Project – Netflix Content Analysis

Trevor Incledon

October 10, 2020

Outline

The goal of this analysis is to review content added to the Netflix platform over time to identify any trends or insightful information by reviewing the historic data. The first step is to explore the data available to me and understand how it is structured. Once I spun up an AWS RDS instance for database processing and connected it to MySQL for analysis, I began diving in.

Data Import Validation

I ran the following queries for basic exploratory analysis on the dataset:

```
--Exploring the dataset-----
SHOW TABLES;

DESCRIBE netflixdata2;

SELECT *
FROM netflixdata2
LIMIT 100;
```

Snapshot of dataset including all columns:

picture_id	picture_type	title	director	cast	country	date_added	release_year	rating	duration	category	picture_description
247747	Movie	Amar Akbar Anthony	Manmohan Desai	Vinod Khanna, Rishi Kapoor, Amitabh Bachchan,...	India	2019-12-31	1977	TV-14	172 min	Action & Adventure	Abandoned in a park b
269880	Movie	Bad Boys	Michael Bay	Will Smith, Martin Lawrence, TÃ©a Leoni, TchÃ...	United States	2019-10-01	1995	R	119 min	Action & Adventure	In this fast-paced actio
281550	Movie	La Bamba	Luis Valdez	Lou Diamond Phillips, Esai Morales, Rosanna De...	United States	2020-01-01	1987	PG-13	109 min	Classic Movies	The plane crash that ki
284890	Movie	Barsaat	Rajkumar Santoshi	Twinkle Khanna, Bobby Deol, Danny Denzongpa...	India	2018-04-01	1995	TV-PG	166 min	Action & Adventure	A naÃ¯ve young man a
292118	Movie	Beavis and Butt-head Do America	Mike Judge	Mike Judge, Bruce Willis, Demi Moore, Cloris Lea...	United States	2019-11-20	1996	PG-13	81 min	Comedies	After realizing that the
296682	Movie	Benji	Joe Camp	Benji, Deborah Walley, Peter Breck, Edgar Buch...	United States	2018-03-06	1974	G	86 min	Children & Family Movies	After lovable abandon
347365	Movie	Candyman	Bernard Rose	Virginia Madsen, Tony Todd, Xander Berkeley, K...	United States, United Kingdom	2019-10-01	1992	R	99 min	Cult Movies	Grad student Helen Lyl
352989	Movie	Carrie	Brian De Palma	Sissy Spacek, Piper Laurie, Amy Irving, William ...	United States	2019-06-01	1976	R	98 min	Classic Movies	An outcast teen with t

The above queries give me a high-level understanding of what data exists within the dataset and how it is structured. It is immediately clear that this data is not structured well for the purposes of analysis (ex. multiple values per cell in *cast* column).

Data Cleansing

The next step is to evaluate data normalcy and consistency to understand what can and cannot be used for this analysis. I will search for NULL values in the dataset to see where a significant percentage of the column values are NULL. The *date_added* column oddly contains 3 NULL values out of the 6,231 rows. While I could ignore these values and discount those rows as incomplete data, I decided to manually add them in to complete the column.

```sql
--- Null value check for date_added ---
SELECT *
FROM netflixdata2
WHERE date_added = '0000-00-00';
```

picture_id	picture_type	title	director	cast	country	date_added
70153404	TV Show	Friends	NULL	Jennifer Aniston, Courteney Cox, Lisa Kudrow, ...	United States	0000-00-00
30201906	Movie	Black Panther	Ryan Coogler	Chadwick Boseman, Michael B. Jordan, Lupita N...	United States	0000-00-00
31016045	Movie	One Day	Banjong Pisanthanakun	Chantavit Dhanasevi, Nittha Jirayungyurn, The...	Thailand	0000-00-00

In a quick google search, I found the dates that these films were added to Netflix and used an UPDATE statement to insert them in the table:

```sql
-- Updating date_added for the 3 missing films ---
UPDATE netflixdata2
SET date_added = (CASE WHEN picture_id = '70153404' THEN '2015-01-01'
                       WHEN picture_id = '80201906' THEN '2018-09-04'
                       WHEN picture_id = '81016045' THEN '2018-09-05'
                       END)
WHERE picture_id IN ('70153404','80201906', '81016045');
```

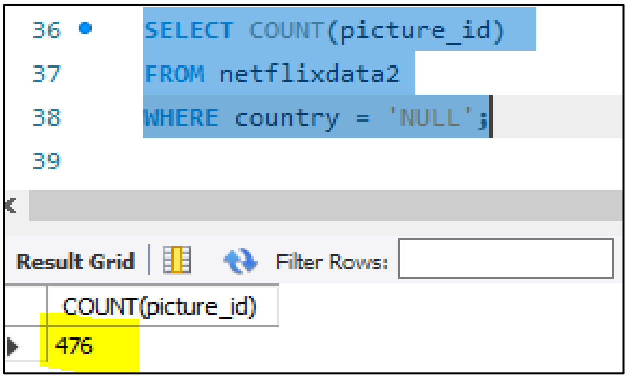
```sql
3 ● SELECT * FROM netflixdata2
4   WHERE picture_id IN ('70153404','80201906', '81016045');
```

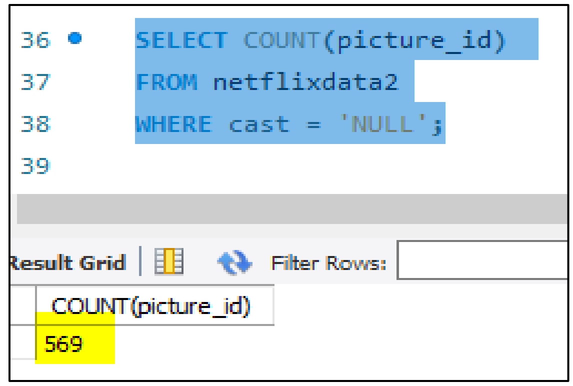
sult Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

picture_id	picture_type	title	director	cast	country	date_added
70153404	TV Show	Friends	NULL	Jennifer Aniston, Courteney Cox, Lisa Kudrow, ...	United States	2015-01-01
80201906	Movie	Black Panther	Ryan Coogler	Chadwick Boseman, Michael B. Jordan, Lupita N...	United States	2018-09-04
81016045	Movie	One Day	Banjong Pisanthanakun	Chantavit Dhanasevi, Nittha Jirayungyurn, The...	Thailand	2018-09-05

Some columns contained many NULL values, making it more difficult to normalize. One option would be to create or import another table with the missing information and use a JOIN clause to link them together. However, it is worth exploring the root cause in more detail to see if that NULL data should exist in the first place.

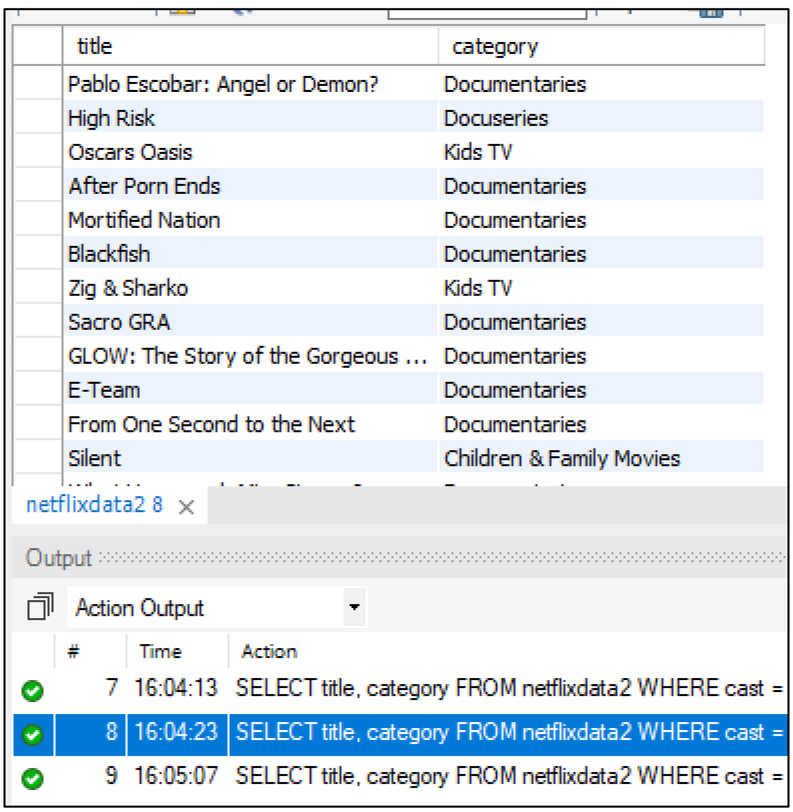
```sql
-- Identifying issues within the dataset ---
SELECT Count(picture_id)
FROM netflixdata2
WHERE country = 'NULL';

SELECT Count(picture_id)
FROM netflixdata2
WHERE cast = 'NULL';
```





The analysis above indicates that there are 476 NULL values in the *country* column and 570 NULL values in the *cast* column. Drilling into the *country* column, I can reasonably assume that the 476 NULL values (7% of dataset) are due to the removal of content from the platform which indicates it is no longer accessible in any country. Looking at the 569 NULL values (~9% of the dataset) in the *cast* column a little further, I notice an interesting trend.

```sql
SELECT title, category
FROM netflixdata2 WHERE cast = 'NULL';
```



It appears that the 569 NULL values for *cast* are largely due to the picture *category*. The categories where *cast* is NULL tend to be animated kids shows, or documentaries where they often will not include cast members. Therefore, we can assume the NULL values are valid and do not require modification.

However, the *category* column contains multiple values per cell, creating redundancy that should be eliminated. It appears that the multiple values listed in the *category* column are sub-categories or ancillary categories that the content may be loosely associated with (i.e. 'The Rugrats Movie' being considered Action & Adventure).

title	rating	duration	category
The Rugrats Movie	G	81	Children & Family Movie, Action & Adven...
Troy	R	163	Action & Adventure, Drama
Trainspotting	R	94	Comedies, Dramas
Agyaat	TV-MA	97	Horror Movies
25 Kille	TV-PG	140	Action & Adventure

For the purposes of normalizing the data, I am only interested in the primary category and thus can use the query below to rectify this column. The query updates the *category* column to only include the first value listed, which is the primary category, when there are multiple:

```
-- Updating Category to only include primary category --
UPDATE netflixdata2
SET category = SUBSTRING_INDEX(category, ',', 1);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

title	rating	duration	category
The Rugrats Movie	G	81	Children & Family Movie
Troy	R	163	Action & Adventure
Trainspotting	R	94	Comedies
Agyaat	TV-MA	97	Horror Movies
25 Kille	TV-PG	140	Action & Adventure

Additionally, I'll need to run a similar query for the *duration* of films. The *duration* currently has a few variations in the way it is displayed. A handful of titles provide the *duration* as an integer while others include an integer followed by "min", for minutes. To normalize these values, I will run a query to truncate any strings of their "min" portion of the value so that I am only dealing with an integer value across all records. An alternative to updating the table would to implement a check constraint that does not allow alpha characters within this column. Given the fact that TV Shows are denoted by season rather than minute, I felt it was easier to update just the movie durations since that is what I'm concerned with.

title	rating	duration
Harud	TV-14	91 min
Welcome Mr. President	TV-14	99
ChatÃ ́: The King of Brazil	NR	163 min
Bombshell	TV-MA	81 min
Outlaw King	R	122

```
--- eliminating 'min' from duration of movies
UPDATE netflixdata2
SET duration = SUBSTRING_INDEX(duration, ' ', 1)
WHERE picture_type = 'Movie';
```

title	rating	duration	category
Harud	TV-14	91	Dramas
Welcome Mr. President	TV-14	99	Comedies
ChatÃ´: The King of Brazil	NR	163	Dramas
Bombshell	TV-MA	81	Dramas
Outlaw King	R	122	Action & Adventure

Now that the data has been evaluated and cleansed, it is time to dig deeper into the dataset to uncover any trends or interesting takeaways.

Analysis

With the data clean and ready to work with, I want to look at the various categories (genres) that exist within this dataset, particularly for movies.

```
-- Movie categories in the data set ----
SELECT COUNT(DISTINCT category)
FROM netflixdata2
WHERE picture_type = 'Movie';
```

COUNT(DISTINCT category)
19

The query returned 19 distinct categories. Next, I want to understand how many movies are in each of those 19 categories in order to get a feel for the distribution of movie categories. To do this, I will use a basic aggregation function to count the unique picture_id per category:

```sql
-- distribution of movies in each of the movie categories ---
SELECT COUNT(picture_id), category
FROM netflixdata2
WHERE picture_type = 'Movie'
GROUP BY category
ORDER BY COUNT(picture_id) DESC;
```

COUNT(picture_id)	category
1077	Dramas
802	Comedies
643	Documentaries
597	Action & Adventure
357	Children & Family Movies
273	Stand-Up Comedy
205	Horror Movies
85	International Movies
61	Classic Movies
56	Movies
40	Thrillers

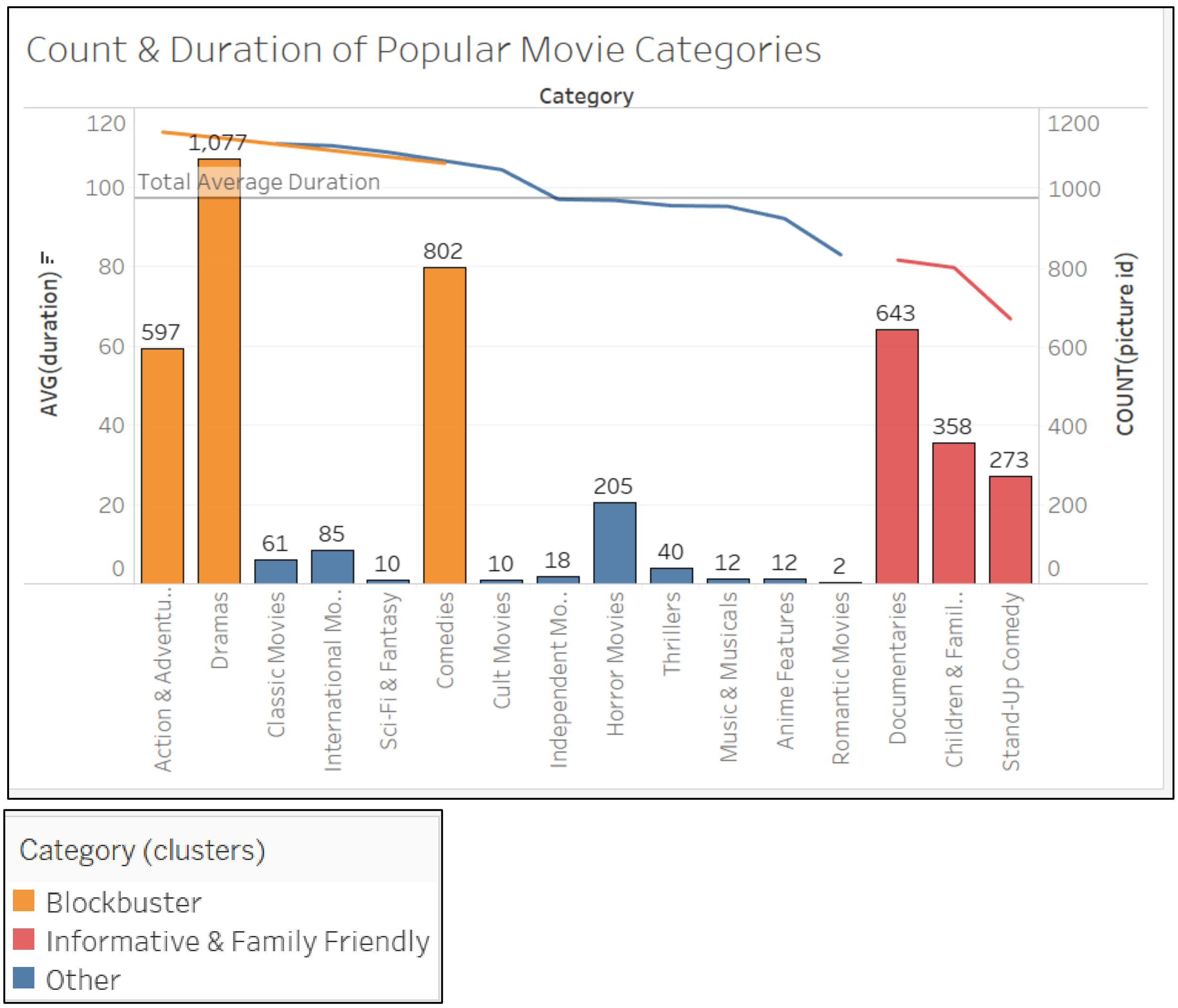
It's clear that of the movies added to Netflix as of 2020, Dramas were the most popular category of movie. I'm curious to see if there is any correlation between the category of movie and the duration of those movies. For example, perhaps Netflix found that shorter films tend to get more views, and therefore they seek to add movies in categories that typically have shorter durations.

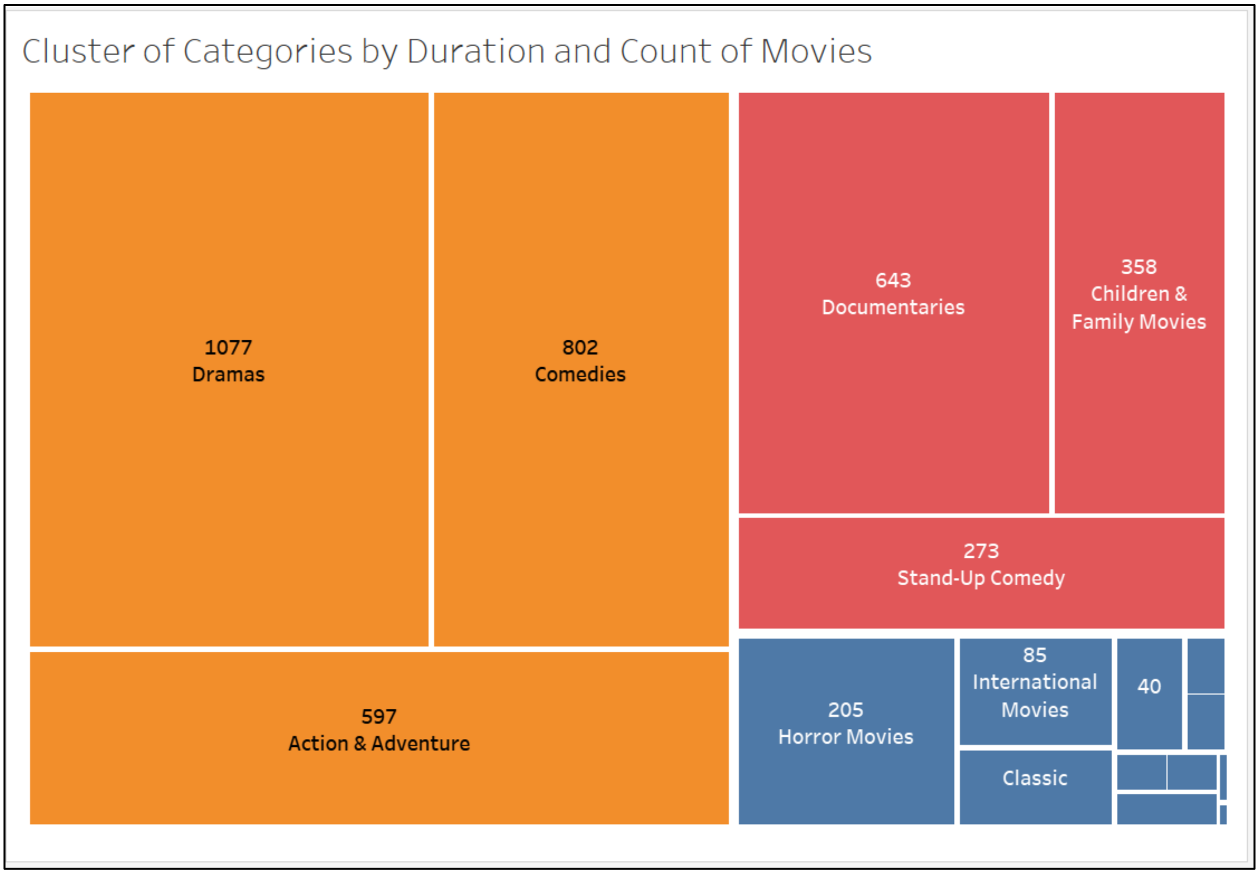
To get some clarity on this question, I used a common table expression (CTE) to aggregate multiple levels of data in one query. While there are several different ways to arrive at the same result, I found this to be best for readability.

```sql
-- Category, count of pictures within each category, and their avg runtime--
WITH t1 AS (
SELECT DISTINCT category, duration, picture_id
FROM netflixdata2
WHERE picture_type = 'Movie')

SELECT category, ROUND(AVG(duration),1) AS average_duration_minutes, COUNT(picture_id)
FROM t1
GROUP BY category
ORDER BY AVG(duration) DESC;
```

category	average_duration_minutes	COUNT(picture_id)
Action & Adventure	114.0	597
Dramas	112.6	1077
Classic Movies	111.0	61
International Movies	110.6	85
Sci-Fi & Fantasy	108.9	10
Comedies	106.1	802
Cult Movies	104.5	10
Independent Movies	97.0	18
Horror Movies	96.8	205
Thrillers	95.4	40
Music & Musicals	95.2	12

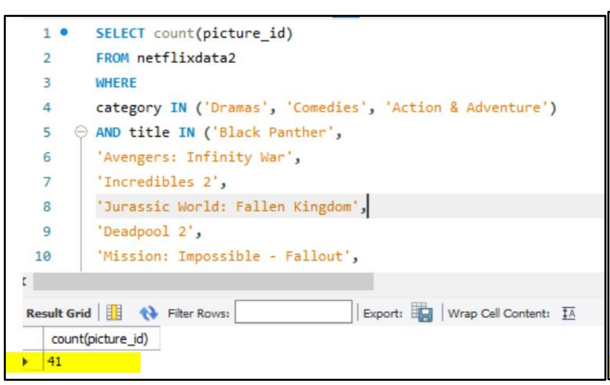
Now that I have a dimension (distinct movie categories) and 2 measures (duration and count of movies), I can leverage Tableau to build visualizations of the findings. My first step is to export the above result into an csv file. From there I can upload it into Tableau and build a few visuals to show the relationship between duration of movies and the count of each movie in those categories:

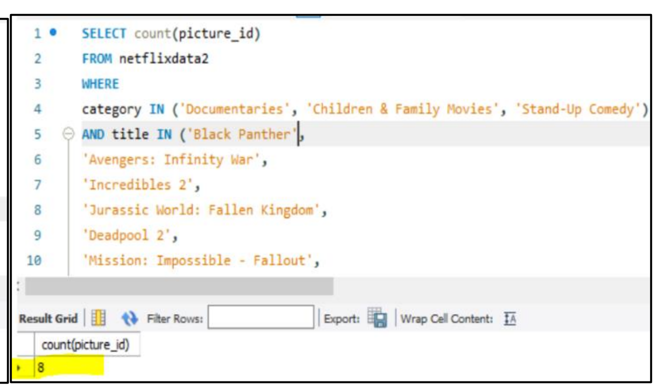


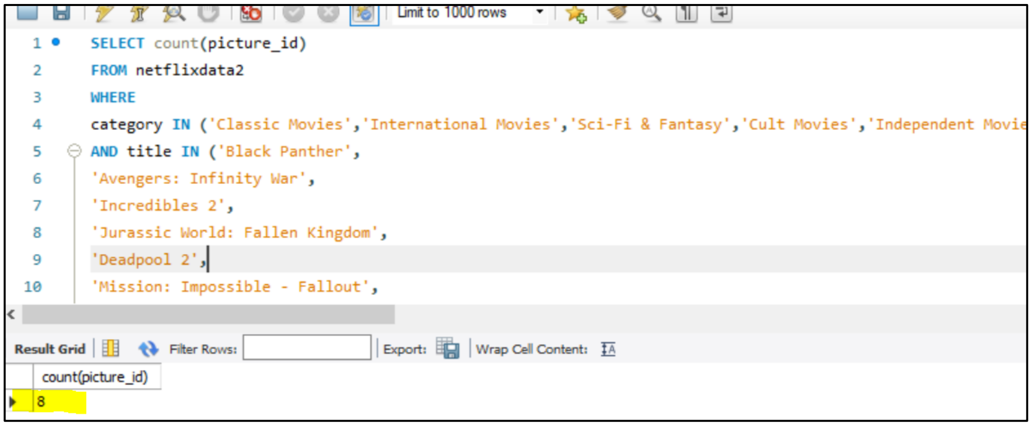


Summary of Visuals

The two visuals above are broken down by categories and grouped into 3 clusters based on the variables – 'Average Duration' and 'Count of Movie'. The first visual is a bar chart showing each of these clusters represented by different colors, overlaid with the average duration for that category. The 3 clusters contain movies that fall most closely to the mean of that cluster. The 3 clusters seem to fall within the following themes: "Blockbusters", "Informative & Family Friendly", and "Other". The movies contained within the orange blockbuster cluster contain 41 blockbuster movies from the [Top 10 Highest Grossing Films (1975-2018)] list from Kaggle. Compared to the 2 other clusters which contain only 8 blockbuster films each.







The second visual shows categories based on the count of movies within that category. The larger the block, the higher the count of movies within it. The Blockbuster cluster is the largest as it contains the most movies. The clusters are determined by the same set of variables as the first visual (i.e. 'Average Duration' and 'Count of Movie'). This visual emphasizes how much each category and cluster make up the total movies in the dataset.

Observations

If we look back to our original hypothesis: 'Netflix considers the duration and genre of movies when choosing what new content to add to their platform', we can make an informed judgement based on our analysis. We can see the 3 clusters in the first visual by noting a spike in the count of movies on the left, a dip in the middle, and a modest spike on the far right. The average duration is trending down from left to right relatively smoothly.

What we notice is that the categories with the most movies, have the longest average duration (Blockbuster cluster). We also notice that the categories that have the second most movies on average, have the shortest average duration (Informative & Family Friendly cluster). Finally, the categories that tend to have a duration somewhere in the middle (around 98 minutes), have significantly less movies on the platform (Other cluster). This indicates that perhaps duration DOES have an impact on the movies added to Netflix. Specifically, longer movies (~110 min on average) and shorter movies (~76 min on average) are added in droves, while movies that are in the middle, are much less popular. This could have to do with Netflix's target audience, catering to both movie enthusiasts who prefer longer movies and younger audiences with shorter attention spans.

To validate this assumption, we can look at the P-value of the independent variable (duration) to assess if it does have a significant impact on the dependent variable (count of movies). The table below shows the p-value for duration is >0.05 which indicates statistical significance (assuming 95% confidence threshold). Therefore, we can reject the null hypothesis and claim that duration of movies per category does affect the number of movies that Netflix adds to their platform within that category.

Analysis of Variance:

Variable	F-statistic	p-value	Model Sum of Squares	DF
Sum of COUNT(picture id)	6.11	0.01237	1.35	2
Sum of AVG(duration)	4.512	0.03073	0.8562	2

Limitation of the Data

While the data was useful for this exercise, there are several limitations worth noting. The first being the true accuracy of this data, it was downloaded from a 3rd party data aggregation site (Kaggle.com) and appears to have some inconsistencies in the data (missing values, skewed numbers, minimal normalization). Furthermore, by discounting the subcategories of movies, we lose some accuracy as to the numbers within each category. Additionally, the categories are not an even comparison to one another, as Stand-Up Comedy will not run as long as a full-length feature film. Ideally, I would take the average duration of each category and compare it to a broader database of all films within that category as a benchmark. Lastly, our dataset only accounts for content added, and not what has been deleted from the platform. Therefore, we are limited in our ability to get a read of total content on the platform at any given time.