Business_Case_Netflix_Data_Exploration_and_Visualisation

Modules

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly
import plotly.express as px
import plotly.graph_objects as go
import os
```

Read Data

In [2]:

```
# Read Data
data = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv")
```

In [3]:

```
# original copy of data
original_data=data.copy()
```

In [4]:

```
data.head()
```

Out[4]:

	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...
					Mayur More,							

EDA

Overview

Data Types of all the Attributes

- Among 12 Attributes only 1 Attribute is type interger that is release_year and all other attributes are type object

In [5]:

```
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
```

In [6]:

```
# View the datatype of each column in the dataset
data.dtypes
```

Out[6]:

```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

Missing Value

- director,cast,country and date_added has missing value respectively

  director-2634 cast-825 country-831 date_added-10 rating-4 duration-3

In [7]:

```
data.isnull().sum()
```

Out[7]:

```
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
```

Summary Statistics of Data

In [8]:

```
# Summary Statistics of the data
# It returned descriptive statistics of all Numerical type attribute release_year
data.describe()
```

Out[8]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

In [9]:

```
#include object gives the summary statistics of the character attributes
data.describe(include=['object'])
```

Out[9]:

	show_id	type	title	director	cast	country	date_added	rating	duration	listed_in	description
count	8807	8807	8807	6173	7982	7976	8797	8803	8804	8807	8807
unique	8807	2	8807	4528	7692	748	1767	17	220	514	8775
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	1	19	19	2818	109	3207	1793	362	4

Clean Data

- Clean Missing value data

In [10]:

```
data.isnull().any()
```

Out[10]:

```
show_id         False
type            False
title           False
director         True
cast             True
country          True
date_added       True
release_year    False
rating           True
duration         True
listed_in       False
description     False
dtype: bool
```

In [11]:

```
data.isnull().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
```

- Attributes date_added,rating and duration has very small number of missing value 10,4 and 3 respectively,so we can directly drop it *

In [12]:

```
data.dropna(subset=['rating', 'duration','date_added'],inplace=True)
```

- Attributes director, cast, and country has missing 2634, 825 and 831 respectively, we can not directly drop it because it will loss of data so we can fill this value with 'UNKNOWN' Entry *

In [13]:

```
data['director'].fillna("UNKNOWN",inplace=True)
data['cast'].fillna("UNKNOWN",inplace=True)
data['country'].fillna("UNKNOWN",inplace=True)
```

In [14]:

```
# No missing value in data
data.isnull().any()
```

Out[14]:

```
show_id         False
type            False
title           False
director        False
cast            False
country         False
date_added      False
release_year    False
rating          False
duration        False
listed_in       False
description     False
dtype: bool
```

Column - show_id

Points

- show_id is just a unique show label across each records (Each Record is either a Movie or Tv show).
- show_id is not having any missing values.

In [15]:

```
# Unique Values
data["show_id"].unique()
```

Out[15]:

```
array(['s1', 's2', 's3', ..., 's8805', 's8806', 's8807'], dtype=object)
```

In [16]:

```
# Total Unique Values
len(data["show_id"].unique())
```

Out[16]:

```
8790
```

In [17]:

```
# Detect duplicates
data["show_id"].duplicated().any()
```

Out[17]:

```
False
```

In [18]:

```
# Detect missing Values
data["show_id"].isna().any()
```

Out[18]:

```
False
```

Column - type

Points

- type has only two unique Values - Movie (representing movie record) and TV Show (representing TV show)
- type does not have any missing values.
- type has discrete categories Movie and Tv Show so it is categorical attribut but here it is present as type object so convert it into type category

In [19]:

```
# Unique Values
data["type"].unique()
```

Out[19]:

```
array(['Movie', 'TV Show'], dtype=object)
```

In [20]:

```
# Detect missing Values
data["type"].isna().any()
```

Out[20]:

```
False
```

In [21]:

```
#attribute type converted into category
data["type"] = data["type"].astype('category')
```

Column - title

Points

- title is not having missing values
- title is not having any duplicated values.

In [22]:

```
# Unique Values
data["title"].unique()
```

Out[22]:

```
array(['Dick Johnson Is Dead', 'Blood & Water', 'Ganglands', ...,
       'Zombieland', 'Zoom', 'Zubaan'], dtype=object)
```

In [23]:

```
# Detect duplicates
data["title"].duplicated().any()
```

Out[23]:

```
False
```

In [24]:

```
# Detect missing Values
data["title"].isna().any()
```

Out[24]:

```
False
```

column - director

Points

- director may have missing values but we fill with UNKNOWN
- director may have duplicated values

In [25]:

```
# Unique Values
data["director"].unique()
```

Out[25]:

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

In [26]:

```
# Detect duplicates
data["director"].duplicated().any()
```

Out[26]:

```
True
```

In [27]:

```
# Detect missing Values
data["director"].isna().any()
```

Out[27]:

```
False
```

Column - cast

Points

- Case may have multiple values (comma sepated) in a record
- Cast may have missing values but we fill with UNKNOWN
- Cast may have duplicates.

In [28]:

```
# Unique Values
data["cast"].unique()
```

Out[28]:

```
array(['UNKNOWN',
       'Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, X
olile Tshabalala, Getmore Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube, Odwa Gwa
nya, Mekaila Mathys, Sandi Schultz, Duane Williams, Shamilla Miller, Patrick Mofokeng',
       'Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia Lesaffre, Salim Kechiouche, Noureddine Far
ihi, Geert Van Rampelberg, Bakary Diombera',
       ...,
       'Jesse Eisenberg, Woody Harrelson, Emma Stone, Abigail Breslin, Amber Heard, Bill Murray, Derek Graf',
       'Tim Allen, Courteney Cox, Chevy Chase, Kate Mara, Ryan Newman, Michael Cassidy, Spencer Breslin, Rip Tor
n, Kevin Zegers',
       'Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, Malkeet Rauni, Anita Sha
bdish, Chittaranjan Tripathy'],
      dtype=object)
```

In [29]:

```
# Detect duplicates
data["cast"].duplicated().any()
```

Out[29]:

```
True
```

In [30]:

```
# Detect missing Values
data["cast"].isna().any()
```

Out[30]:

```
False
```

Column - Country

Points

- Country can have multiple values - comma separated in a record
- Country can have duplicates
- Country can have missing values but we fill with UNKNOWN

In [31]:

```
# Unique Values
data["country"].unique()
```

Out[31]:

```
array(['United States', 'South Africa', 'UNKNOWN', 'India',
       'United States, Ghana, Burkina Faso, United Kingdom, Germany, Ethiopia',
       'United Kingdom', 'Germany, Czech Republic', 'Mexico', 'Turkey',
       'Australia', 'United States, India, France', 'Finland',
       'China, Canada, United States',
       'South Africa, United States, Japan', 'Nigeria', 'Japan',
       'Spain, United States', 'France', 'Belgium',
       'United Kingdom, United States', 'United States, United Kingdom',
       'France, United States', 'South Korea', 'Spain',
       'United States, Singapore', 'United Kingdom, Australia, France',
       'United Kingdom, Australia, France, United States',
       'United States, Canada', 'Germany, United States',
       'South Africa, United States', 'United States, Mexico',
       'United States, Italy, France, Japan',
       'United States, Italy, Romania, United Kingdom',
       'Australia, United States', 'Argentina, Venezuela',
       'United States, United Kingdom, Canada', 'China, Hong Kong',
       'Russia', 'Canada', 'Hong Kong', 'United States, China, Hong Kong',
```

In [32]:

```
# Detect duplicates
data["country"].duplicated().any()
```

Out[32]:

```
True
```

In [33]:

```
# Detect missing Values
data["country"].isna().any()
```

Out[33]:

```
False
```

Column - date_added

Points

- date_added represents the date when movie/tv show is added in Netflix, it might be different from release year.
- date_added format is MONTH DAY, YEAR
- date_added may have duplicates
- date_added may have missing values but we drop it because it is very small number.

In [34]:

```
# Unique Values
data["date_added"].unique()
```

Out[34]:

```
array(['September 25, 2021', 'September 24, 2021', 'September 23, 2021',
       ..., 'December 6, 2018', 'March 9, 2016', 'January 11, 2020'],
      dtype=object)
```

In [35]:

```
# Detect duplicates
data["date_added"].duplicated().any()
```

Out[35]:

```
True
```

In [36]:

```
# Detect missing Values
data["date_added"].isna().any()
```

Out[36]:

```
False
```

Column - release_year

Points

- release_year can have duplicates

In [37]:

```
data["release_year"].unique()
```

Out[37]:

```
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])
```

In [38]:

```
# Detect duplicates
data["release_year"].duplicated().any()
```

Out[38]:

```
True
```

In [39]:

```
# Detect missing Values
data["release_year"].isna().any()
```

Out[39]:

```
False
```

Column - rating

Points

- Cleaning is required as some values does not make any sense
- It may have duplicates
- It may have missing values we drop it because it is very small in number

In [40]:

```
data["rating"].unique()
```

Out[40]:

```
array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', 'NR', 'TV-Y7-FV', 'UR'], dtype=object)
```

In [41]:

```
# Detect duplicates
data["rating"].duplicated().any()
```

Out[41]:

```
True
```

In [42]:

```
# Detect missing Values
data["rating"].isna().any()
```

Out[42]:

```
False
```

Column - duration

Points

- Cleaning is required as format keeps changing.
- It may have duplicates
- It may have missing values we drop it because very small in numbers

In [43]:

```
data["duration"].unique()
```

Out[43]:

```
array(['90 min', '2 Seasons', '1 Season', '91 min', '125 min',
       '9 Seasons', '104 min', '127 min', '4 Seasons', '67 min', '94 min',
       '5 Seasons', '161 min', '61 min', '166 min', '147 min', '103 min',
       '97 min', '106 min', '111 min', '3 Seasons', '110 min', '105 min',
       '96 min', '124 min', '116 min', '98 min', '23 min', '115 min',
       '122 min', '99 min', '88 min', '100 min', '6 Seasons', '102 min',
       '93 min', '95 min', '85 min', '83 min', '113 min', '13 min',
       '182 min', '48 min', '145 min', '87 min', '92 min', '80 min',
       '117 min', '128 min', '119 min', '143 min', '114 min', '118 min',
       '108 min', '63 min', '121 min', '142 min', '154 min', '120 min',
       '82 min', '109 min', '101 min', '86 min', '229 min', '76 min',
       '89 min', '156 min', '112 min', '107 min', '129 min', '135 min',
       '136 min', '165 min', '150 min', '133 min', '70 min', '84 min',
       '140 min', '78 min', '7 Seasons', '64 min', '59 min', '139 min',
       '69 min', '148 min', '189 min', '141 min', '130 min', '138 min',
       '81 min', '132 min', '10 Seasons', '123 min', '65 min', '68 min',
       '66 min', '62 min', '74 min', '131 min', '39 min', '46 min',
       '38 min', '8 Seasons', '17 Seasons', '126 min', '155 min',
       '159 min', '137 min', '12 min', '273 min', '36 min', '34 min',
       '77 min', '60 min', '49 min', '58 min', '72 min', '204 min',
       '212 min', '25 min', '73 min', '29 min', '47 min', '32 min',
       '35 min', '71 min', '149 min', '33 min', '15 min', '54 min',
       '224 min', '162 min', '37 min', '75 min', '79 min', '55 min',
       '158 min', '164 min', '173 min', '181 min', '185 min', '21 min',
       '24 min', '51 min', '151 min', '42 min', '22 min', '134 min',
       '177 min', '13 Seasons', '52 min', '14 min', '53 min', '8 min',
       '57 min', '28 min', '50 min', '9 min', '26 min', '45 min',
       '171 min', '27 min', '44 min', '146 min', '20 min', '157 min',
       '17 min', '203 min', '41 min', '30 min', '194 min', '15 Seasons',
       '233 min', '237 min', '230 min', '195 min', '253 min', '152 min',
       '190 min', '160 min', '208 min', '180 min', '144 min', '5 min',
       '174 min', '170 min', '192 min', '209 min', '187 min', '172 min',
       '16 min', '186 min', '11 min', '193 min', '176 min', '56 min',
       '169 min', '40 min', '10 min', '3 min', '168 min', '312 min',
       '153 min', '214 min', '31 min', '163 min', '19 min', '12 Seasons',
       '179 min', '11 Seasons', '43 min', '200 min', '196 min', '167 min',
       '178 min', '228 min', '18 min', '205 min', '201 min', '191 min'],
      dtype=object)
```

In [44]:

```
# Detect duplicates
data["duration"].duplicated().any()
```

Out[44]:

```
True
```

In [45]:

```
# Detect missing Values
data["duration"].isna().any()
```

Out[45]:

```
False
```

Column - listed_in

Points

- It is representing Genre.
- It can have multiple values in a Record (comma separated)
- It can have duplicates
- It can not have missing values

In [46]:

```
data["listed_in"].unique()
```

Out[46]:

```
array(['Documentaries', 'International TV Shows, TV Dramas, TV Mysteries',
       'Crime TV Shows, International TV Shows, TV Action & Adventure',
       'Docuseries, Reality TV',
       'International TV Shows, Romantic TV Shows, TV Comedies',
       'TV Dramas, TV Horror, TV Mysteries', 'Children & Family Movies',
       'Dramas, Independent Movies, International Movies',
       'British TV Shows, Reality TV', 'Comedies, Dramas',
       'Crime TV Shows, Docuseries, International TV Shows',
       'Dramas, International Movies',
       'Children & Family Movies, Comedies',
       'British TV Shows, Crime TV Shows, Docuseries',
       'TV Comedies, TV Dramas', 'Documentaries, International Movies',
       'Crime TV Shows, Spanish-Language TV Shows, TV Dramas',
       'Thrillers',
       'International TV Shows, Spanish-Language TV Shows, TV Action & Adventure',
       'International TV Shows, TV Action & Adventure, TV Dramas',
       'Comedies, International Movies',
       'Comedies, International Movies, Romantic Movies',
```

In [47]:

```
# Detect duplicates
data["listed_in"].duplicated().any()
```

Out[47]:

```
True
```

In [48]:

```
# Detect missing Values
data["listed_in"].isna().any()
```

Out[48]:

```
False
```

Column - description

Points

- It is representing Short Description.
- It can have duplicates
- It can not have missing values

In [49]:

```
data["description"].unique()
```

Out[49]:

```
array(['As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive and comi
cal ways to help them both face the inevitable.',
       'After crossing paths at a party, a Cape Town teen sets out to prove whether a private-school swimming sta
r is her sister who was abducted at birth.',
       'To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of robbers are p
ulled into a violent and deadly turf war.',
       ...,
       'Looking to survive in a world taken over by zombies, a dorky college student teams with an urban roughnec
k and a pair of grifter sisters.',
       'Dragged from civilian life, a former superhero must train a new crop of youthful saviors when the militar
y preps for an attack by a familiar villain.',
       "A scrappy but poor boy worms his way into a tycoon's dysfunctional family, while facing his fear of music
and the truth about his past."],
      dtype=object)
```

In [50]:

```
# Detect duplicates
data["description"].duplicated().any()
```

Out[50]:

```
True
```

In [51]:

```
# Detect missing Values
data["description"].isna().any()
```

Out[51]:

```
False
```

Cleaning and Defining the Dataset

Cleaning Country Column

In [52]:

```
def mapCountryToShows(df):
    result = []
    for index,row in df.iterrows():
        try:
            for country in row["country"].split(","):
                result.append([row["show_id"],country.strip()])
        except:
              pass

    res_df = pd.DataFrame(result, columns = ["show_id","country"])
    return res_df
```

In [53]:

```
shows_country_df = mapCountryToShows(original_data)
shows_country_df.head()
```

Out[53]:

	show_id	country
0	s1	United States
1	s2	South Africa
2	s5	India
3	s8	United States
4	s8	Ghana

Cleaning Cast Column

In [54]:

```
def mapCastToShows(df):
    result = []
    for index,row in df.iterrows():
        try:
            for country in row["cast"].split(","):
                result.append([row["show_id"],country.strip()])
        except:
            pass

    res_df = pd.DataFrame(result, columns = ["show_id","cast"])
    return res_df
```

In [55]:

```
shows_cast_df = mapCastToShows(original_data)
shows_cast_df.head()
```

Out[55]:

	show_id	cast
0	s2	Ama Qamata
1	s2	Khosi Ngema
2	s2	Gail Mabalane
3	s2	Thabang Molaba
4	s2	Dillon Windvogel

Cleaning Duration Column

Points

- In TV Show, duration is defined in terms of Seasons.
- In Movie Show, duration is defined in terms of mins.

In [56]:

```
data[data["type"]=="TV Show"]["duration"].unique()
```

Out[56]:

```
array(['2 Seasons', '1 Season', '9 Seasons', '4 Seasons', '5 Seasons',
       '3 Seasons', '6 Seasons', '7 Seasons', '10 Seasons', '8 Seasons',
       '17 Seasons', '13 Seasons', '15 Seasons', '12 Seasons',
       '11 Seasons'], dtype=object)
```

In [57]:

```
data[data["type"]=="Movie"]["duration"].unique()
```

Out[57]:

```
array(['90 min', '91 min', '125 min', '104 min', '127 min', '67 min',
       '94 min', '161 min', '61 min', '166 min', '147 min', '103 min',
       '97 min', '106 min', '111 min', '110 min', '105 min', '96 min',
       '124 min', '116 min', '98 min', '23 min', '115 min', '122 min',
       '99 min', '88 min', '100 min', '102 min', '93 min', '95 min',
       '85 min', '83 min', '113 min', '13 min', '182 min', '48 min',
       '145 min', '87 min', '92 min', '80 min', '117 min', '128 min',
       '119 min', '143 min', '114 min', '118 min', '108 min', '63 min',
       '121 min', '142 min', '154 min', '120 min', '82 min', '109 min',
       '101 min', '86 min', '229 min', '76 min', '89 min', '156 min',
       '112 min', '107 min', '129 min', '135 min', '136 min', '165 min',
       '150 min', '133 min', '70 min', '84 min', '140 min', '78 min',
       '64 min', '59 min', '139 min', '69 min', '148 min', '189 min',
       '141 min', '130 min', '138 min', '81 min', '132 min', '123 min',
       '65 min', '68 min', '66 min', '62 min', '74 min', '131 min',
       '39 min', '46 min', '38 min', '126 min', '155 min', '159 min',
       '137 min', '12 min', '273 min', '36 min', '34 min', '77 min',
       '60 min', '49 min', '58 min', '72 min', '204 min', '212 min',
       '25 min', '73 min', '29 min', '47 min', '32 min', '35 min',
       '71 min', '149 min', '33 min', '15 min', '54 min', '224 min',
       '162 min', '37 min', '75 min', '79 min', '55 min', '158 min',
       '164 min', '173 min', '181 min', '185 min', '21 min', '24 min',
       '51 min', '151 min', '42 min', '22 min', '134 min', '177 min',
       '52 min', '14 min', '53 min', '8 min', '57 min', '28 min',
       '50 min', '9 min', '26 min', '45 min', '171 min', '27 min',
       '44 min', '146 min', '20 min', '157 min', '17 min', '203 min',
       '41 min', '30 min', '194 min', '233 min', '237 min', '230 min',
       '195 min', '253 min', '152 min', '190 min', '160 min', '208 min',
       '180 min', '144 min', '5 min', '174 min', '170 min', '192 min',
       '209 min', '187 min', '172 min', '16 min', '186 min', '11 min',
       '193 min', '176 min', '56 min', '169 min', '40 min', '10 min',
       '3 min', '168 min', '312 min', '153 min', '214 min', '31 min',
       '163 min', '19 min', '179 min', '43 min', '200 min', '196 min',
       '167 min', '178 min', '228 min', '18 min', '205 min', '201 min',
       '191 min'], dtype=object)
```

In [58]:

```
def getNumber(input):
    result = None
    try:
        result = int(''.join(filter(str.isdigit, input)))
    except:
        result = np.NaN
    return result
```

In [59]:

```
data['duration'] = data['duration'].apply(getNumber)
```

In [60]:

```
data.head()
```

Out[60]:

	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	UNKNOWN	United States	September 25, 2021	2020	PG-13	90	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	UNKNOWN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	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...	UNKNOWN	September 24, 2021	2021	TV-MA	1	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	UNKNOWN	UNKNOWN	UNKNOWN	September 24, 2021	2021	TV-MA	1	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	UNKNOWN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

Cleaning Listed_in (Genre)

In [61]:

```
def mapListedInToShows(df):
    result = []
    for index,row in df.iterrows():
        try:
            for listed_in in row["listed_in"].split(","):
                result.append([row["show_id"],listed_in.strip()])
        except:
            pass

    res_df = pd.DataFrame(result, columns = ["show_id","listed_in"])
    return res_df
```

In [62]:

```
shows_listed_in_df = mapListedInToShows(data)
shows_listed_in_df.head()
```

Out[62]:

	show_id	listed_in
0	s1	Documentaries
1	s2	International TV Shows
2	s2	TV Dramas
3	s2	TV Mysteries
4	s3	Crime TV Shows

Cleaning date_added

In [63]:

```
def getMonth(s):
    result = None
    try:
        result = s.split(" ")[0].strip()
    except:
        result = np.NaN
    return result
```

In [64]:

```
def getYear(s):
    result = None
    try:
        result = s.split(" ")[-1].strip()
    except:
        result = np.NaN
    return result
```

In [65]:

```
data["date_added_month"] = data["date_added"].apply(getMonth)
data["date_added_year"] = data["date_added"].apply(getYear)
```

Visual Analysis

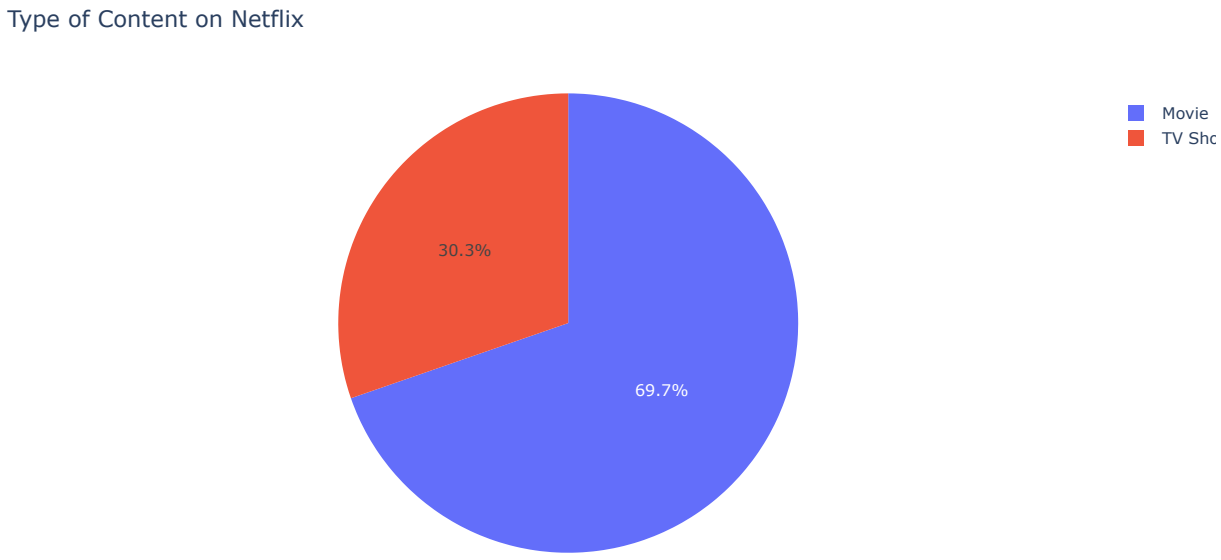
Type of Content in the NetFlix

Type of Content on Netflix

- 30.4 % TV Show and 69.6% Movie Content

In [66]:

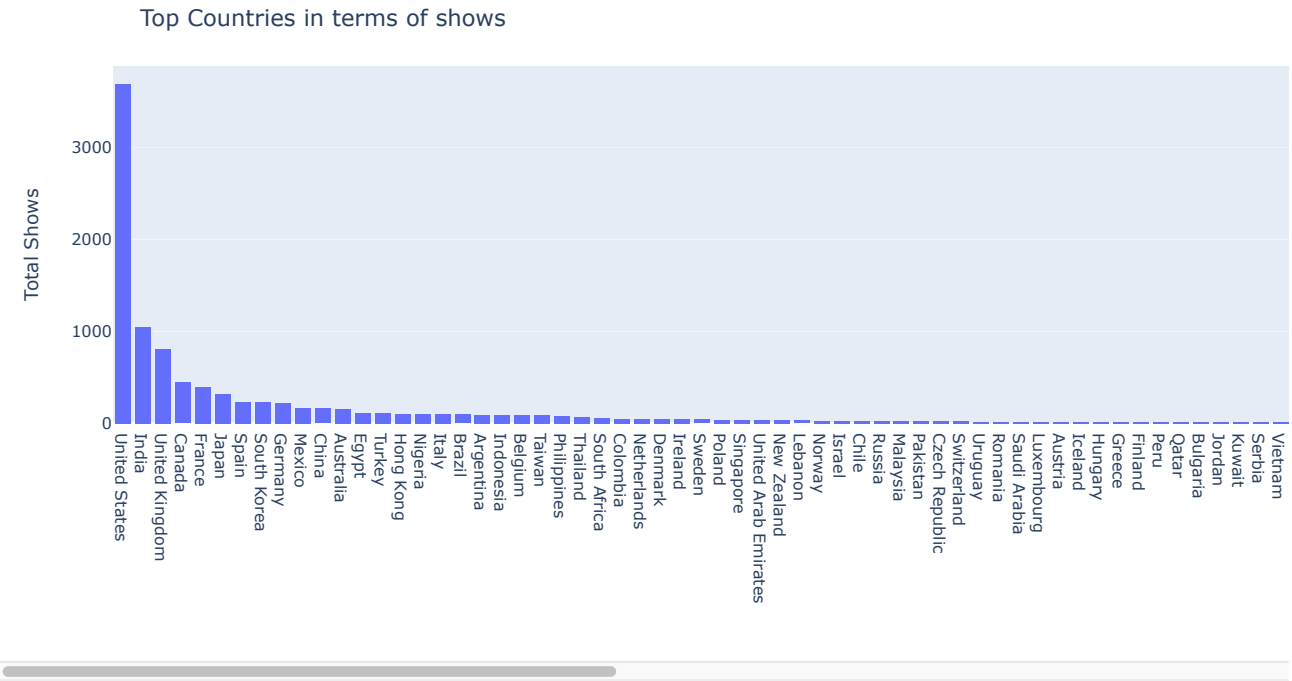
```
df_type_count=data['type'].value_counts().reset_index()
fig = px.pie(df_type_count, values='type', names='index', title='Type of Content on Netflix')
fig.show()
```



Top countries in terms of Shows Produced

In [67]:

```
# plotly-Histogram
fig = px.histogram(shows_country_df, x="country")
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.update_layout(
    autosize=False,
    width=2000,
    height=500,
    title = "Top Countries in terms of shows",

    xaxis=dict(
        title_text="Countries"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)
fig.show()
```



Duration Trend

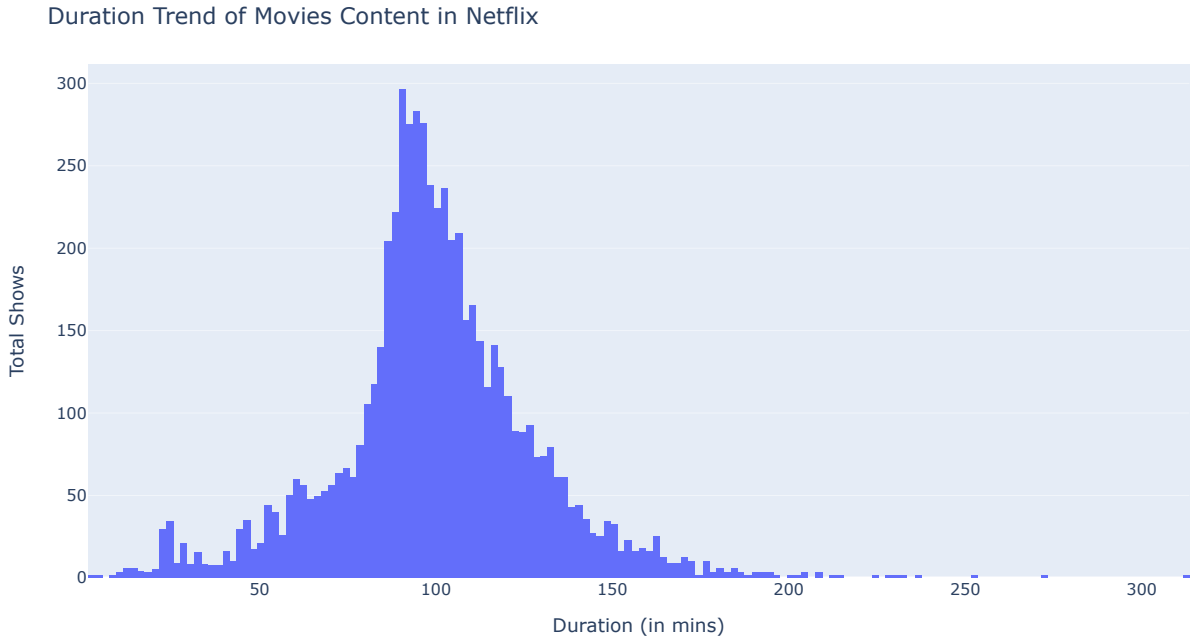
Movies Content Duration Trend

Points

- Mostly, movies duration are around 90 mins. Histogram is dense means that majority of movies are around 90 mins.

In [68]:

```
# plotly-Histogram
movies_duration_data = data[data["type"]=="Movie"][["show_id","duration"]]
fig = px.histogram(movies_duration_data, x="duration")
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.update_layout(

    title = "Duration Trend of Movies Content in Netflix",

    xaxis=dict(
        title_text="Duration (in mins)"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)
#fig.update_xaxes(range=[20,50])
fig.show()
```



In [69]:

```
# stats
movies_duration_data["duration"].describe()
```

Out[69]:

```
count    6126.000000
mean       99.584884
std        28.283225
min         3.000000
25%        87.000000
50%        98.000000
75%       114.000000
max       312.000000
Name: duration, dtype: float64
```

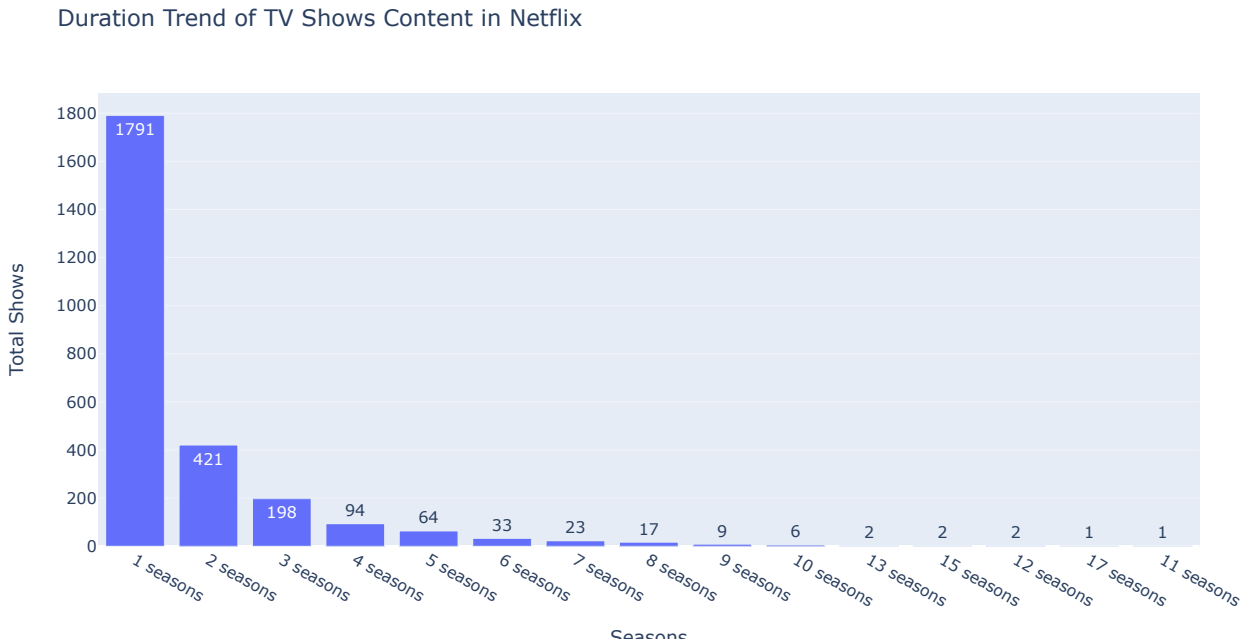
Tv Shows Content Duration Trend

Points

- Majority TV Shows with 1 Seasons are in Netflix having 1793 shows.

In [70]:

```
tv_show_duration_data = data[data["type"]=="TV Show"][["show_id","duration"]]
tv_show_duration_freq = tv_show_duration_data["duration"].value_counts()

val = list(tv_show_duration_freq.index)
x = list(map(int,val))
x = list(map(str,x))
x = list(map(lambda x: x + " seasons", x))

#x = list(tv_show_duration_freq.index)
y = list(tv_show_duration_freq)

# Use textposition='auto' for direct text
fig = go.Figure(data=[go.Bar(
            x=x, y=y,
            text=y,
            textposition='auto',
        )])
fig.update_layout(
    title = "Duration Trend of TV Shows Content in Netflix",

    xaxis=dict(
        title_text="Seasons"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)

fig.show()
```

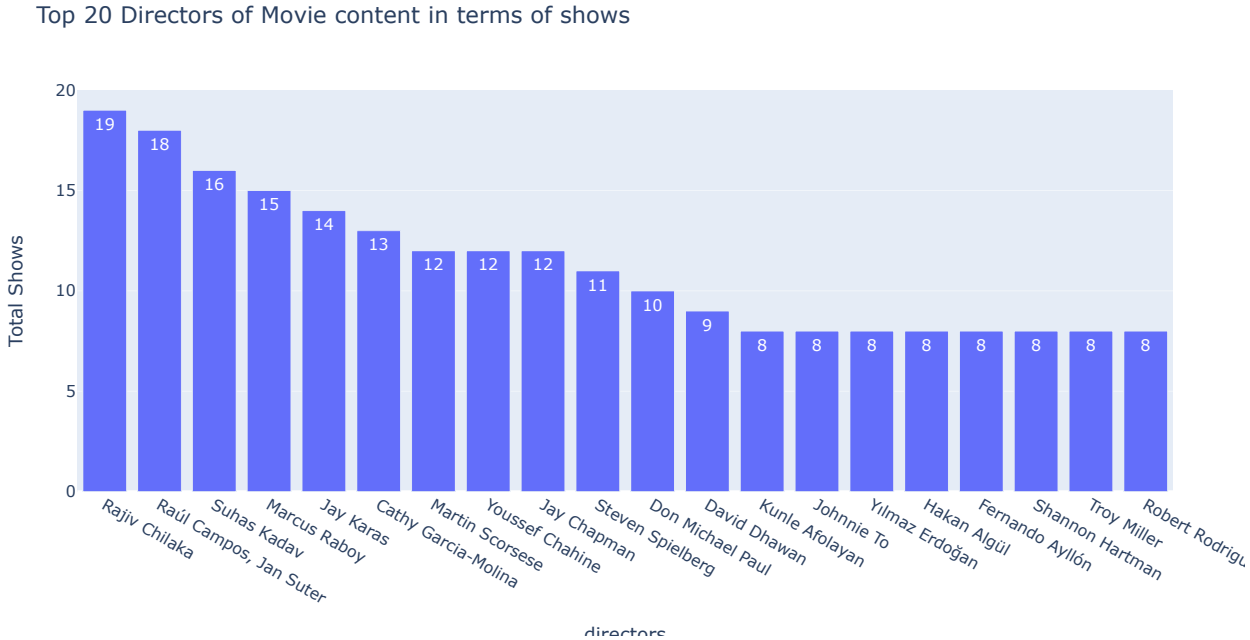


Top 20 Directors in terms of Shows/Movies Produced

Top 20 Directors in Movies Content in terms of Movies produced

In [71]:

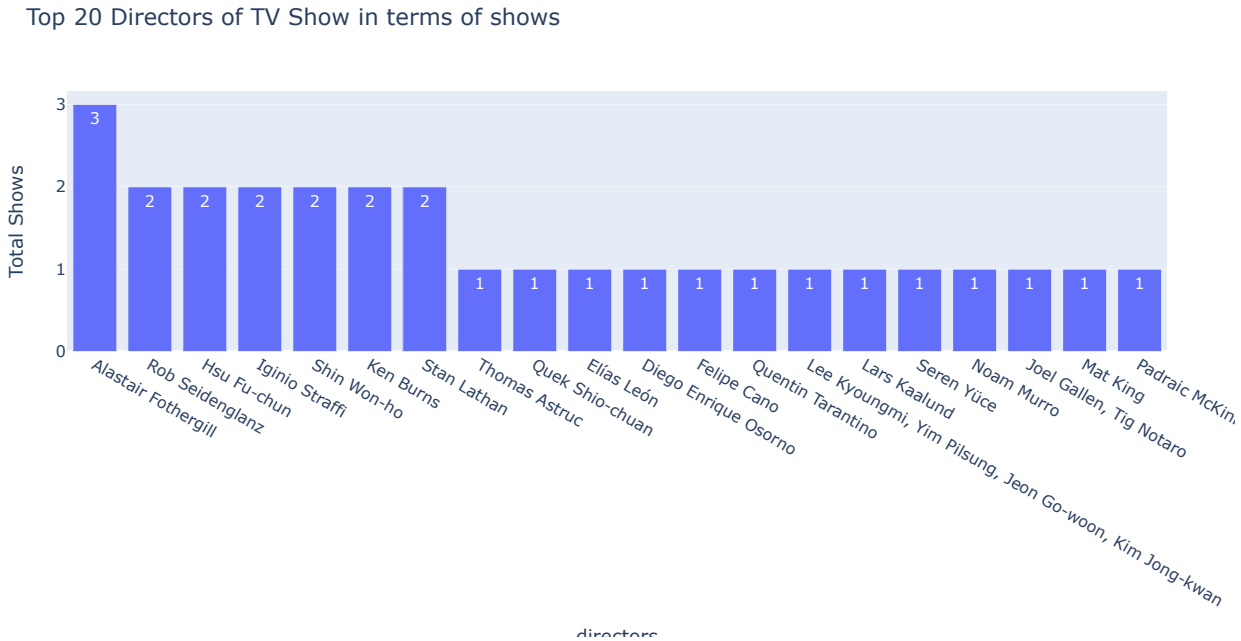
```
# Top 20 directors

movie_director_freq = original_data[original_data["type"]=="Movie"]["director"].value_counts()
x = list(movie_director_freq.index)[:20]
y = list(movie_director_freq)[:20]

# Use textposition='auto' for direct text
fig = go.Figure(data=[go.Bar(
            x=x, y=y,
            text=y,
            textposition='auto',
        )])
fig.update_layout(
    title = "Top 20 Directors of Movie content in terms of shows",

    xaxis=dict(
        title_text="directors"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)

fig.show()
```



Top 20 Directors in TV Shows Content in terms of Shows produced

In [72]:

```
# Top 20 directors

tv_show_director_freq = original_data[original_data["type"]=="TV Show"]["director"].value_counts()
x = list(tv_show_director_freq.index)[:20]
y = list(tv_show_director_freq)[:20]

# Use textposition='auto' for direct text
fig = go.Figure(data=[go.Bar(
            x=x, y=y,
            text=y,
            textposition='auto',
        )])
fig.update_layout(
    title = "Top 20 Directors of TV Show in terms of shows",

    xaxis=dict(
        title_text="directors"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)

fig.show()
```



Top Casts in terms of Shows/Movies Produced

In [73]:

```
combined_show_cast_data = pd.merge(original_data, shows_cast_df, on='show_id')
combined_show_cast_data.head()
```

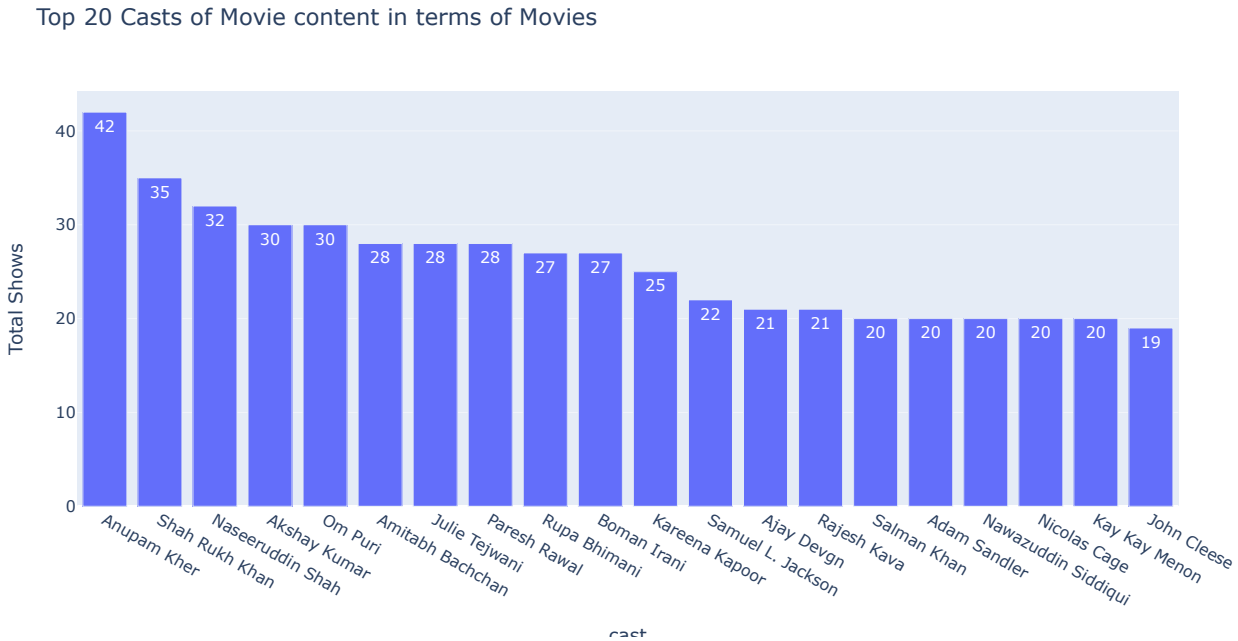
Out[73]:

	show_id	type	title	director	cast_x	country	date_added	release_year	rating	duration	listed_in	description	cast_y
0	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...	Ama Qamata
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...	Khosi Ngema
2	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...	Gail Mabalane
3	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...	Thabang Molaba
4	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...	Dillon Windvogel

Top 20 casts in movies

In [74]:

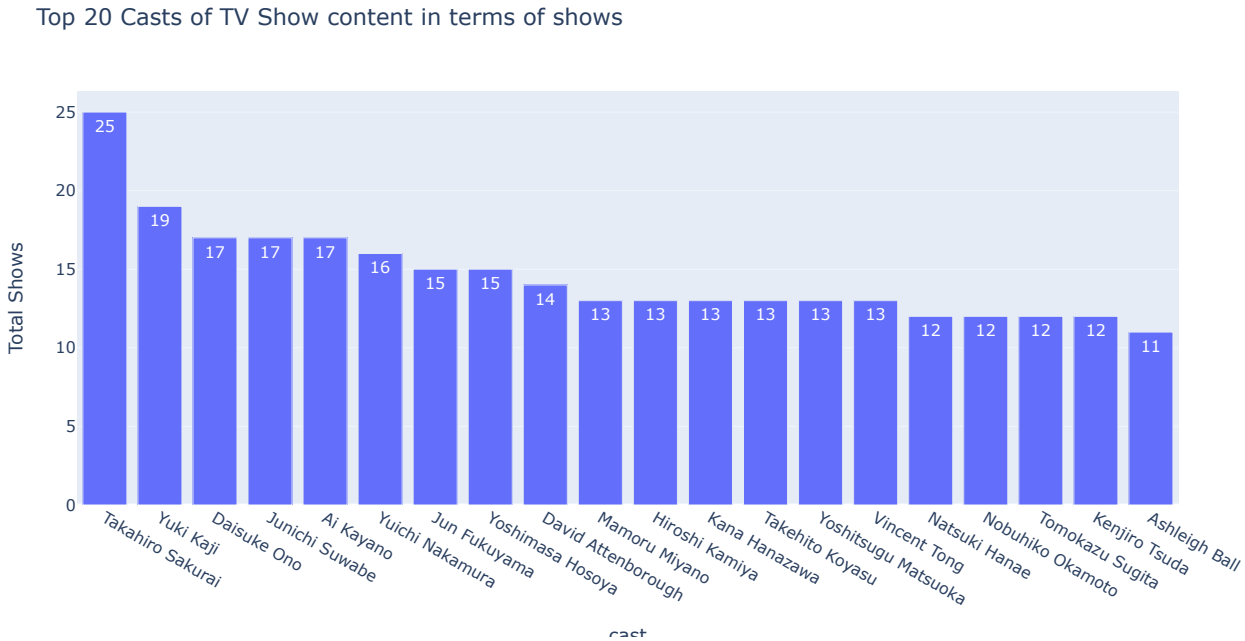
```
# Top 20 Casts

movie_cast_freq = combined_show_cast_data[combined_show_cast_data["type"] == "Movie"]["cast_y"].value_counts()
x = list(movie_cast_freq.index)[:20]
y = list(movie_cast_freq)[:20]

# Use textposition='auto' for direct text
fig = go.Figure(data=[go.Bar(
            x=x, y=y,
            text=y,
            textposition='auto',
        )])
fig.update_layout(
    title = "Top 20 Casts of Movie content in terms of Movies",

    xaxis=dict(
        title_text="cast"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)

fig.show()
```



Top 20 casts in tv show

In [75]:

```
# Top 20 Casts

tv_show_cast_freq = combined_show_cast_data[combined_show_cast_data["type"] == "TV Show"]["cast_y"].value_counts()
x = list(tv_show_cast_freq.index)[:20]
y = list(tv_show_cast_freq)[:20]

# Use textposition='auto' for direct text
fig = go.Figure(data=[go.Bar(
            x=x, y=y,
            text=y,
            textposition='auto',
        )])
fig.update_layout(
    title = "Top 20 Casts of TV Show content in terms of shows",

    xaxis=dict(
        title_text="cast"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)

fig.show()
```



Top countries in terms of Shows/Movies Produced

In [76]:

```
combined_show_country_data = pd.merge(original_data, shows_country_df, on='show_id')
combined_show_country_data.head()
```

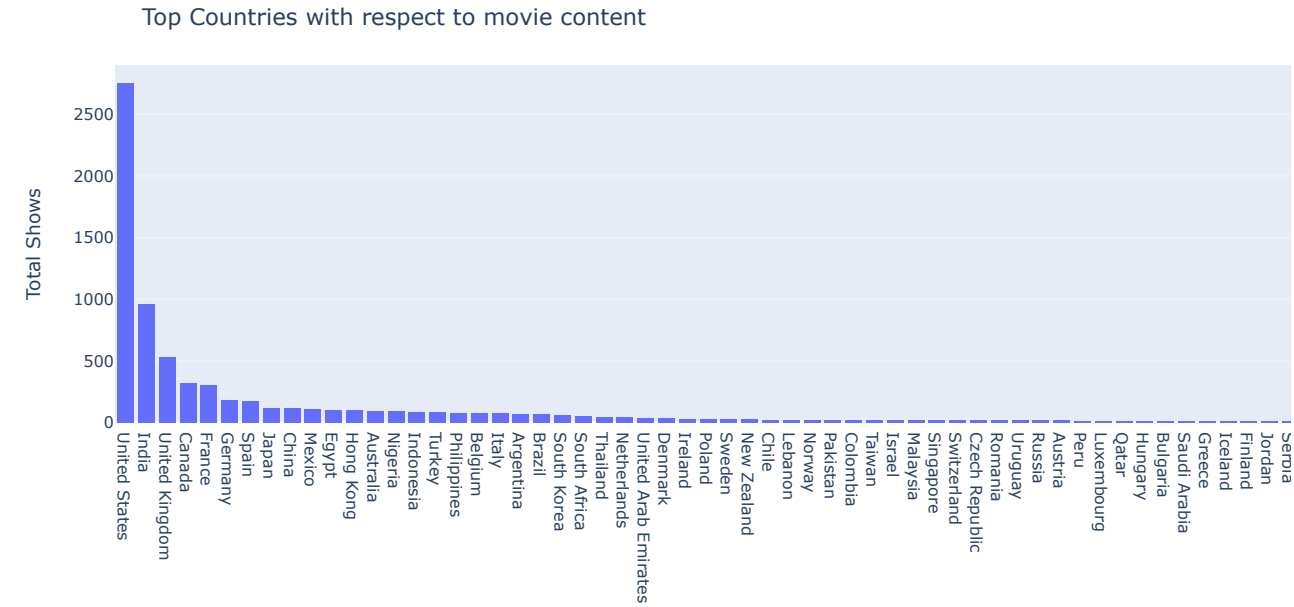
Out[76]:

	show_id	type	title	director	cast	country_x	date_added	release_year	rating	duration	listed_in	description	country_y
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...	United States
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...	South Africa
2	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...	India
3	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...	United States
4	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...	Ghana

Top contries with respect to Movie content

In [77]:

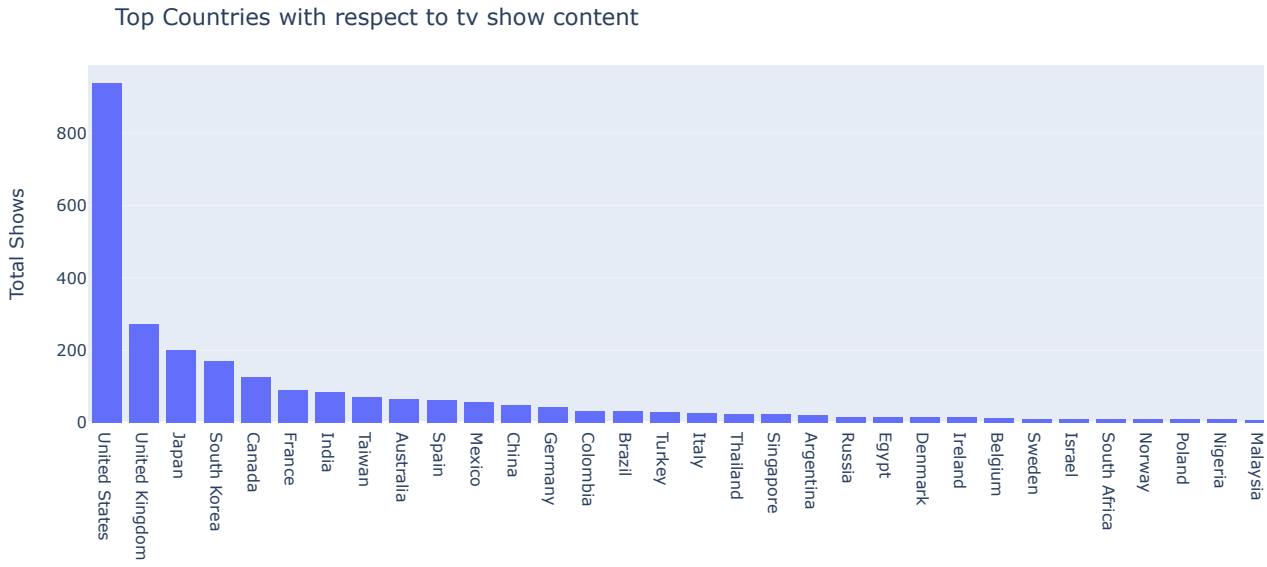
```
# plotly-Histogram
movie_combined_show_country_data = combined_show_country_data[combined_show_country_data["type"]== "Movie"]
fig = px.histogram(movie_combined_show_country_data, x="country_y")
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.update_layout(
    autosize=False,
    width=2000,
    height=500,
    title = "Top Countries with respect to movie content",

    xaxis=dict(
        title_text="Countries"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)
fig.show()
```



Top contries with respect to TV Show content

In [78]:

```
# plotly-Histogram
movie_combined_show_country_data = combined_show_country_data[combined_show_country_data["type"]== "TV Show"]
fig = px.histogram(movie_combined_show_country_data, x="country_y")
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.update_layout(
    autosize=False,
    width=2000,
    height=500,
    title = "Top Countries with respect to tv show content",

    xaxis=dict(
        title_text="Countries"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)
fig.show()
```



Top Genre

In [79]:

```
# Merge data
combined_shows_listed_data = pd.merge(data, shows_listed_in_df, on='show_id')
combined_shows_listed_data.head()
```

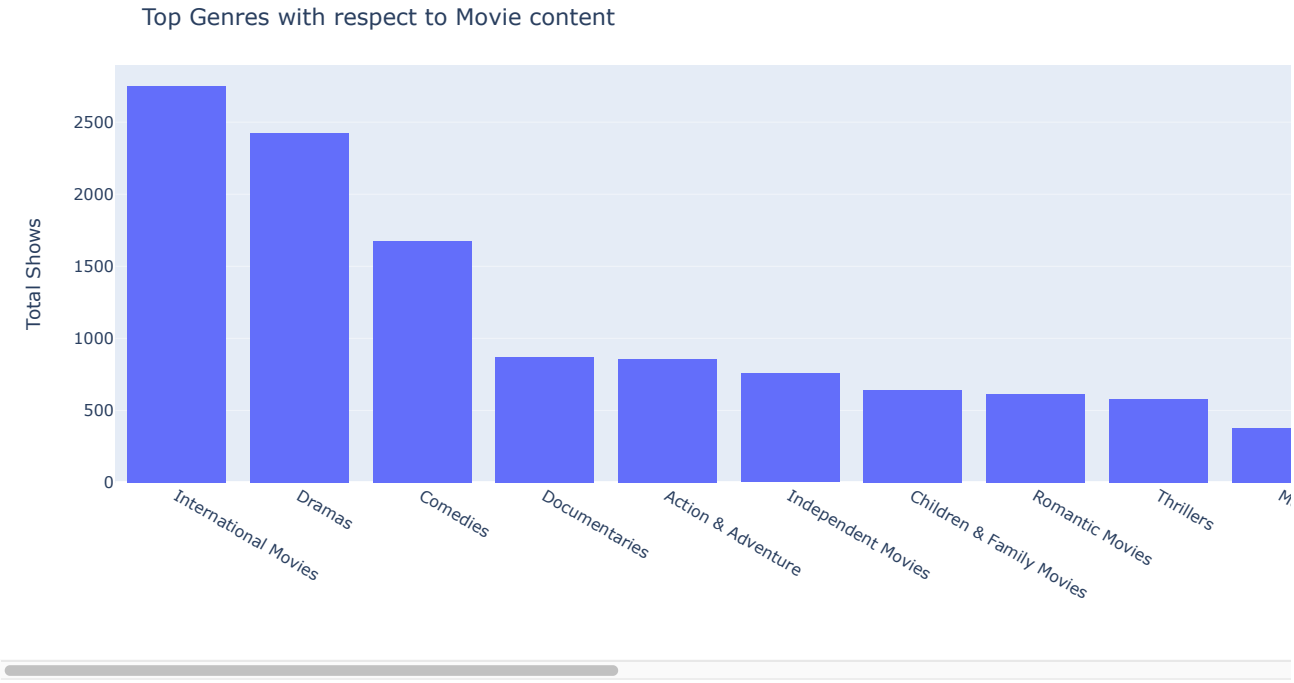
Out[79]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in_x	description	date_added_month	date_added
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	UNKNOWN	United States	September 25, 2021	2020	PG-13	90	Documentaries	As her father nears the end of his life, filmm...	September	
1	s2	TV Show	Blood & Water	UNKNOWN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	September	
2	s2	TV	Blood &	UNKNOWN	Ama Qamata, Khosi Ngema	South	September	2021	TV-	2	International TV Shows, TV	After crossing paths at a	September	

Tope Genre in Movies

In [80]:

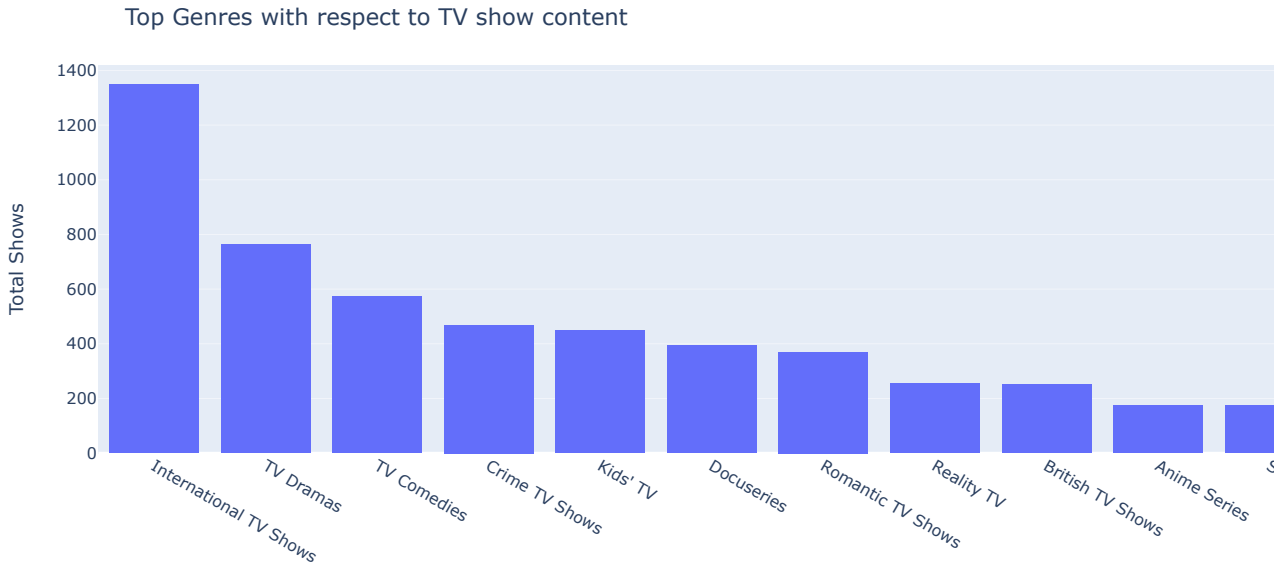
```
# plotly-Histogram
movie_combined_listed_in_data = combined_shows_listed_data[combined_shows_listed_data["type"]== "Movie"]
fig = px.histogram(movie_combined_listed_in_data, x="listed_in_y")
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.update_layout(
    autosize=False,
    width=2000,
    height=500,
    title = "Top Genres with respect to Movie content",

    xaxis=dict(
        title_text="Genre"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)
fig.show()
```



Top Genre in TV Shows

In [81]:

```
# plotly-Histogram
tv_show_combined_listed_in_data = combined_shows_listed_data[combined_shows_listed_data["type"]== "TV Show"]
fig = px.histogram(tv_show_combined_listed_in_data, x="listed_in_y")
fig.update_layout(xaxis={'categoryorder':'total descending'})
fig.update_layout(
    autosize=False,
    width=2000,
    height=500,
    title = "Top Genres with respect to TV show content",

    xaxis=dict(
        title_text="Genre"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)
fig.show()
```



Top 5 Genres per year (date added) - Trend

Top 5 Genres with respect to Movie content per year (date added)

In [82]:

```
top_5_genre_in_movies = list(movie_combined_listed_in_data["listed_in_y"].value_counts().index)[:5]
top_5_genre_in_movies
```

Out[82]:

```
['International Movies',
 'Dramas',
 'Comedies',
 'Documentaries',
 'Action & Adventure']
```

In [83]:

```
# get data for top 5 genere in Movie
df = movie_combined_listed_in_data[movie_combined_listed_in_data["listed_in_y"].isin(top_5_genre_in_movies)]
df.head()
```

Out[83]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in_x	description	date_added_month	da
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	UNKNOWN	United States	September 25, 2021	2020	PG-13	90	Documentaries	As her father nears the end of his life, filmm...	September	
16	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...	September	
18	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...	September	
21	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104	Comedies, Dramas	A woman adjusting to life after a loss contend...	September	
22	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104	Comedies, Dramas	A woman adjusting to life after a loss contend...	September	

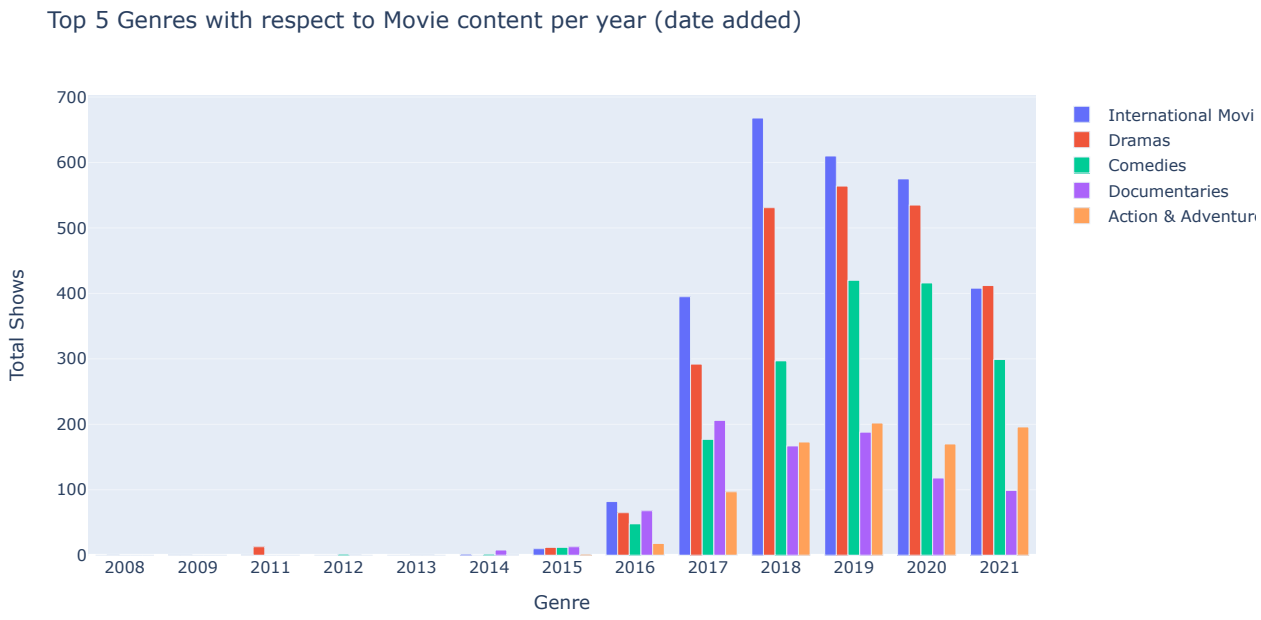
In [84]:

```
date_added_year_list = list(df["date_added_year"].unique())
date_added_year_list.sort()

genre_year_map_list = []
for genre in top_5_genre_in_movies:
    genre_list = []
    for year in date_added_year_list:
        genre_list.append(df[df["date_added_year"] == year][df["listed_in_y"] == genre].shape[0])
    genre_year_map_list.append(genre_list)
```

```
/var/folders/9t/0rf19dxs4xb82c79_12x19ph0000gn/T/ipykernel_2724/1278067857.py:8: UserWarning:

Boolean Series key will be reindexed to match DataFrame index.
```

In [85]:

```
data_list = []
for index,genre in enumerate(top_5_genre_in_movies):
    data_list.append(go.Bar(name=genre, x=date_added_year_list, y=genre_year_map_list[index]))

fig = go.Figure(data=data_list)
# Change the bar mode
fig.update_layout(barmode='group')
fig.update_layout(
    title = "Top 5 Genres with respect to Movie content per year (date added)",

    xaxis=dict(
        title_text="Genre"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)
fig.show()
```



Top 5 Genres with respect to TV Show content per year (date added)

In [86]:

```
top_5_genre_in_tv_show = list(tv_show_combined_listed_in_data["listed_in_y"].value_counts().index)[:5]
top_5_genre_in_tv_show
```

Out[86]:

```
['International TV Shows',
 'TV Dramas',
 'TV Comedies',
 'Crime TV Shows',
 "Kids' TV"]
```

In [87]:

```
# get data for top 5 genere in TV show
df = tv_show_combined_listed_in_data[tv_show_combined_listed_in_data["listed_in_y"].isin(top_5_genre_in_tv_show)]
df.head()
```

Out[87]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in_x	description	date_added_month	d
1	s2	TV Show	Blood & Water	UNKNOWN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	September	
2	s2	TV Show	Blood & Water	UNKNOWN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	September	
4	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	UNKNOWN	September 24, 2021	2021	TV-MA	1	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	September	
5	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	UNKNOWN	September 24, 2021	2021	TV-MA	1	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	September	
9	s5	TV Show	Kota Factory	UNKNOWN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...	September	

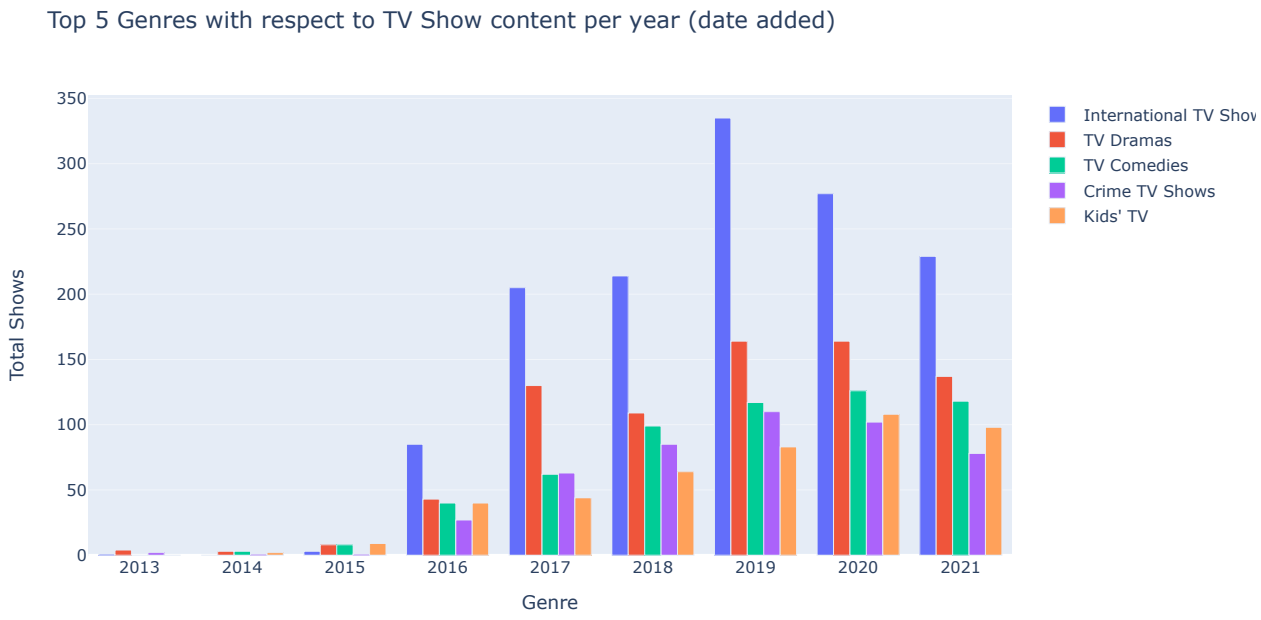
In [88]:

```
date_added_year_list = list(df["date_added_year"].dropna().unique())
date_added_year_list.sort()

genre_year_map_list = []
for genre in top_5_genre_in_tv_show:
    genre_list = []
    for year in date_added_year_list:
        genre_list.append(df[df["date_added_year"] == year][df["listed_in_y"] == genre].shape[0])
    genre_year_map_list.append(genre_list)
```

```
/var/folders/9t/0rf19dxs4xb82c79_12x19ph0000gn/T/ipykernel_2724/685306507.py:8: UserWarning:

Boolean Series key will be reindexed to match DataFrame index.
```

In [89]:

```
data_list = []
for index,genre in enumerate(top_5_genre_in_tv_show):
    data_list.append(go.Bar(name=genre, x=date_added_year_list, y=genre_year_map_list[index]))

fig = go.Figure(data=data_list)
# Change the bar mode
fig.update_layout(barmode='group')
fig.update_layout(
    title = "Top 5 Genres with respect to TV Show content per year (date added)",

    xaxis=dict(
        title_text="Genre"
        ),
    yaxis=dict(
        title_text="Total Shows"
        )

)
fig.show()
```



Delay (in years) in adding the shows/Movies in NetFlix

In [90]:

```
def getDelays(data):
    df = data.copy()
    df["date_added_year"] = df["date_added_year"].apply(float)

    df["delay"] = abs(df["date_added_year"] - df["release_year"])

    return df
```

In [91]:

```
delayed_df = getDelays(data)
delayed_df.head()
```

Out[91]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	date_added_mont
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	UNKNOWN	United States	September 25, 2021	2020	PG-13	90	Documentaries	As her father nears the end of his life, filmm...	Septembe
1	s2	TV Show	Blood & Water	UNKNOWN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	Septembe
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	UNKNOWN	September 24, 2021	2021	TV-MA	1	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	Septembe
3	s4	TV Show	Jailbirds New Orleans	UNKNOWN	UNKNOWN	UNKNOWN	September 24, 2021	2021	TV-MA	1	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...	Septembe
4	s5	TV Show	Kota Factory	UNKNOWN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...	Septembe

In [92]:

```
delayed_df["delay"].dropna().unique()
```

Out[92]:

```
array([ 1.,  0., 28.,  3., 25., 23., 24., 11.,  8.,  4., 46., 43., 38.,
       34.,  9., 20.,  7., 19., 18., 17., 10., 13., 12., 14., 16., 15.,
       27.,  6.,  2.,  5., 39., 32., 31., 30., 22., 35., 29., 37., 41.,
       60., 21., 26., 36., 45., 62., 33., 40., 49., 57., 76., 66., 64.,
       50., 47., 44., 93., 51., 55., 48., 42., 54., 59., 61., 52., 63.,
       72., 71., 75., 65., 73., 70., 74.])
```

Delay (in years) in adding the Movie in NetFlix

In [93]:

```
movie_delayed_df = getDelays(movie_combined_listed_in_data)
movie_delayed_df.head()
```

Out[93]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in_x	description	date_added_month
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	UNKNOWN	United States	September 25, 2021	2020	PG-13	90	Documentaries	As her father nears the end of his life, filmm...	September
15	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	UNKNOWN	September 24, 2021	2021	PG	91	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...	September
16	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...	September
17	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...	September
18	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...	September

In [94]:

```
df = movie_delayed_df[movie_delayed_df["listed_in_y"].isin(top_5_genre_in_movies)]
```

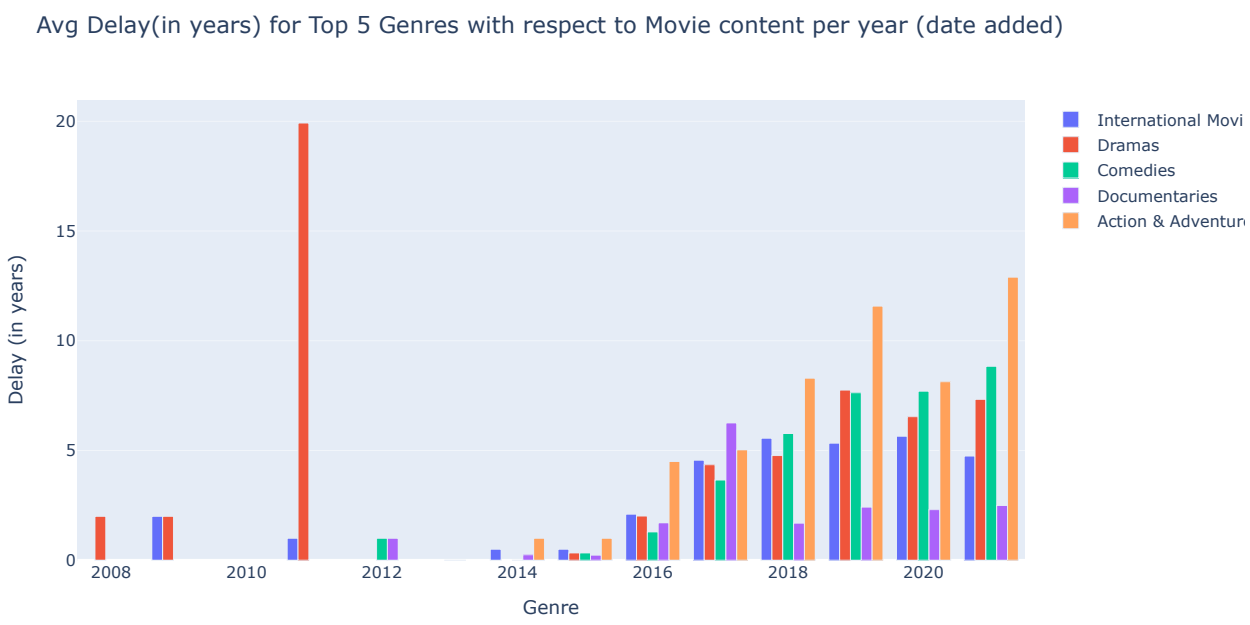
In [95]:

```
date_added_year_list = list(df["date_added_year"].dropna().unique())
date_added_year_list.sort()

genre_year_delay_map_list = []
for genre in top_5_genre_in_movies:
    genre_list = []
    for year in date_added_year_list:
        genre_list.append(df[df["date_added_year"] == year][df["listed_in_y"] == genre]["delay"].mean())
    genre_year_delay_map_list.append(genre_list)
```

```
/var/folders/9t/0rf19dxs4xb82c79_12x19ph0000gn/T/ipykernel_2724/3430086438.py:8: UserWarning:

Boolean Series key will be reindexed to match DataFrame index.
```

In [96]:

```
data_list = []
for index,genre in enumerate(top_5_genre_in_movies):
    data_list.append(go.Bar(name=genre, x=date_added_year_list, y=genre_year_delay_map_list[index]))

fig = go.Figure(data=data_list)
# Change the bar mode
fig.update_layout(barmode='group')
fig.update_layout(
    title = "Avg Delay(in years) for Top 5 Genres with respect to Movie content per year (date added)",

    xaxis=dict(
        title_text="Genre"
        ),
    yaxis=dict(
        title_text="Delay (in years)"
        )

)
fig.show()
```



Delay (in years) in adding the TV Show in NetFlix

In [97]:

```
tv_show_delayed_df = getDelays(tv_show_combined_listed_in_data)
tv_show_delayed_df.head()
```

Out[97]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in_x	description	date_added_month	d
1	s2	TV Show	Blood & Water	UNKNOWN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	September	
2	s2	TV Show	Blood & Water	UNKNOWN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	September	
3	s2	TV Show	Blood & Water	UNKNOWN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	September	
4	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	UNKNOWN	September 24, 2021	2021	TV-MA	1	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	September	
5	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	UNKNOWN	September 24, 2021	2021	TV-MA	1	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	September	

In [98]:

```
df = tv_show_delayed_df[tv_show_delayed_df["listed_in_y"].isin(top_5_genre_in_tv_show)]
```

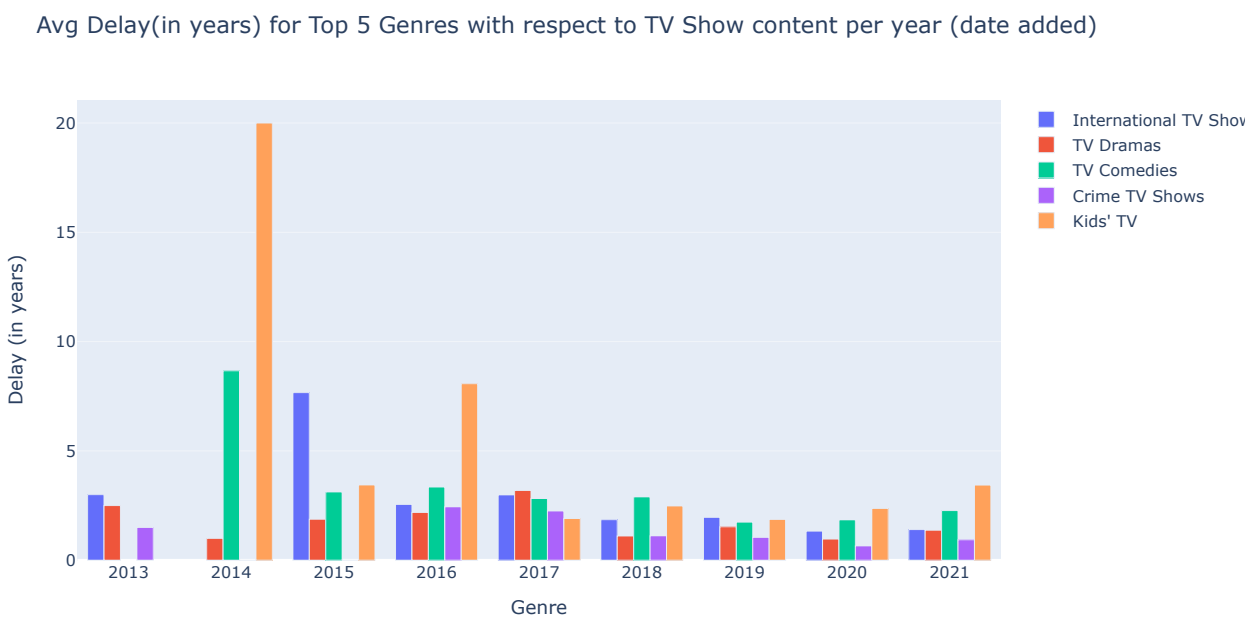
In [99]:

```
date_added_year_list = list(df["date_added_year"].dropna().unique())
date_added_year_list.sort()

genre_year_delay_map_list = []
for genre in top_5_genre_in_tv_show:
    genre_list = []
    for year in date_added_year_list:
        genre_list.append(df[df["date_added_year"] == year][df["listed_in_y"] == genre]["delay"].mean())
    genre_year_delay_map_list.append(genre_list)
```

```
/var/folders/9t/0rf19dxs4xb82c79_12x19ph0000gn/T/ipykernel_2724/4077610881.py:8: UserWarning:

Boolean Series key will be reindexed to match DataFrame index.
```

In [100]:

```
data_list = []
for index,genre in enumerate(top_5_genre_in_tv_show):
    data_list.append(go.Bar(name=genre, x=date_added_year_list, y=genre_year_delay_map_list[index]))

fig = go.Figure(data=data_list)
# Change the bar mode
fig.update_layout(barmode='group')
fig.update_layout(
    title = "Avg Delay(in years) for Top 5 Genres with respect to TV Show content per year (date added)",

    xaxis=dict(
        title_text="Genre"
        ),
    yaxis=dict(
        title_text="Delay (in years)"
        )

)
fig.show()
```



Summary

- In Netflix 30.4 % TV Show and 69.6% Movie Content, Netflix has most of movies data base in There Platform.
- Unites State is Top country in terms of Shows Produced as well as Movies Produces.
- Mostly, movies duration are around 90 mins. Histogram is dense means that majority of movies are around 90 mins. In future Movie production Netflix can consider movie duration one of parameter to increase bussiness
- Majority TV Shows with 1 Seasons are in Netflix having 1793 shows. Means most of show stopped producing more seasons so need analysis to understand reason behind it.

- Consider Top 20 Directors in terms of Shows/Movies Produced for Future movie and shows production
- Consider Top 20 Casts in terms of Shows/Movies Produced for Future Movie and shows production, Here top casts in movie is totally different than top cast in shows

- Consider Top countries in terms of Shows Produced as well as movies produced for future production.
- For future movie gerne Netflix should consider top Gerne of movie which is International, Dramas and Comedies to increase bussiness.
- For future shows gerne Netflix should consider top Gerne of shows which is International TV Shows, TV Dramas and TV Comedies to increase bussiness.
- We can observe Top 5 Genres that is International Movies, Dramas, Comedies, Documentaries, Action & Adventure with respect to Movie content per year, we can see year wise content added per genres, Each year at top International genre Movies get added. We observe that each year content added per genres is varies so we should consider this factor.

- we can observe Top 5 Genres that is International TV Shows, TV Dramas, TV Comedies, Crime TV Shows, Kids TV with respect to show content per year, we can see year wise content added per genres, Each year at top International TV Shows genre shows get added.

- Here we are intoducing Delay (in years) in adding the shows or movies in NetFlix which is difference between date added year on Netflix Platform and release year for particular movie or show.

-Avg Delay (in years) for top 5 Genres International Movies, Dramas, Comedies, Documentaries, Action & Adventure release date with respect date added on Netflix Platform must be less for Movies, some delays are valid for Movies which are older when Netflix Platform was not available but for other cases to increase bussiness Netflix should keep this delay less.