Netflix Movies and TV Shows

Netflix is a well-known media and video streaming platform. They have over 8000 movies or television series available on their site, and as of mid-2021, they have over 200 million subscribers worldwide. This report provides visualisation of Netflix data and its popular in each country.

About Data

The data used in this report comes from https://www.kaggle.com/datasets/shivamb/netflix-shows (https://www.kaggle.com/datasets/shivamb/netflix-shows) and was updated till 2021. This tabular dataset contains 8807 rows of all Netflix movies and TV series, together with 12 columns information such as actors, directors, ratings, release year, duration, and so on.

Attribute	Description
show_id	Unique ID for every Movie / Tv Show
type	Identifier - A Movie or TV Show
title	Title of the Movie / Tv Show
director	Director of the Movie
cast	Actors involved in the movie / show
country	Country where the movie / show was produced
date_added	Date it was added on Netflix
release_year	Actual Release year of the move / show
rating	TV Rating of the movie / show
duration	Total Duration - in minutes or number of seasons
listed_in	Genere
description	The summary description

Netflix Data Set

> I've imported 4 libraries: * Numpy is used for working with arrays and maths operations * Pandas is used for manipulating data and for all plots not created with seaborn. * Matplotlib is used to create pie plot and editing the visulisation of plots. * Seaborn is used to create the Heatmap, countplot plot and bar plot.

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Read data

In [ ]:

```
# read data
df = pd.read_csv('netflix_titles.csv')

#see the first 5 rows of data table
df.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA

Cleaning data

> I use Isnull() function to find missing data in each column then removes the rows that contains NULL values

In [ ]:

```
# counting null data
df.isnull().sum()
```

Out[ ]:

```
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
```

In [ ]:

```
# Replacments

df['director'].replace(np.nan, 'NaN', inplace  = True)
df['cast'].replace(np.nan, 'NaN', inplace  = True)
df['country'] = df['country'].fillna(df['country'].mode()[0])


# removes the rows that contains NULL values

df.dropna(inplace  = True)

# Drop Duplicates

df.drop_duplicates(inplace= True)
```

In [ ]:

```
# counting null data again to check if there is any null value in data set
df.isnull().sum()
```

Out[ ]:

```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

Null rows is clear. Now we need to convert the date_added as a date time object

In [ ]:

```
df['date_added'] = pd.to_datetime(df['date_added'])
df.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	2021	TV-MA
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	United States	2021-09-24	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA

The distribution of TV shows and Movies on Netflix

> First, let's explore: Will Netlix has more TV shows or more Movies? To see that I will use Pandas groupby to group my data by 'type' and use count() function to calculate the number of products in each type, then in turn divide it to the total number of both types, and times 100 to take the percentage. To plot the value, I use mathplot pie chart.

In [ ]:

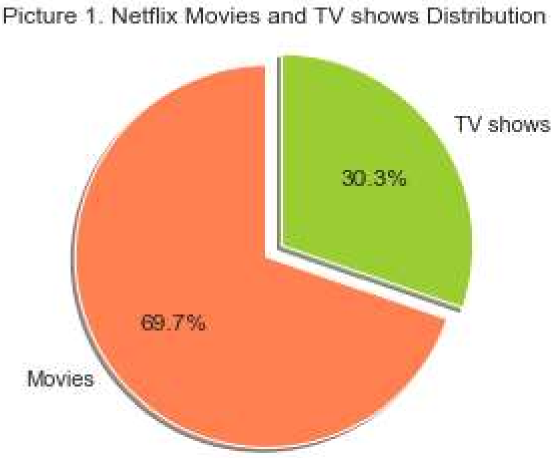
```
x=df.groupby(['type'])['type'].count() # calculate the number of products i
n each type
y=len(df)                              # the number of all movies and TV sh
ows in data set
percentage = ((x/y)).round(3)*100      #calculate percentage of each type

ratio = pd.DataFrame(percentage).T
```

```
In [ ]: # Using bar plot to indicate the distribution of TV shows and Movies

mylabels = ["Movies", "TV shows"]
colors = ["coral","yellowgreen"]
explode = (0.1, 0)

plt.pie(np.array(ratio).ravel(), explode=explode, labels = mylabels, colors
= colors, autopct='%1.1f%%', shadow=True, startangle=90)
plt.title('Picture 1. Netflix Movies and TV shows Distribution')
plt.axis('equal')
plt.show()
```



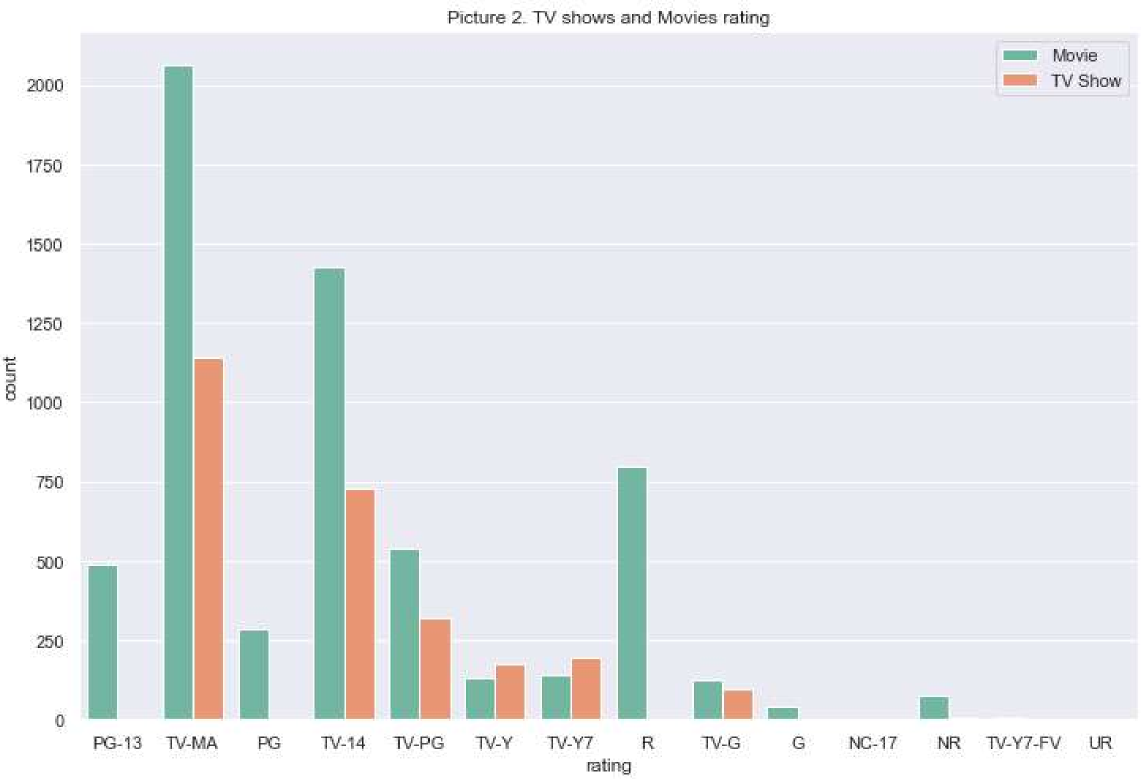
According to the Picture 1, it is clear that there are more movies (69.7%) on Netflix rather than TV shows (30.3%)

> Now, we will dig more into TV shows and Movies to see what is the most rating type by using seaborn countplot

```
In [ ]: plt.figure(figsize=(12,8))
        sns.set(style="darkgrid")

        rating_plot = sns.countplot(df['rating'], hue='type', data=df, palette="Set
        2").set(title='Picture 2. TV shows and Movies rating')
        plt.legend(loc='upper right')
```

```
Out[ ]: <matplotlib.legend.Legend at 0x220220de820>
```



We can see in Picture 2, the 'TV-MA' (Mature Audiences Only) classification is used in the majority of films and TV shows. This program is specifically designed to be viewed by adults and therefore may be unsuitable for children under 17.

The second largest is 'TV-14,' which stands for programming that may be unsuitable for minors under the age of 14.

The third most common movie is the well-known 'R' rating. The Motion Picture Association of America defines an R-rated picture as one that contains material that may be inappropriate for minors under the age of 17; the MPAA states, "Under 17 needs accompanying parent or adult guardian."

Meanwhile, TV-PG (TV Parental Guidelines) is the third common rating in TV shows which contains material that parents may find unsuitable for younger children.

Top countries have content produces on Netflix

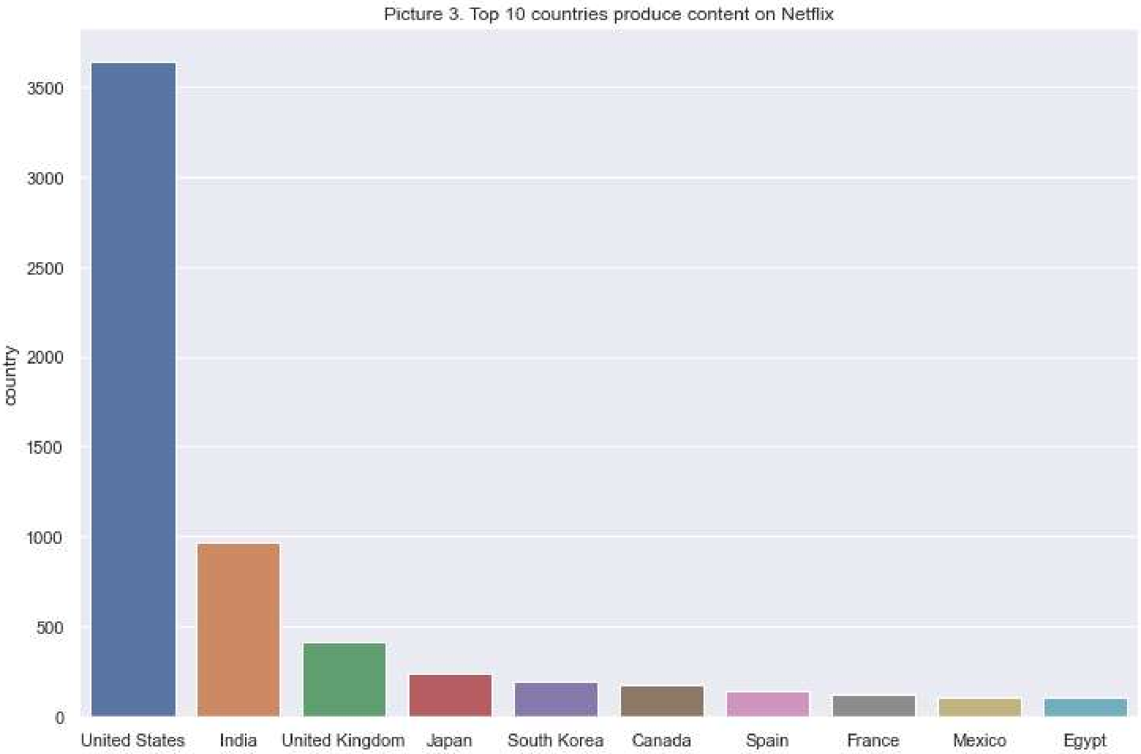
> To see which country produces more content on Netflix, I use value_counts() and plot the value by seaborn barplot

```
In [ ]: count_country = df['country'].value_counts().head(10)
        count_country
```

```
Out[ ]: United States     3638
        India              972
        United Kingdom     418
        Japan              243
        South Korea        199
        Canada             181
        Spain              145
        France             124
        Mexico             110
        Egypt              106
        Name: country, dtype: int64
```

```
In [ ]: plt.figure(figsize=(12,8))
        sns.barplot(x = count_country.index, y=count_country, data=df).set(title='P
        icture 3. Top 10 countries produce content on Netflix')
```

```
Out[ ]: [Text(0.5, 1.0, 'Picture 3. Top 10 countries produce content on Netflix')]
```



In Picture 3, we can see the US produces the most content on Netflix which is 3638. India and the UK is far more behind as producers of content which accounts for 972 and 418, respectively. It is resonable as Netlix is a US company

The number of Movies and TV shows added on Netflix through years

> Now the question is: Is there a growth in the number of movies/TV series over time? What about movies and television shows on their own?

> I add month and day information to my current data frame, which presently just contains dates. I'll then need to filter my data such that I deal with TV Show and Movie data separately. I accomplish this by constructing a dataframe from the entire dataset and choosing just the rows where the type == "TV Show" and type == "Movie". The data is then grouped by added year, and the data frame Movies and TV shows are selected, and the value_counts() function is applied to them. This tells me the number of TV shows or Movies was added on Netlix for each year.

In [ ]:

```
# Add date columns to datetime object

df['added_year'] = df['date_added'].dt.year
df['added_month'] = df['date_added'].dt.month
df['month_name_added'] = df['date_added'].dt.month_name()
```

In [ ]:

```
# Filtering data from Tv Show and Movie
df_tv = df[df["type"] == "TV Show"]
df_movies = df[df["type"] == "Movie"]
```

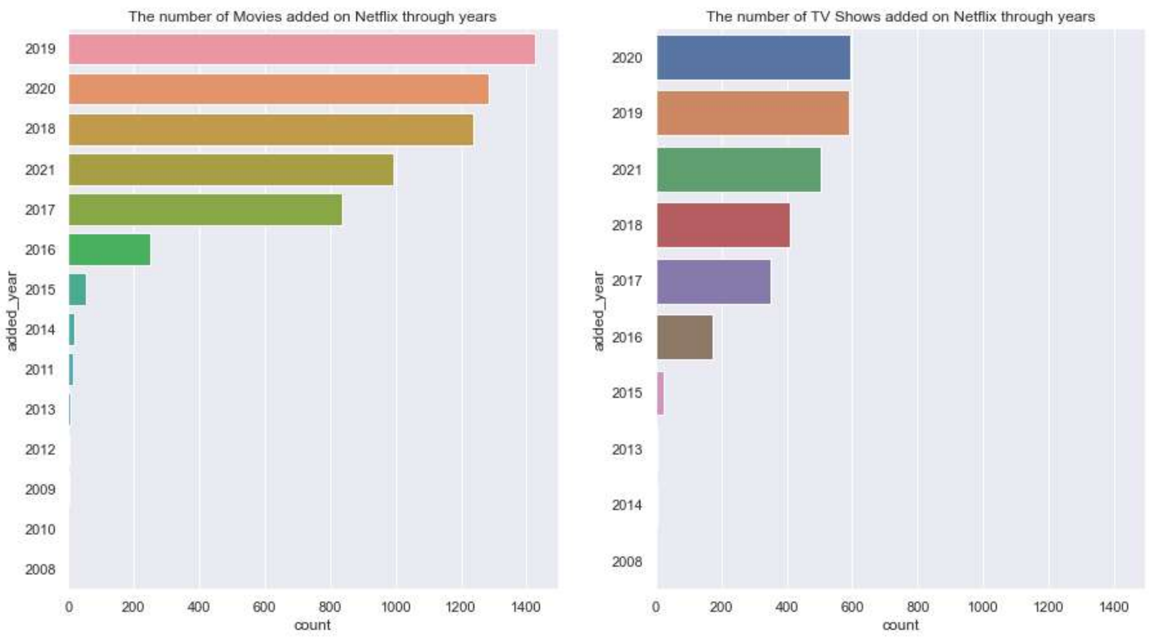
```
In [ ]: fig, axes = plt.subplots(1, 2, sharex=True, figsize=(15,8))
        fig.suptitle('Picture 4. The number of Movies and TV Shows added on Netflix through years')

        # Movies
        sns.countplot(ax=axes[0], y ='added_year', data = df_movies, order = df_movies['added_year'].value_counts().index[0:15])
        axes[0].set_title("The number of Movies added on Netflix through years")

        # TV Shows
        sns.countplot(ax=axes[1], y ='added_year', data = df_tv, order = df_tv['added_year'].value_counts().index[0:15])
        axes[1].set_title("The number of TV Shows added on Netflix through years")
```

```
Out[ ]: Text(0.5, 1.0, 'The number of TV Shows added on Netflix through years')
```

Picture 4. The number of Movies and TV Shows added on Netflix through years



The number of Movies added on Netflix through years

The number of TV Shows added on Netflix through years

Now we will see exactly how many Movies and TV shows were added on Netflix over time

In [ ]:

```
type_by_year = df.groupby(['type','added_year']).count()['date_added']

# unstack to present data with each different data variable in a separate c
olumn
unstacked = type_by_year.unstack(level=0)
unstacked
```

Out[ ]:

type	Movie	TV Show
added_year		
2008	1.0	1.0
2009	2.0	NaN
2010	1.0	NaN
2011	13.0	NaN
2012	3.0	NaN
2013	6.0	5.0
2014	19.0	5.0
2015	56.0	26.0
2016	251.0	175.0
2017	836.0	349.0
2018	1237.0	411.0
2019	1424.0	592.0
2020	1284.0	595.0
2021	993.0	505.0

As it can be seen from Picture 4, both types climbs year after year until it reaches a high in 2019 with over 1400 new films and nearly 600 new TV shows uploaded to the Netflix database, after which it begins to decline.

Genre correlation

> Let's take a look more about each genre. I want to see the relationship between each category in a type. I use function to allow the same piece of code to run two times: one for TV shows and another one for Movies, which helps me break long programs up into smaller components. Then I use seaborn heatmap to indicate the relationship as covered in Module 6. The greater the association, the brighter the color.

In [ ]:

```
# Genres
from sklearn.preprocessing import MultiLabelBinarizer # to encode multiple labels per instance

# Function
def heatmap(df, genre):
    df['genre'] = df['listed_in'].apply(lambda x : x.replace(' ,',',').replace(', ',',').split(','))

    df_genre = df['genre']
    multi_lable = MultiLabelBinarizer()

    # Get correlation of genre
    res = pd.DataFrame(multi_lable.fit_transform(df_genre), columns = multi_lable.classes_, index = df_genre.index)
    corr = res.corr()

    # Create a mask for the upper triangle
    #If passed, data will not be shown in cells where mask is True. Cells with missing values are automatically masked.

    mask = np.zeros_like(corr, dtype = np.bool)
    mask[np.triu_indices_from(mask)] = True


    # Color bar range from -0.3 to 0.3
    plot_heatmap = sns.heatmap(corr, vmin=-0.3, vmax=0.3, mask=mask, square=True, linewidths=1.5)

    plt.show()
```

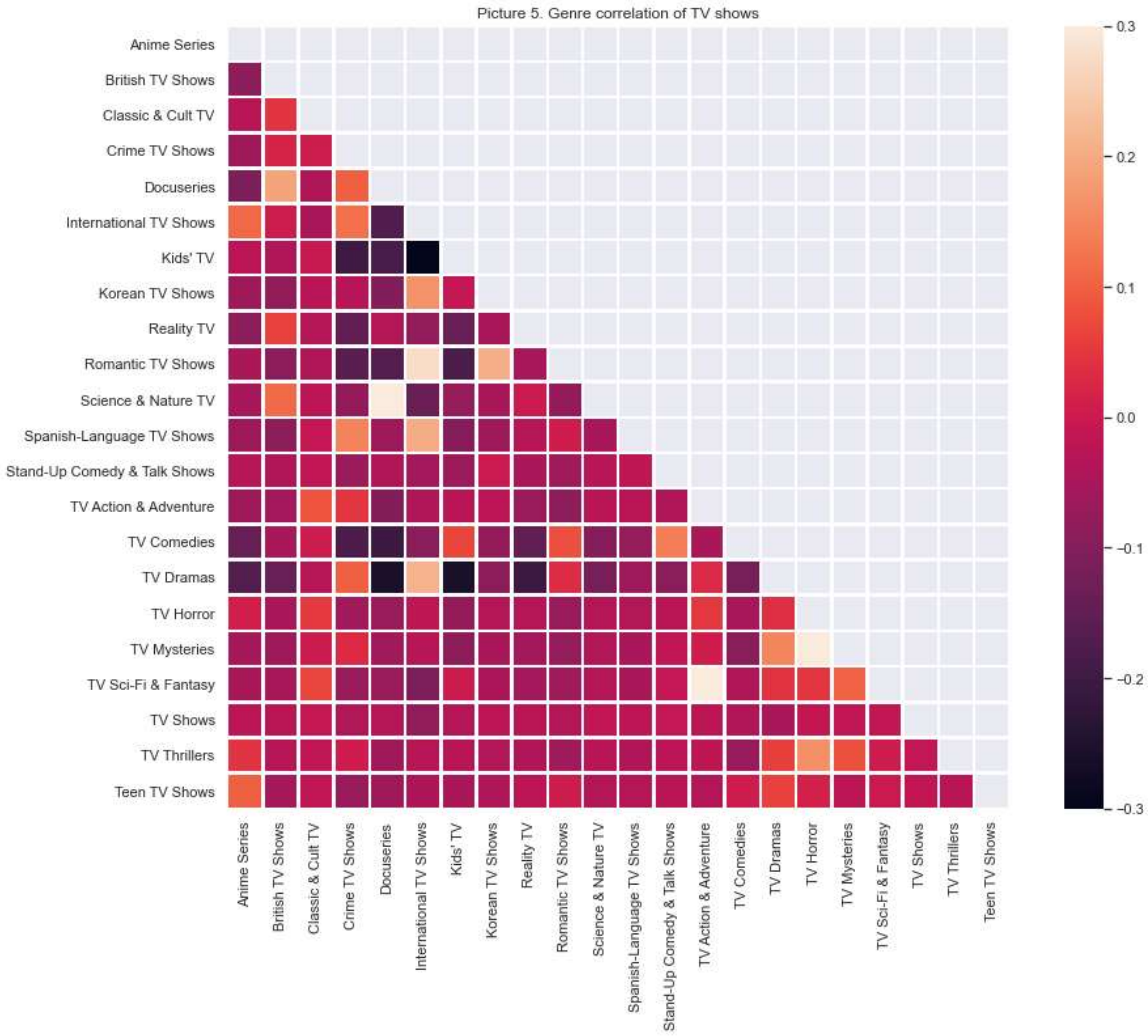
In [ ]:

```
# Plotting heatmap for TV show

plt.figure(figsize = (15,11))
plt.title('Picture 5. Genre correlation of TV shows')
heatmap(df_tv, 'TV Show')
plt.show()
```

```
<ipython-input-65-22a6fc3b709c>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['genre'] = df['listed_in'].apply(lambda x : x.replace(' ,',',').repla
ce(', ',',').split(','))
```



The Netflix TV Shows Dataset has 22 different categories. It can be seen in Picture 5 that TV Sci-Fi & Fantasy is common in TV Action & Adventure. Meanwhile, Kid's TV is uncommon in International TV shows.

In [ ]:

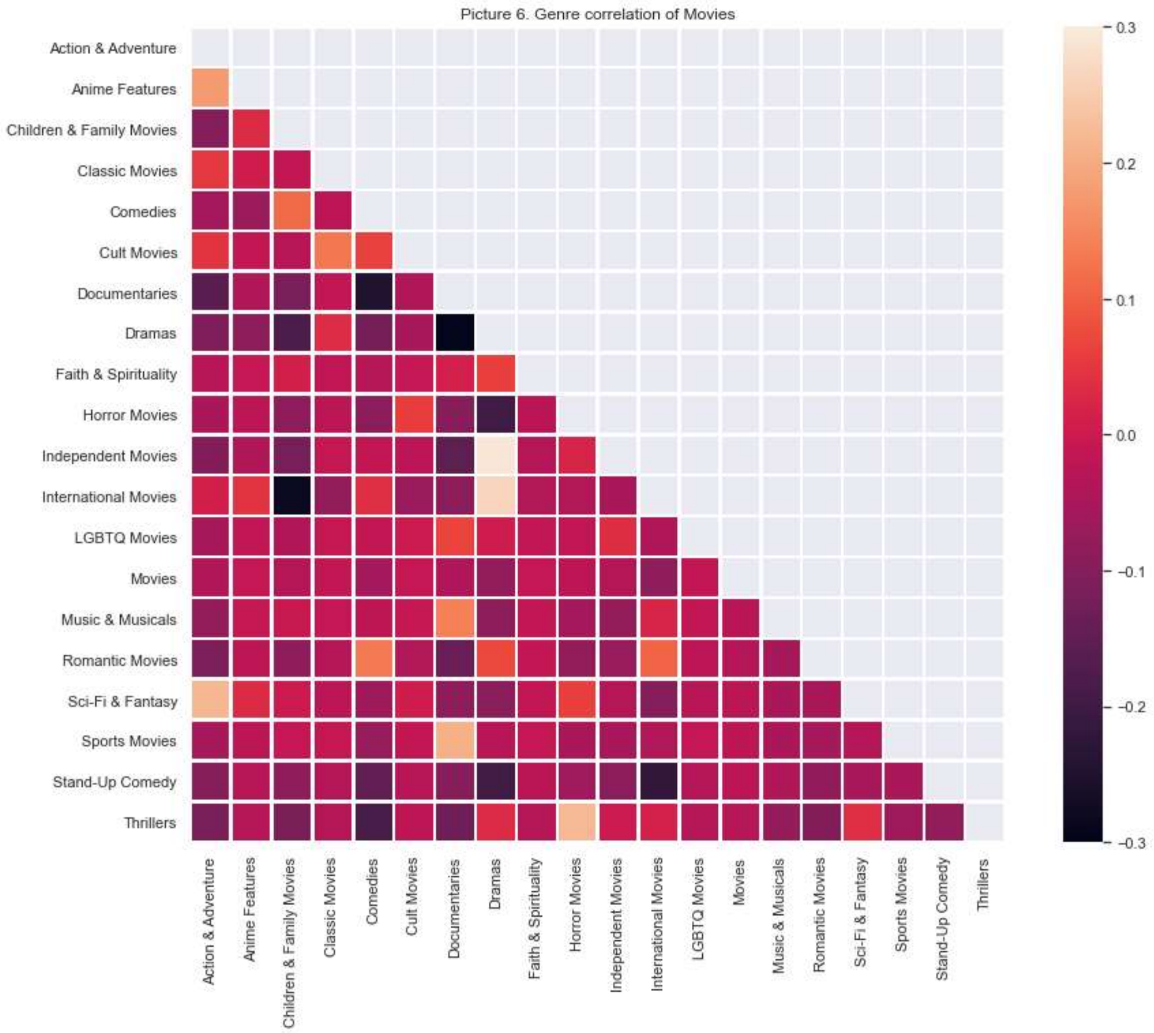
```
# Plotting heatmap for Movie

plt.figure(figsize = (15,11))
plt.title('Picture 6. Genre correlation of Movies')
heatmap(df_movies, 'Movie')

plt.show()
```

```
<ipython-input-65-22a6fc3b709c>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['genre'] = df['listed_in'].apply(lambda x : x.replace(' ,',',').repla
ce(', ',',').split(','))
```



The Netflix TV Shows Dataset has 20 different categories. According to Picture 6, it's interesting to note that most independent films are dramas. Another finding is that International Movies in the Children and Famaily's category are uncommon.