HarvardX - Data Science Professional Certificate
Capstone Project
- MovieLens -

Anass Latif

March 31, 2019

Contents

1. Introduction / Overview / Executive Summary

Background and Motivation

Recommendation systems are one of the most famous machine learning models. They are extensively used by many companies (eg: Netflix, Amazon, Facebook, etc.) to improve and enhance user experiences and increase revenues by recommending the most relevant products to their customers.

This **Harvard Data Science Capstone** is the final assignment for HarvardX - Data Science Professional Certificate from Harvard University.

This project is motivated by the Netflix challenge that was organized in October, 2006. Netflix offered one million dollars reward to anyone that could improve their recommendation systems by 10%.

Dataset

For this assignment, we will go throught all the steps to create a movie recommendation system using the MovieLens dataset, collected by GroupLens Research as the Netflix Datasets are private.

We will be using the 10M version of the MovieLens dataset to make the computation a little easier.

Goal

The objective of this report is to predict, in the most accurate and comprehensive way, the user ***movie ratings*** by implementing, testing and validating different machine learning models.

The outcome is to provide a minimal typical error or RMSE (Root Mean Square Error) on the validation dataset with ***`RMSE lower or equal to 0.87750`***.

The RMSE is defined as:

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

We define $y_{u,i}$ as the rating for movie i by user u and denote our prediction with $\hat{y}_{u,i}$ with N being the number of user/movie combinations and the sum occurring over all these combinations.

Key Steps

To achieve the project objectives, we will follow a comprehensive machine learning workflow:

1. Run the R code provided by Edx staff to generate the datasets. The script execute the following steps:

- Download the MovieLens 10M dataset
- Split the MovieLens dataset into training (`edx`) and test (`validation`) datasets.

2. Explore the `edx` dataset to discover the data and the available features. We will use some exploratory techniques such as data description, preparation, exploration and visualization.
3. Develop and train different predictive models and algorithms in order to find a recommendation model with the best possible outcome (RMSE) that meets our goals.

4. Explain the results and conclude.

All the project will be made through RStudio (version 3.5.3) using some useful packages (eg: dplyr, tidyverse, lubridate, caret, etc.).

This report doesn't display the R code used to generate the information. All scripts are available in My GitHub Repository.

2. Methods / Analysis

Data Preparation

Dataset Generation

The generated datasets are divided into two subsets:

- a training subset to train our algorithm, called `edx`. This subset represents `90%` of the generated dataset.
- a validation subset to predict the movie ratings, called `validation`. This subset represents `10%` of the generated dataset.

Dataset Description

The `edx` dataset contains ***9000055*** observations (rows) of ***6*** variables (columns). There is no missing values (***0*** na).

The `validation` dataset contains ***999999*** observations (rows) of ***6*** variables (columns). There is no missing values (***0*** na).

The features / variables identified in both datasets are:

- `userId`: `integer` variable that represents the unique identification number for each user.
- `movieId`: `numeric` variable that represents the unique identification number for each movie.
- `rating`: `numeric` variable that represents the rating of one movie by one user. Ratings are made on a 5-star scale (0 to 5), with half-star increments.
- `timestamp`: `integer` variable that represents the number of seconds since midnight UTC (1970-JAN-01).
- `title`: `character` variable that represents the movie title including the year of the release.
- `genres`: `character` variable that represents a pipe-separated list of genres affected to each movie.

Let's display a sample of the `edx` dataset.

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

Dataset Preprocessing (Feature Selection and Engineering)

We notice that the `genres` are pipe-separated values. We might need this feature `movieGenre` to predict the movie rating. However, we should extract individual value for more consistent and robust estimate.

We also observe that the movie title contains the year where the movie was released. We might need this feature `releaseYear` to predict the movie rating.

In addition, we can extract the year where the movie was rated from `timestamp` to define the feature as `ratingYear`.

After preprocessing the data, the `edx` dataset looks like:

userId	movieId	rating	title	releaseYear	genres	ratingYear
1	122	5	Boomerang	1992	Comedy|Romance	1996
1	185	5	Net, The	1995	Action|Crime|Thriller	1996
1	292	5	Outbreak	1995	Action|Drama|Sci-Fi|Thriller	1996
1	316	5	Stargate	1994	Action|Adventure|Sci-Fi	1996
1	329	5	Star Trek: Generations	1994	Action|Adventure|Drama|Sci-Fi	1996
1	355	5	Flintstones, The	1994	Children|Comedy|Fantasy	1996

To summarize, the features that could be selected in the machine learning models to predict the `rating` are:

- `movieId`
- `userId`
- `movieGenre`
- `releaseYear`
- `ratingYear`

Exploratory Data Analysis (EDA)

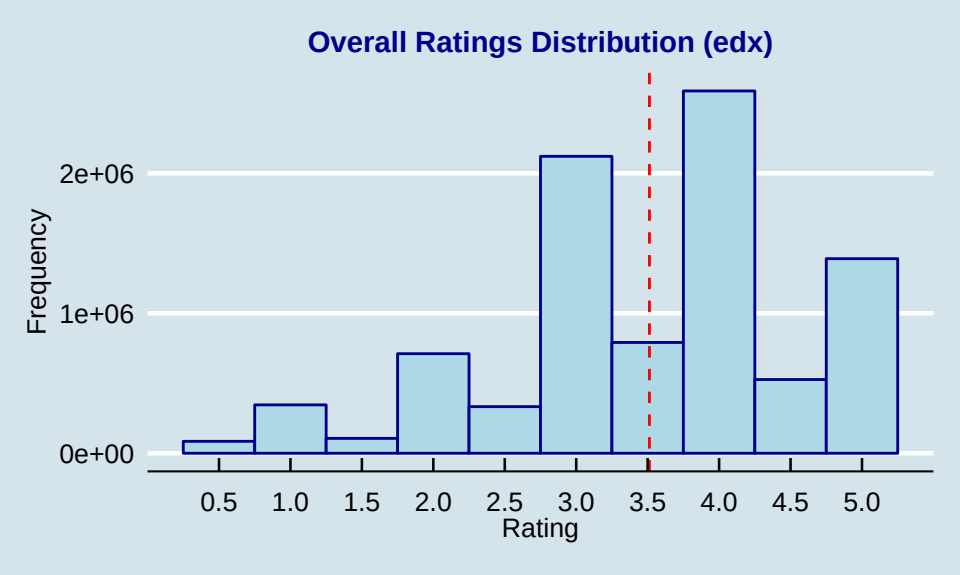
The `edx` dataset contains ***69878*** distinct users who rated ***10677*** distinct movies that are classified in ***20*** distinct genres.

The `validation` dataset contains ***68534*** distinct users who rated ***9809*** distinct movies that are classified in ***19*** distinct genres.

Let's explore our `edx` dataset using some visualization techniques to build more comprehensive understanding of the data. So, here are some questions that we raise:

Rating Analysis

What is the overall ratings distribution?

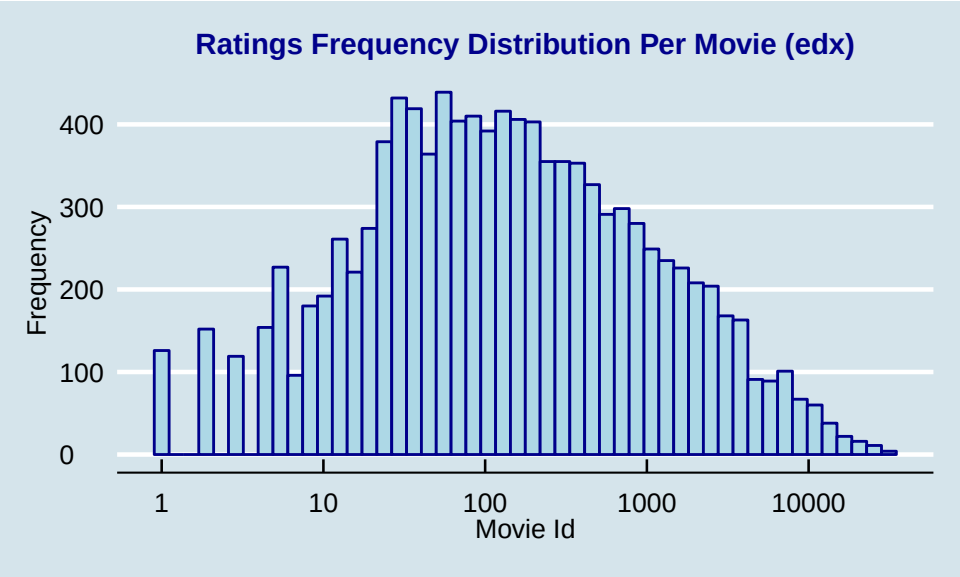


The figure *"Overall Ratings Distribution (edx)"* shows the rating distribution in the `edx` dataset. The `rating` variable has a left-skewed distribution which shows that there are more *"good"* ratings than *"bad"* ratings.

The vertical dashed line represents the overall rating average μ (***3.5124652***) across all users and all movies. We notice also that the rating range from 0.5 to 5 with the most common rating is 4.0 followed closely by 3.0. It also appears that we have a predilection for whole numbered ratings (full star rates) instead of half-star increment ratings.

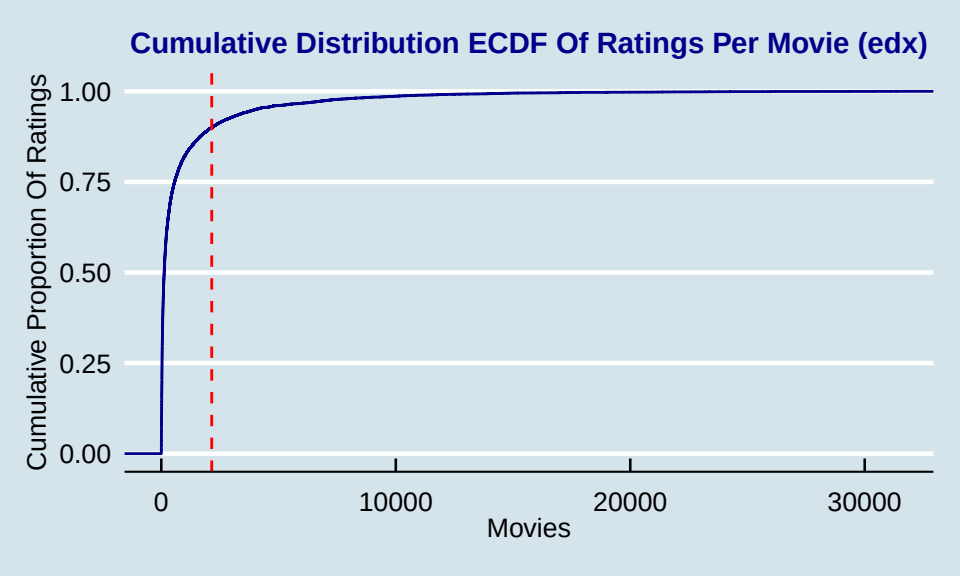
Movie Analysis

How frequently are movies rated?



The figure *"Ratings Frequency Distribution Per Movie (edx)"* shows that some movies are more popular than others, so they are much more rated.

What is the cumulative rating distribution by movie?



The figure *"Cumulative Distribution ECDF Of Ratings Per Movie (edx)"* shows that ***2150*** distinct movies receive ***90%*** of the ratings.

What are the top 10 rated movies?

movieId	title	**Rating_Count**	**Rating_Average**
3226	Hellhounds on My Trail	**1**	**5.00**
33264	Satan's Tango (Sátántangó)	**2**	**5.00**
42783	Shadows of Forgotten Ancestors	**1**	**5.00**
51209	Fighting Elegy (Kenka erejii)	**1**	**5.00**
53355	Sun Alley (Sonnenallee)	**1**	**5.00**
64275	Blue Light, The (Das Blaue Licht)	**1**	**5.00**
5194	Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva)	**4**	**4.75**
26048	Human Condition II, The (Ningen no joken II)	**4**	**4.75**
26073	Human Condition III, The (Ningen no joken III)	**4**	**4.75**
65001	Constantine's Sword	**2**	**4.75**

The top 10 rated movies (based on the mean rating) had received only a few by exclusively excellent ratings. We should avoid ranking the movies if we have only few ratings. Arbitrarly, we will pick only the movies that have been rated more than 100 times.

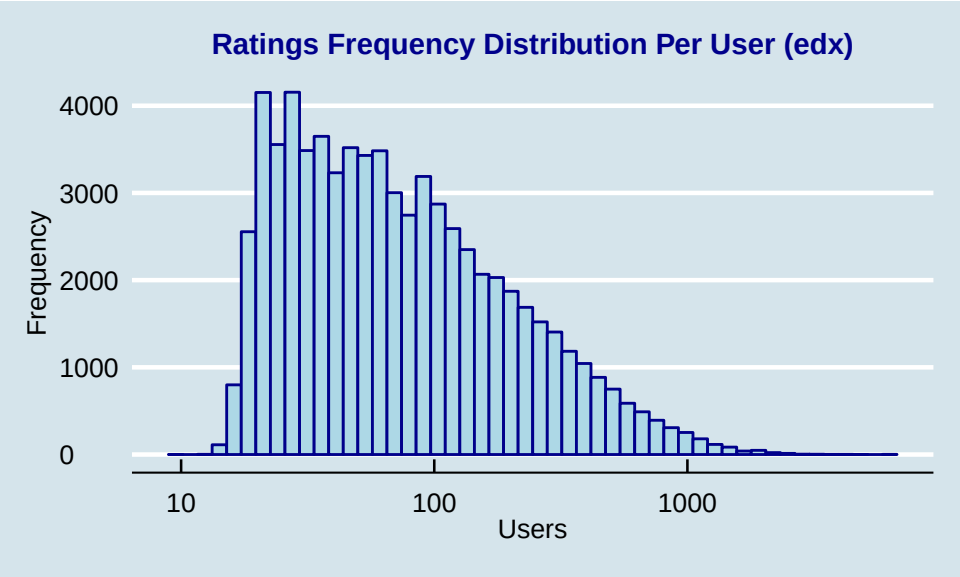
What are the top 10 rated movies with at least 100 ratings?

movieId	title	**Rating_Count**	**Rating_Average**
318	Shawshank Redemption, The	**28015**	**4.45513**
858	Godfather, The	**17747**	**4.41537**
50	Usual Suspects, The	**21648**	**4.36585**
527	Schindler's List	**23193**	**4.36349**
912	Casablanca	**11232**	**4.32042**
904	Rear Window	**7935**	**4.31865**
922	Sunset Blvd. (a.k.a. Sunset Boulevard)	**2922**	**4.31588**
1212	Third Man, The	**2967**	**4.31143**
3435	Double Indemnity	**2154**	**4.31082**
1178	Paths of Glory	**1571**	**4.30872**

The table above shows that the top 10 rated movies are very famous. However, there is a substantial difference in the numbers of ratings that a specific movie receives.

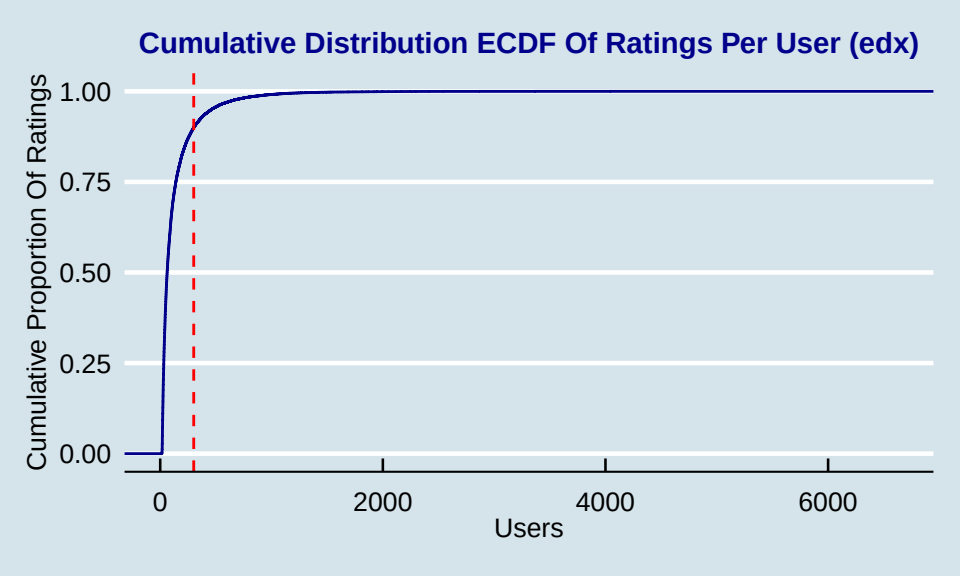
User Analysis

How frequently do the users rate movies?



The figure *"Ratings Frequency Distribution Per User (edx)"* shows that some users rate movies more often than others, so they are much more active.

What is the cumulative rating distribution by user?



The figure *"Cumulative Distribution ECDF Of Ratings Per User (edx)"* shows that only ***301*** distinct users give ***90%*** of the ratings.

What are the top 10 Users/Raters by frequency?

userId	**Rating_Count**	**Rating_Average**
59269	**6616**	**3.26459**
67385	**6360**	**3.19772**
14463	**4648**	**2.40361**
68259	**4036**	**3.57693**
27468	**4023**	**3.82687**
19635	**3771**	**3.49881**
3817	**3733**	**3.11251**
63134	**3371**	**3.26817**
58357	**3361**	**3.00074**
27584	**3142**	**3.00143**

The table *"Top 10 Users/Raters By frequency (edx)"* shows that some user are more active rating the movies than others.

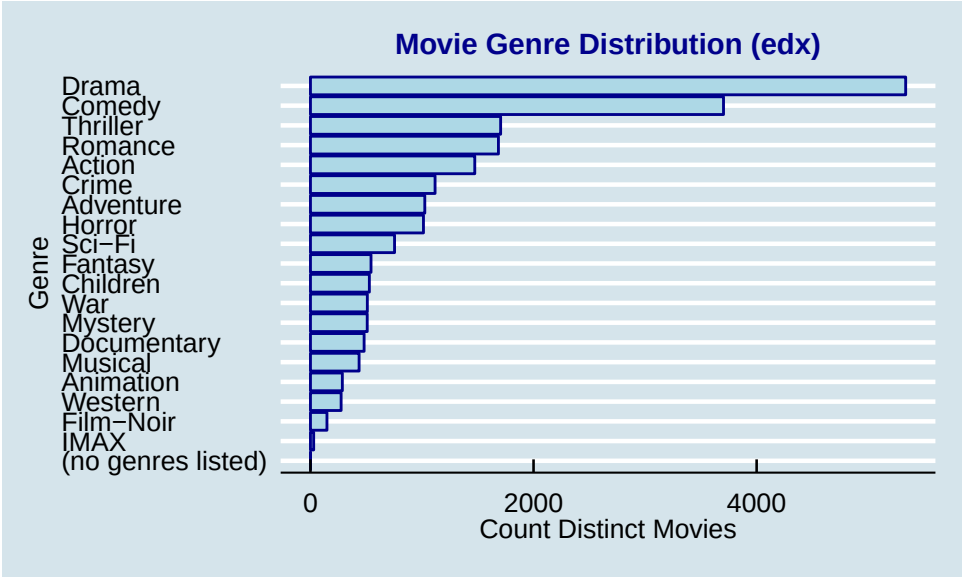
What are the top 10 Users/Raters with at least 100 ratings?

userId	**Rating_Count**	**Rating_Average**
5763	**214**	**4.93458**
59987	**202**	**4.89604**
36896	**149**	**4.89262**
19010	**140**	**4.85000**
16033	**102**	**4.84314**
48518	**130**	**4.83846**
49082	**118**	**4.82203**
20931	**192**	**4.80469**
46262	**307**	**4.79967**
69672	**142**	**4.79577**

The table above shows that the top 10 users / raters are very active. However, there is a substantial difference in the numbers of ratings that a specific user rates.

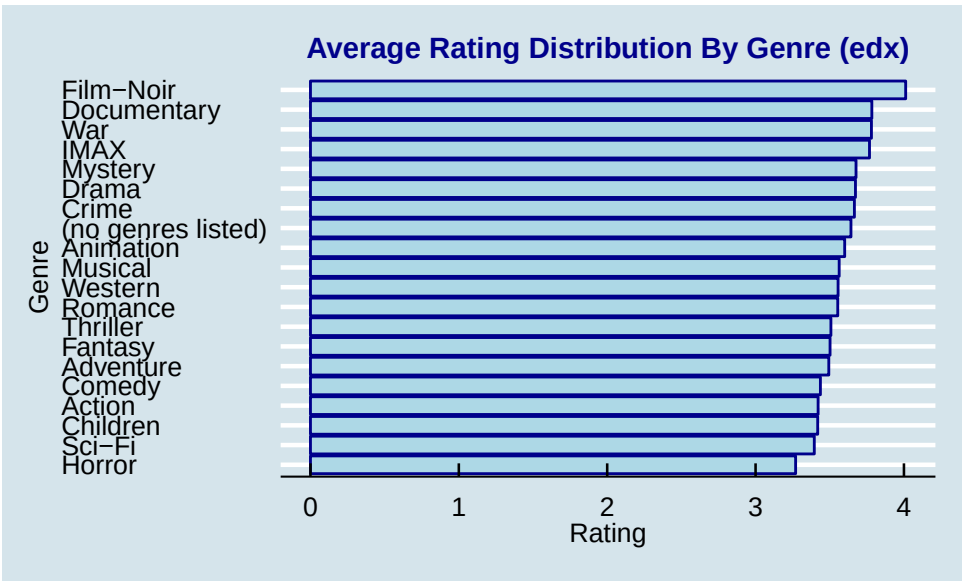
Genre Analysis

What is the genre ratings distribution?



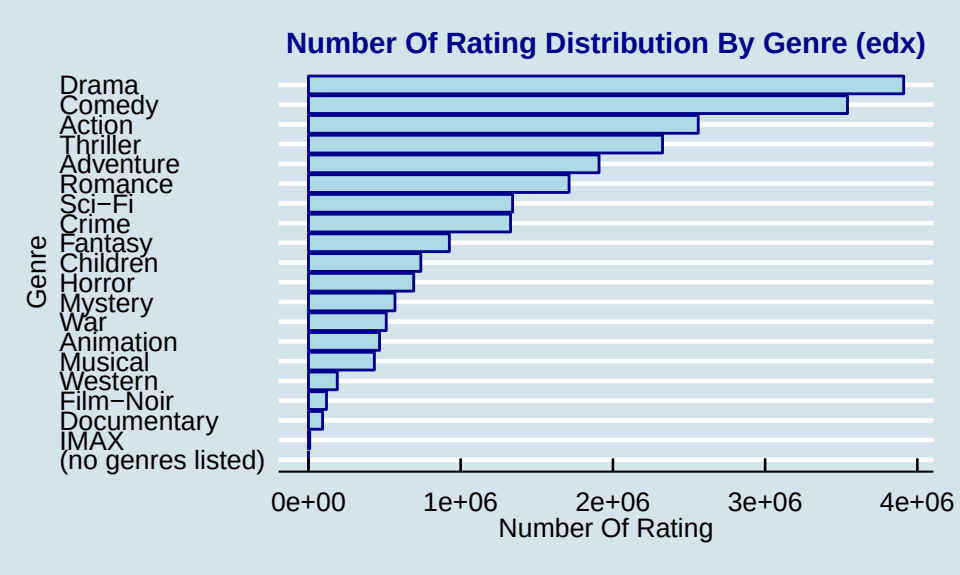
The table above shows that the most common and popular genres are `Drama` and `Comedy`.

What is the average ratings by genre?



The table above shows that the average rating by genre is between ***3.27*** stars and ***4.01*** stars.

What is the number of ratings by genre?



The table above shows that there is a substantial difference in the numbers of ratings that a specific genre receives. Definetely, there is a `genre` effect.

Model Building, Training and Validation

Key steps

After an in-depth Exploratory Data Analysis, we are ready to build, train and test different algorithms and models to reach our goal that provide a minimal reported RMSE (Root Mean Square Error) on the validation dataset with ***`RMSE lower or equal to 0.87750`***.

To build and train the different models, we will proceed, for each model, in five sequential steps:

1. Define and Build the model
2. Train the algorithm in the training dataset `edx`
3. Tune the algorithm using techniques such as Regularisation (optimizing `lambda`)
4. Validate the algorithm by runing the predictions in the validation dataset `validation` and comparing the predicted RMSE against our goal
5. Iterate over the models until goal satisfaction

Based on our Exploratory Data Analysis, we decided to test the following models:

- Model 1: Model-based approach (Naive Baseline)
- Model 2: Content-based approach (Movie Effects)
- Model 3: User-based approach (Movie Effects + User Effects)
- Model 4: Regularized Content-based approach (Movie Effects + Regularisation)
- Model 5: Regularized User-based approach (Movie Effects + User Effects + Regularisation)

Model 1 - Model-based approach (Naive Baseline)

This model is the simplest possible recommendation algorithm. The model predicts the same rating for all movies regardless of other features.

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

With $\varepsilon_{i,u}$ 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.

Model_Id	Model_Method	**Predicted_RMSE**
Model 1	Model-based approach (Naive Baseline)	**1.0612**

The predicted RMSE on the `validation` dataset for the ***Model-based approach (Naive Baseline)*** is about ***1.0612*** which is far from our goal ***0.8775***. We definetly can perform better prediction.

Model 2 - Content-based approach (Movie Effects)

We know from experience that some movies are just generally rated higher than others. The Exploratory Data Analysis performed on the movies confirm that different movies are rated differently. We can add a new term b_i (item / movie feature) to our previous model 1 to represent average ranking for movie i

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

With b_i the bias effect for the movie i.

Model_Id	Model_Method	**Predicted_RMSE**
Model 2	Content-based approach (Movie Effects)	**0.94391**

The predicted RMSE on the `validation` dataset for the ***Content-based approach (Movie Effects)*** is about ***0.94391*** which is better than the ***Model 1***. We still do not reach our goal ***0.8775***. We can perform better prediction.

Model 3 - User-based approach (Movie Effects + User Effects)

From our Exploratory Data Analysis performed on the users, we know that there is a substantial variability across the users. This implies that we can improve our previous model. We can add a new term b_u (user feature) to our previous model 2 to represent the average ranking for users u

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

With b_u the bias effect for the user u.

Model_Id	Model_Method	**Predicted_RMSE**
Model 3	User-based approach (Movie Effects + User Effects)	**0.86535**

The predicted RMSE on the `validation` dataset for the ***User-based approach (Movie Effects + User Effects)*** is about ***0.86535*** which is better than the ***Model 2***. Now, we do reach our goal ***0.8775***. But, could we tune the previous model to perform better prediction?

Model 4 - Regularized Content-based approach (Movie Effects + Regularisation)

We have seen in the EDA section that the supposed "best" and "worst" movies were rated by very few users, in most cases just 1. This is because with just a few users, we have more uncertainty. Therefore, larger estimates of b_i, negative or positive, are more likely.

These are noisy estimates that we should not trust, especially when it comes to prediction. Large errors can increase our RMSE, so we would rather be conservative when unsure.

Regularization permits us to penalize large estimates that are formed using small sample sizes. It has commonalities with the Bayesian approach that shrunk predictions. The general idea of penalized regression is to control the total variability of the movie effects. Specifically, instead of minimizing the least square equation, we minimize an equation that adds a penalty:

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

The first term is just least squares and the second is a penalty that gets larger when many b_i are large. Using calculus we can actually show that the values of b_i that minimize this equation are:

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

where n_i is the number of ratings made for movie i. This approach will have our desired effect: when our sample size n_i is very large, a case which will give us a stable estimate, then the penalty λ is effectively ignored since $n_i+\lambda\approx n_i$. However, when the n_i is small, then the estimate $\hat{b}_i(\lambda)$ is shrunken towards 0. The larger λ, the more we shrink.

So, let's apply the regularisation on the ***Content-based approach (Movie Effects)***.

Model_Id	Model_Method	**Predicted_RMSE**
Model 4	Regularized Content-based approach (Movie Effects + Regularisation)	**0.94385**

The predicted RMSE on the `validation` dataset for the ***Regularized Content-based approach (Movie Effects + Regularisation)*** is about ***0.94385*** which is better than the ***Model 2*** after applying the regularisation using λ equal to ***2.4*** that minimize the RMSE. We still do not reach our goal ***0.8775***. We can perform better prediction.

Model 5 - Regularized User-based approach (Movie Effects + User Effects + Regularisation)

Let's apply the same regularisation technique on the ***User-based approach (Movie Effects + User Effects)***.

Model_Id	Model_Method	**Predicted_RMSE**
Model 5	Regularized User-based approach (Movie Effects + User Effects + Regularisation)	**0.86482**

The predicted RMSE on the `validation` dataset for the ***Regularized User-based approach (Movie Effects + User Effects + Regularisation)*** is about ***0.86482*** which is better than the ***Model 3*** after applying the regularisation using λ equal to ***5.2*** that minimize the RMSE.

Now, we do reach our goal ***0.8775***.

3. Results

Here is the summary of the RMSEs After building, training and validating different models on the `validation` dataset:

Model_Id	Model_Method	**Predicted_RMSE**
Model 1	Model-based approach (Naive Baseline)	**1.06120**
Model 2	Content-based approach (Movie Effects)	**0.94391**
Model 3	User-based approach (Movie Effects + User Effects)	**0.86535**
Model 4	Regularized Content-based approach (Movie Effects + Regularisation)	**0.94385**
Model 5	Regularized User-based approach (Movie Effects + User Effects + Regularisation)	**0.86482**

We can observe that a better RMSE is obtained from ***Regularized User-based approach (Movie Effects + User Effects + Regularisation)***. We can conclude that this is our definitive model as it is achieving our goal.

We achieved an **RMSE = 0.86482** which is more than the expected goal of ***Target RMSE = 0.8775***.

4. Conclusion

In this project, we have used an iterative applied machine learning approach to implement a movie recommendation system to predict a movie rating by a given user. We have build, train and validate naive approach, movie effects, user effects and tuned the models using regularisation technique to predict the movie ratings by users with a targeted RMSE.

As we have seen the Exploratory Data Analysis, the MovieLens dataset contains some other features that we could use as predictors like `genres`, `releaseYear (from title)` and `ratingYear (from timestamp)`.

We could also build and train other models such as KNN, Kmeans, Random Forest, Matrix Factorization, SVD, PCA, Slope One. Some of these algorithms could improve the RMSEs, others could improve the Accuracy.

The most challenging impediment that we might encounter when implementing those additional models and going further in the overall optimizations are both the dataset size and the machine computation resources.

However, using only two predictors `movieId` and `userId`, we wre able to reach our objective.