beta_k$) that minimizes:

$$\sum \varepsilon_n^2 = \sum (y_n - \beta_0 - \beta_1 x_{1,n} - \ldots - \beta_k x_{k,n})^2$$

This is called the least squares estimation because it gives the least value for the sum of squared errors. Finding the best estimates of the coefficient is often called fitting the model to the data.

It is pertinent to mention that prediction errors cannot be annihilated but minimized, thus the error estimate will always be visible because a user may either overrate or underrate a particular movie due to individual preferences "a fan of comedy genre may underrate a horror movie and overrate a comedy movie and vice versa". It is also possible for the movie attributes to add bias to rating "a movie to showcases fancy or popular artiste may be overrated by those that appeals to fashion, secondly a low visual quality may affect movie rating rather than content, etc". In this context, the relationship between the variables of estimation (user and movie effects) can be given as:

$$y_n = \beta_0 + \beta_1 x_{1,n} + \ldots + \beta_k x_{k,n} + \varepsilon_n$$

Where y_n is the predictive rating index, β_0 is the average rating, β_1 is the rating bias due user effect, β_k is the rating bias due to movie effect, ε_n is the rating error, while $x_{1,k,n}$ are estimation variables.

2.5 THE LOSS FUNCTION

Loss function is a machine learning algorithm that helps to minimize the predictive error by ensuring that the predicted outcome is minimally deviated from the actual values. It is a method of evaluating how well specific algorithm models the given data. If predictions deviates too much from actual results, loss function would reveal very large number. Gradually, with the help of some optimization function, loss function learns to reduce the error in prediction. There is no one-size-fits-all loss function to algorithms in machine learning. There are various factors involved in choosing a loss function for specific problem such as type of machine learning algorithm chosen, ease of calculating the derivatives and to some degree the percentage of outliers in the data set.

Machines learn by means of a loss function. Broadly, loss functions can be classified into two major categories depending upon the type of learning task. Regression losses and Classification losses. In classification, the task is to

predict output from set of finite categorical values i.e given large data set of images of hand written digits, categorizing them into one of 0 – 9 digits. While in Regression losses, on the other hand, deals with predicting a continuous value for example given floor area, number of rooms, size of rooms, predict the price of room or a movie rating.

The loss function can be represented mathematically a the square root of the mean squared error

Mean Squared Error $$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

Thus the loss function can be evaluated as the residual MSE by taking the squared root.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value and n is the size of data.

Note that the RMSE corresponds to the root mean square error formula as given in table 1.0 above.

2.6 MODELING

From the foregoing, it is obvious that to build a functional model for movie recommendation system from the edx dataset, the average rating must be determined which is now presumably is the expected rating for all users irrespective of individual differences.

2.6.1 RANDOM PREDICTION

A very simple model is just randomly predicting the rating using the probability distribution observed during the data exploration. For example, if we know the probability of all users giving a movie a rating of 3 is 10%, then we may guess

that 10% of the ratings will have a rating of 3. Such prediction sets the worst error possible, so any other model should provide better result. In this study, the random value was selected by Monte Carlo process. Since the training set is a sample of the entire population and the real distribution of ratings is unknown, thus Monte Carlo simulation with replacement provides a good approximation of the rating distribution.

2.6.2 PREDICTION WITH CENTRAL TENDENCIES

The result of predicting with the mean value, upper and lower limits of the spread, the first and third quartiles further explains some characteristics about the dataset. There was considerable improvement in prediction with the quartiles. This is a new idea which need to be thoroughly investigated in building machine learning algorithms, although the mean performed better.

2.6.3 PREDICTION WITH LINEAR MODEL

The simplest model predicts all users will give the same rating to all movies and assumes the movie to movie variation is the randomly distributed error. Although the predicted rating can be any value, statistics theory says that the average minimizes the RMSE, so the initial prediction is just the average of all observed ratings, as described in this formula:

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

Where $\hat{Y}$ is the predicted rating, μ is the mean of observed data and $\epsilon_{i,u}$ is the error distribution. Any value other than the mean increases the RMSE, so this is a good initial estimation.

Part of the movie to movie variability can be explained by the fact that different movies have different rating distribution. This is easy to understand, since some movies are more popular than others and the public preference varies. This is called movie effect or movie bias, and is expressed as b_i in this formula:

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

The movie effect can be calculated as the mean of the difference between the observed rating y and the mean μ.

$$\hat{b}_i = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{\mu})$$

Similar to the movie effect, different users have different rating pattern or distribution. For example, some users like most movies and consistently rate 4 or 5, while other users dislike most movies rating 1 or 2. This is called user effect or user bias and is expressed in this formula:

$$\hat{b}_u = \frac{1}{N}\sum_{i=1}^{N}(y_{u,i} - \hat{b}_i - \hat{\mu})$$

The prediction model that includes the user effect becomes:

$$\hat{Y}_{u,i} = \mu + b_i + b_u + \epsilon_{u,i}$$

The distribution of the respective movie and user bias (b_i and b_u) of the model is graphically represented by (figures 6 and 7). It is evident in (figure 6) that the bias due to movie effect is majorly skewed towards negative values, while (figure 7), reveals that the User bias is slightly skewed towards positive values.

However, movies can be grouped into genres, with different distributions. In general, movies in the same genre get similar ratings but that is beyond the scope of this project.

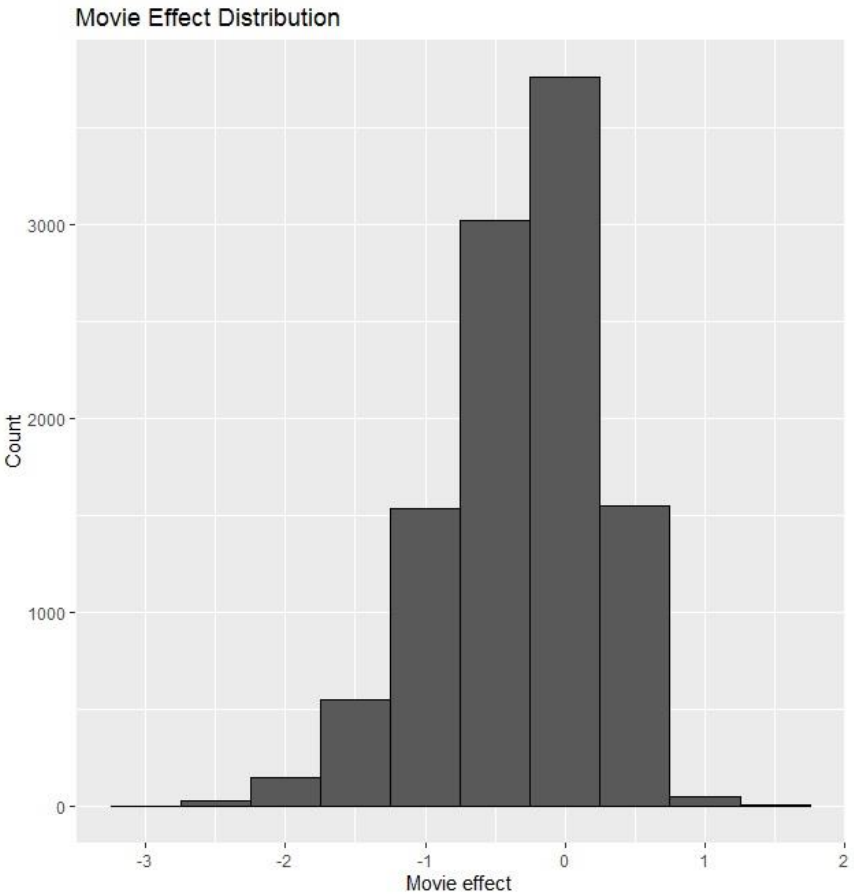


Figure 6.0 Distribution of Movie Bias (b_i)

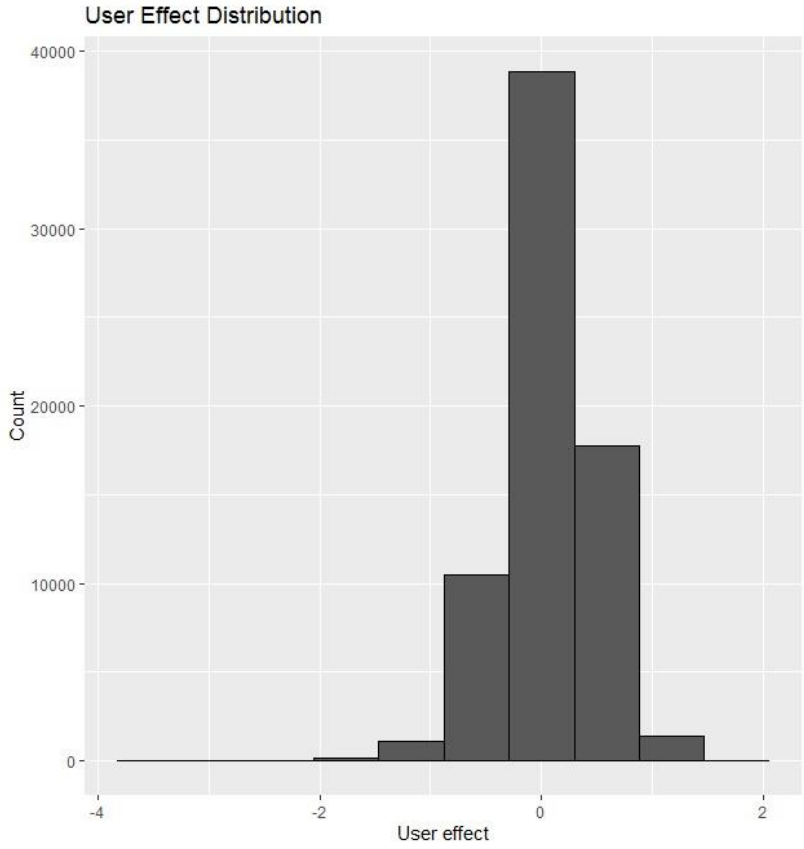


Figure 7.0 Distribution of User Bias (b_u)

2.6.4 REGULARIZATION

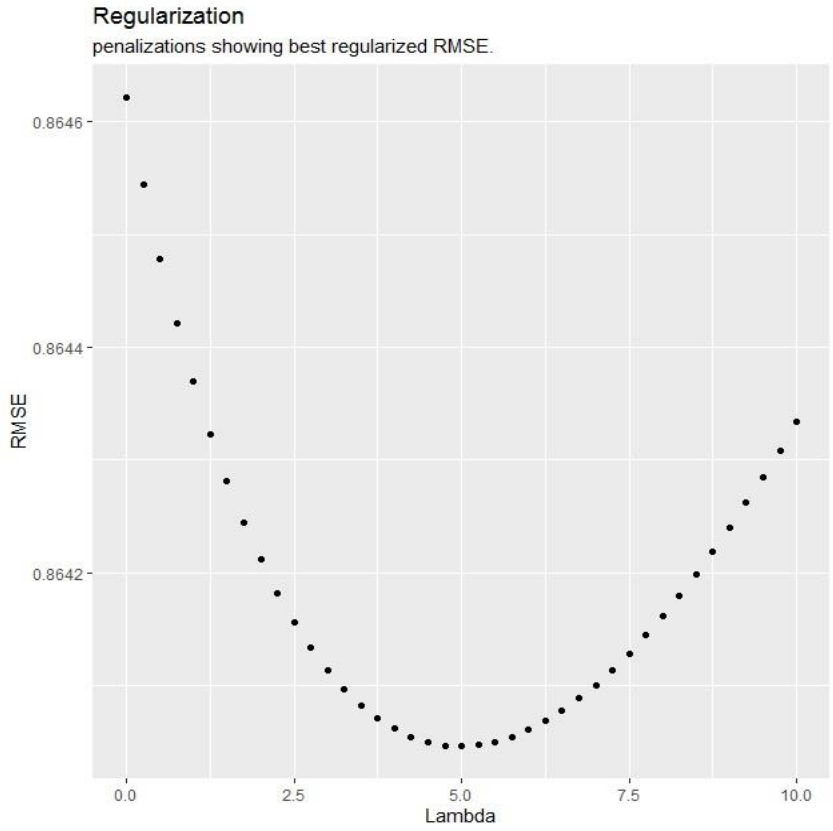
The linear model provides a good estimation for the ratings, but does not consider that many movies have very few number of ratings, and some users rate very few movies. This means that the sample size is very small for these movies and these users. Statistically, this leads to large estimated error.

The estimated value can be improved adding a factor that penalizes small sample sizes and have little or no impact otherwise. Thus, estimated movie and user effects can be calculated with these formulas:

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

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

For values of N smaller than or similar to λ, $\hat{b}_i$ and $\hat{b}_u$ is smaller than the original values, whereas for values of N much larger than λ, $\hat{b}_i$ and $\hat{b}_u$ change very little. An effective method to choose λ that minimizes the RMSE is running simulations with several values of λ otherwise known as penalizations.



CHAPTER THREE

RESULT

To show model result, first the model evaluation function ought to be defined. In this study, three error terms will be evaluated, the Mean Absolute Error (MAE), Mean Squared Error (MSE) and the Root Mean Squared Error (RMSE) but only the RMSE will be investigated.

Note that the result tables are continuously updated to add the new model which makes it easy to compare performance of the model. This study assumes that at the initial stage, the error term is zero until proven otherwise, which is also reflected the result table as *Initial Assumption for Prediction Error*.

The naïve RMSE for model_result 1 to 9 examined different algorithms using the training set and tested with the test_set. The least squares estimation using linear model identified and applied biases due to movie and user effect. The biases was penalized by regularization with lambda, which seem to produce the best RMSE.

NAIVE-RMSE-RESULT USING RANDOM PREDICTION

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.000000	0.000000	0.000000
➡ Model_Result_1 (Random Prediction)	1.497961	2.243888	1.164816

NAIVE-RMSE-RESULT USING TRAINING DATA

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.000000	0.000000	0.000000
Model_Result_1 (Random Prediction)	1.497961	2.243888	1.164816
➡ Model_Result_2 (Training Data)	1.498786	2.246360	1.165267

NAIVE-RMSE-RESULT USING LOWER LIMIT OF THE DATA SPREAD
(Average rating minus 1 sd)

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.000000	0.000000	0.000000
Model_Result_1 (Random Prediction)	1.497961	2.243888	1.164816
Model_Result_2 (Training Data)	1.498786	2.246360	1.165267
➡ Model_result_3 (Lower Limit of Spread)	1.498904	2.246712	1.302233

NAIVE-RMSE-RESULT USING UPPER LIMIT OF THE DATA SPREAD
(Average rating plus 1 sd)

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.000000	0.000000	0.000000
Model_Result_1 (Random Prediction)	1.497961	2.243888	1.164816
Model_Result_2 (Training Data)	1.498786	2.246360	1.165267
Model_result_3 (Lower Limit of Spread)	1.498904	2.246712	1.302233
➡ Model_result_4 (Upper Limit of Spread)	1.499218	2.247654	1.192231

NAIVE-RMSE-RESULT USING THE 1ST QUARTILE

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.000000	0.000000	0.0000000
Model_Result_1 (Random Prediction)	1.497961	2.243888	1.1648163
Model_Result_2 (Training Data)	1.498786	2.246360	1.1652666
Model_result_3 (Lower Limit of Spread)	1.498904	2.246712	1.3022327
Model_result_4 (Upper Limit of Spread)	1.499218	2.247654	1.1922312
➡ Model_result_5 (First Quartile)	1.176911	1.385120	0.9434307

NAIVE-RMSE-RESULT USING THE THIRD QUARTILE

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.000000	0.000000	0.0000000
Model_Result_1 (Random Prediction)	1.497961	2.243888	1.1648163
Model_Result_2 (Training Data)	1.498786	2.246360	1.1652666
Model_result_3 (Lower Limit of Spread)	1.498904	2.246712	1.3022327
Model_result_4 (Upper Limit of Spread)	1.499218	2.247654	1.1922312
Model_result_5 (First Quartile)	1.176911	1.385120	0.9434307
➡ Model_Result_6 (Third Quartile)	1.166442	1.360588	0.8547722

NAIVE-RMSE-RESULT USING OBSERVED RATING MEAN

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.000000	0.000000	0.0000000
Model_Result_1 (Random Prediction)	1.497961	2.243888	1.1648163
Model_Result_2 (Training Data)	1.498786	2.246360	1.1652666
Model_result_3 (Lower Limit of Spread)	1.498904	2.246712	1.3022327
Model_result_4 (Upper Limit of Spread)	1.499218	2.247654	1.1922312
Model_result_5 (First Quartile)	1.176911	1.385120	0.9434307
Model_Result_6 (Third Quartile)	1.166442	1.360588	0.8547722
➡ Model_Result_7 (Rating Mean)	1.059577	1.122704	0.8548865

ADDING MOVIE BIAS (bi) TO THE LINEAR MODEL

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.0000000	0.0000000	0.0000000
Model_Result_1 (Random Prediction)	1.4979612	2.2438877	1.1648163
Model_Result_2 (Training Data)	1.4987862	2.2463600	1.1652666
Model_result_3 (Lower Limit of Spread)	1.4989037	2.2467122	1.3022327
Model_result_4 (Upper Limit of Spread)	1.4992178	2.2476541	1.1922312
Model_result_5 (First Quartile)	1.1769113	1.3851202	0.9434307
Model_Result_6 (Third Quartile)	1.1664424	1.3605878	0.8547722
Model_Result_7 (Rating Mean)	1.0595771	1.1227036	0.8548865
➡ Model_Result_8 (Mean + bi)	0.9432467	0.8897143	0.7375429

ADDING USER BIAS (bu) TO THE LINEAR MODEL

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.0000000	0.0000000	0.0000000
Model_Result_1 (Random Prediction)	1.4979612	2.2438877	1.1648163
Model_Result_2 (Training Data)	1.4987862	2.2463600	1.1652666
Model_result_3 (Lower Limit of Spread)	1.4989037	2.2467122	1.3022327
Model_result_4 (Upper Limit of Spread)	1.4992178	2.2476541	1.1922312
Model_result_5 (First Quartile)	1.1769113	1.3851202	0.9434307
Model_Result_6 (Third Quartile)	1.1664424	1.3605878	0.8547722
Model_Result_7 (Rating Mean)	1.0595771	1.1227036	0.8548865
Model_Result_8 (Mean + bi)	0.9432467	0.8897143	0.7375429
➡ Model_Result_9 (Mean + bi + bu)	0.8646215	0.7475703	0.6684608

REGULARIZATION OF MOVIE AND USER BIASES

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.0000000	0.0000000	0.0000000
Model_Result_1 (Random Prediction)	1.4979612	2.2438877	1.1648163
Model_Result_2 (Training Data)	1.4987862	2.2463600	1.1652666
Model_result_3 (Lower Limit of Spread)	1.4989037	2.2467122	1.3022327
Model_result_4 (Upper Limit of Spread)	1.4992178	2.2476541	1.1922312
Model_result_5 (First Quartile)	1.1769113	1.3851202	0.9434307
Model_Result_6 (Third Quartile)	1.1664424	1.3605878	0.8547722
Model_Result_7 (Rating Mean)	1.0595771	1.1227036	0.8548865
Model_Result_8 (Mean + bi)	0.9432467	0.8897143	0.7375429
Model_Result_9 (Mean + bi + bu)	0.8646215	0.7475703	0.6684608
➡ Regularized model (bi and bu)	0.8640459	0.7465753	0.6687106

FINAL TESTING USING FINAL HOLD OUT SET (VALIDATION)

Method	RMSE	MSE	MAE
Initial Assumption for Prediction Error	0.0000000	0.0000000	0.0000000
Model_Result_1 (Random Prediction)	1.4979612	2.2438877	1.1648163
Model_Result_2 (Training Data)	1.4987862	2.2463600	1.1652666
Model_result_3 (Lower Limit of Spread)	1.4989037	2.2467122	1.3022327
Model_result_4 (Upper Limit of Spread)	1.4992178	2.2476541	1.1922312
Model_result_5 (First Quartile)	1.1769113	1.3851202	0.9434307
Model_Result_6 (Third Quartile)	1.1664424	1.3605878	0.8547722
Model_Result_7 (Rating Mean)	1.0595771	1.1227036	0.8548865
Model_Result_8 (Mean + bi)	0.9432467	0.8897143	0.7375429
Model_Result_9 (Mean + bi + bu)	0.8646215	0.7475703	0.6684608
Regularized model (bi and bu)	0.8640459	0.7465753	0.6687106
➡ Final Model Testing (edx vs validation)	**0.8648182**	0.7479105	0.6693494

As seen from the model result tables, the RMSE for random prediction, training data and deviations produced similar result (approximately `1.49`). But the central values such as the mean, median/upper quartile and lower quartile produced a much better result, however controlling bias such as movie effect (bi) and user effect (bu) as well as regularization produced more valuable results. Consequently, using final holdout set (validation data) to test the model produced the desired result though slightly above the regularized model for the test set.

EVALUATION OF RESULT USING VALIDATION SET

The best ten(10) rated movies using prediction model from validation set are:

```
 1 Usual Suspects, The (1995)
 2 Shawshank Redemption, The (1994)
 3 Shawshank Redemption, The (1994)
 4 Shawshank Redemption, The (1994)
 5 Eternal Sunshine of the Spotless Mind (2004)
 6 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)
 7 Schindler's List (1993)
 8 Donnie Darko (2001)
 9 Star Wars: Episode VI - Return of the Jedi (1983)
10 Schindler's List (1993)
```

The worst ten(10) rated movies using prediction model from validation set are:

```
 1 Battlefield Earth (2000)
 2 Police Academy 4: Citizens on Patrol (1987)
 3 Karate Kid Part III, The (1989)
 4 PokÃ©mon Heroes (2003)
 5 Turbo: A Power Rangers Movie (1997)
 6 Kazaam (1996)
 7 PokÃ©mon Heroes (2003)
 8 Free Willy 3: The Rescue (1997)
 9 Shanghai Surprise (1986)
10 Steel (1997)
```

EVALUATION OF RESULT USING TEST SET

The best ten (10) rated movies using prediction model from test set are:

```
 1 Shawshank Redemption, The (1994)
 2 Star Wars: Episode V - The Empire Strikes Back (1980)
 3 Silence of the Lambs, The (1991)
 4 Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark)(1981)
 5 Manchurian Candidate, The (1962)
 6 Matrix, The (1999)
 7 Butch Cassidy and the Sundance Kid (1969)
 8 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)
 9 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)
10 Shawshank Redemption, The (1994)
```

The worst ten(10) rated movies using prediction model from test set are:

```
 1 Battlefield Earth (2000)
 2 Barney's Great Adventure (1998)
 3 Grease 2 (1982)
 4 From Justin to Kelly (2003)
 5 Shanghai Surprise (1986)
 6 Children of the Corn III (1994)
 7 From Justin to Kelly (2003)
 8 Grease 2 (1982)
 9 Glitter (2001)
10 Gigli (2003)
```

CHAPTER FOUR

4.1 SUMMARY AND CONCLUSION

The idea of building a movie recommendation system is to simulate rating pattern of users from databases and subsequently make recommendation based on some results generated from machine learning algorithms. The main aim of a recommendation system are:

1. A recommendation system provides its users with relevant contents and substitutes based on their preferences and likes.
2. A recommendation system takes the information about the user as an input and analyze it with machine learning algorithms to predict the user's behaviour
3. The primary goal of recommendation systems is to help users find what they want based on their preferences and previous interactions, and predicting the rating for a new item

Nevertheless, to achieve these aims, Movielens data was assembled using "provided" code. The collected data was processed, cleaned and applied to carry out the analysis, then an exploratory data analysis was conducted by visualizing the dataset to get more insight into the data structure, which might was very useful in model building.

A random model was created to predict the rating based on the probability distribution of each rating. The central tendencies or measures such as mean, upper and lower quartiles and the spread were also modeled to determine which produces the best RMSE result. The random prediction actually gave the worst result, while the mean gave a better result, the upper and lower quartiles gave an exceptional results which need to be further studied for more insight.

Finally, a simple linear model with the mean of the observed ratings was created. Then bias due to movie effect (bi) was added and lastly bias due to user effect (bu) was added. The model was then penalized by adding lambda with regularization which added a penalty value (optimal lambda) for the movies and users with few number of ratings. The linear model achieved the RMSE of 0.8648182.

Note that in this study, other error estimates such as the Mean Squared Error (MSE) and Mean Absolute Error (MAE) were also investigated with results displayed. But these two error terms produced lesser error values with every model - a clear indication that both the MSE and MAE tends to downplay the error terms while the RMSE seems to give a better error estimate, hence it was the main focus for building prediction model for the recommendation system in this study.

4.2 LIMITATIONS OF THE STUDY

The movie recommendation system is only effective for archived dataset because the system is not equipped to automatically collect data for implementation. Thus The model works only for existing users, movies and rating values, so the algorithm must run every time a new user or movie is included, or when the rating changes.

Another delineating factor is that there is no initial recommendation for a new user or for users that usually do not rate movies, whereas algorithms that use several features as predictors can overcome this issue.

There are enormous challenges that impedes actualization of this project, because the machine learning algorithms computationally overwhelmed my computer

system even though only two predictors are used, the movie and user information, not considering other features. Meanwhile, in reality, modern recommendation system models might as well use many predictors, such as genres, bookmarks, playlists, etc. Since most commodity laptops are limited in capacity, running the codes are sometimes frustrating and daunting. The required amount of memory far exceeded the available in a commodity laptop, even with increased virtual memory.

4.3 FUTURE WORK

This study only examines a model for recommendation system that works only for existing users, movies and rating values. To improve the system, the algorithm should consist of Artificial Intelligence (AI) architecture whereby sensors and actuators are applied to collect and utilize data in real time. The database and knowledge-base should be well implemented for modelling machine learning algorithms.

Since the Least Squares Estimate (lm function) could not be implemented in this study due to system limitation, it is recommended the model derived in this study is analyzed using the lm function for learning purpose.

Since there was no initial recommendation for a new user or for users that usually do not rate movies, in the future it is recommended that all other features or factors that may directly or indirectly affect rating should be considered as predictors.

REFERENCES

Georgios Drakos (2018). How to select the Right Evaluation Metric for Machine Learning Models: Part 2 Regression Metrics

Michael Hahsler (2019). recommendationlab: Lab for Developing and Testing recommendation Algorithms. R package version 0.2-5.

Parmar, R., (2018). Common Loss functions in machine learning. Towards data science. Available at https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23

Rafael A. Irizarry (2019). Introduction to Data Science: Data Analysis and Prediction Algorithms with R

Witten I. H., Frank E., Hall M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques. Third Edition. Morgan Kaufmann Series, USA. Elsevier, ISBN 978-0-12-374856-0