Importing Library

In [1]:
```
import pandas as pd
```

Loading Dataset

In [2]:
```
df= pd.read_csv("Netflix.csv")
```

In [3]:
```
df.head(10)
```

Out[3]:

	type	title	date_added	release_year	rating	listed_in
0	Movie	**Dick Johnson Is Dead**	9/25/2021	2020)	,?PG-13	Documentaries
1	TV Show	**Ganglands**	9/24/2021	2021C	,?TV-MA	Crime TV Shows, International TV Shows, TV Act...
2	TV Show	**Midnight Mass**	9/24/2021	2021c	,?TV-MA	TV Dramas, TV Horror, TV Mysteries
3	Movie	**Confessions of an Invisible Girl**	9/22/2021	2021	,?TV-PG	Children & Family Movies, Comedies
4	Movie	**Sankofa**	9/24/2021	1993S	,?TV-MA	Dramas, Independent Movies, International Movies
5	TV Show	**The Great British Baking Show**	9/24/2021	2021s	,?TV-14	British TV Shows, Reality TV
6	NaN	**The Starling**	9/24/2021	2021)	,?PG-13	Comedies, Dramas
7	Movie	**Motu Patlu in the Game of Zones**	05/01/2021	2019C	,?TV-Y7	Children & Family Movies, Comedies, Music & Mu...
8	Movie	**Je Suis Karl**	9/23/2021	2021S	,?TV-MA	Dramas, International Movies
9	Movie	**Motu Patlu in Wonderland**	05/01/2021	2013	,?TV-Y7	Children & Family Movies, Music & Musicals

We can see that there are lot of discrepancies in the dataset \ For example, 'title' , 'release_year' , 'rating' columns contain messy values (bad char) \ 'type' column has null values

Identifying columns with missing values

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4001 entries, 0 to 4000
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   type          3934 non-null   object
 1   title         4001 non-null   object
 2   date_added    4001 non-null   object
 3   release_year  4001 non-null   object
 4   rating        4001 non-null   object
 5   listed_in     4001 non-null   object
dtypes: object(6)
memory usage: 187.7+ KB
```

identified column with missing values : 'type'

Handling Missing Data

In [5]:
```
df['type'] = df['type'].astype(str)
```

In [6]:
```
df['type'] = df['type'].replace(['nan' , 'Nan'] , 'Unknown')
```

checking if there is any missing value left

In [8]:
```
df.isna().sum()
```

Out[8]:
```
type            0
title           0
date_added      0
release_year    0
rating          0
listed_in       0
dtype: int64
```

No more missing values

Handling Messy Data

Removing "**" from the "title" column.

In [9]:
```
df['title'] = df['title'].str.strip("**")
df['title']
```

Out[9]:
```
0                    Dick Johnson Is Dead
1                               Ganglands
2                           Midnight Mass
3       Confessions of an Invisible Girl
4                                 Sankofa
                       ...
3996                            One 2 Ka 4
3997       Jim Norton: Mouthful of Shame
3998                            100 Meters
3999                         Burning Sands
4000                The Butterfly's Dream
Name: title, Length: 4001, dtype: object
```

Converting the date format from "9/25/2021" to "9-25-2021".

In [10]:
```
df['date_added'] = df['date_added'].str.replace('/','-')
df['date_added']
```

Out[10]:
```
0        9-25-2021
1        9-24-2021
2        9-24-2021
3        9-22-2021
4        9-24-2021
           ...
3996     3-15-2017
3997     3-14-2017
3998    03-10-2017
3999    03-10-2017
4000    03-10-2017
Name: date_added, Length: 4001, dtype: object
```

Converting the dtype into 'datetime'

In [19]:
```
df['date_added'] = pd.to_datetime(df['date_added'] , format = '%m-%d-%Y')
df['date_added']
```

Out[19]:
```
0      2021-09-25
1      2021-09-24
2      2021-09-24
3      2021-09-22
4      2021-09-24
          ...
3996   2017-03-15
3997   2017-03-14
3998   2017-03-10
3999   2017-03-10
4000   2017-03-10
Name: date_added, Length: 4001, dtype: datetime64[ns]
```

Cleaning the "release_year" column by removing bad characters

In [12]:
```
bad_char = ["(" , ")" , "c" , "C" , "." , " " , "S" , "s"]

for i in bad_char:
    df["release_year"] = df["release_year"].str.replace(i, "")
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_16672\546611212.py:4: FutureWarning: The default value of regex will c
hange from True to False in a future version. In addition, single character regular expressions will *not* be t
reated as literal strings when regex=True.
  df["release_year"] = df["release_year"].str.replace(i, "")
```

In [13]:
```
df["release_year"] = df["release_year"].astype(int)
df["release_year"]
```

Out[13]:
```
0       2020
1       2021
2       2021
3       2021
4       1993
        ...
3996    2001
3997    2017
3998    2016
3999    2017
4000    2013
Name: release_year, Length: 4001, dtype: int32
```

Cleaning the "rating" column by removing ",?"

In [14]:
```
df['rating'] = df['rating'].str.strip(',?')
df['rating']
```

Out[14]:
```
0       PG-13
1       TV-MA
2       TV-MA
3       TV-PG
4       TV-MA
        ...
3996    TV-14
3997    TV-MA
3998    TV-MA
3999    TV-MA
4000    TV-PG
Name: rating, Length: 4001, dtype: object
```

Keeping Only the Relevant Data`

Split the last column ("listed_in") using commas.

In [15]:
```
df['listed_in'].str.split(',' , expand = True)
```

Out[15]:

	0	1	2
0	Documentaries	None	None
1	Crime TV Shows	International TV Shows	TV Action & Adventure
2	TV Dramas	TV Horror	TV Mysteries
3	Children & Family Movies	Comedies	None
4	Dramas	Independent Movies	International Movies
...	...	...	...
3996	Action & Adventure	Comedies	Dramas
3997	Stand-Up Comedy	None	None
3998	Dramas	International Movies	Sports Movies
3999	Dramas	Independent Movies	None
4000	Dramas	International Movies	Romantic Movies

4001 rows × 3 columns

Keeping only the first category from the split list.

In [16]:
```
df['listed_in'].str.split(',' , expand = True)[0]
df['listed_in'] = df['listed_in'].str.split(',' , expand = True)[0]
df['listed_in']

# cross checking the 'listed_in' column.
df
```

Out[16]:

	type	title	date_added	release_year	rating	listed_in
0	Movie	Dick Johnson Is Dead	9-25-2021	2020	PG-13	Documentaries
1	TV Show	Ganglands	9-24-2021	2021	TV-MA	Crime TV Shows
2	TV Show	Midnight Mass	9-24-2021	2021	TV-MA	TV Dramas
3	Movie	Confessions of an Invisible Girl	9-22-2021	2021	TV-PG	Children & Family Movies
4	Movie	Sankofa	9-24-2021	1993	TV-MA	Dramas
...	...	...	...	...	...	...
3996	Movie	One 2 Ka 4	3-15-2017	2001	TV-14	Action & Adventure
3997	Movie	Jim Norton: Mouthful of Shame	3-14-2017	2017	TV-MA	Stand-Up Comedy
3998	Movie	100 Meters	03-10-2017	2016	TV-MA	Dramas
3999	Movie	Burning Sands	03-10-2017	2017	TV-MA	Dramas
4000	Movie	The Butterfly's Dream	03-10-2017	2013	TV-PG	Dramas

4001 rows × 6 columns

Now, when we look at our data, we can see that there are no missing or messy values. Lastly, our data is in the tidy data format.