SI 618 Project Part 1 Report

Using PySpark to Explore the Movie Datasets

Motivation

People nowadays love watching movies. Whether a movie is a success is always determined by its director, as directors are in charge of many important aspects of the movies. So in this project, I want to focus my analysis on movie directors. I will explore the movie datasets by PySpark to find out which directors made the most average revenues and which directors produced the highest rated movies. We could learn by merging the relevent datasets together and taking a look at the differences between the directors' average revenues and average movie ratings. Besides, I would also like to discuss the trends of female proportions in movie directors for each genre over time.

Data Sources

I used two different datasets which are described below to conduct data analysis.

1. Movie Basic Information Dataset

Source: https://www.kaggle.com/rounakbanik/the-movies-dataset#movies_metadata.csv

This dataset regarding basic information of movies is available on kaggle and can be downloaded as a CSV file. It contains 24 variables and 45,466 records which cover movies released from 1874 to 2017. Considering the completeness of the movie information, I decided to use the 29,373 records of the movies that were released from 1985 to 2015. Each record mainly shows genres, budget, revenue, original language, original title, overview, popularity, production companies, production countries, release date, runtime, release status, vote average and vote count of a movie.

The variables of interest are *genres*, *revenue*, *release_date* and *vote_average*. Each movie may have several genres, such as adventure, animation, comedy and etc. I chose to use the *vote_average* variable which shows the weighted average ratings of movies for average movie rating comparison instead of calculating the average ratings myself. Vote average applies filters to eliminate and reduce attempts at vote stuffing, which makes the ratings more accurate.

2. Cast and Crew Information Dataset

Source: https://www.kaggle.com/rounakbanik/the-movies-dataset#credits.csv

The dataset that consists of the cast and crew information of movies is also available on kaggle and could be downloaded in CSV format. It has 45,476 records and contains 3 variables which are *id*, *cast* and *crew*. Each record shows a movie's cast and crew information. I am interested in the crew information which mainly shows the department, gender, job, name of every person in the crew. Note that gender has three encoded values 0, 1, 2 which correspondingly represents 'not specified', 'female' and 'male'. I would regard the percentage of females in the total count of females and males as female proportion in movie directors.

Data Manipulation

The datasets were not in a clean and nice format and I still needed to do some data manipulations to prepare the dataset for the analysis and visualization part.

The source code can be found in **si618_project1.py**. The comments indicate the location of code for each specific part.

Step 1: Clean the data

Firstly, I kept only the necessary variables that I am interested in for convenience. I simply dropped the incomplete records and removed the movie records with no specified genres, vote average or crew infomation from the datasets, as they only account for a small part of the whole data. Furthermore, I changed all the variables to the right types because the datasets somehow had unreasonable types for some variables such as *id*.

As I mentioned above, I decided to use the records of the movies that were released from 1985 to 2015 due to the consideration for information completeness. I picked out the movies that have *status* "Released" and created a new variable *release_year* based on *release_date* to select the movies that were released in the chosen time period.

Most movies have unnormal 0 *revenue* values which probably represent missing values out of the difficulty in information acquisition. I chose to retain these movie records because of their large amount. When it comes to calculating the average revenue of each director, I would ignore these records and only take the data with positive recorded revenues into consideration.

All the manipulations in step 1 were accomplished by using the basic functions and operations in pandas.

Step 2: Find the director name and gender for each movie

I applied the eval() function and map operation in pandas to create two new variables called *director* and *gender* which show the name and gender of the director of a movie based on the variable *crew* in the Cast and Crew Information Dataset. For future analysis, I dropped those movie records with no director information.

Step 3: Join the two datasets by shared movie id column

I created RDDs for the datasets by sc.parallelize(df.values.tolist()) and formed key-value pairs by map operation. To find new insights for the movie datasets, I joined the two processed datasets by their shared id column using join operation in PySpark. The merged dataset would be used to support the 3 tasks in the next part.

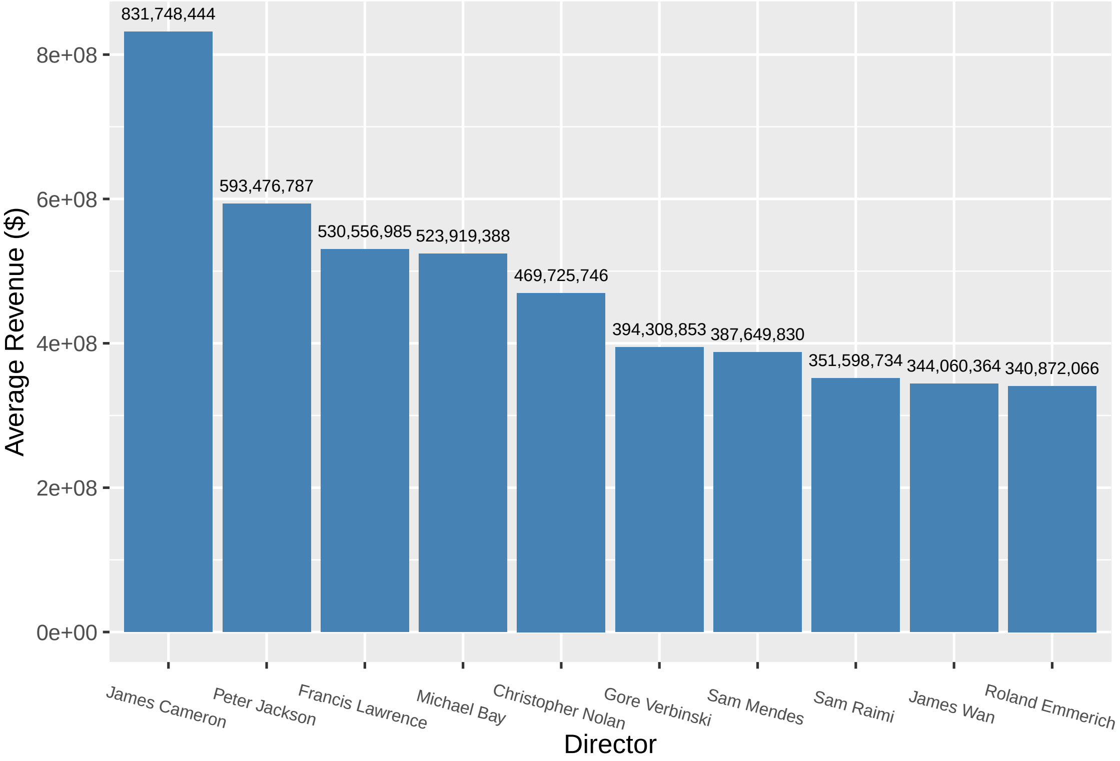

Figure 1: The top 10 average movie revenues of directors.

Analysis and Visualization

The three tasks in this part were accomplished by using large-scale computation techniques PySpark. The source code can be found in the part commented as ‘analysis’ from **si618_project1.py**. The visualization work was done in R using package ggplot2. This part of code is in **si618-project-part-1-xinyej.Rmd**.

Task 1: Which directors made the most average movie revenues?

The first task was to find out which directors made the most average revenue. The code corresponds to the ‘analysis task 1’ part in **si618_project1.py**.

For convenience, I only kept the needed variables and generated (director, (revenue, count)) pairs by map. As I mentioned before, I retained all the missing values recorded as 0 in *revenue*. So I ruled out these records before computing the average revenue of each director by filter operation. Then I used reduceByKey and mapValues operations to compute the desired statistic and applied sortBy operation to sort the results in descending order. Considering the amount of computation, I only reserved the directors which produced more than 5 movies in the time period of 1985 to 2015 by filter.

Figure 1 shows the top 10 average movie revenues and the corresponding directors who directed more than 5 movies from 1985 to 2015. The top 5 directors are James Cameron, Peter Jackson, Francis Lawrence, Michael Bay and Christopher Nolan.

We see that James Cameron, who directed ‘Titanic’ and ‘Avatar’, made the highest average profit about 832 million dollars during the period from 1985 to 2015. The average revenue that he made was around 140% the amount that the second person Peter Jackson made. It was also almost twice as much as the amount that Christopher Nolan who ranked 6th made. This shows that even though many directors are outstanding, there is great difference between the profits that various directors can make. Moreover, the average profit difference between directors tends to decrease when the average revenue declines.

Task 2: Which directors produced the movies that have the highest average ratings?

In the second task, I tried to figure out which directors made movies with the best average ratings. The code can be found in the 'analysis task 2' part from **si618_project1.py**. I used the *vote_average* variable which shows the weighted average ratings of movies for average rating comparison.

I generated (director, (vote average, count)) tuples by map and again used reduceByKey, filter, mapValues and sortBy operations to get the sorted average movie ratings of directors in descending order. The filter operation that was mentioned above removed the directors which produced less than or equal to 5 movies from 1985 to 2015.

The output is shown in Table 1 below.

Table 1: The top 10 average movie ratings of directors.

Director	Average Rating
Don Hertzfeldt	8.067
Rocco Urbisci	7.822
Hayao Miyazaki	7.717
João César Monteiro	7.683
Krzysztof Kieślowski	7.537
Christopher Nolan	7.536
Quentin Tarantino	7.490
Lance Bangs	7.460
Dominic Brigstocke	7.443
Louis C.K.	7.425

From Table 1, we can see the top 10 average movie ratings and the directors. People gave high evaluations to the movies of Don Hertzfeldt, Rocco Urbisci, Hayao Miyazaki and so on. The highest average rating of Don Hertzfeldt was over 8, and all the top 10 average ratings were bigger than 7.4. The rating levels were quite close. It is interesting to see that the top 10 directors in the average revenue rankings and the top 10 directors in the average rating rankings barely overlapped. It may suggest that the movies that most people love to purchase are not necessarily the ones that people think highly of.

Task 3: The trends of female proportions in movie directors for each genre over time

The last task was focused on the trends of female proportions in movie directors for different genres over time. The code can be seen in the 'analysis task 3' part from **si618_project1.py**.

Firstly I needed to reorganize the dataset to produce one line of movie information for each genre by flatMap. The female proportion in a given genre was computed as the proportion of females in the total count of females and males. In order to calculate the statistic, I first formed ((genre, release year), (gender, count)) pairs by flatMap and then utilized reduceByKey and mapValues to get the desired female percentages in movie directors for each genre and each year. The trends of female proportions are shown in Figure 2 and Figure 3 which are both on the next page.

Considering the genres' large amount (20), I plotted the female proportions for the first ten genres in Figure 2 and the ones for the remaining 10 genres in Figure 3. The division did not have special meaning except for

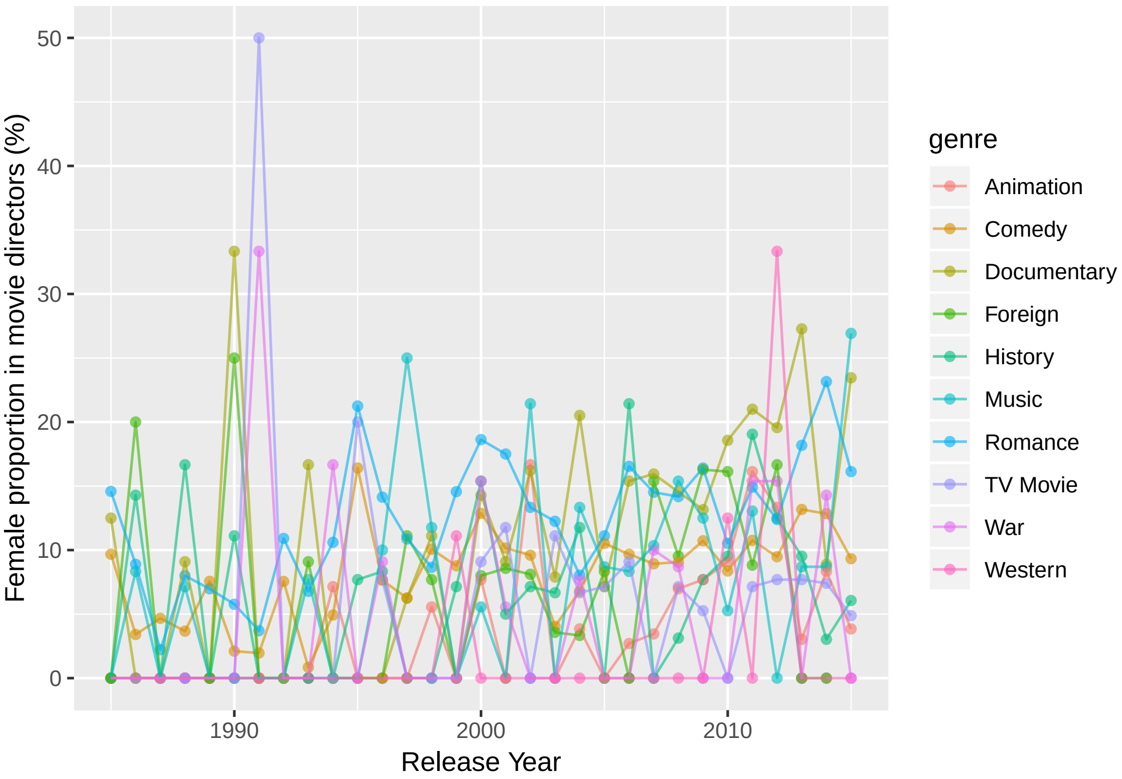

Figure 2: Female proportions in movie directors for each genre (part 1) over time.

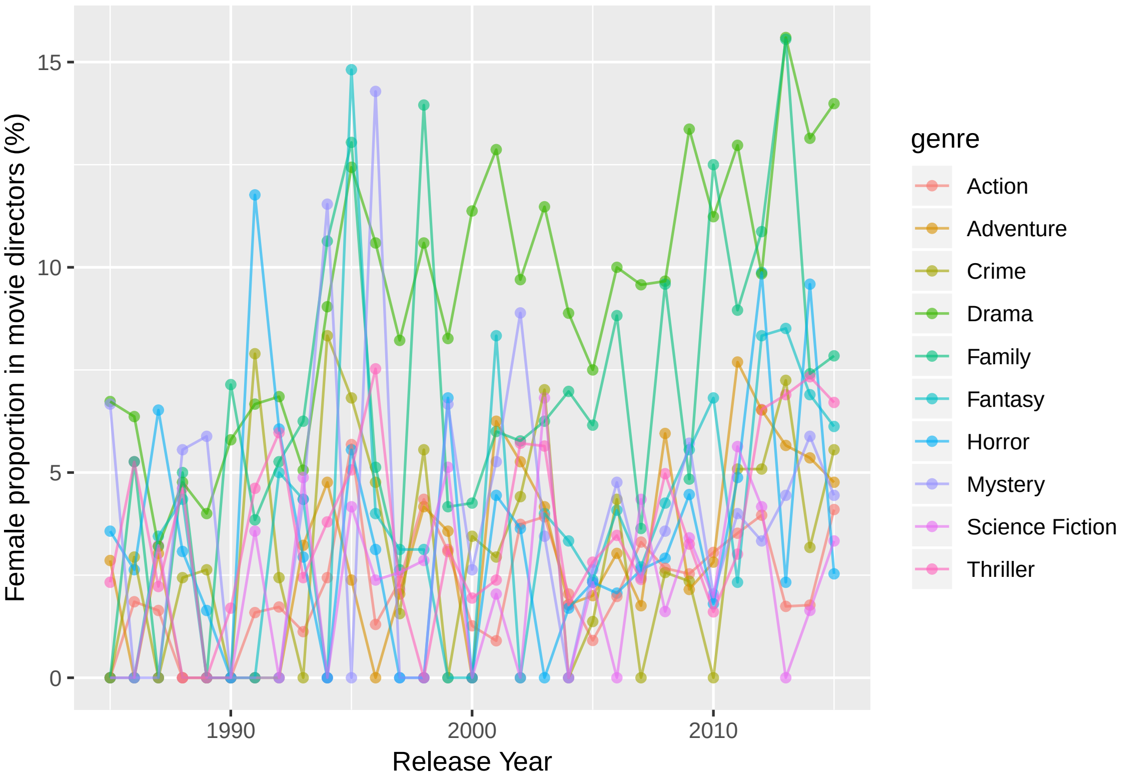

Figure 3: Female proportions in movie directors for each genre (part 2) over time.

better observation. Figure 2 covers genres such as Animation, Comedy, Documentary and so on. Figure 3 covers genres like Action, Adventure, Crime, and etc.

It is easy to see that almost all the female proportions in directors for different genres in each year were below 30%. But the overall female proportions did increase over time. For example, the female proportions for different genres barely passed 10% before 1990, yet quite a few of them were over 10% after 2010. Less genres had 0% female proportions in directors as time went on. We could see that more and more female directors directed movies of Romance, Documentary, Drama, Family, and etc.

Challenges

The biggest challenge was the difficulty in data cleaning. The datasets not only had obvious missing values, but also had missing values recorded as '[]' or 0. Besides, some variables such as id had unreasonable data types when they were read in. It is because some original records somehow mixed a few features. In order to avoid weird bugs in the following parts, I checked each variable of interest extremely carefully and got rid of all the wrong records. The difficulty in data cleaning was also the reason why I first applied pandas to read and process the datasets and then loaded them to RDDs in PySpark by sc.parallelize(df.values.tolist()). I had to investigate the datasets after every single step by interactive computing in jupyter notebook. It was much easier for me to pursue this goal in pandas dataframes than in PySpark RDDs.

The next trouble was caused by the fact that many operations in PySpark such as join and reduceByKey are only defined on Pair RDDs. To detect the problem origin, I logged onto the terminal, entered the interactive PySpark interface and tested my code line by line using the first 100 records of the datasets. After I found out the causes, I solved the problems and ran my code to get the final results.