Netflix Investigation

Valentín Feliu

Query N1: Find the information about the movie with the highest IMDB Score off all times.

Process:

Having a first glance into the data, I first was intrigue to know which one was the movie with the highest IMDB Score from my dataset. To get to this value I made a 4 steps process:

1. First, I checked the Schema to see if aggregations where possible.
2. Second, I changed the type of the column IMDB_Score so that the aggregation function that I wanted to implement was possible.
3. Thirdly, I found the maximum value of the column by applying group by and max functions to the original data frame.
4. Lastly, as there was only one unique highest value for the IMDB_Score column, I decided to apply a filter by the maximum value so that it would return me the information related to that value.

Conclusion:

The movie with the maximum IMDB Score of all times is “David Attenborough: A Life on Our Planet”. This movie is a 83 minute Documentary, launched on the 4th of Octouber, 2020. The main language of the movie is English, and the IMDB Score is equal to 9 points out of 10.

df_IMDB_Score: pyspark.sql.dataframe.DataFrame = [Title: string, Genre: string ... 4 more fields]

Title	Genre	Premiere	Runtime	IMDB_Score	Language
David Attenboroug...	Documentary	October 4, 2020	83	9	English

Query N2: Create a ranking of the Top IMDB Scored movies.

Assumption:

We consider Top IMDB Scored movies those who have a IMDB Score higher or equal to 7/10.

Process:

The ranking was created as a new data frame to make further analysis on the best movies of the market.

To get to the ranking I applied the filter function and filtered the original dataser by IMDB_Score that where >= 7. Latter, I also applied the order by function to the IMDB_Score column, to have the information in descending order acording to the score.

Finally, I checked if the data frame was created properly by applying print(type()) function to the name of the new data frame.

Conclusion:

I used the comand display() to better visualize the query. As a result I got a table with 152 rows and all the original columns of the data frame.

```
1 print(type(df_top_movies))
```

```
<class 'pyspark.sql.dataframe.DataFrame'>
```

Query N3: Understanding the top Genre of the Top IMDB Scored movies.

Process:

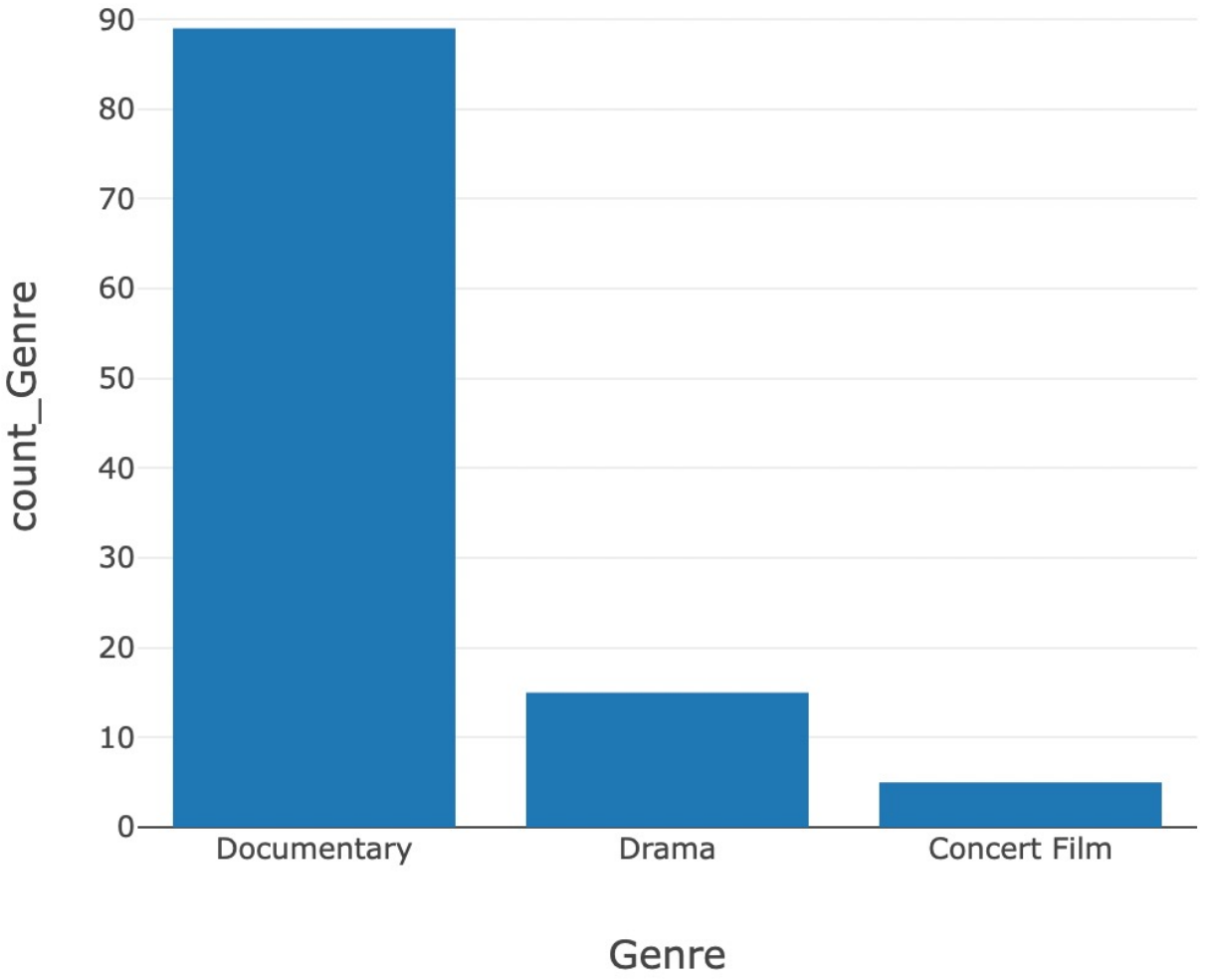
First, I wanted to have an idea of the amount of Genre they where on the 152 top movies selected. Therefore, I imported the function countDistinct from pyspark sql functions and applied it to my previous data frame created. The result was of 31 different Genre in the top movies list.

Secondly, I created a new data frame called df_genre_of_top_movies. This data frame contained two columns, the firstone with the names of the 31 Genres extracted before, and the second one the number of times (count) that the Genre appeared in each column.

Lastly, I ordered the data frame in descending order and limited the number of rows to 3. Therefore, as a result I got the 3 most used Genres of the top movies.

Conclusion:

The Top 3 Genre are Documentary, Drama and Concert Films.



Query N4: Extract the average runtime of the Top IMDB Scored movies

Process:

Now that I know what is the most watched Genre, it might be usefull as well to understand what is the average runtime of the top scored movies. In this way, a person interested in making a movie must know that this are good qualities to get a better IMDB Score.

To extract the average runtime of the top IMDB Dcored Movies I simply applied and aggregation function extracting the average value of the column “Runtime” for the data frame df_top_movies.

Conclusion:

The average runtime of the Top IMDB Scored movies is equal to 89.99 minutes as shown on the picture below.

```
+-----------------+
|     avg(Runtime)|
+-----------------+
|89.99342105263158|
+-----------------+
```

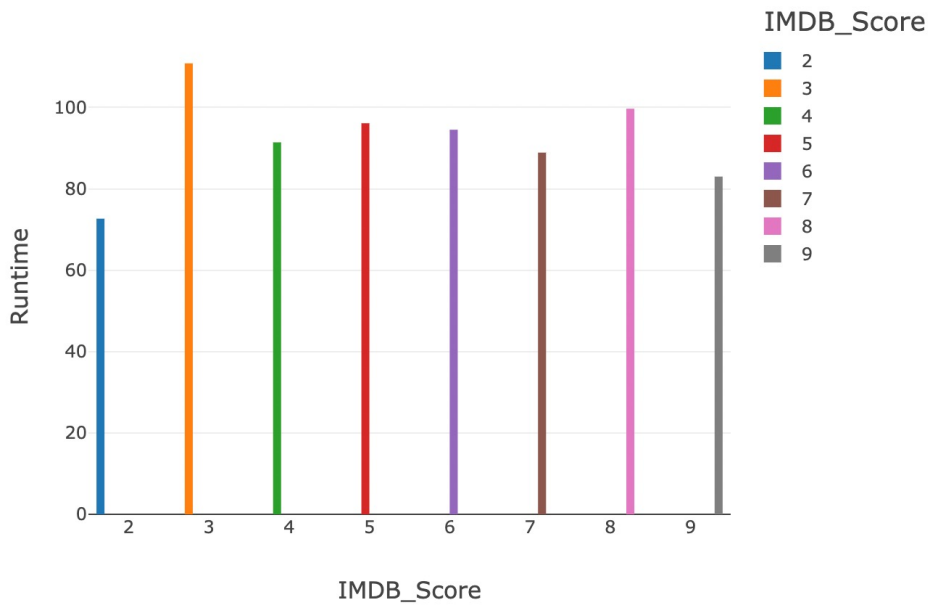
Query N5: Understand if there is a relationship between IMDB Score and Runtime

Process:

I wanted to understand if there was a relationship between the IMDB Score and the Runtime. Therefore, I worked on the original dataset and created a new data frame with three columnes: Title, Runtime and IMDB Score.

Secondly, I decided to change the type of the columnes Runtime and IMDB Score as they were as Strings and not Integers.

Lastly, I displayed the new data frame and created a graph showing the average runtime for each different IMDB Score puntuation.



Conclusion:

As you can see on the graph, the IMDB Score punctuation which has the highest runtime average is 3. Also, as analyzed on Query N4 we can see that the scores 7, 8, 9 are all in between 80 and 100 minutes of runtime.

Further Analysis

While analyzing Query N5, I also figuer out that with the same data frame I could create a graph representing the quantity of titles for each IMDB Score punctuation.

The result was suprisingly interesting as the shape of the grpah plotted is very similar to the graph of a Gaussian function. Just by looking at it, we can identify the punctuation 6 as the expected value of the function.

Morover, we could say that if a new movie is released it will be rated with a puntuaction in between 5 and 7 with a probability of 88%.