Building a Movie Rating Recommendation System

By Deepika Sridhar

Table of Contents

[Problem Statement 4](#_Toc35181463)

[Project Flow 4](#_Toc35181464)

[Business Understanding 5](#_Toc35181465)

[Data Understanding 5](#_Toc35181466)

[Data Wrangling 7](#_Toc35181467)

[Missing Values Treatment 7](#_Toc35181468)

[Exploratory Data analysis 9](#_Toc35181469)

[Total movies the individual users have rated 9](#_Toc35181470)

[Mean Ratings 10](#_Toc35181471)

[Machine Learning Analysis 12](#_Toc35181472)

[Steps 12](#_Toc35181473)

[Data Preparation 12](#_Toc35181474)

[Removing unnecessary features 12](#_Toc35181475)

[Machine Learning Model 12](#_Toc35181476)

[Evaluation 13](#_Toc35181477)

[Cross Validation 13](#_Toc35181478)

[Hyper Parameter Tuning 13](#_Toc35181479)

[Model Accuracy Analysis 14](#_Toc35181480)

[Conclusion 16](#_Toc35181481)

[Recommendation 16](#_Toc35181482)

# Problem Statement

## Project Flow



1. Start with the Business Problem
   1. Low Customer Retention Rate for the entertainment platform
2. Convert the Business Problem into a Data Problem
   1. User has rated movies
3. Solve the Data Problem
   1. Build a collaborative filtering recommendation model
   2. Rate the movies for the user based on user's history of rating previous movie
   3. Rate the movies for the user based on other users who have rated the movies similarly
4. Convert the Data Solution into a Business Solution
   1. User engagement in the platform increases
   2. User retention in the platform increases

## Business Understanding

We need to build a Collaborative Filtering Systems based Recommender Systems that will predict the rating of a movie to the user based on the history of ratings of the user.

Suppose a user logs into an entertainment platform and wants to watch a movie, system need to suggest the movie rating based on the ratings the user has provided for other movies before and what rating other users who have watched similar movies have given to the current movie. The business reasons to do this are many. Few of the reasons are:

* **Relevancy** - The rating for the movie must be relevant for the user, i.e. based on the user's past ratings and thereby interests, if the user has rated horror movies highly and romantic movies poorly in the past, then a similar relation must exist here also.
* **Retaining the customers** - The entertainment platform must put in its best efforts to retain its customers. For example, if a user watches a movie rated highly and if the movie is indeed relevant to the user, the user may come back to the platform to watch more content.

Recommender systems are an important class of machine learning algorithms that offer "relevant" suggestions to users. They can be categorized as either collaborative filtering or a content-based system.

#### Collaborative Filtering Systems

Collaborative filtering methods for recommender systems are methods that are solely based on the past interactions between users and the target movies. Thus, the input to a collaborative filtering system will be all historical data of user interactions with target movies. This data is typically stored in a matrix where the rows are the users, and the columns are the movies.

The core idea behind such systems is that the historical data of the users should be enough to make a prediction. I.e we don’t need anything more than that historical data, no extra push from the user, no presently trending information, etc.

#### Content Based Filtering Systems

Content here refers to the content or attributes of the products you like. So, the idea in content-based filtering is to tag products using certain keywords, understand what the user likes, look up those keywords in the database and recommend different products with the same attributes.

## 

## Data Understanding

The primary data objective is to determine the ratings of the movies that increases user engagement rate. The only way to understand this process further is to digitally connect each user to the next one, piece by piece and begin to understand the relationship between them. The dataset has 2 tables namely ratings and movies. The feature details can be found below:

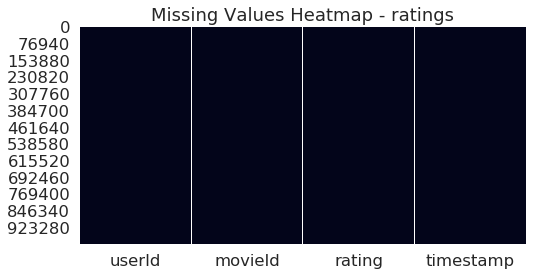


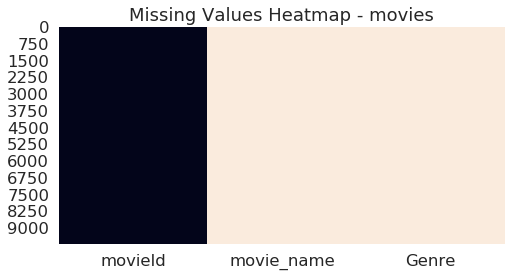


# Data Wrangling

## Missing Values Treatment

The machine learning models cannot work on null data or missing values. Hence we need to check whether there are any missing values and work on them.



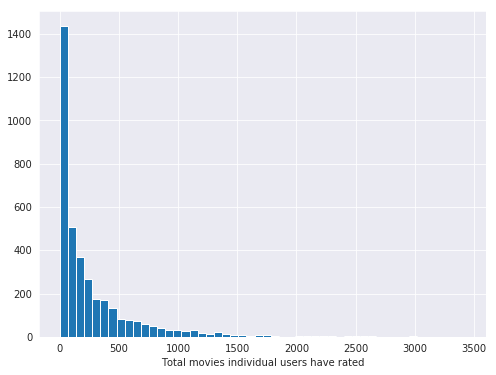


There are no missing values in both users and movies dataframe. And we can hence move on to the Exploratory Data Analysis.

# Exploratory Data analysis

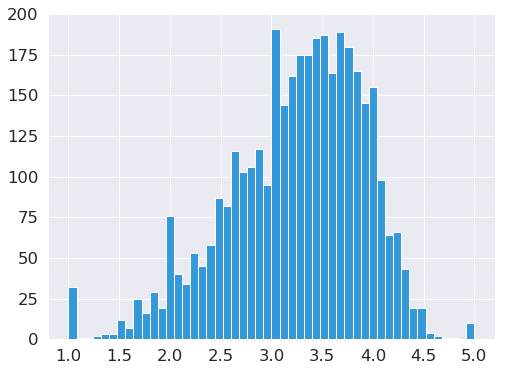
## **Total movies the individual users have rated**

The chart below describes a histogram of the total movies that have been rated by the individual users. As you can see from the above chart, more than 25% of the users have rated less than 100 movies. And very few users have rated more than 1000 movies.

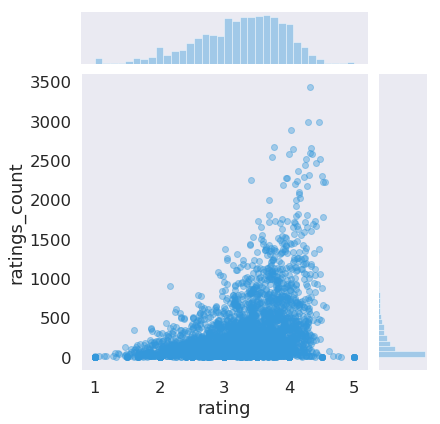


## **Mean Ratings Distribution**

As you can see from the plot, around 50% movies have been rated in the range 3-4. Very few movies are rated at around 5.



As you can see from the below plot, movies that got a rating of 1 or 5 were rated by very few people. This makes sense because some individuals may have extreme views of some movies not many people have watched/rated.



# Machine Learning Analysis

## Steps

1. Data Prep
   1. Train Test split
   2. Removing unnecessary features
2. Model
   1. SVD
   2. Cross Validation
   3. Hype Parameter Tuning
3. Model Accuracy Analysis
   1. Fraction of Concordant Pair
   2. Mean Absolute Error
   3. Mean Squared Error
   4. Root Mean Squared Error

## Data Preparation

### Splitting the Data into training and test set

We now split our data into Training and Test set in a ratio of 70/30. We make use of the pre-existing function called ***train\_test\_split*** to accomplish this. The training set contains the known output in this case, ‘Survived’ column and the model learns on this data to be generalized on other data later on. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset.

### Removing unnecessary features

Timestamp is not very useful for our model and hence need to be removed from our model

## Machine Learning Model

### SVD – Singular Value Decomposition

The Prediction algorithm is given as follows:

𝑟̂ 𝑢𝑖=𝜇+𝑏𝑢+𝑏𝑖+𝑞𝑇𝑖𝑝𝑢

Where, rui is the rating for user u, movie i.

#### Evaluation

We use the root mean squared error to predict the accuracy of our SVD model. After training and testing our algorithm for our data, we see that the accuracy of the algorithm is around 87.8%. This result indicates that our model's prediction ability is good.

We can also consider a few more factors and methods to evaluate the model and improve the performance of our model.

### Cross Validation

Cross Validation is performed in order to reduce the bias that may have happened when splitting the data between test set and training set. Cross validation divided the data into a specified number of sets, n (usually a default of 5 sets) and performs training on n-1 sets and uses 1 set as the test data to evaluate the performance of the model. This step is repeated till all the sets are used as test and train data.



According to the results seen above, RMSE score is around 87% which is very close to the results obtained during the model training and testing without cross validation.

### Hyper Parameter Tuning

Hyper Parameter Tuning is the process of choosing a set of optimal hyperparameters for a machine learning algorithm. The same kind of machine learning algorithms can require different constraints or weights to generalize different data patterns. Hence, we need to tune these parameters so that our model can optimally solve the machine learning problem. We use ***GridSearchCV*** to find our hyperparameters.

We used the following values for finding the optimal parameters:



The attribute ***best\_params\_*** gives the optimal parameter values the machine has chosen for us. The optimal parameter values here are:



Using these values in our SVD algorithm, we need to check for the accuracy rate of our model again. The new rmse accuracy score now is 93.04%. As you can see from the above results, after finding the optimal parameters for the model using GridSearchCV, the performance of the model improved 600 basis points, which is very good.

## Model Accuracy Analysis

Prediction accuracy answers the question how well the recommender does at estimating preference

Decision support metrics answers how well the recommender does at finding good things.

Rank accuracy metrics look at how well the recommender estimates relative preference

**Metrics Families:**

1. Fraction of Concordant Pair
2. Mean Absolute Error
3. Mean Squared Error
4. Root Mean Squared Error

### Fraction of Concordant Pair

#### Looks at the fraction of all pairs that it puts in the correct order. That is, in a concordant pair, both elements of one pair are either greater than, equal to, or less than the corresponding elements of the other pair.

The fcp i.e. the fraction of concordant pair value for our model is 72.22%. Hence according to this metrics it puts around 72.2% of the pairs in the correct order.

### Mean Absolute Error

MAE=1|𝑅̂ \* ∑ r𝑢𝑖∈𝑅̂ |𝑟𝑢𝑖−𝑟̂𝑢𝑖|

This gives us the absolute mean error of predicted values and actual values. The mae i.e. the mean absolute error value for our model is 74.49%. Hence according to this metrics the prediction accuracy is around 74.5%.

### Mean Squared Error

This gives us the absolute mean squared error of predicted values and actual values.

MSE=1|𝑅̂ \* ∑ r𝑢𝑖∈𝑅̂ |𝑟𝑢𝑖^2−𝑟̂𝑢𝑖^2|

This gives us the absolute mean squared error of predicted values and actual values. The mse i.e. the mean absolute squared error value for our model is 86.5%. Hence according to this metrics the prediction accuracy is around 86.5%.

### Root Mean Squared Error

This gives us the squared root of absolute mean squared error of predicted values and actual values.

RMSE=(1|𝑅̂ \* ∑ r𝑢𝑖∈𝑅̂ |𝑟𝑢𝑖^2−𝑟̂𝑢𝑖^2|) ^ 0.5

If you observe the formula, this is the square root of the Mean squared error computed above. This is indeed the square root of .865. The accuracy of this model is around 93%.

The below table summarizes all the accuracy scores computed above:

|  |  |
| --- | --- |
| **Metric** | **Accuracy Score** |
| **Fraction of Concordant Pair** | **72.20** |
| **Mean Absolute Error** | **74.5** |
| **Mean Squared Error** | **86.5** |
| **Root Mean Squared Error** | **93.00** |

# Conclusion

* We build a Collaborative Filtering based Recommender System which has a very good accuracy of around 93%. Which will help the business retain customers and increase customer engagement in the entertainment platform.
* There is definitely scope for improvement by taking more information about users and movies.

# Recommendation

When the user uses the imdb database for the first time, it is recommended that the system asks the user to rate a set of movies and this helps to prevent the cold start problem, wherein, we do not have enough information about the user preferences and hence recommendation becomes very difficult.