Netflix Business Case Study

1. Defining Problem Statement and Analysing basic metrics

Prblem Statement : Analyse Netfilx OTT Platform Data,Derive which type of content can be produced to grow business in different countries

In [ ]:

```

```

Basic metrics

In [119]:

```
#import required libraries to work with
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
```

In [2]:

```
#read the data
Nt_df=pd.read_csv(r'C:\Users\lenovo\Downloads\netflixdata.csv')
Nt_df.head()
```

Out[2]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
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...
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...
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...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

1.1 we can see that in the year 2019 most number of movies were realeased by Netflix

In [4]:

```
#1.1 In which year most numbe of movies were released
print("Number of Movies Released")
Nt_df['date_added']=pd.to_datetime(Nt_df["date_added"])   #convert the date type object
Nt_df["date_added"].dt.year.value_counts()
```

```
Number of Movies Released
```

Out[4]:

```
2019.0    2016
2020.0    1879
2018.0    1649
2021.0    1498
2017.0    1188
2016.0     429
2015.0      82
2014.0      24
2011.0      13
2013.0      11
2012.0       3
2009.0       2
2008.0       2
2010.0       1
Name: date_added, dtype: int64
```

1.2. Total number of movies Vs Tv Shows released by Netflix

In [5]:

```
#NUMBER OF MOVIES AND TV SHOWS RELEASED
Nt_df["type"].value_counts()
```

Out[5]:

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

1.3 which genre movies are poduced most

In [6]:

```
Nt_df.groupby(["listed_in"])["show_id"].count().sort_values(ascending=False)
```

Out[6]:

```
listed_in
Dramas, International Movies                                 362
Documentaries                                                359
Stand-Up Comedy                                              334
Comedies, Dramas, International Movies                       274
Dramas, Independent Movies, International Movies             252
                                                            ... 
Cult Movies, Dramas, International Movies                      1
Cult Movies, Dramas, Music & Musicals                          1
Cult Movies, Dramas, Thrillers                                 1
Cult Movies, Horror Movies, Thrillers                          1
Crime TV Shows, TV Action & Adventure, TV Sci-Fi & Fantasy     1
Name: show_id, Length: 514, dtype: int64
```

1.4 What are the different ratings assigned

In [8]:

```
Nt_df.groupby(['rating'])["show_id"].count().sort_values(ascending=False)
```

Out[8]:

```
rating
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         1
84 min         1
66 min         1
Name: show_id, dtype: int64
```

1.5 Movie releaase year

In [91]:

```
Nt_df['release_year'].unique()
```

Out[91]:

```
array([2020, 2021, 1993, 2018, 1996, 1998, 1997, 2010, 2013, 2017, 1975,
       1978, 1983, 1987, 2012, 2001, 2014, 2002, 2003, 2004, 2011, 2008,
       2009, 2007, 2005, 2006, 1994, 2015, 2019, 2016, 1982, 1989, 1990,
       1991, 1999, 1986, 1992, 1984, 1980, 1961, 2000, 1995, 1985, 1976,
       1959, 1988, 1981, 1972, 1964, 1945, 1954, 1979, 1958, 1956, 1963,
       1970, 1973, 1925, 1974, 1960, 1966, 1971, 1962, 1969, 1977, 1967,
       1968, 1965, 1946, 1942, 1955, 1944, 1947, 1943], dtype=int64)
```

2. Statistical Summary and Basic infomation about the data

2.1 Shape of the data

In [9]:

```
Nt_df.shape
```

Out[9]:

```
(8807, 12)
```

2.2 Infomation about the data

In [10]:

```
Nt_df.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   datetime64[ns]
 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: datetime64[ns](1), int64(1), object(10)
memory usage: 825.8+ KB
```

2.3 Missing value detection

In [11]:

```
Nt_df.isna().sum()
```

Out[11]:

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

2.4 Conversion of Categoriacal attributes to category

In [15]:

```
Nt_df['type'] = Nt_df['type'].astype('category')
Nt_df['rating'] = Nt_df['rating'].astype('category')
Nt_df['country'] = Nt_df['country'].astype('category')
Nt_df['listed_in'] = Nt_df['listed_in'].astype('category')
Nt_df.dtypes
```

Out[15]:

```
show_id                 object
type                  category
title                   object
director                object
cast                    object
country               category
date_added      datetime64[ns]
release_year             int64
rating                category
duration                object
listed_in             category
description             object
dtype: object
```

2.5 Statistical summary of the dataset

In [22]:

```
Nt_df.describe(include='all')
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_11948\3512536337.py:1: FutureWarning: Treating datetime data as
categorical rather than numeric in `.describe` is deprecated and will be removed in a future version of pand
as. Specify `datetime_is_numeric=True` to silence this warning and adopt the future behavior now.
  Nt_df.describe(include='all')
```

Out[22]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
count	8807	8807	8807	6173	7982	7976	8797	8807.000000	8803	8804	8807	8807
unique	8807	2	8807	4528	7692	748	1714	NaN	17	220	514	8775
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	2020-01-01 00:00:00	NaN	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	1	19	19	2818	110	NaN	3207	1793	362	4
first	NaN	NaN	NaN	NaN	NaN	NaN	2008-01-01 00:00:00	NaN	NaN	NaN	NaN	NaN
last	NaN	NaN	NaN	NaN	NaN	NaN	2021-09-25 00:00:00	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.180198	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.819312	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.000000	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.000000	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2017.000000	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2019.000000	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021.000000	NaN	NaN	NaN	NaN

3. Non-Graphical Analysis: Value counts and unique attributes

3.1 Value counts for type cloumn

In [23]:

```
Nt_df['type'].value_counts()
```

Out[23]:

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

3.2 Value counts for rating column

**we can observe that Mostly TV-MA (Mature-Audience)content is produced

In [24]:

```
Nt_df['rating'].value_counts()
```

Out[24]:

```
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

3.3 Value counts for country column

United States tops the list of countries with Highest content on the plot form followed by india

In [27]:

```
Nt_df['country'].value_counts().sort_values(ascending=False)
```

Out[27]:

```
United States                                        2818
India                                                 972
United Kingdom                                        419
Japan                                                 245
South Korea                                           199
                                                     ... 
United Kingdom, Spain, Belgium                          1
United Kingdom, Spain                                   1
United Kingdom, South Africa, France                    1
United Kingdom, South Africa, Australia, United States  1
Zimbabwe                                                1
Name: country, Length: 748, dtype: int64
```

3.4 Value counts for director column

Rajiv Chilaka is the most popular director

In [29]:

```
Nt_df['director'].value_counts()
```

Out[29]:

```
Rajiv Chilaka                  19
Raúl Campos, Jan Suter         18
Marcus Raboy                   16
Suhas Kadav                    16
Jay Karas                      14
                               ..
Raymie Muzquiz, Stu Livingston  1
Joe Menendez                    1
Eric Bross                      1
Will Eisenberg                  1
Mozez Singh                     1
Name: director, Length: 4528, dtype: int64
```

3.5 Unique directors

In [30]:

```
Nt_df['director'].unique()
```

Out[30]:

```
array(['Kirsten Johnson', nan, 'Julien Leclercq', ..., 'Majid Al Ansari',
       'Peter Hewitt', 'Mozez Singh'], dtype=object)
```

3.6 Unique countries

In [31]:

```
Nt_df['country'].unique()
```

Out[31]:

```
['United States', 'South Africa', NaN, 'India', 'United States, Ghana, Burkina Faso, United Ki..., ..., 'Rus
sia, Spain', 'Croatia, Slovenia, Serbia, Montenegro', 'Japan, Canada', 'United States, France, South Korea,
Indonesia', 'United Arab Emirates, Jordan']
Length: 749
Categories (748, object): [', France, Algeria', ', South Korea', 'Argentina', 'Argentina, Brazil, France, Po
land, Germany, D..., ..., 'Venezuela, Colombia', 'Vietnam', 'West Germany', 'Zimbabwe']
```

3.7 unique Ratings

In [32]:

```
Nt_df['rating'].unique()
```

Out[32]:

```
['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', ..., '66 min', 'NR', NaN, 'TV-Y7-FV', 'UR']
Length: 18
Categories (17, object): ['66 min', '74 min', '84 min', 'G', ..., 'TV-Y', 'TV-Y7', 'TV-Y7-FV', 'UR']
```

4. Visual Analysis - Univariate, Bivariate after pre-processing of the data

4.1 Pre-Processing of the data

In [42]:

```
# convert the column values to string and split them based on the comma sepeerator,use explode function to un-nest the da
Nt_df['cast'] = Nt_df['cast'].str.split(',').explode('cast')
Nt_df['director'] = Nt_df['director'].str.split(',').explode('director')
Nt_df['country'] = Nt_df['country'].str.split(',').explode('country')
Nt_df['listed_in']=Nt_df['listed_in'].str.split(',').explode('listed_in')
#Reset the index after unnesting
Nt_df.reset_index(drop=True, inplace=True)
Nt_df.head(5)
```

Out[42]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Khosi Ngema	NaN	2021-09-24	2021	TV-MA	1 Season	TV Dramas	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	Gail Mabalane	NaN	2021-09-24	2021	TV-MA	1 Season	TV Mysteries	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Thabang Molaba	India	2021-09-24	2021	TV-MA	2 Seasons	Crime TV Shows	In a city of coaching centers known to train I...

In [69]:

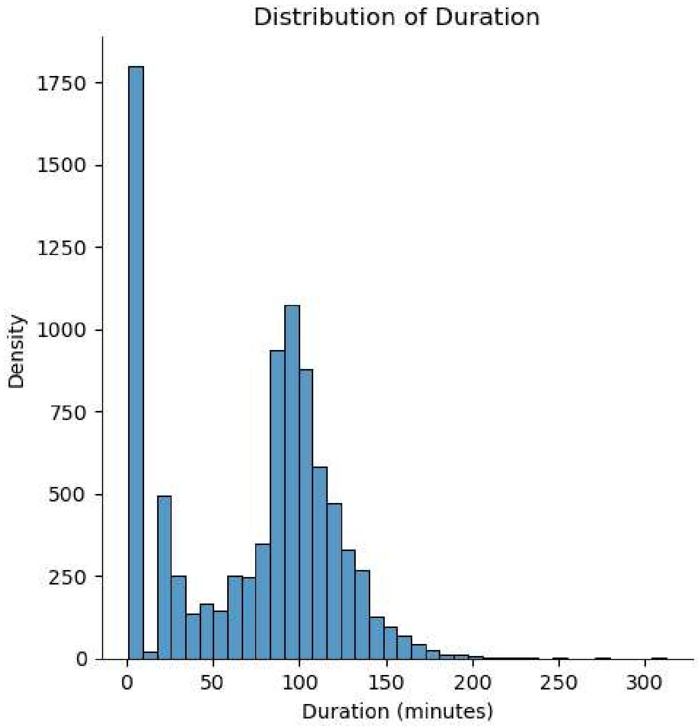
```
Nt_df[['duration_value', 'duration_unit']] = Nt_df['duration'].str.split(' ', 1, expand=True)
Nt_df['duration_value'] = pd.to_numeric(Nt_df['duration_value'], errors='coerce')
print(Nt_df[['duration', 'duration_value', 'duration_unit']])
Nt_df.loc[Nt_df['duration_unit'] == 'Seasons', 'duration_value'] *= 10
print(Nt_df[['duration', 'duration_value', 'duration_unit']].head(10))
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_11948\3062981337.py:1: FutureWarning: In a future version of pa
ndas all arguments of StringMethods.split except for the argument 'pat' will be keyword-only.
  Nt_df[['duration_value', 'duration_unit']] = Nt_df['duration'].str.split(' ', 1, expand=True)

       duration  duration_value duration_unit
0        90 min            90.0           min
1     2 Seasons             2.0       Seasons
2      1 Season             1.0        Season
3      1 Season             1.0        Season
4     2 Seasons             2.0       Seasons
...         ...             ...           ...
8802    158 min           158.0           min
8803  2 Seasons             2.0       Seasons
8804     88 min            88.0           min
8805     88 min            88.0           min
8806    111 min           111.0           min

[8807 rows x 3 columns]
    duration  duration_value duration_unit
0     90 min            90.0           min
1  2 Seasons            20.0       Seasons
2   1 Season             1.0        Season
3   1 Season             1.0        Season
4  2 Seasons            20.0       Seasons
5   1 Season             1.0        Season
6     91 min            91.0           min
7    125 min           125.0           min
8  9 Seasons            90.0       Seasons
9    104 min           104.0           min
```

4.2 # Univariate analysis - Distplot

In [70]:

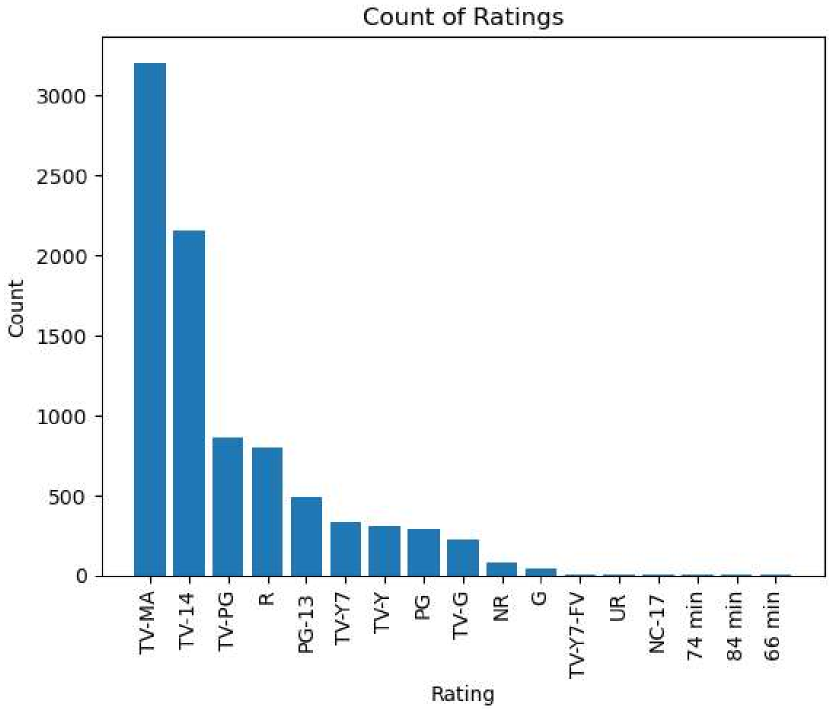
```
sns.displot(Nt_df['duration_value'].dropna())
plt.title('Distribution of Duration')
plt.xlabel('Duration (minutes)')
plt.ylabel('Density')
plt.show()
```



4.3 Univariate analysis - Countplot

In [71]:

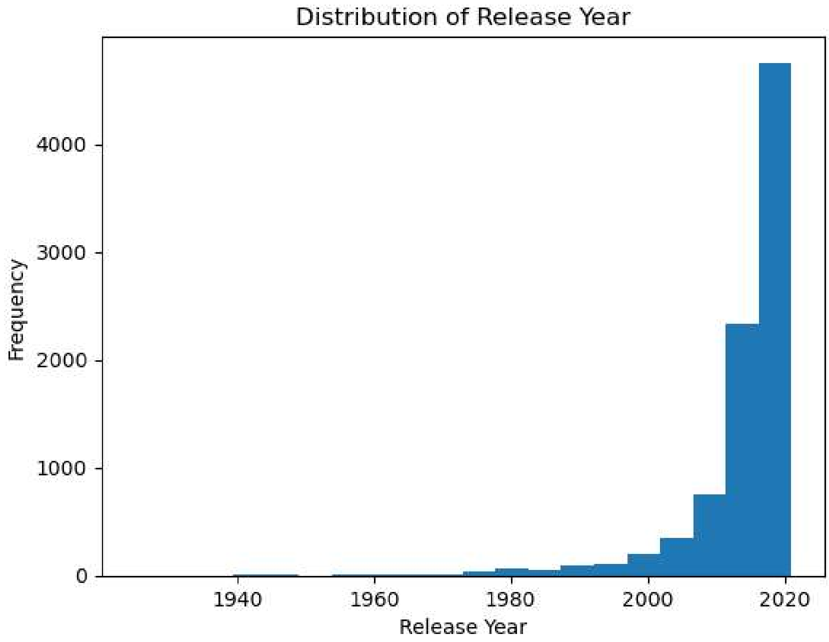
```
rating_counts = Nt_df['rating'].value_counts()
plt.bar(rating_counts.index, rating_counts.values)
plt.title('Count of Ratings')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



4.4 Univariate analysis - Histogram

In [72]:

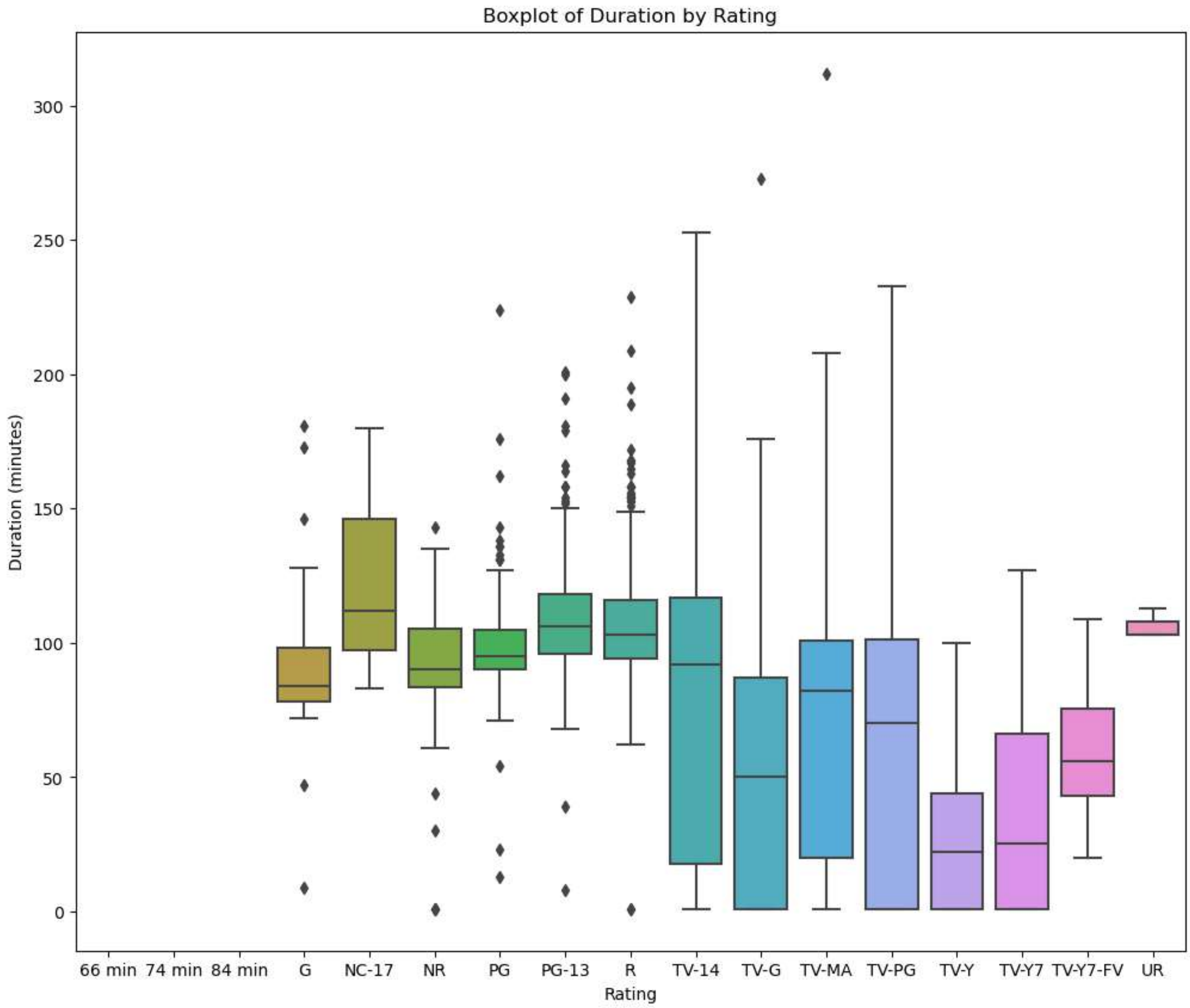
```
plt.hist(Nt_df['release_year'].dropna(), bins=20) # Drop missing values before plotting
plt.title('Distribution of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.show()
```



4.3 For categorical variable(s): Boxplot

In [75]:

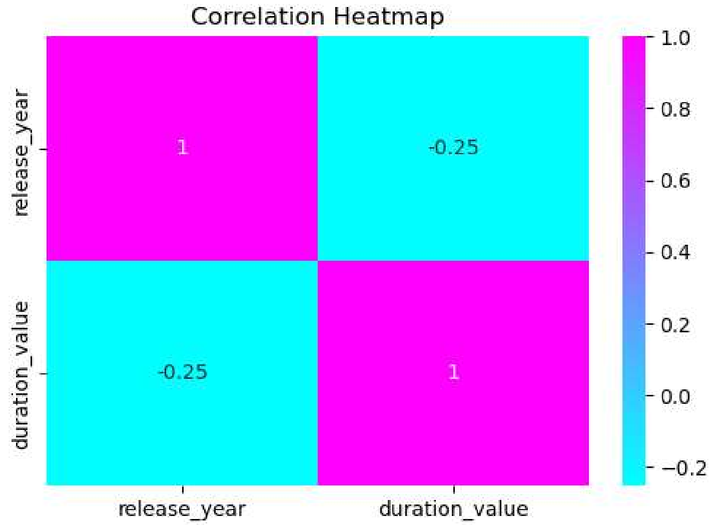
```
# Boxplot
plt.figure(figsize = (12,10))
sns.boxplot(x='rating', y='duration_value', data=Nt_df)
plt.title('Boxplot of Duration by Rating')
plt.xlabel('Rating')
plt.ylabel('Duration (minutes)')
plt.show()
```



4.4 For correlation: Heatmaps, Pairplots

In [83]:

```
correlation_matrix = Nt_df.corr(numeric_only =True)
plt.figure(figsize=(6, 4))
sns.heatmap(correlation_matrix, annot=True, cmap='cool')
plt.title('Correlation Heatmap')
plt.show()
```

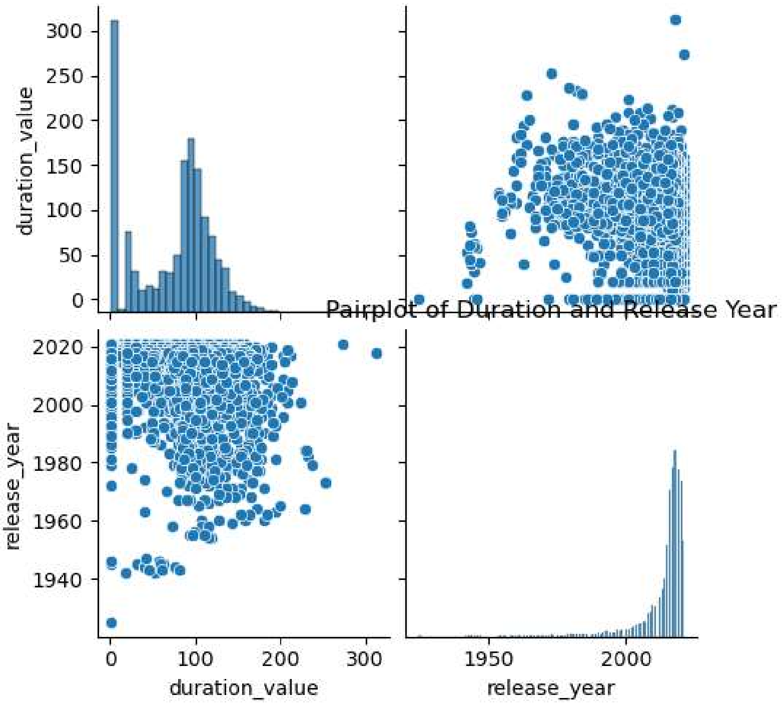


4.5 pair plot

In [85]:

```
plt.figure(figsize = (16, 14))
sns.pairplot(Nt_df, vars=['duration_value', 'release_year'])
plt.title('Pairplot of Duration and Release Year')
plt.show()
```

```
<Figure size 1600x1400 with 0 Axes>
```



5. Missing Value & Outlier check

5.1 Missing values filling

In [87]:

```
Nt_df['director'].fillna('unknown_director',inplace=True)
Nt_df['country'].fillna('unknown_country',inplace=True)
Nt_df['cast'].fillna('unknown_cast',inplace=True)
Nt_df.isnull().sum()
```

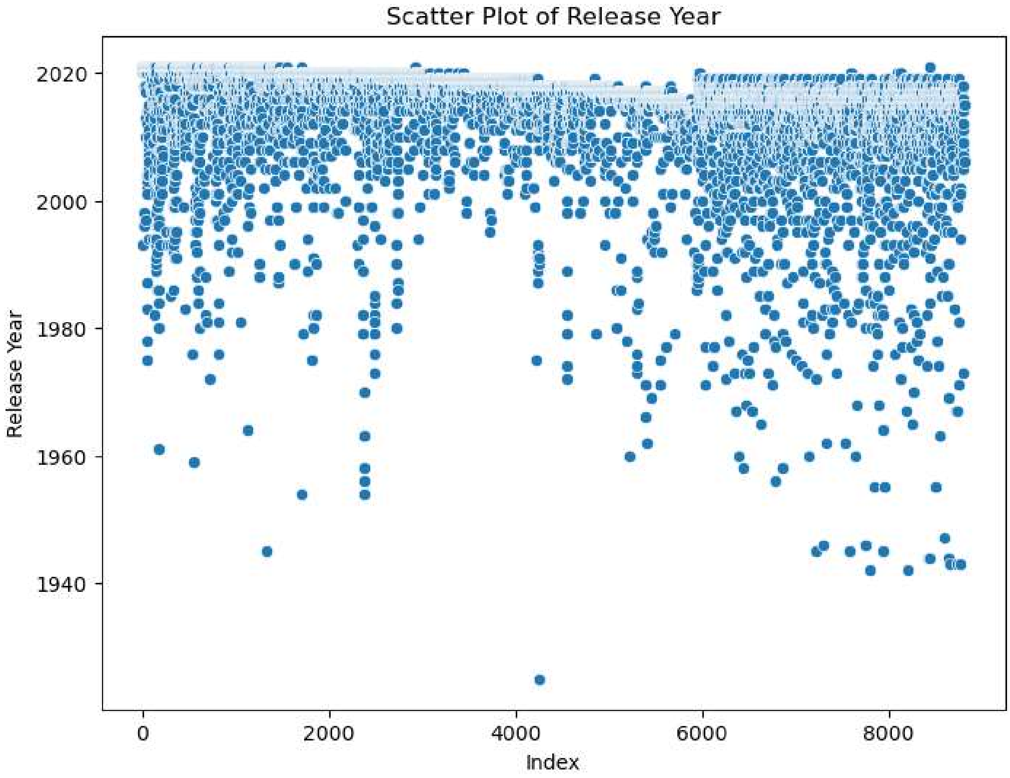
Out[87]:

```
show_id            0
type               0
title              0
director           0
cast               0
country            0
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
duration_value     3
duration_unit      3
dtype: int64
```

5.2 Outlier check

In [106]:

```
# Select the numerical columns for outlier check
numerical_columns = ['release_year', 'duration']
# Create scatter plots to visualize outliers
plt.figure(figsize=(8, 6))
sns.scatterplot(x=Nt_df.index, y=Nt_df['release_year'])
plt.title('Scatter Plot of Release Year')
plt.xlabel('Index')
plt.ylabel('Release Year')
plt.show()
```



6. Insights based on Non-Graphical and Visual Analysis

6.1 Comments on the range of attributes

```
The Type attribute is a categorical measure which defines weather a perticula record belongs to Movie or a TV-Show, the
country attribute provides infomation about the countries where the content was released,most content was released in
United states followed by india,date_added attribute defines from which year the conent was added to Netflix ,as per the
data first conent was added in the year  2010, the rating attibute defines the rating for a peticular content based on
the genre like R (restricted),Tv-MA(mature adult content)etc...
```

6.2 Comments on the distribution of the variables and relationship between them

In [ ]:

```
The 'release_year' variable appears to have a relatively uniform distribution across the range of years, suggesting a rel
movie and TV show releases over time. The 'duration' variable shows a distribution with multiple peaks, indicating the pr
runtime categories in the dataset. Some content may have shorter durations, while others may have longer durations. The '
exhibits different categories with varying frequencies, suggesting that certain ratings are more prevalent in the dataset
```

6.3 Comments for each univariate and bivariate plot

In [ ]:

```
The distplot of 'release_year' shows the frequency distribution of movie and TV show releases over the years. It provides
overall trend and concentration of releases in specific time periods. The distplot of 'duration' reveals the distribution
allowing us to identify the most common runtime categories and assess the overall spread of durations. The countplot of '
frequency of each rating category in the dataset, indicating the popularity and prevalence of different content ratings. 
'release_year' and 'duration' helps visualize the relationship between these variables.
```

7. Business Insights

In [ ]:

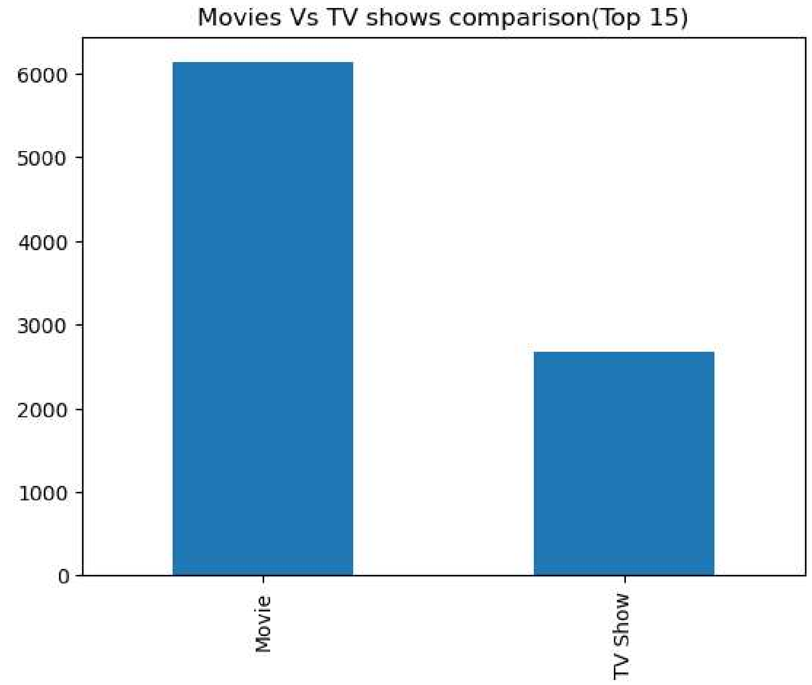
```
1. The Summary  of the data set shows Netflixis currently focusing  more on the Movies than Tv_shows with the count of mo
6131 and Tv shows 2676, which shows that nearly 70% of the content in Netflix is about movies,Netflix showing less inter
est in Tv_shows content.
```

In [110]:

```
Nt_df['type'].value_counts().plot(kind='bar')
plt.title('Movies Vs TV shows comparison')
```

Out[110]:

```
Text(0.5, 1.0, 'Movies Vs TV shows comparison(Top 15)')
```



In [ ]:

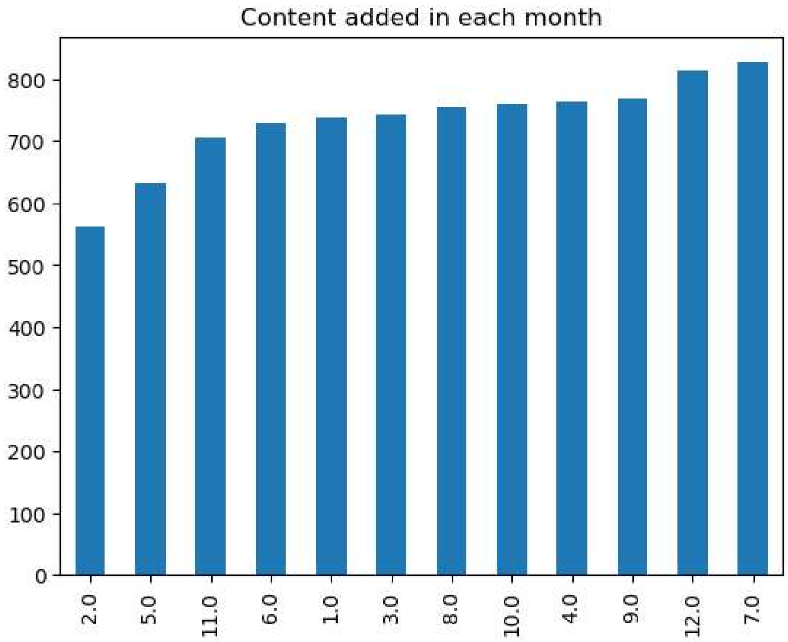
```
2. looking at the bar graph we can say that in months of December and july more number of content was
released into the plotform,also it is observed that Netflix adds more new Content during second half of the year from jun
to December.
```

In [105]:

```
Nt_df["date_added"].dt.month.value_counts().sort_values(ascending=True).plot(kind='bar')
plt.title('Content added in each Month')
```

Out[105]:

```
Text(0.5, 1.0, 'Content added in each month')
```



In [ ]:

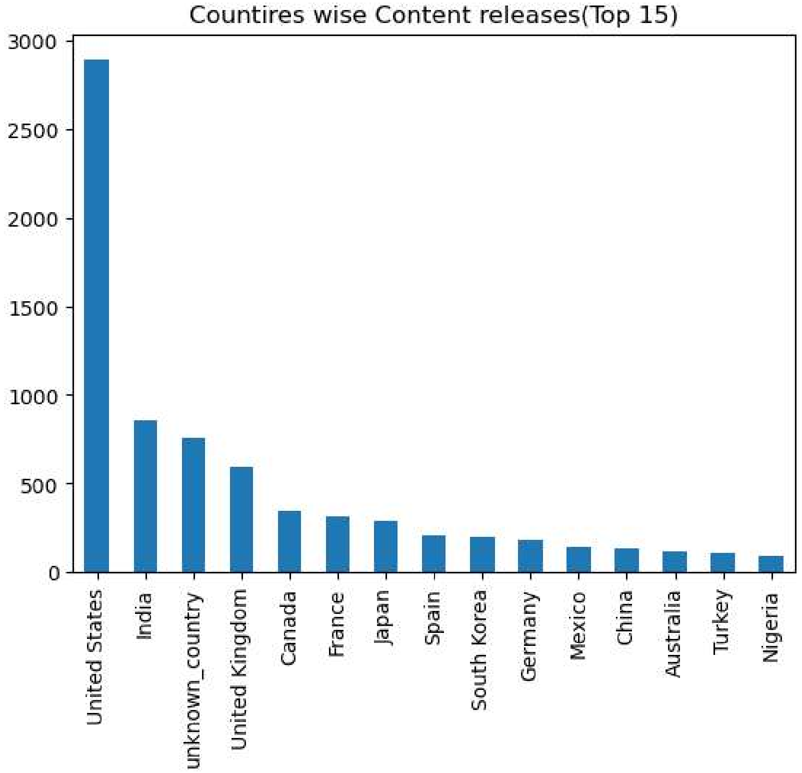
```
3.looking at the bar graoh from the countries wise comparison United stated stands at top as U.S is the primary
Contnet contributor and india stands at the 2nd position.
```

In [111]:

```
Nt_df["country"].value_counts().head(15).plot(kind='bar')
plt.title('Countires wise Content releases(Top 15)')
```

Out[111]:

```
Text(0.5, 1.0, 'Countires wise Content releases(Top 15)')
```



In [ ]:

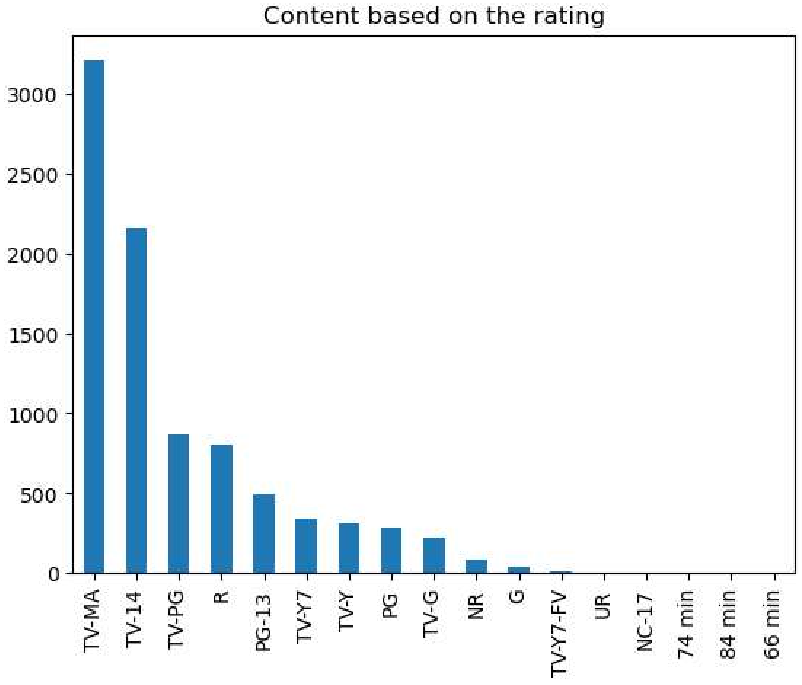
```
4.looking at the rating bar grapgh it is clear that most of  the   content is relatedto  TV_MA rating so the taget audien
for Netflix is Teenagers and adults
```

In [151]:

```
Nt_df["rating"].value_counts().plot(kind='bar')
plt.title('Content based on the rating')
```

Out[151]:

```
Text(0.5, 1.0, 'Content based on the rating')
```

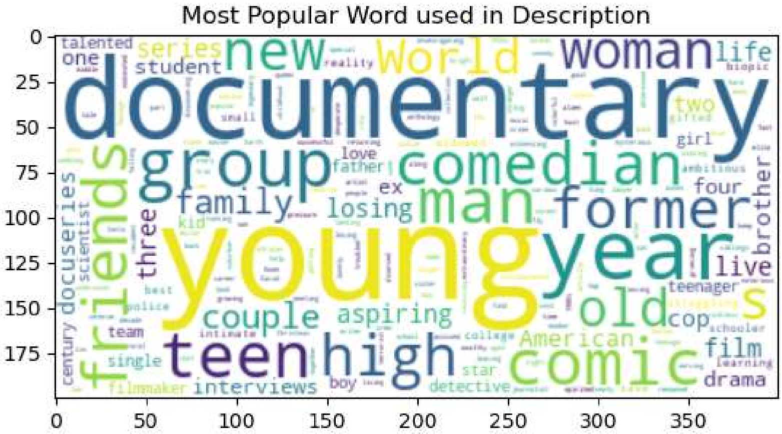


In [ ]:

```
5.the wod cloud for the desciption is drawn and young,Documentory,Group,family are most
popular words
```

In [128]:

```
text = " ".join(cat.split()[1] for cat in Nt_df.description)
word_cloud = WordCloud(collocations = False, background_color = 'white').generate(text)
plt.imshow(word_cloud, interpolation='bilinear')
plt.title('Most Popular Word used in Description')
plt.show()
```



8. Actionable items for business.

In [ ]:

```
The data shows that movies with TV_MA  ratings were watched more by users across all countries . This indicates a potentia
market opportunity in India. Netflix could consider focusing on acquiring and producing more such content to acquire more
subsriptions from india
```

In [ ]:

```
Since the data includes ratings from 110 countries it is clear  that Netflix has a diverse user base,NetFlix  can focus o
content localization by providing subtitles or dubbing options in multiple languages. This can enhance the user experienc
and attract a wider range of subsriptions
```

In [153]:

```
Nt_df['country'].nunique()
```

Out[153]:

```
110
```