CA4022 Assignment 1

Joseph Oluwasanya

October 13, 2022

1 Data Cleaning

This section contains the data processing steps I undertook using apache Pig on the Movielens-small dataset. **Note:** All the source code for this assignment is located (here). Pig code for Sections 1 and 2 is in the movie_clean.Pig file.

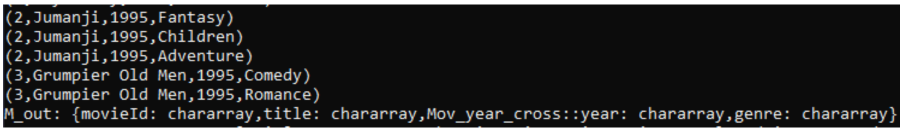
Cleaning the Movie table:

- Remove headers from data files and change data file type to tab delimited text files rather than csv.
- Remove timestamp column from ratings data file. First 2 steps were done in Excel before reading the data into Pig.
- Split the ”genres” field by ”|” characters into a tuple.
- Separated the year from movie title using substring, resulting in a ”year” field.
- Left joined IMDB and TMBD links from the Links table to the movie table (found that this was redundant later, didn’t add links data to Hive).
- Created an n_rating column using the Ratings table, left joined this to the Movie table.

To test for whether the ”year” field had the correct values, any year with length not equal to 4 or null were considered erroneous.

Result: 12 films had no year field, 1 film had a length 9 year field: (171749,Death Note: Desu nôto, 2006â“2007)

Movies have a one-to-many relationship with ratings, so this is not joined here. I changed the delimeters from comma separated values to tab separated values as film titles containing commas were problematic for the preprocessing Pig script. Since movies have a one-to-many relationship with genres, I extracted a separate movie-genre table which has a separate row for each genre of each movie. (Example below)

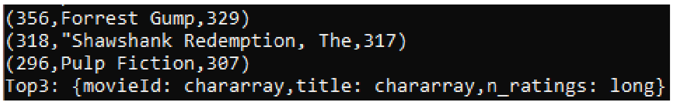


```
(2,Jumanji,1995,Fantasy)
(2,Jumanji,1995,Children)
(2,Jumanji,1995,Adventure)
(3,Grumpier Old Men,1995,Comedy)
(3,Grumpier Old Men,1995,Romance)
M_out: {movieId: chararray,title: chararray,Mov_year_cross::year: chararray,genre: chararray}
```

2 Data Analysis using Pig

1. **What is the title of the movie with the highest number of ratings?**

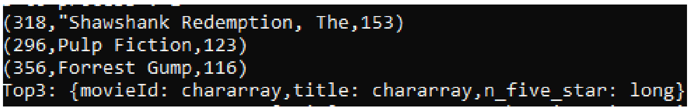
   ”Forrest Gump” is the movie with the highest number of ratings in the dataset with 329 ratings.



```
(356,Forrest Gump,329)
(318,"Shawshank Redemption, The,317)
(296,Pulp Fiction,307)
Top3: {movieId: chararray,title: chararray,n_ratings: long}
```

2. **What is the title of the most liked movie?**

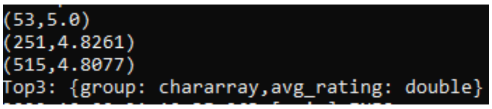
   There are several ways to rank the movies to get the "most liked". Here I interpret the most liked movie as the movie with the most 5 star ratings. The movie with the highest number of 5 star ratings, and hence the most liked is "Shawshank Redemption" with 153 five star ratings.



```
(318,"Shawshank Redemption, The,153)
(296,Pulp Fiction,123)
(356,Forrest Gump,116)
Top3: {movieId: chararray,title: chararray,n_five_star: long}
```

3. **Who is the User with the highest average rating?**

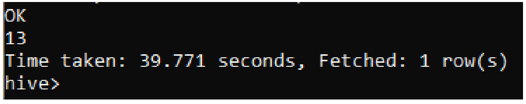
   As the users in the Rating Table are anonymous, the goal is to find the userId belonging to the user with the highest average rating. The resulting userId was userId 53, with a mean rating of 5. Here are the top 3 average ratings given by users to 4 decimal places:



```
(53,5.0)
(251,4.8261)
(515,4.8077)
Top3: {group: chararray,avg_rating: double}
```

3 Data Analysis using Hive

Once I was done cleaning and analysing the data in Pig, I stored the movies, ratings, and movie-genre cross table into tab delimited text files and uploaded this to the HDFS. I then created tables named movies, ratings, and moviesgenre on the Hive default database. A quick check I ran was to see if there are still 13 missing "year" fields when the movies are loaded into Hive. This is verified below:

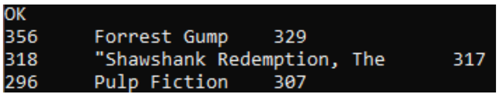


```
OK
13
Time taken: 39.771 seconds, Fetched: 1 row(s)
hive>
```

The first 3 queries are located in queries1.hql file.

1. **What is the title of the movie with the highest number of ratings?**

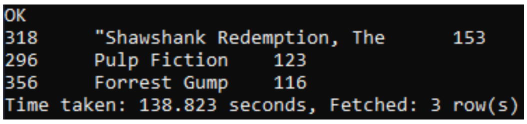
   I used a nested query in Hive to get the top 3 movies with the highest number of ratings.



```
OK
356     Forrest Gump    329
318     "Shawshank Redemption, The      317
296     Pulp Fiction    307
```

2. **What is the title of the most liked movie?**

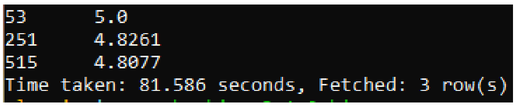
   Here I used the same definition of "most liked" as with pig. That is, the movie with the highest number of five star ratings. The query was similar to the first.



```
OK
318     "Shawshank Redemption, The      153
296     Pulp Fiction    123
356     Forrest Gump    116
Time taken: 138.823 seconds, Fetched: 3 row(s)
```

3. **Who is the user with the highest average rating?**

   The results of these first three queries were consistent with Pig.



```
53      5.0
251     4.8261
515     4.8077
Time taken: 81.586 seconds, Fetched: 3 row(s)
```

Now let's look at a distribution of average ratings by users. Since getting the average of each users' ratings is like getting sample means from a population of ratings, we expect the distribution to be approximately normal. Although the CLT assumptions are not properly met, since the samples are not random (each sample is 1 user's ratings), and the sample sizes can vary from small (1) to large (>30). In the figure below we see that despite these violations, the distribution is still approximately normal. The mean being 3.5 and standard deviation 0.5. The average of these sample means is approximately equal to the population mean.

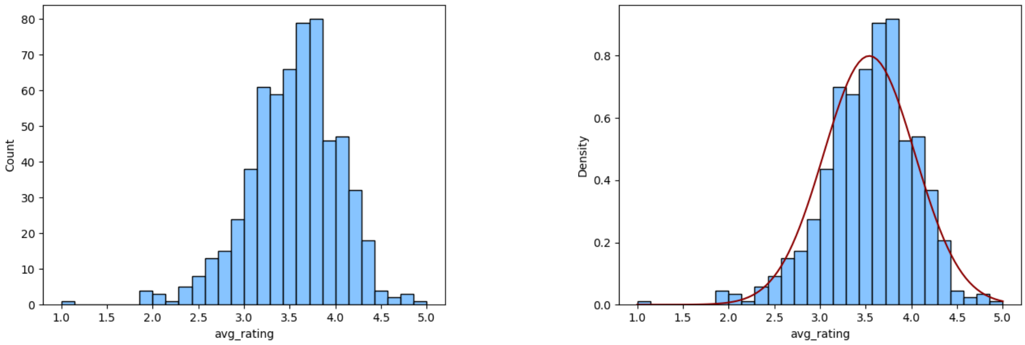
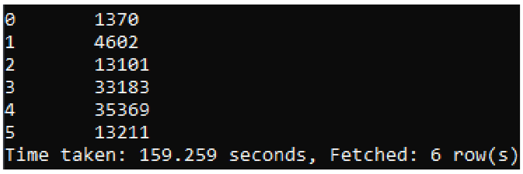


Figure 1: Histograms of average movie ratings by users (1) Count histogram, (2) Density histogram with fitted Normal PDF with parameters $\overline{x} = 3.5394$ and $\overline{s} = 0.4991$.

1. **Count the number of ratings for each star level. What is the most popular rating?**

   The following screenshot shows the number of ratings at each star level. It is the output produced by the getratings.hql script.

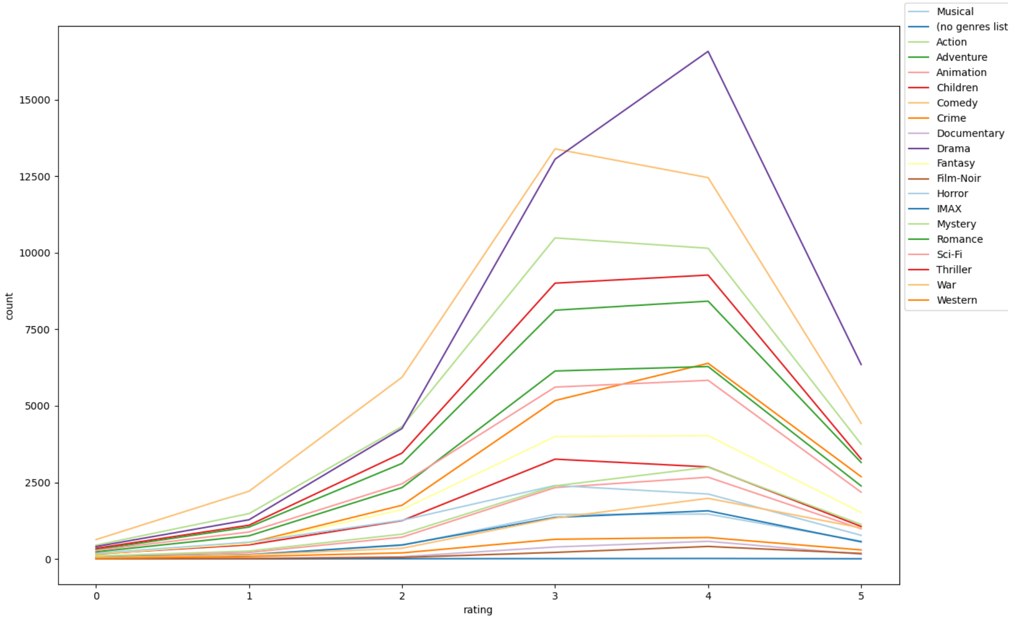


```
0       1370
1       4602
2       13101
3       33183
4       35369
5       13211
Time taken: 159.259 seconds, Fetched: 6 row(s)
```

   4 stars is the most popular rating, with 35,369 ratings at this level.

2. **How are the ratings distributed by genre?**

   For this task I used the movie-genre cross table and grouped by genre and star rating. The script is located in ratingsbygenre.hql.



   From this graph, we can see that the Drama and Comedy genres recieved the most ratings in general, with Drama recieving the most 4-5 ratings. On the lower end, the most popular genre in terms of 1-2 star ratings was comedy. The least popular movie genres in terms of ratings are Film Noir, Documentary, and Western.