School of Continuing Studies

**University of Toronto**

**Term Project**

**3252 – Big data management systems and tool set**

**Analysis of movie ratings data set using Spark MLlib and building a movie recommendation system**

**Author: Sridhar Mani**

**Date: Nov 30, 2019**

[1. Introduction 3](#_Toc26132464)

[2. Objectives 3](#_Toc26132465)

[3. Data 4](#_Toc26132466)

[3.1. Loading the Data Set 5](#_Toc26132467)

[3.2. About the data 5](#_Toc26132468)

[3.3. Data Transformation 5](#_Toc26132469)

[4. Findings from detailed Data Analysis 6](#_Toc26132470)

[5. Building a Recommender System 11](#_Toc26132471)

[6. Findings from testing the recommender system 11](#_Toc26132472)

[7. Conclusion 12](#_Toc26132473)

[8. References 12](#_Toc26132474)

[9. Appendices 13](#_Toc26132475)

[9.1. Appendix A – Sample of Movie and Rating dataset 13](#_Toc26132476)

# Introduction

A recommender system or a recommendation system is a subclass of [information filtering system](https://en.wikipedia.org/wiki/Information_filtering_system) that seeks to predict the "rating" or "preference" a user would give to an item. Recommender systems are utilized in a variety of areas, and are most commonly recognized as playlist generators for video and music services like Netflix, YouTube and Spotify, product recommenders for services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and [online dating](https://en.wikipedia.org/wiki/Online_dating). There are two main approaches to recommendation systems:

* Collaborative filtering – produces recommendations based on user’s past behavior or based on decisions made by similar users
* Content-based filtering – focuses on attributes of an item in order to recommend items having similar properties

In general, Collaborative filtering (CF) is more commonly used than content-based systems because it usually gives better results and is relatively easy to understand (from an overall implementation perspective). spark.ml currently supports model-based collaborative filtering, in which users and products are described by a small set of latent factors that can be used to predict missing entries. spark.ml uses the alternating least squares (ALS) algorithm to

learn these latent factors. ALS is a Matrix Factorization approach to implement recommendation algorithm you decompose your large user/item matrix into lower dimensional user factors and item factors.

# Objectives

The objective of this project is two-fold:

1. Download and analyse the movie ratings data set using Spark libraries
2. Build a simple implementation of movie recommendation using Spark MLlib

The movie ratings data set is analysed in an attempt to answer the following questions.

* What are the properties of data?
  + 1. Number of distinct users who rated
    2. Number of distinct movies rated
    3. Number of distinct users, movies rated
    4. Yearly split of number of movies rated by users
* What are the movies that are reviewed the greatest number of times?
* What are the movies that are reviewed least number of times?
* What are the top 10 highest rated movies with a minimum of 1000 reviews?
* What are the top 10 lowest rated movies with a minimum of 1000 reviews?
* What are the genres of top rated / worst rated movies?
* How is the trend in rating movies from 1995 to 2015?

# Data

Source of the data is the Group Lens – movie review data set. This data set has information associated with 138493 users rating 26744 movies over a period of ~10 years between Jan 09, 1995 to Mar 31, 2015 adding to a total of 20000263 ratings. The data set is scattered across 6 CSV files as below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File name | File type | # of rows | Info about | Size |
| genome\_scores | CSV | 11709768 | movie-tag relevance data:   * **movieId** * **tagId** * **relevance** | 214.3 MB |
| genome\_tags | CSV | 1128 | tag descriptions:   * **tagId** * **tag** | 20 KB |
| link | CSV | 27278 | identifiers that can be used to link to other sources:   * **movieId** * **imdbId** * **tmbdId** | 539 KB |
| movie | CSV | 27278 | movie information:   * **movieId** * **title** * **genres** | 1.5 MB |
| rating | CSV | 20000263 | ratings of movies by users:   * **userId** * **movieId** * **rating** * **timestamp** | 690.4 MB |
| tag | CSV | 465564 | tags applied to movies by users:   * **userId** * **movieId** * **tag** * **timestamp** | 21.7 MB |

## Loading the Data Set

Data was loaded into Databricks notebook for analysis and model building. Data from CSV was loaded into a Spark data frame object using spark.read function.

## About the data

The loaded data was analysed for completeness and integrity. The two key datasets required for data analysis and model building are movie and rating. There were no nulls in both the data sources. Rest of the following data sets were analysed and deemed not required for data analysis and model building:

* genome\_scores
* genome\_tags
* link
* rating
* tag

## Data Transformation

For ease of data analysis, movie and rating datasets are joined using the key “movieID” under the data frame “movieAndRating”

val movieAndRating = rating.join(movie, Seq("movieID"))

For building the recommendation system, following are the key attributes – userId, movieId, rating. As these attributes are already numerical, there is no need for transforming these attributes

# Findings from detailed Data Analysis

The data was analysed using Spark libraries and functions for significant data characters. Some interesting facts were uncovered. Listed below are some of the key findings.

* Number of distinct users who rated – 138493
* Number of distinct movies rated – 26744
* Total number of ratings - 20000263
* View of total number of ratings grouped by rating band – 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5



Fig(a) - total number of ratings grouped by rating band

The above analysis indicates most number of movies were rated 3 (above average) while least number of movies were rated 1.5 (below average)

* Yearly split of number of movies rated by users



Fig(b) - Yearly split of number of movies rated by users

The above analysis indicates year 2000 had most number of reviews and year 1998 had less number of reviews.

* What are the top 20 movies that are reviewed the greatest number of times?

|  |  |
| --- | --- |
| Title | review count |
| Pulp Fiction (1994) | 67310 |
| Forrest Gump (1994) | 66172 |
| Shawshank Redemption, The (1994) | 63366 |
| Silence of the Lambs, The (1991) | 63299 |
| Jurassic Park (1993) | 59715 |
| Star Wars: Episode IV - A New Hope (1977) | 54502 |
| Braveheart (1995) | 53769 |
| Terminator 2: Judgment Day (1991) | 52244 |
| Matrix, The (1999) | 51334 |
| Schindler's List (1993) | 50054 |
| Toy Story (1995) | 49695 |
| Fugitive, The (1993) | 49581 |
| Apollo 13 (1995) | 47777 |
| Independence Day (a.k.a. ID4) (1996) | 47048 |
| Usual Suspects, The (1995) | 47006 |
| Star Wars: Episode VI - Return of the Jedi (1983) | 46839 |
| Batman (1989) | 46054 |
| Star Wars: Episode V - The Empire Strikes Back (1980) | 45313 |
| American Beauty (1999) | 44987 |
| Twelve Monkeys (a.k.a. 12 Monkeys) (1995) | 44980 |

Table(a) - top 20 movies that are reviewed the greatest number of times

Out of the ~20 million ratings datasets analysed, there were at least 5 movies which had ~60,000 ratings. Pulp fiction which is a famous movie, topped the table with 67,310 ratings followed by Forest Gump with 66172 ratings and Shawshank redemption with 63366 ratings.

* What are the movies that are reviewed least number of times?

|  |  |
| --- | --- |
| Title | review count |
| The Gang That Sold America (1979) | 1 |
| Marilena de la P7 (2006) | 1 |
| Small Town of Anara (Qalaqi Anara) (1978) | 1 |
| Mondo Trasho (1969) | 1 |
| Kidnapping of the President, The (1980) | 1 |
| Destruction Force (1977) | 1 |
| Isi & Disi: Amor a lo bestia (2004) | 1 |
| Lovers and Lollipops (1956) | 1 |
| T.N.T. (1997) | 1 |
| Wicked Blood (2014) | 1 |
| Bakhita (2009) | 1 |
| Majority of One, A (1961) | 1 |
| Double, Double, Toil and Trouble (1993) | 1 |
| Beauty and the Boss (1932) | 1 |
| Career (1959) | 1 |
| Higher and Higher (1943) | 1 |
| Otaku (1994) | 1 |
| An Empress and the Warriors (2008) | 1 |
| Genocide (Konch√ª daisens√¥) (1968) | 1 |

Table(b) - top 20 movies that are reviewed the least number of times

Out of the ~20 million ratings datasets analysed, there were many movies which were reviewed only once. This can be mostly attributed to less popularity of the movie.

* What are the top 10 highest rated movies with a minimum of 1000 reviews?

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **genres** | **avg\_rating** | **count** |
| Shawshank Redemption, The (1994) | Crime|Drama | 4.4469905 | 63366 |
| Godfather, The (1972) | Crime|Drama | 4.364732197 | 41355 |
| Usual Suspects, The (1995) | Crime|Mystery|Thriller | 4.334372208 | 47006 |
| Schindler's List (1993) | Drama|War | 4.310175011 | 50054 |
| Godfather: Part II, The (1974) | Crime|Drama | 4.275640558 | 27398 |
| Seven Samurai (Shichinin no samurai) (1954) | Action|Adventure|Drama | 4.274179657 | 11611 |
| Rear Window (1954) | Mystery|Thriller | 4.271333601 | 17449 |
| Band of Brothers (2001) | Action|Drama|War | 4.263182346 | 4305 |
| Casablanca (1942) | Drama|Romance | 4.258326831 | 24349 |
| Sunset Blvd. (a.k.a. Sunset Boulevard) (1950) | Drama|Film-Noir|Romance | 4.256934866 | 6525 |

Table(c) - top 10 highest rated movie

Out of all movies rated at least a minimum of 1000 times, Shawshank redemption had the highest average rating of 4.4. The minimum criteria were included to avoid any biases.

* What are the top 10 lowest rated movies with a minimum of 1000 reviews?

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **genres** | **avg\_rating** | **count** |
| Battlefield Earth (2000) | Action|Sci-Fi | 1.60055374 | 3973 |
| Baby Geniuses (1999) | Comedy | 1.70300214 | 1399 |
| Problem Child 2 (1991) | Comedy | 1.76 | 1125 |
| Friday the 13th Part VIII: Jason Takes Manhattan (1989) | Horror | 1.76006711 | 1192 |
| Stop! Or My Mom Will Shoot (1992) | Action|Comedy | 1.76839237 | 1835 |
| Spice World (1997) | Comedy | 1.77031603 | 2658 |
| Universal Soldier: The Return (1999) | Action|Sci-Fi | 1.79119548 | 1238 |
| Police Academy 6: City Under Siege (1989) | Comedy|Crime | 1.79489559 | 2155 |
| Dumb and Dumberer: When Harry Met Lloyd (2003) | Comedy | 1.79621849 | 1428 |
| Home Alone 3 (1997) | Children|Comedy | 1.81998517 | 2697 |

Table(d) - top 10 poorly rated movie

Out of all movies rated at least a minimum of 1000 times, Battlefield Earth (2000) had the poorest rating of 1.6. The minimum criteria were included to avoid any biases.

* What are the genres of top rated / worst rated movies?

**Genres of top-rated movies:** Drama, Crime, Mystery, Thriller, War, Action, Adventure, Romance

**Genres of lowest-rated movies:** Comedy, Sci-Fi, Children, Horror, Action

This suggests movies based off of Drama, Crime, Action, Adventure had more popularity among the viewers while movie based off of Comedy, Sci-fi are least popular among the viewers. There is also a possibility that Comedy, Sci-fi & children movies are least popular because of its low-quality content, less production value, less popular star cast etc.

* How is the trend in rating movies from 1995 to 2015?



Fig(c) – Trend in rating movies

The count of users reviewing the movies has been following a trend of spike and decline over the last 20 years. The above trend graph suggests that the count of users reviewing movies and number of movies being reviewed have started declining from 2014 onwards. This could be due to numerous factors. More users may have started posting their comments on social media as opposed to rating movies on dedicated websites.

# Building a Recommender System

In this study, Collaborative based filtering approach is used to build a recommendation system based on ~20 million user rating data. Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. The system generates recommendations using only information about rating profiles for different users or items. By locating peer users/items with a rating history similar to the current user or item, they generate recommendations using this neighborhood. The user- and item-based [nearest neighbor algorithms](https://en.wikipedia.org/wiki/Nearest_neighbour_algorithm) can be combined to deal with the [cold start problem](https://en.wikipedia.org/wiki/Cold_start_(computing)) and improve recommendation results using this data. In this approach, Alternating Least Squares (ALS) a matrix factorization algorithm is used to build the recommender system.

Following steps outline building the recommender system using collaborative filtering approach:

* Import required libraries
  + ml.evaluation.RegressionEvaluator
  + ml.recommendation.ALS
* Load the data set containing following attributes into a data frame
  + User ID
  + Movie ID
  + Rating
* Split the data set into test and train with the following ratio
  + Training – 80%
  + Test – 20%
* Initiate the ALS model with the following parameters:
  + Maximum Iterations (maxIter) – 5
  + Reg Param – 0.01
  + userID as User column
  + movieId as item column
  + rating as rating column
* Fit the model using training data
* Transform the model using test data
* Calculate the accuracy of the model

# Findings from testing the recommender system

The model was tested by comparing the actual ratings of the movie id and user id combinations with that of the prediction. In order to eliminate the skewness due to negative differences between rating values, absolute value of the differences was used. Below is the summary of the rating differences between actual value and prediction value.



Fig(d) – Model results

The result indicates the model performed good as the mean value of the absolute rating differences between predictions and actual were only 0.6.

# Conclusion

Data set having information on ~20 million user ratings of movies was analysed with two key objectives:

1. Develop insights into trends of users rating movies
2. Develop and implement a movie recommender system using ALS algorithm

There were number of key observations that came to light based on the analysis of movie ratings data set such as 1) Rating movies on rating site is on the decline since 2014. This suggests users could be more inclined towards posting their feedback on social media rather than movie rating site 2) Top 10 movies with highest average ratings belong to action, crime and drama genres whereas top 10 movies with lowest average ratings belong to comedy, sci-fi and children genres.

A movie recommender system was developed using collaborative filtering approach based on ALS model. When tested the model with 20% test data set, the results appear to be good as the mean difference between prediction vs actual rating was only 0.6.

# References

* <https://en.wikipedia.org/wiki/Recommender_system>
* <https://spark.apache.org/docs/2.2.0/ml-collaborative-filtering.html>
* <https://www.kaggle.com/grouplens/movielens-20m-dataset>

# Appendices

# Appendix A – Sample of Movie and Rating dataset

