

# **Group AB - Report**

Alex Nesteruk  
Stanislav Cekunov  
Kar-Sing Zak Carr  
Jackil Rajnicant

## **MOVIE RECOMENDATIONS FOR FAVOURITE GENRES**

## INTRODUCTION

As was stated in the specification we were going to produce the recommendations of 3 movies for each top five genre that a user watches. Several changes to the specification had taken place during the development of the system, which were vital for an appropriate extraction of data from datasets (details can be found further). The datasets were downloaded from <http://grouplens.org/datasets/movielens/>, which is the MovieLens 10M Dataset and it contains three datasets, which are: Ratings, Movies, and Tags. For this assignment we used the Hadoop framework and a set of MapReduce programs to process the data. This is because our group was proficient in using java and the datasets weren't that big. Therefore using hadoop was the best approach.

The overall principles of our extractions were:

We would extract the movies that a user had watched and establish the genres of those. Then we would manually sort the results to get the genres of movies from the most movies to least. Then we would extract five star movies excluding the ones that the user had already watched. These would be further reduced to exclude those that had genres not relevant to the user's genre selections. Finally, we would choose 3 movies from that list, which have relevant genres that are in our top 5-genre list and present their name and the genre to the user.

## IMPLEMENTATION

During our development, to get the desired result we were trying to answer 6 questions and get the output answers for them.

### QUESTION ONE

Use a simple MapReduced algorithm to extract from ratings.dat the movies, for user “500”. Note, as the ids of users were encrypted and in implementing needs, this “500” value was hardcoded; to extract a correct id from the first field of that dataset and to get the movies the user had watched. Output was thrown into the folder called Out. Below, there is a screen shot showing some of the output for question 1.

1041	1
1050	1
1059	1
1060	1
1095	1
110	1
111	1
1179	1
1213	1
1245	1
1248	1
125	1
1252	1
1267	1
1268	1
1278	1
1284	1
1344	1
1357	1
1358	1
1393	1
144	1

The output from question 1 shows all the movies that the user has watched and a count of how many times the user has watched it.

## QUESTION TWO

Now we need the genres of those movies. So we check the movie id from the Out folder and check it against the ids in movies.dat to extract the genre of that movie. The result from question 1 was added to the cache in order to read the file in mapper. Results were saved in Out2. The numbers were recorded against the name of the genre. Then the output was manually sorted (as the sorting by the value of numbers of movies, would require an additional complicated MapReduce job). The screen shot below shows the full output for question 2.

Drama	73
Comedy	45
Crime	30
Thriller	26
Romance	22
Film-Noir	11
Action	8
Mystery	7
War	6
Adventure	5
Fantasy	4
Horror	4
Sci-Fi	4
Documentary	2

The output from question 2 shows all genres that the user has watched and it is sorted in the most watched genre by user.

## QUESTION THREE

Here we were trying to answer which movies the user has not watched. Simply checking the id of the movie from movies.dat to the result of Out2 and Out3 (results from question 2 and 3). Here we cached two files, one of them is the result from question 1, which is all the movies the user has watched. Second file is the result from question 2, which contains the genre the user has watched. A simple MapReduce job to get the extracted results and save them to Out3. Below there is a screen shot showing some of the output for question 3.

1000	1
1001	1
1002	1
1006	1
102	1
1020	1
1038	1
104	1
1040	1
1044	1
1051	1
1052	1
1054	1
1055	1
1056	1
1058	1
1061	1
1062	1
1063	1
1082	1
109	1
1091	1
1096	1
1098	1

The output from question 3 shows all the movies id, which are in the top 5 genres for the user that the user hasn't watched.

#### QUESTION 4

In this question we extracted the movie id's from ratings.dat. The selection was also restricted by comparing the field “rating” to a hard coded value of “5”, so that we could get the movies of the 5 star category. The output of this selection will all the 5 star movies. Below, there is a screen shot showing some of the output for question 4.

1	7005
10	1589
100	138
1000	19
1001	1
1002	30
1003	55
1004	64
1005	43
1006	100
1007	48
1008	42
1009	100
101	429
1010	//
1011	35
1012	325
1013	293
1014	118
1015	139
1016	48
1017	224
1018	56
1019	430

The output from question 4 shows that all the movies, which are 5 star rated movies.

#### QUESTION 5

Here we are trying to establish all the movies that the user hasn't watched that are 5 star rated and that they belong to the genres that are present in our user's selection. Note that no datasets were involved in this MapReduce job as we are comparing the output of question 3 (unwatched movies that belong to the users top 5 genres) and the output of question 4 (5 star rated movies). So here we had to cache one file, which is the result from question 4 and for the dataset, the used the result from question 3. The result is the 5 star movies that haven't been watched by the user and that belong to their top 5 genres. Below, there is a screen shot showing some of the output for question 5.

```

1000    19
1001    1
1002    30
1006    100
102     46
1020    182
1038    26
104     1504
1040    26
1044    39
1051    123
1052    12
1054    93
1055    19
1056    71
1058    16
1061    633
1062    7
1063    16
1082    127
1091    104
1096    601
1098    14
1099    312
1103    659

```

The output from question 5 shows all the movies that the user hasn't watched, that are 5 star rated and they belong to the top 5 genres in our user's selection.

### Question 6

This is a slightly odd MapReduce job, as here we are not interested in mapping of the data, but rather interested in reducing the number of output tokens. We take the output of Question 5 and movies.dat to map unwatched movies and arrange them in top 3 movies for each user genre in the Mapper class. Then we emit the key-value to the reducer where we are carrying out the following: We are reading the output of Question 2 to keep track of all the top genres. At the same time we are checking that the number of key-value pairs of each genre is not exceeding three, by loading it into a loop. As the three numbers of each pair is exceeded, we simply discard all the further pairs that belong to that genre and go to the next genres. Here we had to cache two files, which were the results from question 2 and 5 and movies dataset used to get the names of the movies.

The output is 5 sets of three movies of the user's genres, which they have not watched. The screen shot below shows the full output for question 6.

```

Comedy 65133 - Blackadder Back & Forth (1999)
Comedy 64969 - Yes Man (2008)
Comedy 63479 - Sex Drive (2008)
Crime 4945 - Enforcer, The (1976)
Crime 1662 - Gang Related (1997)
Crime 59418 - American Crime, An (2007)
Drama 5496 - Osessione (1943)
Drama 261 - Little Women (1994)
Drama 5494 - Earth Trembles, The (La Terra Trema) (1948)
Romance 4157 - Price of Milk, The (2000)
Romance 3414 - Love Is a Many-Splendored Thing (1955)
Romance 4024 - House of Mirth, The (2000)
Thriller 39427 - Stay (2005)
Thriller 4093 - Cop (1988)
Thriller 3005 - Bone Collector, The (1999)

```

The output from question 6 shows 5 sets of three movies of the user's genres, which they have not

watched. The output shows the genre, movie id and title.

## CHANGES MADE TO THE INITIAL SPECIFICATION

Note, as was stated in the specification we thought to take the average of all the ratings for movies in our ratings.dat. This approach was discarded, as we had noticed that not all the movies of our genres would make it to the output. This is because not all the movies of the top 5 genres may have been rated 5 stars (or not that many people rated them as 5 stars). It was decided that it will be not fair to the user if we don't recommend the movies for all the genres that they like, as this could occur if the movies for one of their top 5 genres may not get many ratings. Therefore, it was decided to use the approach described in Questions 4 to Question 6.