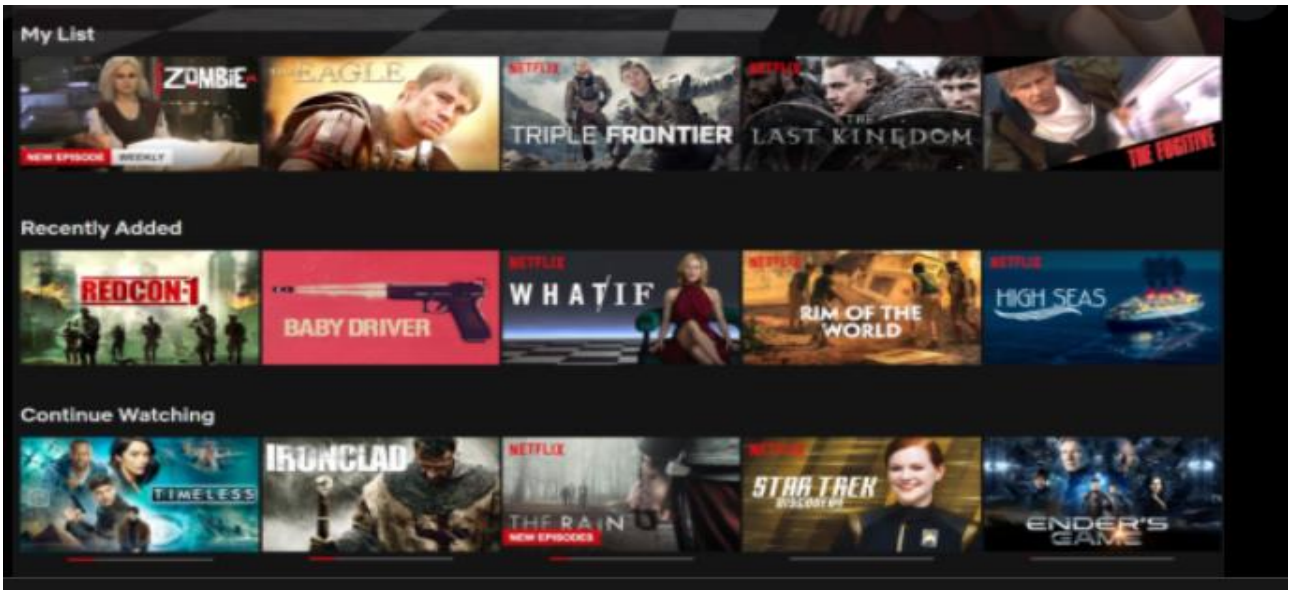
NETFLIX TOP TENS (1925-2020)
EXPLORATORY DATA ANALYSIS USING SQL



Introduction

This is a data analysis done on a very interesting dataset of Netflix. Netflix is one of the most famous entertainment platforms almost available in all countries in world. It contains countless movies and TV shows for all age groups and of all interests. Movies/ Tv shows in Netflix are given a rating that tells the viewer if that's appropriate for particular age group or not. In pandemic time, Netflix was the best friend of all people working from home.
So, lets jump in & find out some interesting sights from this data
Tools used during this analysis includes SQL, Microsoft SQL server Management Studio, SQL import/export utility, Tableau.

Exploratory Data Analysis mainly include following steps:

1. Business task/ Problem statement
2. Data collection
3. Data wrangling
4. Data Analysis
5. Data visualization
6. Sharing

Business Task

EDA was performed to extract 'Top Tens' for all important attributes in data:

1. Top 10 years in which maximum movies were released
2. Top 10 directors with maximum number of movies

3. Top 10 Actor/actress on the basis of maximum movies done
4. Top 10 longest series/movies & other interesting facts

Data Collection

Data was downloaded from public source. It was provided in form of separate table but for analysis, SQL was use to join all useful attributes to answer business task. Dataset included 4 tables. By using primary key and foreign keys, all tables were compiled into one main table that was having 139942 rows and 13 columns. Dataset included data from 1925-2020 (almost 100years!!!)

Data Wrangling

Data wrangling is a process transform raw dirty data to clean data that can be used for actual analysis. Depending upon project, the methods & tools can vary.

Firstly, missing values, empty rows, invalid data & data types and duplicate values were removed. 'show_Id' was the primary key and was helpful in relating different tables. It was observed that 4 rows were without any 'show_id' , so those were removed. Also , other columns were having incomplete data but keeping primary key in mind, other data was used as it is.

So now, here are few snapshots of my work with outputs. So firstly, exploring first table.

```
select * from projects..netflix_titles$
```

Results | Messages

	duration_minutes	duration_seasons	type	title	date_added	release_year	rating	description	show_id
1	90	NULL	Movie	Norm of the North: King Sized Adventure	2019-09-09 00:00:00.000	2019	TV-PG	Before planning an awesome wedding for his grand...	81145628
2	94	NULL	Movie	Jandino: Whatever it Takes	2016-09-09 00:00:00.000	2016	TV-MA	Jandino Asporaat riffs on the challenges of raising ki...	80117401
3	NULL	1	TV Show	Transformers Prime	2018-09-08 00:00:00.000	2013	TV-Y7-FV	With the help of three human allies, the Autobots on...	70234439
4	NULL	1	TV Show	Transformers: Robots in Disguise	2018-09-08 00:00:00.000	2016	TV-Y7	When a prison ship crash unleashes hundreds of D...	80058654
5	99	NULL	Movie	#realityhigh	2017-09-08 00:00:00.000	2017	TV-14	When nerdy high schooler Dani finally attracts the int...	80125979
6	NULL	1	TV Show	Apaches	2017-09-08 00:00:00.000	2016	TV-MA	A young journalist is forced into a life of crime to sav...	80163890
7	110	NULL	Movie	Automata	2017-09-08 00:00:00.000	2014	R	In a dystopian future, an insurance adjuster for a tec...	70304989
8	60	NULL	Movie	Fabrizio Copano: Solo pienso en mi	2017-09-08 00:00:00.000	2017	TV-MA	Fabrizio Copano takes audience participation to the ...	80164077
9	NULL	1	TV Show	Fire Chasers	2017-09-08 00:00:00.000	2017	TV-MA	As California's 2016 fire season rages, brave backco...	80117902
10	90	NULL	Movie	Good People	2017-09-08 00:00:00.000	2014	R	A struggling couple can't believe their luck when the...	70304990
11	78	NULL	Movie	Joaquín Reyes: Una y no más	2017-09-08 00:00:00.000	2017	TV-MA	Comedian and celebrity impersonator Joaquín Reye...	80169755
12	95	NULL	Movie	Kidnapping Mr. Heineken	2017-09-08 00:00:00.000	2015	R	When beer magnate Alfred "Freddy" Heineken is kid...	70299204
13	58	NULL	Movie	Krish Trish and Baltiboy	2017-09-08 00:00:00.000	2009	TV-Y7	A team of minstrels, including a monkey, cat and do...	80182480
14	62	NULL	Movie	Krish Trish and Baltiboy: Battle of Wits	2017-09-08 00:00:00.000	2013	TV-Y7	An artisan is cheated of his payment, a lion of his thr...	80182483
15	65	NULL	Movie	Krish Trish and Baltiboy: Best Friends Forever	2017-09-08 00:00:00.000	2016	TV-Y	A cat, monkey and donkey team up to narrate folktal...	80182596
16	61	NULL	Movie	Krish Trish and Baltiboy: Comics of India	2017-09-08 00:00:00.000	2012	TV-Y7	In three comic-strip-style tales, a boy tries to sell wis...	80182482
17	65	NULL	Movie	Krish Trish and Baltiboy: Oversmartness Never Pays	2017-09-08 00:00:00.000	2017	TV-Y7	A cat, monkey and donkey learn the consequences ...	80182597

For checking uniqueness of data:

```
select count(distinct [show_id])
from projects..netflix_titles$
```

	(No column name)
1	6232

Output shows 6232 which is 4 less than 6236 (total records in raw data). That means primary key has some null values. So,rows were removed.Next checked duplicates

```
SELECT [show_id], COUNT(*) as n
FROM projects..netflix_titles$
GROUP BY [show_id]
HAVING COUNT(*) > 1

select [type], [title], [show_id]
from projects..netflix_titles$
where [show_id] is null

delete from projects..netflix_titles$
where [show_id] is null
```

Data Analysis

After clearing out duplicates, NULL values from ‘show_id’, checking data types, finally analysis was done. First, started checking how many ratings with how many movies are on Netflix.

```
select distinct[rating]
from projects..netflix_titles$
```

	rating
1	TV-Y
2	TV-G
3	NC-17
4	TV-MA
5	NR
6	PG
7	TV-Y7-FV
8	NULL
9	TV-Y7
10	TV-14
11	PG-13
12	G
13	UR
14	R
15	TV-PG

So total 14 rating, with one showing NULL. As our focus was finding TOP tens, not all NULLs were removed.

```
select sum([duration_minutes]) as total_movietime, sum([duration_seasons]) as
total_seasons,count([type]) as num_of_movies_shows, [rating]
from projects..netflix_titles$
group by [rating]
order by num_of_movies_shows desc
```

Checking which rating has maximum movies under it. Later, exploring years in which maximum movies released

```
select round (max([duration_minutes])/60,2)  as longest_movie_duration, [release_year]
from projects..netflix_titles$
group by [release_year]
order by [longest_movie_duration] desc
```

Results | Messages

	longest_movie_duration	release_year
1	5.2	2018
2	3.8	1964
3	3.73	2001
4	3.57	2008
5	3.48	2019
6	3.42	2015
7	3.38	2009
8	3.35	2003
9	3.33	1965
10	3.27	2006
11	3.25	1993
12	3.22	1994
13	3.2	1989
14	3.17	2014
15	3.15	1999
16	3.12	1997
17	3.08	1962

Output shows longest movie was in 2018 with duration of 5.2 hrs. Also to check which searies has maximum number of seasons.

```
select count([title]), max([duration_minutes]), max([duration_seasons])
from projects..netflix_titles$
```

```
elect [title], [release_year]
from projects..netflix_titles$
where [duration_seasons] = 15
```

Results | Messages

	title	release_year
1	Grey's Anatomy	2018
2	NCIS	2017

Two series NCIS and Grey's Anatomy found to have 15 seasons.

```
select TOP 10 ([title]), [duration_minutes],[release_year]
from projects..netflix_titles$
order by [duration_minutes] desc
```

Top 10 longest movies with year in which it was released

Results | Messages

	title	duration_minutes	release_year
1	Black Mirror: Bandersnatch	312	2018
2	Sangam	228	1964
3	Lagaan	224	2001
4	Jodhaa Akbar	214	2008
5	The Irishman	209	2019
6	The Gospel of Luke	205	2015
7	What's Your Raashee?	203	2009
8	The Lord of the Rings: The Return of the King	201	2003
9	Doctor Zhivago	200	1965
10	Elephants Dream 4 Hour	196	2006

```
select TOP 10 ([title]), [duration_seasons],[release_year]
from projects..netflix_titles$
order by [duration_seasons] desc
```

Top Ten series on the basis of the number of seasons with year in which released on Netflix

Results | Messages

	title	duration_seasons	release_year
1	Grey's Anatomy	15	2018
2	NCIS	15	2017
3	Supernatural	14	2019
4	COMEDIANS of the world	13	2019
5	Red vs. Blue	13	2015
6	Trailer Park Boys	12	2018
7	Criminal Minds	12	2017
8	Cheers	11	1992
9	Heartland	11	2017
10	Frasier	11	2003

Checking other tables & then joined on the basis of primary key.

```
select * from projects..netflix_titles_cast$

select* from projects..netflix_titles_category$
```

Finally created a VIEW after joining table for later analysis and visualization.

```
create view final_table
as
(select main_table.* , cat.listed_in, countries.country , dir.director
from main_table
left join projects..netflix_titles_category$ as cat
on main_table.show_id= cat.show_id
left join  projects..netflix_titles_countries$  as countries
on main_table.show_id= countries.show_id
left join projects..netflix_titles_directors$ as dir
on main_table.show_id= dir.show_id)
```

Top 10 categories of movies that were famous on Netflix

```
select top 10 [listed_in], count([type]) as num_of_movies
from projects..final_table
group by [listed_in]
order  by count([type]) desc
```

100 %

Results | Messages

	listed_in	num_of_movies
1	Dramas	19831
2	International Movies	19659
3	Comedies	13952
4	International TV Shows	8928
5	Action & Adventure	8331
6	Independent Movies	7146
7	TV Dramas	6587
8	Children & Family Movies	5874
9	Thrillers	4624
10	Romantic Movies	3883

```
select distinct([cast])
from projects..final_table

select distinct([director])
from projects..final_table
```

Now on the basis of number of movies on Netflix done by actor/actress, rank was given. For this windows function were used. RANK(), DENSE_RANK(), LEAD(), LAG() were used in following SQL queries.

```
select [cast], count([title]) as total_movies, rank ()
over (order by count([title]) desc) as rank_cast
from projects..final_table
group by [cast]
having [cast] is not null
```

Rank() always give same rank to same value and skip the next rank, so Dense_Rank() was used.

```
select [cast], count([title]) as total_movies, dense_rank ()
over (order by count([title]) desc) as rank_cast
from projects..final_table
```

```
group by [cast]
having [cast] is not null
```

Top Actor/Actress on Netflix with maximum number of movies

Results | Messages

	cast	total_movies	rank_cast
1	Alfred Molina	144	1
2	Liam Neeson	131	2
3	John Krasinski	123	3
4	John Rhys-Davies	122	4
5	Frank Langella	114	5
6	Salma Hayek	108	6
7	David Attenborough	105	7
8	Quvenzhané Wallis	100	8
9	Anupam Kher	99	9
10	Radhika Apte	95	10
11	Shah Rukh Khan	94	11
12	Naseeruddin Shah	90	12
13	Ben Whishaw	83	13
14	Jim Broadbent	83	13
15	Paresh Rawal	82	14
16	Om Puri	82	14
17	Luci Christian	81	15

Query executed successfully.

To check just any randon rank, like 10th rank , following query was used.

```
WITH row_table AS (
select [cast], count([title]) as total_movies, dense_rank ()
over (order by count([title]) desc) as rank_cast
from projects..final_table
group by [cast]
having [cast] is not null)
SELECT *
FROM row_table
where [rank_cast]= 10
```

Results | Messages

	cast	total_movies	rank_cast
1	Radhika Apte	95	10

Used LAG() & LEAD() to compare number of movies in previous & forward years.

```
select [release_year], LAG(count([title])) OVER (ORDER BY [release_year]) as
previous_yr_movies, count([title]) as current_movies
FROM projects..final_table
GROUP BY [release_year]
ORDER BY [release_year]


--using lead () function to check next year number of movies
select [release_year], Lead(count([title])) OVER (ORDER BY [release_year]) as
next_yr_movies, count([title]) as current_movies
FROM projects..final_table
GROUP BY [release_year]
ORDER BY [release_year]
```

Results | Messages

	release_year	next_yr_movies	current_movies
1	1925	6	1
2	1942	5	6
3	1943	22	5
4	1944	12	22
5	1945	30	12
6	1946	8	30
7	1947	27	8
8	1954	20	27
9	1955	21	20
10	1956	76	21
11	1958	12	76
12	1959	129	12
13	1960	120	129
14	1962	2	120
15	1963	24	2
16	1964	130	24
17	1965	24	130

Query executed successfully.

Ranks was also provided to director with maximum movies.

```
select [director], count([title]) as total_movies, dense_rank ()
over (order by count([title]) desc) as rank_dir
from projects..final_table
group by [director]
having [director] is not null
```

Top Ten director famous on Netflix

Results | Messages

	director	total_movies	rank_dir
1	Lars von Trier	336	1
2	Saul Dibb	270	2
3	Tom Hooper	246	3
4	David Dhawan	243	4
5	Noah Baumbach	242	5
6	McG	241	6
7	Steven Spielberg	241	6
8	Martin Scorsese	230	7
9	Umesh Mehra	228	8
10	Lilly Wachowski	226	9
11	Lana Wachowski	226	9
12	Terry Gilliam	219	10
13	Noam Murro	219	10
14	Cathy Garcia-Molina	210	11
15	Ari Folman	210	11
16	Robert Altman	210	11
17	Johnnie To	209	12

Query executed successfully.

Lars Van Trier was on top of the list with 336 movies released on Netflix.

```
with rank_dir
as (select [director], count([title]) as total_movies, dense_rank ()
over (order by count([title]) desc) as rank_dir
from projects..final_table
group by [director]
having [director] is not null)
select * from rank_dir
where [rank_dir]=30
```

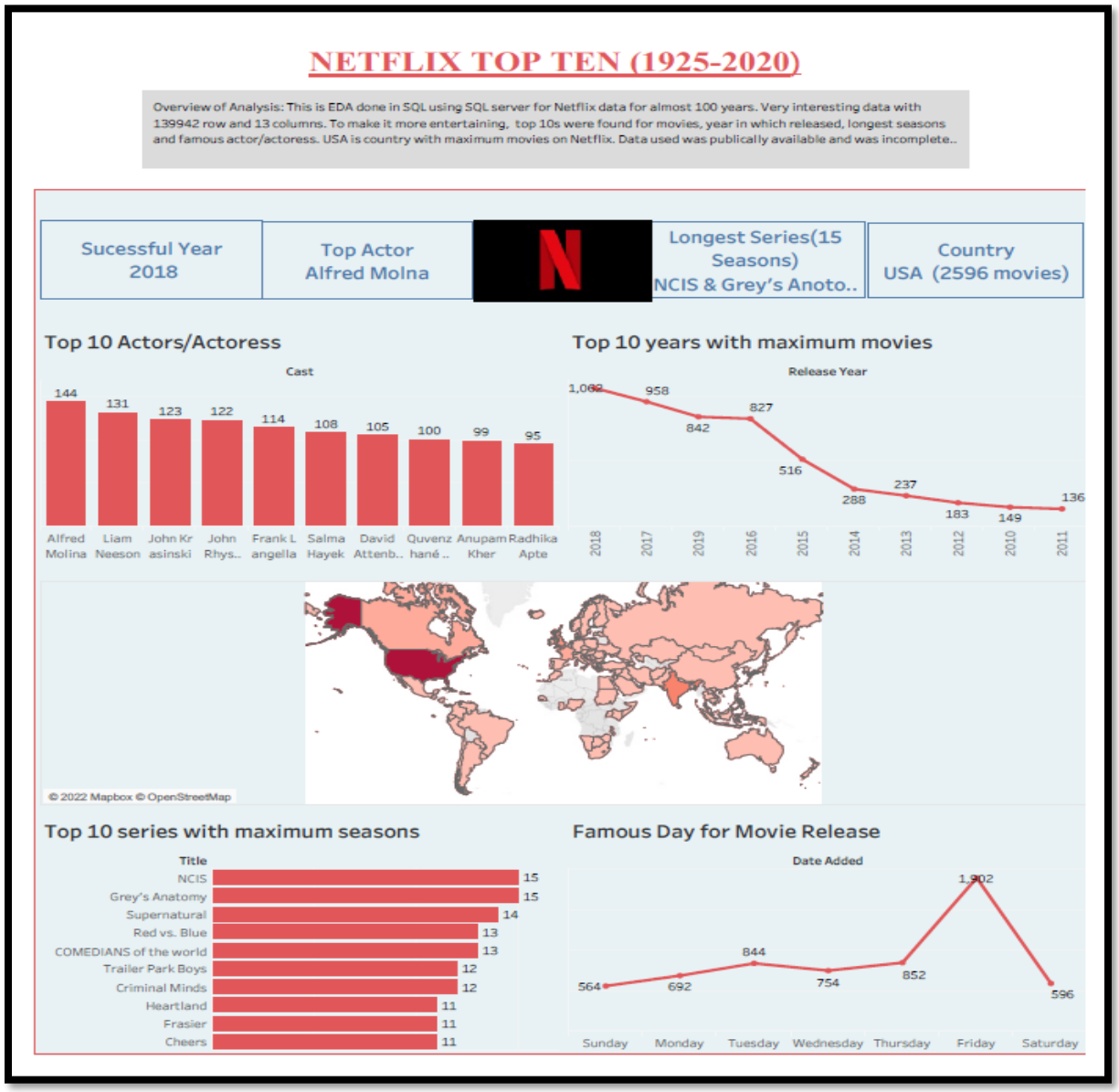
Results | Messages

	director	total_movies	rank_dir
1	Bong Joon Ho	156	30
2	Petra Costa	156	30

Data Visualization & Sharing

Once the crucial step of analysis is over, next comes how to present it in form of report or a dashboard and share with team or stakeholder. As this project was done for self practise and learning, it was shared on LinkedIn & Tableau Public for valuable feedback. For creating dashboard, Tableau Public was used and tried to narrate beautiful Top Ten story of Netflix from 1925-2020.

Tableau is very powerful BI tool famous these days. Its impeccable features make it stronger than other BI tools available in market. After various tries, following was the dashboard created and published on tableau.



So here comes the end to this project. I am sure you must have learned something new and interesting about Netflix.

Summary

This project was done for my own learning and practise on one of the most powerful tools in data analytical field: SQL & Tableau. This project has given me even more confidence and comfort level working with these tools. Learned many new features and SQL functions during analysis.