Business Case Study: Popular OTT Streaming Platform

EDA Using Python

Context:

This particular business case focuses on the operations of a popular media and video streaming platform, that have over 10000 movies or tv shows available with more than 200 million subscribers globally. This case study aims to analyze the dataset consisting of listings of all the movies and tv shows available on the platform, along with cast, directors, ratings, release year, duration, etc to provide data driven insights and actionable business recommendations to help business decide which type of shows/movies to produce and how they can grow the business in different countries.

This case study report contains the solutions to the problem statements (using Python queries), sample output of the queries, followed by insights and recommendations. As part of the confidentiality agreement, the name of the streaming platform, the actual dataset and problem statements are not included in this report.

Google Colab Notebook-Python File - This Python project involves exploratory data analysis (EDA) of a dataset from this streaming platform.

****

1. Import the dataset in pandas and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

Shape of the dataset and Data type of all columns –

Code:

```
data = pd.read_csv("platform.csv")
```

There are a total of 8807 rows and 12 columns. The following columns – show_id, type, title, release_year, listed_in and description have non-null entries. Other columns have null entries in the range of 3- 260. Except release_year, rest of the columns have datatype as object. Release_year has integer datatype.

```
[137] data.shape

(8807, 12)

data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

To move further, we first need to convert the date_added column in the dataset to a datetime formatted column and extract year and month from it for further data fetch.

Code:

```
data['date_added'] = data['date_added'].str.strip()

data['release_date'] = pd.to_datetime(data['date_added']).dt.strftime('%Y-%m-%d')

data['release_date'] = pd.to_datetime(data['release_date'])

data["year"] = data['release_date'].dt.year

data["month"] = data['release_date'].dt.month_name()
```

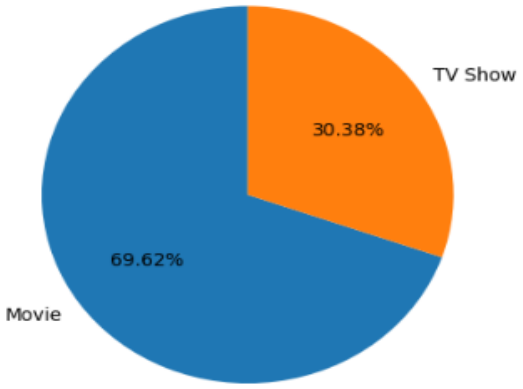
Output: We now have 3 new columns created from date_added column – release_date, year, and month

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	release_date	year	month
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	2021-09-25	2021.0	September
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	2021-09-24	2021.0	September
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	2021-09-24	2021.0	September

2. Comparison of TV Shows vs Movies
   - Number of TV shows and Movies added to the platform so far and their percentage split

Code:
```
type_count = data.groupby("type")["title"].count()
```
Code Output:

```
type
Movie      6131
TV Show    2676
Name: title, dtype: int64
```



Percentage Split using Pie Chart = `plt.pie(type_count, labels = type_count.index, startangle = 90, autopct = "%.2f%%")`

Insights: There are 6,131 Movies and 2,676 TV shows added. Movies constitute of 69.6% of total content available.

   - Get the following year range available in the dataset:
     i. Release_year range of all tv shows and movies
     ii. release date range of all tv shows and movies

Code:
(i)
```
data.groupby("type").agg(min_year = ("release_year", "min"),
max_year = ("release_year", "max"))
```
(ii)
```
data.groupby("type").agg(min_year = ("release_date", "min"),
max_year = ("release_date", "max"))
```

Output

(i)

type	min_year	max_year
Movie	1942	2021
TV Show	1925	2021

(ii)

type	min_year	max_year
Movie	2008-01-01	2021-09-25
TV Show	2008-02-04	2021-09-24

Insights: All movies were released between 1942 and 2021 and TV shows between 1925 and 2021. They were released to the platform between 2008 and 2021. The platform started releasing Movies and TV shows almost around the same time. Still, Movies constitute of 69.6% of total content available.

- Does this streaming platform has more focus on TV Shows than movies in recent years?

Code:

```
data.groupby("year")["type"].value_counts().sort_index(ascending=False)[:10]
```

Insights: In the last 5 years, from 2021-2017, there were more movies added to the platform than TV shows. In 2021, there were 993 movies and 505 TV shows added, means 488 more movies than TV shows. In 2020, there were 689 more movies than TV shows added. In 2019, it was 832 more movies than TV shows. The number of movies added to the platform has drastically reduced from 2019 to 2021 – from 1424 to 993. But so has number of TV shows – from 592 in 2019 to 505 in 2021. Hence, we can say the platform still focuses more on movies than TV shows.

```
year    type
2021.0  TV Show     505
        Movie       993
2020.0  TV Show     595
        Movie      1284
2019.0  TV Show     592
        Movie      1424
2018.0  TV Show     412
        Movie      1237
2017.0  TV Show     349
        Movie       839
Name: count, dtype: int64
```

- What time of the year is the best time to launch a TV show?

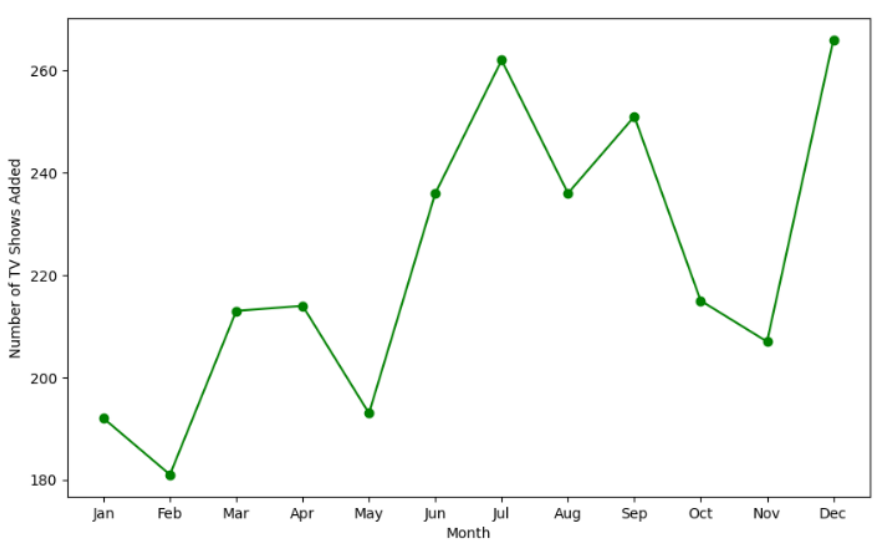
Code:

```
data[data["type"] == "TV Show"].groupby("month")["title"].count().
sort_values(ascending=False)
```

Code for Line Chart:

```
data["monthly"] = data["release_date"].dt.month
new = data[data["type"] == "TV Show"].groupby("monthly")["title"].count()
plt.figure(figsize=(10, 6))
plt.plot(new.index, new.values, marker='o', color = 'g')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows Added')
plt.xticks(ticks=range(1, 13), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May',
'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov',
'Dec'])
plt.show()
```

Output ->




```
month
December     266
July         262
September    251
August       236
June         236
October      215
April        214
March        213
November     207
May          193
January      192
February     181
Name: title, dtype: int64
```

Insights: December has seen the highest number of releases on TV shows on the platform, followed by July. December coincides with the holiday season, including Christmas and New Year's celebrations. During this period, people typically have more free time, are on vacation. This leads to an increase in streaming activity as people look for entertainment options. The platform may also aim to boost its year-end performance metrics by releasing a significant amount of content in December. July is part of the summer vacation period for many people, including school and university students. This time off leads to increased leisure time and higher demand for entertainment.

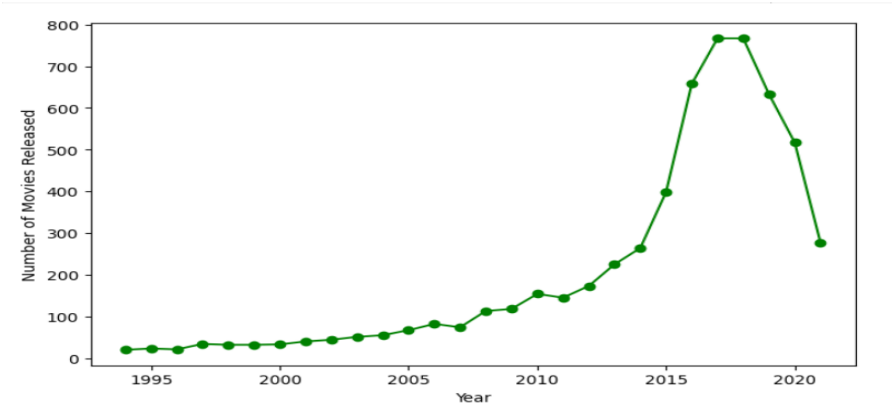
- How has the number of movies released per year changed over the last 20-30 years?
  Code:

```
from datetime import datetime
current_year = datetime.now().year
threshold_year = current_year - 30

last_30_years = data[data['release_year'] >= threshold_year]
yearly_trend = last_30_years[last_30_years["type"] ==
"Movie"].groupby("release_year")["title"].count()

plt.figure(figsize=(8, 5))
plt.plot(yearly_trend.index, yearly_trend.values, marker='o', color =
'g')
plt.xlabel('Year')
plt.ylabel('Number of Movies Released')
plt.show()
```

  Output:



Insights: There was a slow but steady increase in the number of movies released per year from 1995 to around 2010. There is a significant and rapid increase in the number of movies released per year from 2010 to around 2018. Technological advancements and the rise of streaming platforms like this one, which started producing and acquiring a large volume of content contributed to this trend. The sharp drop in the number of movies released in the last few years since 2019 likely reflects the impact of the COVID-19, which caused production halts, and delays.

3. Analysis of actors/directors of different types of shows/movies
   - Top 5 cast on the basis of number of TV shows and movies

Code:

```
#Firstly, split cast column values into multiple columns
cast_split = data["cast"].str.split(', ', expand=True)

#Convert it to a pandas dataframe
cast_split = pd.DataFrame((cast_split.values)

#Add column headers for all new columns
cast_split.columns = [f'cast_{i+1}' for i in range(cast_split.shape[1])]

#Merge the newly split cast columns with the title column
cast_split = pd.concat([data["title"], cast_split], axis=1)

#Unpivot or melt the data to transform it from wide format to long format
cast_melted = cast_split.melt(id_vars = ["title"],var_name = "cast_num",
                    value_name= "cast_name")
```

data sample of cast_melted ->

	title	cast_num	cast_name
0	Dick Johnson Is Dead	cast_1	NaN
1	Blood & Water	cast_1	Ama Qamata
2	Ganglands	cast_1	Sami Bouajila
3	Jailbirds New Orleans	cast_1	NaN
4	Kota Factory	cast_1	Mayur More
...	...	...	...
440345	Zodiac	cast_50	None
440346	Zombie Dumb	cast_50	NaN
440347	Zombieland	cast_50	None
440348	Zoom	cast_50	None
440349	Zubaan	cast_50	None

440350 rows × 3 columns

```
#Now, drop cast_num column and remove NaN
values

cast_melted=cast_melted.drop("cast_num",axis = 1)

cast_melted = cast_melted.dropna()
```

Final Output: No of rows has now increased to 64,126

	title	cast_name
1	Blood & Water	Ama Qamata
2	Ganglands	Sami Bouajila
4	Kota Factory	Mayur More
5	Midnight Mass	Kate Siegel
6	My Little Pony: A New Generation	Vanessa Hudgens
...	...	...
417703	Black Mirror	Jon Hamm
424590	Social Distance	Ayize Ma'at
426510	Black Mirror	Oona Chaplin
433397	Social Distance	Lovie Simone
435317	Black Mirror	Rafe Spall

64126 rows × 2 columns

Top 5 Cast - Code:

```
cast_melted.groupby(["cast_name"])["title"].nunique().
sort_values(ascending=False).head()
```

Output ->

```
cast_name
Anupam Kher          43
Shah Rukh Khan       35
Julie Tejwani        33
Naseeruddin Shah     32
Takahiro Sakurai     32
Name: title, dtype: int64
```

Insights: Top 5 casts are Anupam Kher, Shah Rukh Khan, Julie Tejwani, Naseeruddin Shah and Rakahiro Sakurai – majorly dominated by Indian actors.

Tops 5 Cast (from a specific Content Type) – Code:

```
cast_melted.groupby(["cast_name","type"])["title"].
nunique().sort_values(ascending=False)
```

Output ->

```
cast_name          type
Anupam Kher        Movie      42
Shah Rukh Khan     Movie      35
Naseeruddin Shah   Movie      32
Om Puri            Movie      30
Akshay Kumar       Movie      30
                              ..
Jack Gilpin        Movie       1
Jack Foley         Movie       1
Jack Fisher        TV Show     1
                   Movie       1
Ṣọpẹ́ Dìrísù       Movie       1
Name: title, Length: 40814, dtype: int64
```

Insights: Top 5 cast is still topped by Anupam Kher (for movie category) followed by Shah Rukh Khan, Naseeruddin Shah, Om Puri and Akshay Kumar. Both actors are prominent figures in Indian cinema, suggesting this platform's recognition of their cultural influence and the appeal of their content to global audiences. Their frequent appearances may reflect high viewer engagement and interest in their work, prompting the platform to feature more of their productions to cater to audience preferences.

- Top 5 cast-director combination on the basis of number of TV shows and movies
  Code:

```
#Follow the same split, melt and concat method used for unpivoting cast
director_split = data["director"].str.split(', ', expand=True)

director_split = pd.DataFrame(director_split.values)

director_split.columns = [f'director_{i+1}' for i in
range(director_split.shape[1])]

director_subset = data[["title", "type"]]

director_split  = pd.concat([director_subset, director_split], axis=1)

director_melted = director_split.melt(id_vars = ["title", "type"],
                    var_name = "director_num",
                    value_name= "director_name")
```

```
director_melted = director_melted.drop("director_num", axis = 1)
director_melted = director_melted.dropna()
```

Output ->

	title	type	director_name
0	Dick Johnson Is Dead	Movie	Kirsten Johnson
2	Ganglands	TV Show	Julien Leclercq
5	Midnight Mass	TV Show	Mike Flanagan
6	My Little Pony: A New Generation	Movie	Robert Cullen
7	Sankofa	Movie	Haile Gerima
...	...	...	...
95585	Movie 43	Movie	Rusty Cundieff
102764	Walt Disney Animation Studios Short Films Coll...	Movie	Mike Gabriel
103787	HALO Legends	Movie	Hiroshi Yamazaki
104392	Movie 43	Movie	James Gunn
111571	Walt Disney Animation Studios Short Films Coll...	Movie	Mark Henn

3978 rows × 3 columns

```
#Now, merge the result with the melted_cast data above and get the unique title count for each actor-director combination

dir_cast_merged = cast_melted.merge(director_melted, how = "inner", on = ["title", "type"])

dir_cast_merged[dir_cast_merged["director_name"] == 'Priyadarshan']

dir_cast_merged.groupby(["cast_name", "director_name"])["title"].nunique().sort_values(ascending=False).head()
```

Output:

```
cast_name       director_name
Rajesh Kava     Rajiv Chilaka    19
Julie Tejwani   Rajiv Chilaka    19
Rupa Bhimani    Rajiv Chilaka    18
Jigna Bhardwaj  Rajiv Chilaka    18
Vatsal Dubey    Rajiv Chilaka    16
Name: title, dtype: int64
```

Insights: The Top cast - director combination is Rajesh Kava and Rajiv Chilaka, majorly for the Indian cartoon series - Chhota Bheem

Outlier Checks:

- Average duration (in mins) for movies and average number of seasons for TV shows and create a boxplot for outlier checks

  Code:

```
data["duration"] = data["duration"].str.split().str[0].astype(float)
data.groupby("type")['duration'].mean()
```

Output ->

```
type
Movie      99.577187
TV Show     1.764948
Name: duration, dtype: float64
```

Insights: Movies have an average duration of 99 mins and TV shows have an average number of 1.7 seasons

Boxplot code:

(i) Movie_plot:

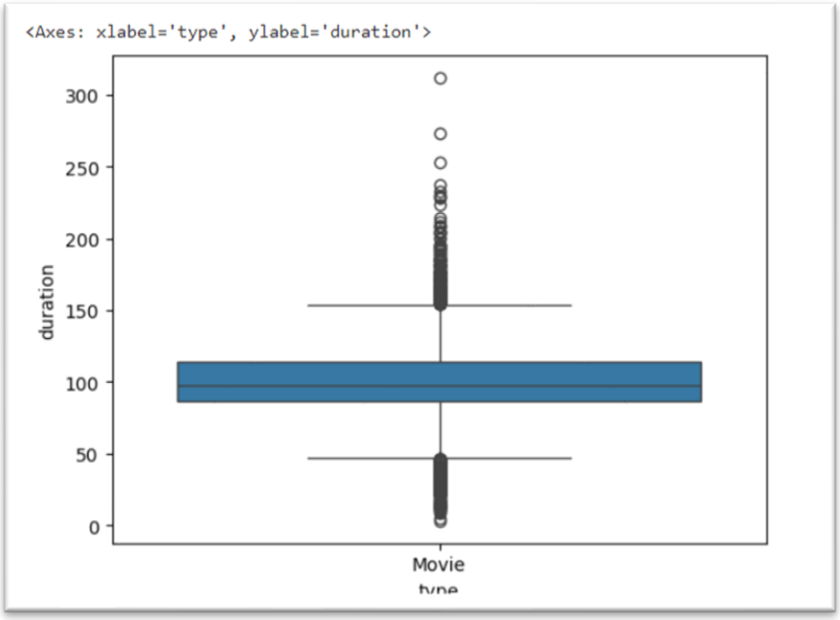
```
Movie_plot = data[data["type"] == "Movie"]
sns.boxplot(data = Movie_plot, x = "type", y = "duration")
```

(ii) TV Show_plot:

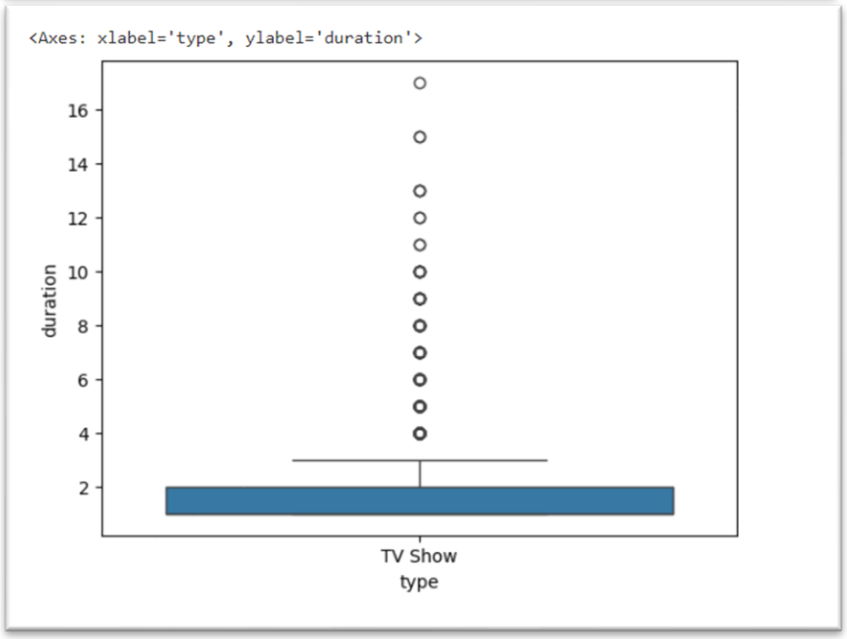
```
TVSHOW_plot = data[data["type"] == "TV Show"]
sns.boxplot(data = TVSHOW_plot, x = "type", y = "duration")
```

Output:

(i)



(ii)



Insights: The median TV show duration is 2 seasons, indicating that most TV shows have relatively short runs. There are numerous outliers in TV show durations. These outliers represent TV shows with durations significantly longer than the typical range. The outliers extend up to 16 seasons. These long-running shows could be popular series with multiple seasons, possibly due to high viewership and demand. For the platform, understanding the typical duration and the presence of outliers can help in content planning. It can inform decisions about the expected lifecycle of new shows and the allocation of resources for long-running series.

The median duration of movies is approximately 100 minutes. The middle 50% of movie durations range from about 80 to 120 minutes.

Recommendations:

1. Adding more movies can diversify the content library for the platform, catering to a broad audience base with different time availabilities. It offers a one-time, shorter engagement, which can appeal to viewers looking for a quick entertainment fix. Given the potential for long-term viewer engagement, the platform should focus on producing high-quality TV shows (even if the number of shows are limited) that have the potential to become long-running series. This can build a dedicated audience base.
2. With actors like Anupam Kher and Shah Rukh Khan topping the list of cast, there's a clear interest in Indian cinema. Continue to produce and acquire Indian content. South Indian movies