

Milestone Report

1. Introduction:

The problem I want to solve for the capstone project 1 is the Movie Recommendation system. Because, they are becoming one of the most popular applications of machine learning which has gained importance in recent years.

This project does not have a definitive client. But the analysis performed could be of use to anyone in the field of movie, music rendering sites, e-commerce sites etc.

I am using a 20 million Movielens dataset for this project. This dataset (ml-20m) describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 20000263 ratings and 465564 tag applications across 27278 movies. These data were created by 138493 users between January 09, 1995 and March 31, 2015. This dataset was generated on October 17, 2016.

The purpose of a recommender system is to suggest relevant items to users. To achieve this task, there exist two major categories of methods: content based methods and collaborative filtering methods. I am using these two methods to solve the recommendation problem.


























2. Problem Statement:

For building a recommender system from scratch, we face several different problems. Currently there are a lot of recommender systems based on the user information, so what should we do if the website has not gotten enough users. After that, we will solve the representation of a movie, which is how a system can understand a movie. That is the precondition for comparing similarity between two movies. Movie features such as genre, actor and director is a way that can categorize movies. But for each feature of the movie, there should be different weight for them and each of them plays a different role for recommendation. So we get these questions:

- How to recommend movies when there are no user information.
- What kind of movie features can be used for the recommender system?

- How to calculate the similarity between two movies.
- Is it possible to set weight for each feature?

Collaborative filtering algorithms try to solve the prediction problem. In other words, we are given a matrix of i users and j items. The value in the i th row and the j th column (denoted by r_{ij}) denotes the rating given by user i to item j .

Matrix of i users and j items

Our job is to complete this matrix. In other words, we need to predict all the cells in the matrix that we have no data for. For example, in the preceding diagram, we are asked to predict whether user E will like the music player item. To accomplish this task, some ratings are available. For doing this we are using the Matrix Factorization Technique to predict the ratings, so that to complete this matrix.

3. Description of Dataset:

This section describes the various data cleaning and data wrangling methods applied on the Movielens datasets to make it more suitable for further analysis. The following sections are divided based on the procedures followed:

Cleaning:

The Movielens data are contained in six files, genome-scores.csv, genome-tags.csv, links.csv, movies.csv, ratings.csv and tags.csv. It contains 20000263 ratings and 465564 tag applications across 27278 movies. These data were created by 138493 users between January 09, 1995 and March 31, 2015. This dataset was generated on October 17, 2016.

The dataset had a lot of features which had 0s for values it did not possess. These values were converted to NaN. These NaN values are filled in two ways according to the type of the data

1. Categorical type was filled with the mode values.
2. Numerical type was filled with mean values.

Removing Unnecessary Features:

This process was done in different ways

Number of missing values:

If a particular feature has more than 50% of missing values in it then most of the times that particular feature will not play any significant role in learning the model. But in my dataset I am not at all having a single feature with more than 50 % of missing values. But there are some features with fewer NaNs. Those features are tag, year and tmdbId but I am not removing those features.

In this Movielens dataset I am having almost 12 features. But I am not using all those 12 features whichever the features are important to me for solving the recommendation problem, I am using those features only.

Significance:

MovieLens dataset bases its recommendations on input provided by users of the website, such as movie ratings. For each user, MovieLens predicts how the user will rate any given movie on the website. Based on these predicted ratings, the system recommends movies that the user is likely to rate highly.

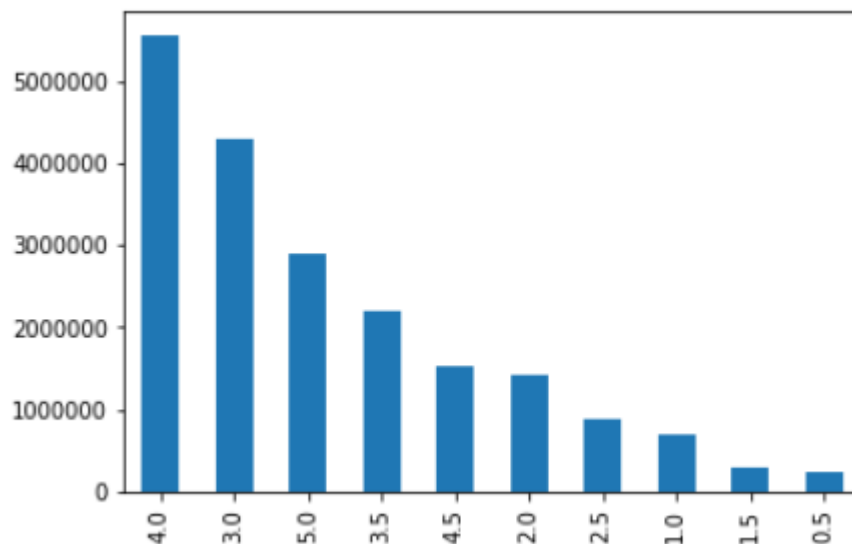
Outliers:

For features like Ratings, whose rating-value is > 5 or negative values, those values are called as Outliers and removing those values will help a lot in predicting the accurate output. And here in this case there are no outliers present in this dataset.

4. Initial findings from exploratory analysis:

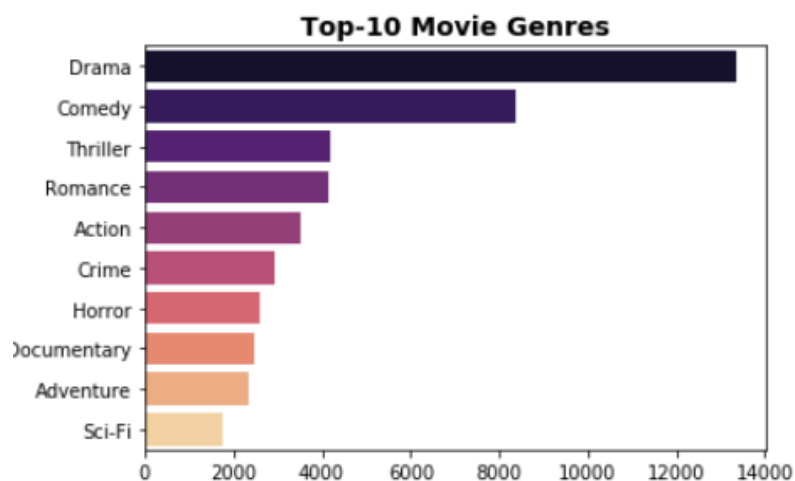
Data storytelling is about communicating your insights effectively, giving your data a voice. Data storytelling is a methodology for communicating information, tailored to a specific audience, with a compelling narrative.

1. Counting the Ratings:



From the above bar graph, we are counting the ratings given by the users for the movies. Most of the users are given rating value as 4. It seems to be good. And the least rating value given by the users are 0.5. From this we conclude that the rating tends to be relatively positive (>3). This may be due to the fact that unhappy customers tend to just leave instead of making efforts to rate. We can keep this in mind - low rating movies mean they are generally really bad

2. Top 10 Movie Genres:



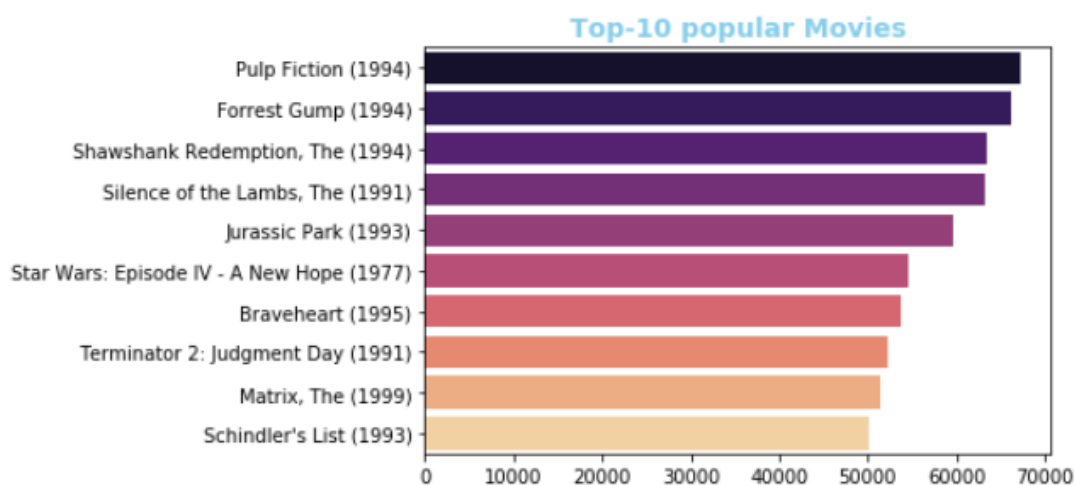
Genre column has various genres associated with that movie but we need to process on each genre separately and hence we will split the genre with the '|' operator and then use explode so that every distinct genre will be in their own row. This will ignore null or empty genres if present in the dataset and if for some reason you want them to persist then you can go for explode outer as it will also store null or empty values.

We have to group the movies by their genre and counted the number of rows to know how many movies are present in different genres. As we can see here drama won the race by some distance.

Drama is the most commonly occurring genre with almost half the movies identifying itself as a drama film. Comedy comes in at a distant second with 25% of the movies having adequate doses of humour. Other major genres represented in the top 10 are Thriller, Romance, Action, Crime, Horror, Documentary, Adventure and Sci-Fi.

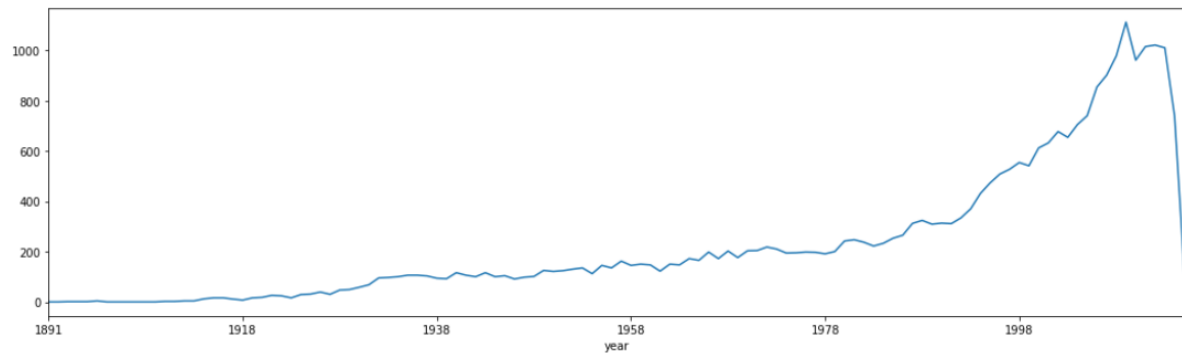
We will consider only those themes that appear in the top 10 most popular genres.

3. Top 10 Popular Movies:



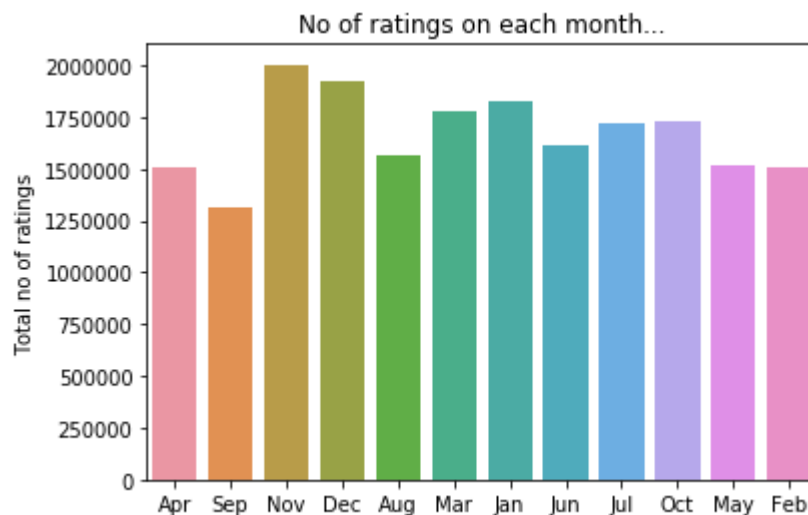
There are almost 27278 movies available with us. From which these are the top 10 popular movies we shown in the above graph. And we notice that Pulp Fiction movie which was released in the year 1994, this is the most popular movie among all the movies. That means this movie is watched by the most number of the users, and then followed by Forrest Gump, Shawshank Redemption etc. From this graph, we conclude that these are the top 10 movies watched by the users.

4. Number of movies released by the year:



From The Movie lens Dataset there are 27278 movies available with us. From that movies we are looking at the number of movies released by the year. We notice that there is a sharp rise in the number of movies starting the 1990s decade.

5. Number of ratings on each month:



From this we wanted to understand that the no of ratings differ by each month. Here, X-axis will be Month and Y-axis will be Total no of ratings. We'll notice that actually September month is far fewer than the other months. And also we noticed that lots of ratings are given by the November month.

Basic Statistics (#Ratings, #Users, and #Movies):

Total data

```
Total no of ratings : 20000263
Total No of Users   : 138493
Total No of movies  : 26744
```

By seeing the above information I'm just understanding that how many ratings, users and movies are there in my dataset. I've almost 20000263 ratings given by the 138493 users on 26744 movies. And also this gives me very high level overview.

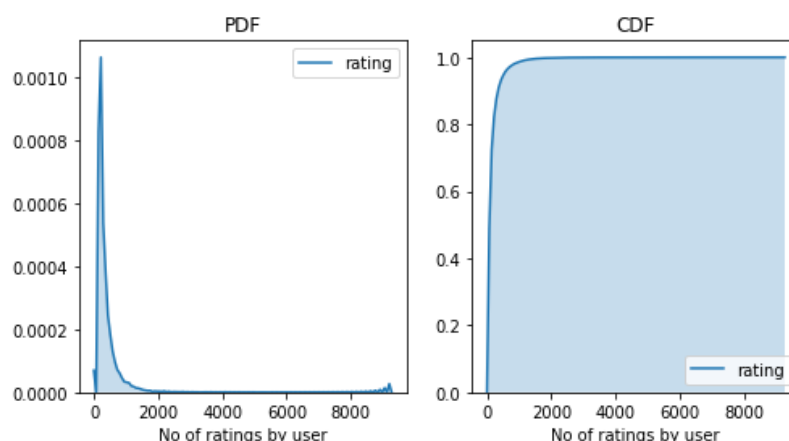
Number of movies rated by a user:

```
: no_of_rated_movies_per_user = ratings_data.groupby(by='userId')['rating'].count().sort_values(ascending=False)
no_of_rated_movies_per_user.head()

: userId
118205    9254
8405      7515
82418     5646
121535    5520
125794    5491
Name: rating, dtype: int64
```

The above data says that the number of movies rated by a given user. If you see the above user Id with 118205 gave 9254 movies that seems to be very large to me for a single user to give. Likewise for every user how many movies he rated with corresponding count we calculated.

Finding PDF & CDF for no of ratings per user:



We plotted CDF & PDF here, we quickly noticed that the peak here tells us that, most of the users rate only a few movies. But there are few users here giving lots of rating. And if you look at the CDF also, almost 90% of people gave very few ratings.

```

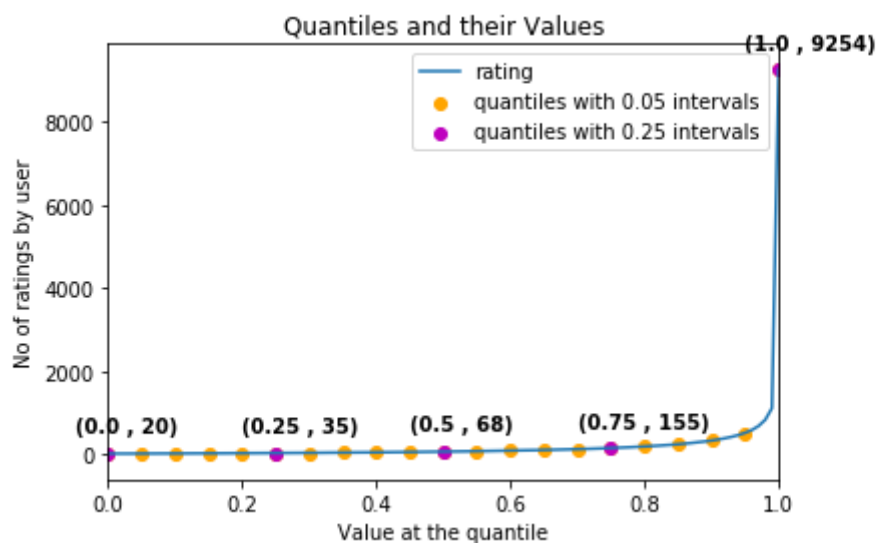
: no_of Rated_movies_per_user.describe()
: count      138493.000000
  mean       144.413530
  std        230.267257
  min        20.000000
  25%        35.000000
  50%        68.000000
  75%       155.000000
  max       9254.000000
  Name: rating, dtype: float64

```

From the above data, we observe that the average number of movies that are rated per user is about 144. This shows that most of the users rated lot of movies. Max number of ratings are 9254 and minimum number of ratings are 20. And if you see the median number of movie rated by a user are 68. That means almost 50% of customers have rated more than 68 movies.

We thought by looking at PDF & CDF, we're not able to get it so well. So we went on understanding about percentiles.

Let's get all the percentile values.



If you notices, what's happening here is, each of the violet circle represents 0.25 intervals. And also we plotted yellow circle it represents 0.05 interval. At 0.25 percentile there are 35 movies rated by the users. And if you closely observe, at 0.95 percentile is also quite low, only 100% percentile is very large.

We actually printed those values.


```
quantiles[::5]
```

```
0.00    20
0.05    21
0.10    24
0.15    27
0.20    30
0.25    35
0.30    39
0.35    45
0.40    51
0.45    59
0.50    68
0.55    79
0.60    93
0.65   108
0.70   127
0.75   155
0.80   193
0.85   246
0.90   334
0.95   520
1.00  9254
Name: rating, dtype: int64
```

At 0.95 percentile also, it is showing 520 movies, there are 5% of users who rated more than 520 movies.

```
print('\n No of ratings at last 5 percentile : {}'.format(sum(no_of Rated_movies_per_user>= 520)) )
```

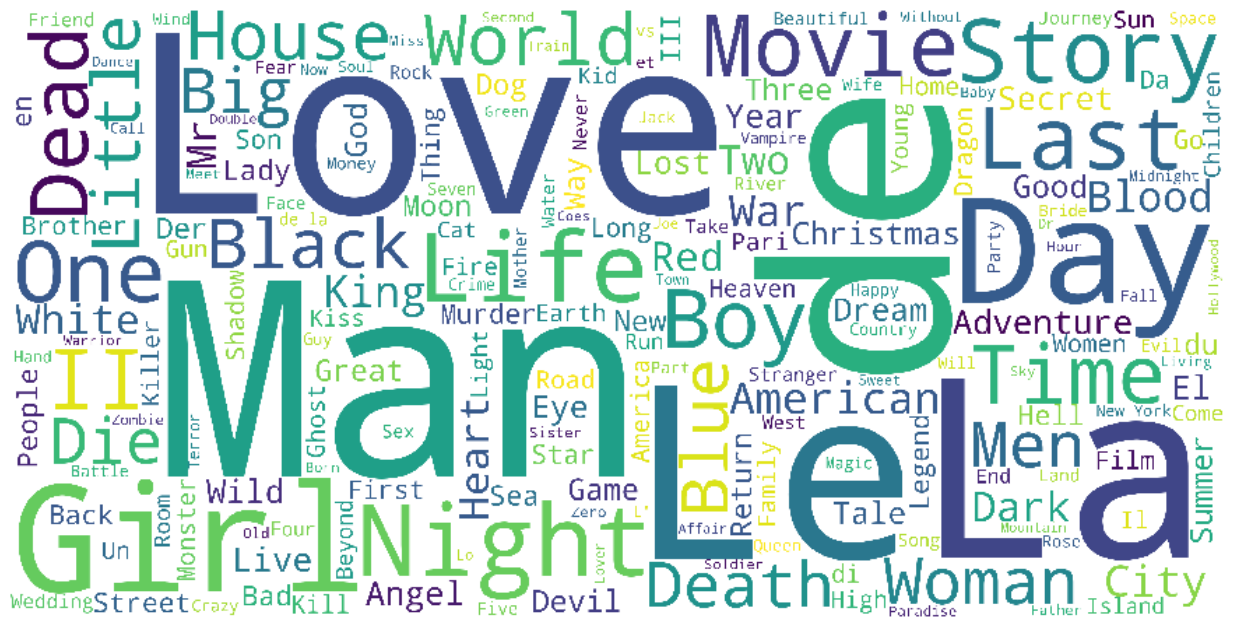
```
No of ratings at last 5 percentile : 6940
```

We also saw that No of ratings at last 5% percentile are 6940 movies. But this gave us good sense on how many ratings that each user has provide.

Analysis of rating of a movie given by a user:

For a given movie let's find the no of users who rated a movie. Because there will be some movies like titanic which are liked by millions of people across the world and hence there will be millions of the rating for a movie like that. But there are some other movies which are liked by very few of them.

Title and overview wordclouds:



Love is the most commonly used word in Movie titles. **Man** and **Girl** are also popular in Movie Blurbs. Together with **Love**, **Man** and **Girl**, these wordclouds give us a pretty good idea of the most popular themes present in movie.