NETFLIX Exploratory Data Analysis - Understanding Dataset/Mohammad Mojahid

```
#importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Tasks

> - Understand the datasets, types and missing values
> - Claening Data sets, Handling missing values, formating datasets
> - Perform Data Visualisation
> - Create Final report summary

```
df = pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv')
```

+ Code + Text

```
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descrip
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her f near end life, fil
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	cro path party, a Tov
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 prote family fr pov drug
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	F flirtation toilet ta down a

```
df.tail()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descr
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A carto repo
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	2 Seasons	Kids' TV, Korean TV Shows, TV Comedies	Wh al spool a yo
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Lo surv wor over
					Tim Allen,							

```
df.info() #gives more insight and aggregate dataset
```

```
<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
```

```
df.shape #gives information about number of rows and column in dataset
```

```
(8807, 12)
```

```
df.describe()
```

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000

```
df['type'].value_counts(normalize=True)*100
```

```
Movie      69.615079
TV Show    30.384921
Name: type, dtype: float64
```

```
df.isna().sum() #missing values
```

```
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
```

Formating Data types and filling missing values

release_year are object, strings are expected **The following data type do not require any fills**

> - type
> - title
> - release_year
> - listed_in
> - description

The following are missing data

> - duration
> - rating
> - date_added
> - cast
> - country
> - director
> - check data types where needed and proceed

data_added to be updated to datetime "unavailable" will be substituted in for any nulls of fields with object/string data types. this applies to everything except for release_year

Update date_added to determine and check

```
# converting data type
df['date_added'] = pd.to_datetime(df['date_added'])
```

```
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descrip
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her f near end life, fil
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	cro paths party, a Tov
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To prote family fr pov drug
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV	Fo flirtations toilet ta down a

Handling missing Values

> - rating, cast, country and director have the nulls filled with 'unavailable'

```
df.fillna({'rating': 'Unavailable', 'cast': 'Unavailable', 'country': 'Unavailable', 'director': 'Unavailable'}, inplace=True)
df.isna().sum()
```

```
    show_id          0
    type             0
    title            0
    director         0
    cast             0
    country          0
    date_added      10
    release_year     0
    rating           0
    duration         3
```

```
listed_in        0
description      0
dtype: int64
```

For nulls date_added, missing date_added is to be substituted in with the most recent date from date_added. this is because Netflix has the tendancy to add more content over time. Other option would be finding actual dates and inputting them manually or dropping the data from results since the amount of missing data is rather small.

```
df[df.date_added.isnull()]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
			and Other		Daniel Adam						TV Dramas

```
most_recent_entry_date = df['date_added'].max()
df.fillna({'date_added':most_recent_entry_date}, inplace=True)
```

```
#checking line of code if change is done
df[df.show_id =='s8183']
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descr
8182	s8183	TV Show	The Adventures of Figaro Pho	Unavailable	Luke Jurevicius, Craig Behenna, Charlotte Haml...	Australia	2021-09-25	2015	TV-Y7	2 Seasons	Kids' TV, TV Comedies	Imagi wors then r tl

Hyde

Additional Data Cleaning

Duration data input error

The missing durations are all movies by Louis C. K. Normally, we would likely fill the duration with the mean duration of movies from the table. In this case it apears that the actual duration was input into the rating column, so one solution is to move the rating data into the duration and the rating information 'Unavailable' like the other nulls

		Show	Girl		Mitsuhashi,					Seasons	Crime TV

```
df[df.duration.isnull()]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descripti
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017	74 min	NaN	Movies	Louis C.K. mus on religio eternal love, g
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	2016-09-16	2010	84 min	NaN	Movies	Emmy-winni comedy wri Louis C.K. brin h
			Louis C.K.: Live at	Louis	Louis	United						The comic pu

Marc,

Check to make sure there is no other content with the same director to avoid accidental overwriting

Josh

```
df[df.director == 'Louis C.K.'].head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descripti
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017	74 min	NaN	Movies	Louis C.K. mus on religic eternal love, g
			Louis									Emmy-winni

```
#overwrite and check with loc
df.loc[df['director']== 'Louis C.K.', 'duration'] = df['rating']
df[df.director == 'Louis C.K.'].head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descripti
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017	74 min	74 min	Movies	Louis C.K. mus on religic eternal love, g
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	2016-09-16	2010	84 min	84 min	Movies	Emmy-winni comedy wri Louis C.K. brin h
			Louis C.K.: Live at	Louis	Louis	United						The comic pu

```
df.loc[df['director']== 'Louis C.K.', 'duration'] = 'Unavailable'
df[df.director == 'Louis C.K.'].head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descript
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017	74 min	Unavailable	Movies	Louis C.K. mu on relig eternal love,
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	2016-09-16	2010	84 min	Unavailable	Movies	Emmy-win comedy w Louis C.K. br
			Louis C.K.: Live at	Louis	Louis	United						The comic

Visualisations

> Type of shows that has been watched on *Netflix*

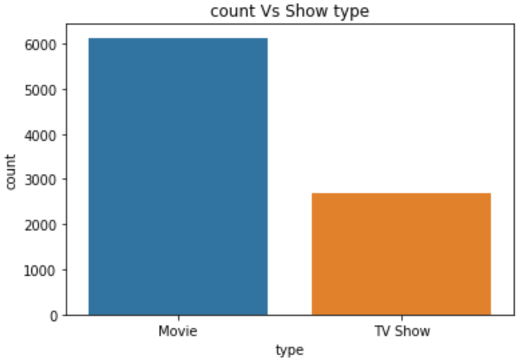
```
df.type.value_counts()
```

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

```
sns.countplot(x = 'type', data =df)
#count plot help to visualise the category
```

```
plt.title('count Vs Show type')
```

```
Text(0.5, 1.0, 'count Vs Show type')
```



Country wise Analysis

```
df['country'].value_counts().head(10) #top ten country
```

```
United States     2818
India              972
Unavailable        831
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Name: country, dtype: int64
```

```
plt.figure(figsize = (12, 6))
sns.countplot(y='country', order=df['country'].value_counts().index[0:10], data=df)
plt.title('Countrywise content on Netflix')
```

```
Text(0.5, 1.0, 'Countrywise content on Netflix')
```

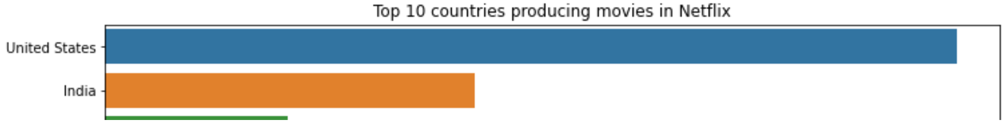
Type of content country wise

```
movie_countries = df[df['type']=='Movie']
tv_show_countries = df[df['type']=='TV Show']
```

```
plt.figure(figsize = (12, 6))
sns.countplot(y='country', order=df['country'].value_counts().index[0:10], data=movie_countries)
plt.title('Top 10 countries producing movies in Netflix')

plt.figure(figsize = (12, 6))
sns.countplot(y='country', order = df['country'].value_counts().index[0:10], data= tv_show_countries)
plt.title('Top 10 countries producing TV Shows in Netflix')
```

```
Text(0.5, 1.0, 'Top 10 countries producing TV Shows in Netflix')
```



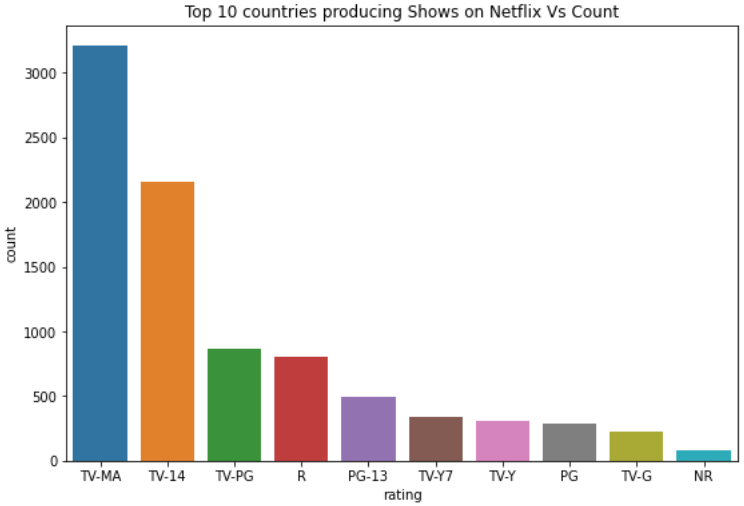
Top 10 countries producing movies in Netflix

```
df.rating.value_counts()
```

```
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
Unavailable       4
NC-17             3
UR                3
74 min            1
84 min            1
66 min            1
Name: rating, dtype: int64
```

```
plt.figure(figsize = (9, 6))
sns.countplot(x='rating', order = df['rating'].value_counts().index[0:10], data=df)
plt.title('Top 10 countries producing Shows on Netflix Vs Count')
```

```
Text(0.5, 1.0, 'Top 10 countries producing Shows on Netflix Vs Count')
```



Top 10 countries producing Shows on Netflix Vs Count

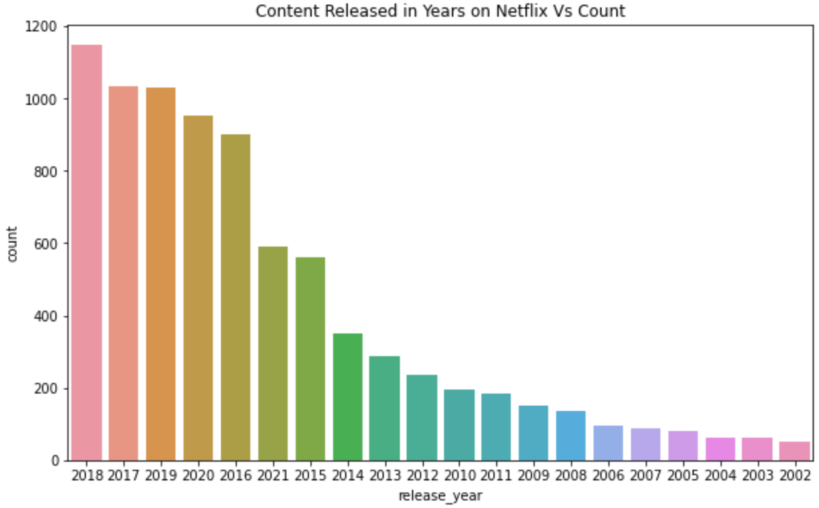
Most of the shows has TV-MA and TV-14 rating

```
df.release_year.value_counts()[:20]
```

```
    2018    1147
    2017    1032
    2019    1030
    2020     953
    2016     902
    2021     592
    2015     560
    2014     352
    2013     288
    2012     237
    2010     194
    2011     185
    2009     152
    2008     136
    2006      96
    2007      88
    2005      80
    2004      64
    2003      61
    2002      51
    Name: release_year, dtype: int64
```

```
plt.figure(figsize = (10, 6))
sns.countplot(x='release_year', order = df['release_year'].value_counts().index[0:20], data= df)
plt.title('Content Released in Years on Netflix Vs Count')
```

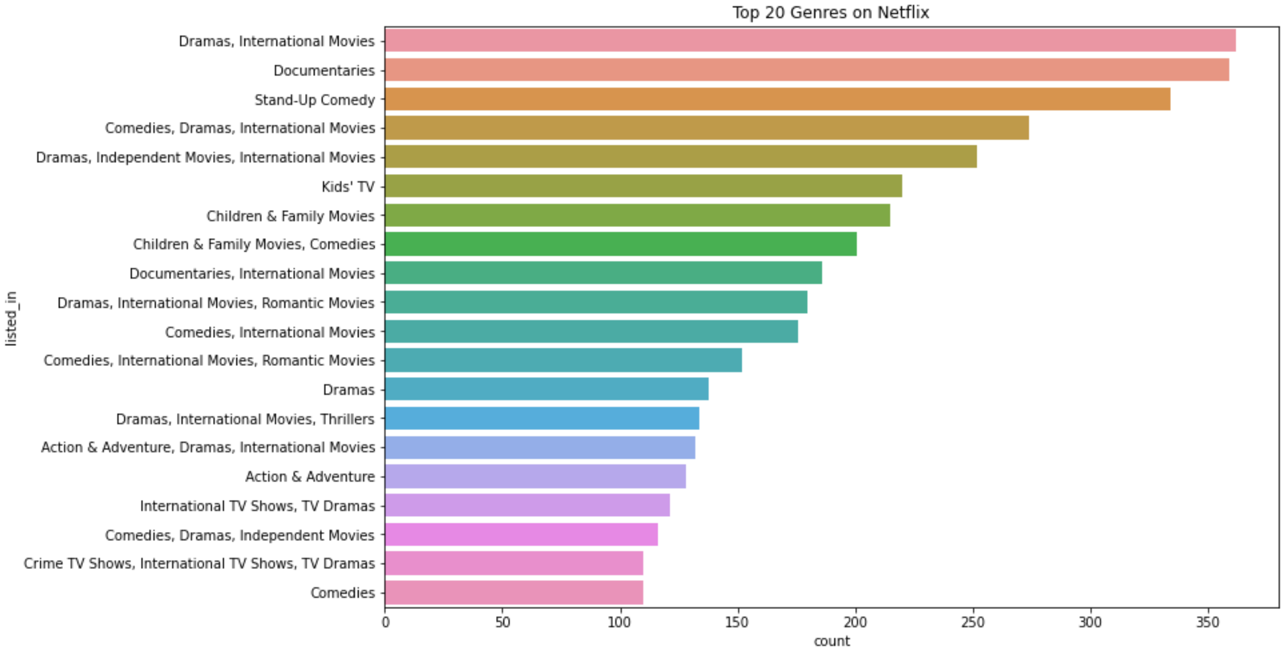
```
    Text(0.5, 1.0, 'Content Released in Years on Netflix Vs Count')
```



Popular Genre Analysis

```
plt.figure(figsize = (12, 8))
sns.countplot(y='listed_in', order = df['listed_in'].value_counts().index[0:20], data= df)
plt.title('Top 20 Genres on Netflix')
```

```
    Text(0.5, 1.0, 'Top 20 Genres on Netflix')
```



Summary:

As per current basis understanding, I performed operations over the dataset to find out some useful insights from it. Below are the major observation I found out with the dataset.

> - Netflix is gaining more popularity over the TV shows hence Netflix has more movies than TV shows
> - Large number of Movies and TV shows are produced in United States, followed by India has produced more movies on Netflix
> - Large number of Movies and TV shows for Netflix are produced under Mature Audiences
> - 2018 onwards Netflix released more content as compared to other years
> - International Movies and Dramas are the most popular Genres on Netflix

✓ 2s completed at 1:20 AM ● ×