Importing Libraries

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

Data Loading

```
df=pd.read_csv("C:\\Users\\Kalpesh\\Downloads\\netflix1.csv")
```

Data Overview

```
df
```

```
     show_id     type                                title
director  \
0         s1    Movie               Dick Johnson Is Dead  Kirsten
Johnson
1         s3  TV Show                          Ganglands  Julien
Leclercq
2         s6  TV Show                      Midnight Mass    Mike
Flanagan
3        s14    Movie  Confessions of an Invisible Girl    Bruno
Garotti
4         s8    Movie                            Sankofa     Haile
Gerima
...      ...      ...                               ...               .
..
8785   s8797  TV Show                         Yunus Emre        Not
Given
8786   s8798  TV Show                          Zak Storm        Not
Given
8787   s8801  TV Show                 Zindagi Gulzar Hai        Not
Given
8788   s8784  TV Show                               Yoko        Not
Given
8789   s8786  TV Show                                YOM        Not
Given

             country  date_added  release_year rating   duration  \
0     United States   9/25/2021          2020  PG-13     90 min
1            France   9/24/2021          2021  TV-MA   1 Season
2     United States   9/24/2021          2021  TV-MA   1 Season
3            Brazil   9/22/2021          2021  TV-PG     91 min
4     United States   9/24/2021          1993  TV-MA    125 min
...             ...         ...           ...    ...        ...
8785         Turkey   1/17/2017          2016  TV-PG  2 Seasons
8786  United States   9/13/2018          2016  TV-Y7  3 Seasons
```

```
8787       Pakistan  12/15/2016          2012  TV-PG   1 Season
8788       Pakistan   6/23/2018          2016   TV-Y   1 Season
8789       Pakistan    6/7/2018          2016  TV-Y7   1 Season

                                                  listed_in
0                                           Documentaries
1     Crime TV Shows, International TV Shows, TV Act...
2                   TV Dramas, TV Horror, TV Mysteries
3                   Children & Family Movies, Comedies
4      Dramas, Independent Movies, International Movies
...                                                  ...
8785                  International TV Shows, TV Dramas
8786                                          Kids' TV
8787  International TV Shows, Romantic TV Shows, TV ...
8788                                          Kids' TV
8789                                          Kids' TV

[8790 rows x 10 columns]
```

```
df.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'country',
'date_added',
       'release_year', 'rating', 'duration', 'listed_in'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8790 non-null   object
 1   type          8790 non-null   object
 2   title         8790 non-null   object
 3   director      8790 non-null   object
 4   country       8790 non-null   object
 5   date_added    8790 non-null   object
 6   release_year  8790 non-null   int64 
 7   rating        8790 non-null   object
 8   duration      8790 non-null   object
 9   listed_in     8790 non-null   object
dtypes: int64(1), object(9)
memory usage: 686.8+ KB
```

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
df.describe()
```

```
       release_year
count   8790.000000
mean    2014.183163
std        8.825466
min     1925.000000
25%     2013.000000
50%     2017.000000
75%     2019.000000
max     2021.000000
```

```
df.isnull().sum()
```

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

```
df.isna().sum()
```

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

Data Cleaning

```
df.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'country',
'date_added',
       'release_year', 'rating', 'duration', 'listed_in'],
      dtype='object')
```

```
#changing the datatype of date added from object to datetime format

df['date_added']=df['date_added'].astype('datetime64[ns]')
```

```
#ensuring datatype change or not
df['date_added'].dtypes
```

```
dtype('<M8[ns]')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       8790 non-null   object        
 1   type          8790 non-null   object        
 2   title         8790 non-null   object        
 3   director      8790 non-null   object        
 4   country       8790 non-null   object        
 5   date_added    8790 non-null   datetime64[ns]
 6   release_year  8790 non-null   int64         
 7   rating        8790 non-null   object        
 8   duration      8790 non-null   object        
 9   listed_in     8790 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 686.8+ KB
```

```
df.dropna(subset=['director','country','title'], inplace=True)
```

```
df
```

```
     show_id     type                                title  
director  \
0         s1    Movie              Dick Johnson Is Dead  Kirsten 
Johnson
1         s3  TV Show                          Ganglands  Julien 
Leclercq
2         s6  TV Show                      Midnight Mass    Mike 
Flanagan
3        s14    Movie  Confessions of an Invisible Girl    Bruno 
Garotti
4         s8    Movie                            Sankofa     Haile 
Gerima
...      ...      ...                               ...               .
..
8785   s8797  TV Show                         Yunus Emre        Not 
Given
8786   s8798  TV Show                          Zak Storm        Not 
Given
8787   s8801  TV Show                 Zindagi Gulzar Hai        Not 
Given
8788   s8784  TV Show                               Yoko        Not 
Given
```

```
8789   s8786  TV Show                                YOM        Not
Given

            country date_added  release_year rating   duration  \
0     United States 2021-09-25          2020  PG-13     90 min
1            France 2021-09-24          2021  TV-MA   1 Season
2     United States 2021-09-24          2021  TV-MA   1 Season
3            Brazil 2021-09-22          2021  TV-PG     91 min
4     United States 2021-09-24          1993  TV-MA    125 min
...             ...        ...           ...    ...        ...
8785         Turkey 2017-01-17          2016  TV-PG  2 Seasons
8786  United States 2018-09-13          2016  TV-Y7  3 Seasons
8787       Pakistan 2016-12-15          2012  TV-PG   1 Season
8788       Pakistan 2018-06-23          2016   TV-Y   1 Season
8789       Pakistan 2018-06-07          2016  TV-Y7   1 Season

                                                listed_in
0                                          Documentaries
1     Crime TV Shows, International TV Shows, TV Act...
2                    TV Dramas, TV Horror, TV Mysteries
3                    Children & Family Movies, Comedies
4      Dramas, Independent Movies, International Movies
...                                                 ...
8785                  International TV Shows, TV Dramas
8786                                          Kids' TV
8787  International TV Shows, Romantic TV Shows, TV ...
8788                                          Kids' TV
8789                                          Kids' TV

[8790 rows x 10 columns]
```

Data Extraction

```
df.columns

Index(['show_id', 'type', 'title', 'director', 'country',
'date_added',
       'release_year', 'rating', 'duration', 'listed_in'],
      dtype='object')
```

Count of Movies , TV Show

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

type
Movie      6126
TV Show    2664
Name: count, dtype: int64
```

```
type_counts=df['type'].value_counts()
type_counts
```

```
type
Movie      6126
TV Show    2664
Name: count, dtype: int64
```

Top 10 Directors

```
#top 10 directors
top_10_director=df['director'].value_counts().head(10)
print(top_10_director)
print("-----------------------------------------------------------------
-----")

# here is not given is high so filtering
filtered=df[df['director']!= 'Not Given']
top_5_director=filtered['director'].value_counts().head(5)
top_5_director
```

```
director
Not Given                  2588
Rajiv Chilaka                20
Alastair Fothergill          18
Raúl Campos, Jan Suter       18
Marcus Raboy                 16
Suhas Kadav                  16
Jay Karas                    14
Cathy Garcia-Molina          13
Jay Chapman                  12
Martin Scorsese              12
Name: count, dtype: int64
----------------------------------------------------------------------

director
Rajiv Chilaka              20
Raúl Campos, Jan Suter     18
Alastair Fothergill        18
Suhas Kadav                16
Marcus Raboy               16
Name: count, dtype: int64
```

Month,Year(Feature Engineering)

```
#monthly data
df['month']=df['date_added'].dt.month

#yearly data
df['year']=df['date_added'].dt.year
```

Monthly release Movies,TV Show

```
monthly_added_movie=df[df['type']== 'Movie']
Monthly_release_movies=monthly_added_movie.groupby('month').size().res
et_index(name='count')
print("monthly_added_movie")
print(Monthly_release_movies)


#monthly release Tv Show
monthly_release_Tv_show=df[df['type']=='TV Show']
monthly_release_tv_show=monthly_release_Tv_show.groupby('month').size(
).reset_index(name='count')
print("monthly_release_Tv_show")
print(monthly_release_tv_show)
```

```
monthly_added_movie
    month  count
0       1    545
1       2    382
2       3    528
3       4    549
4       5    439
5       6    492
6       7    565
7       8    518
8       9    518
9      10    545
10     11    498
11     12    547
monthly_release_Tv_show
    month  count
0       1    192
1       2    180
2       3    213
3       4    214
4       5    193
5       6    236
6       7    262
7       8    236
8       9    251
9      10    215
10     11    207
11     12    265
```

count of content added duration Wise(Top 5)

```
top5_duration_added=df['duration'].value_counts().head(5)
top5_duration_added
```

```
duration
1 Season     1791
2 Seasons     421
3 Seasons     198
90 min        152
94 min        146
Name: count, dtype: int64
```

yearly release Movies,TV Show

```
yearly_release_movies=df[df['type']=='Movie']
movies_by_year =
yearly_release_movies.groupby('year').size().reset_index(name='count')
print("Yearly Added Movies")
print(movies_by_year)

print("----------------------------------------------------------------
--------------------")
yearly_release_tv_show=df[df['type']=='TV Show']
tv_shows_by_year=
yearly_release_tv_show.groupby('year').size().reset_index(name='count'
)
print("yearly_release_tv_show")
print(tv_shows_by_year)
```

```
Yearly Added Movies
    year  count
0   2008      1
1   2009      2
2   2010      1
3   2011     13
4   2012      3
5   2013      6
6   2014     19
7   2015     56
8   2016    251
9   2017    836
10  2018   1237
11  2019   1424
12  2020   1284
13  2021    993
--------------------------------------------------------------------------
-------------
yearly_release_tv_show
   year  count
0  2008      1
1  2013      5
2  2014      5
```

```
3  2015     26
4  2016    175
5  2017    349
6  2018    411
7  2019    592
8  2020    595
9  2021    505
```

Top 10 Ratings

```
top_10_ratings=df['rating'].value_counts().head(10)
top_10_ratings
```

```
rating
TV-MA    3205
TV-14    2157
TV-PG     861
R         799
PG-13     490
TV-Y7     333
TV-Y      306
PG        287
TV-G      220
NR         79
Name: count, dtype: int64
```

Top 10 countries which added content most

```
top_10_countries=df['country'].value_counts().head(10)
top_10_countries
```

```
country
United States     3240
India             1057
United Kingdom     638
Pakistan           421
Not Given          287
Canada             271
Japan              259
South Korea        214
France             213
Spain              182
Name: count, dtype: int64
```

Top 10 Movie Genres

```
top10_Movie_genres=df[df['type']=='Movie']
['listed_in'].value_counts().head(10)
top10_Movie_genres
```

```
listed_in
Dramas, International Movies                          362
Documentaries                                         359
Stand-Up Comedy                                       334
Comedies, Dramas, International Movies                274
Dramas, Independent Movies, International Movies      252
Children & Family Movies                              215
Children & Family Movies, Comedies                    201
Documentaries, International Movies                   186
Dramas, International Movies, Romantic Movies         180
Comedies, International Movies                        176
Name: count, dtype: int64
```

Top 10 TV Show Genres

```
top10_tv_show_genres=df[df['type']=='TV Show']
['listed_in'].value_counts().head(10)
top10_tv_show_genres
```

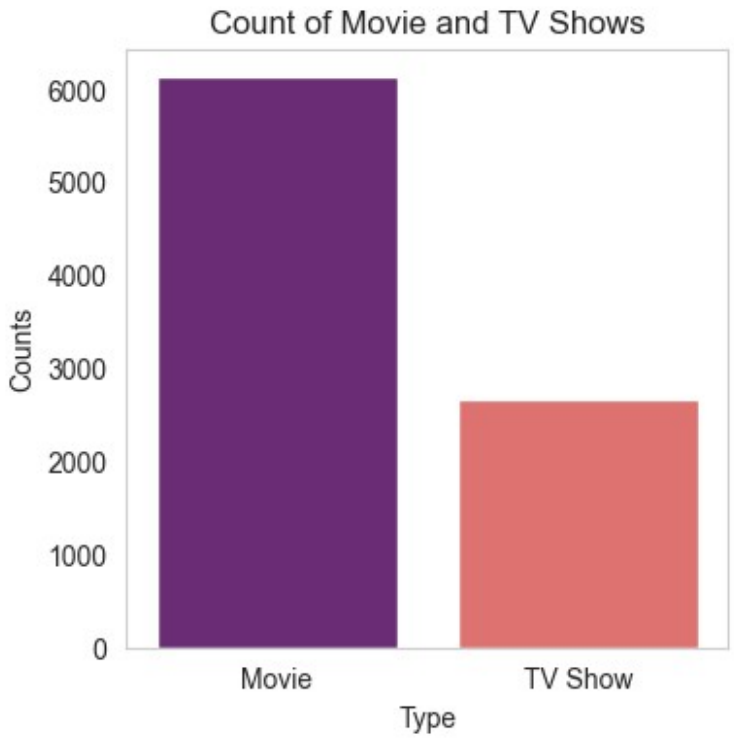
```
listed_in
Kids' TV                                                  219
International TV Shows, TV Dramas                         121
Crime TV Shows, International TV Shows, TV Dramas         110
Kids' TV, TV Comedies                                      97
Reality TV                                                 95
International TV Shows, Romantic TV Shows, TV Comedies     94
International TV Shows, Romantic TV Shows, TV Dramas       90
Anime Series, International TV Shows                       87
Docuseries                                                 84
TV Comedies                                                68
Name: count, dtype: int64
```

Exploratory Data Analyis

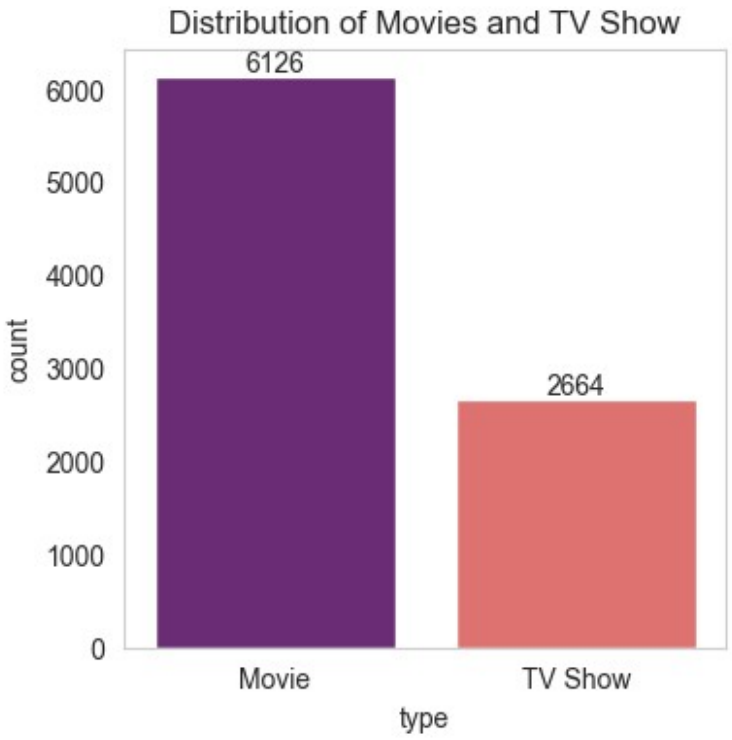
```
import warnings
warnings.filterwarnings("ignore")
```

Count of Movies and TV Show

```
plt.figure(figsize=(4,4))
sns.barplot(x=type_counts.index , y=type_counts.values,
palette='magma')
plt.xlabel("Type")
plt.ylabel("Counts")
plt.title("Count of Movie and TV Shows")
plt.grid(False)
plt.show()
```

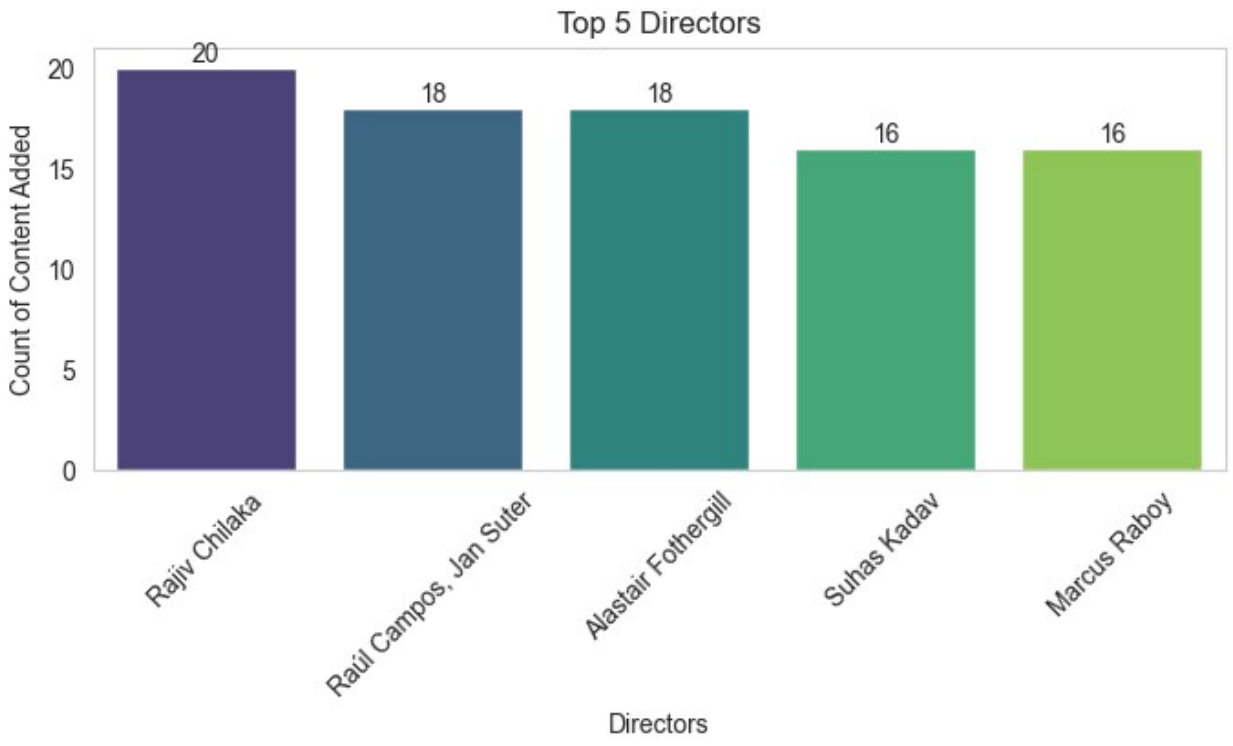



```
plt.figure(figsize=(4,4))
plt.title("Distribution of Movies and TV Show")
count_data=sns.countplot(x='type',data=df,hue='type',palette='magma')
plt.grid(False)
for i in count_data.containers:
    count_data.bar_label(i)
plt.savefig('Distribution_of_Movies_and_TV_Show.jpg')
plt.show()
```



Top 5 Directors

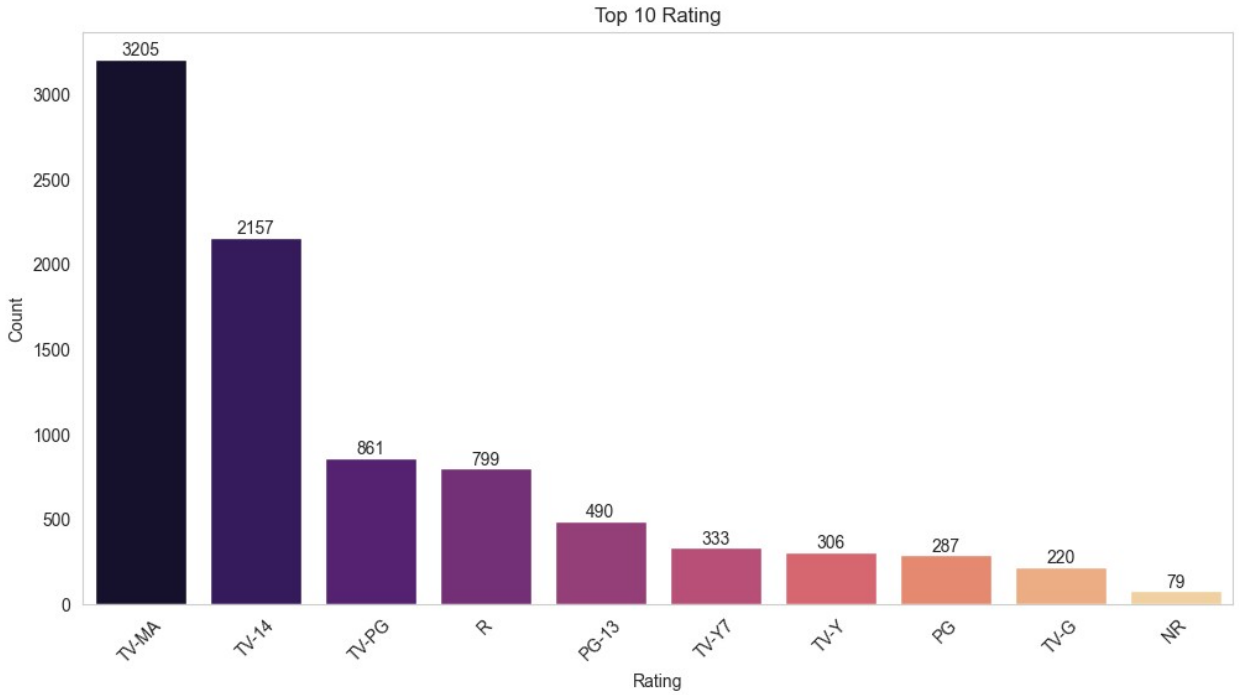
```
plt.figure(figsize=(8,3))
ax=sns.barplot(x=top_5_director.index , y=top_5_director.values ,
palette='viridis')
plt.xlabel('Directors')
plt.ylabel('Count of Content Added')
plt.title('Top 5 Directors')
plt.xticks(rotation=45)
plt.grid(False)
for i in ax.containers:
    ax.bar_label(i)
plt.savefig('Top_5_Directors.jpg')
plt.show()
```



Top 10 Rating

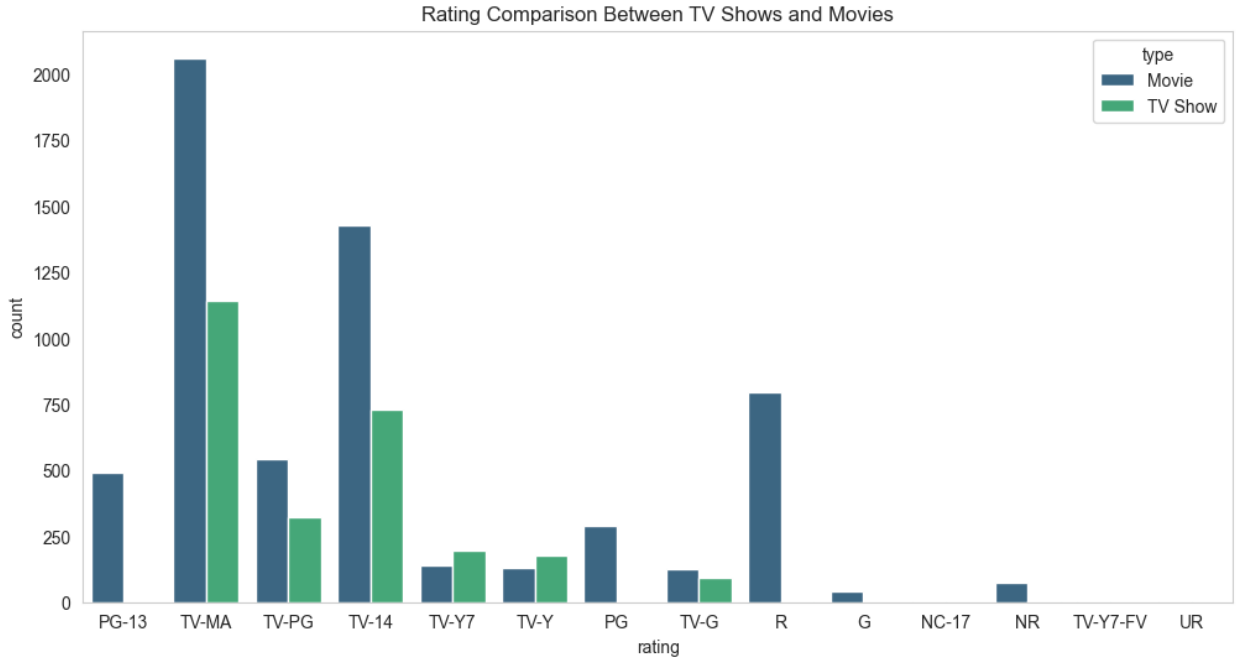
```
plt.figure(figsize=(12,6))
ax=sns.barplot(x=top_10_ratings.index , y=top_10_ratings.values ,
palette='magma')
plt.xlabel("Rating")
plt.ylabel("Count")
plt.title("Top 10 Rating")
plt.xticks(rotation=45)
plt.grid(False)
for i in ax.containers:
    ax.bar_label(i)
plt.savefig('Top_10_Rating.jpg')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



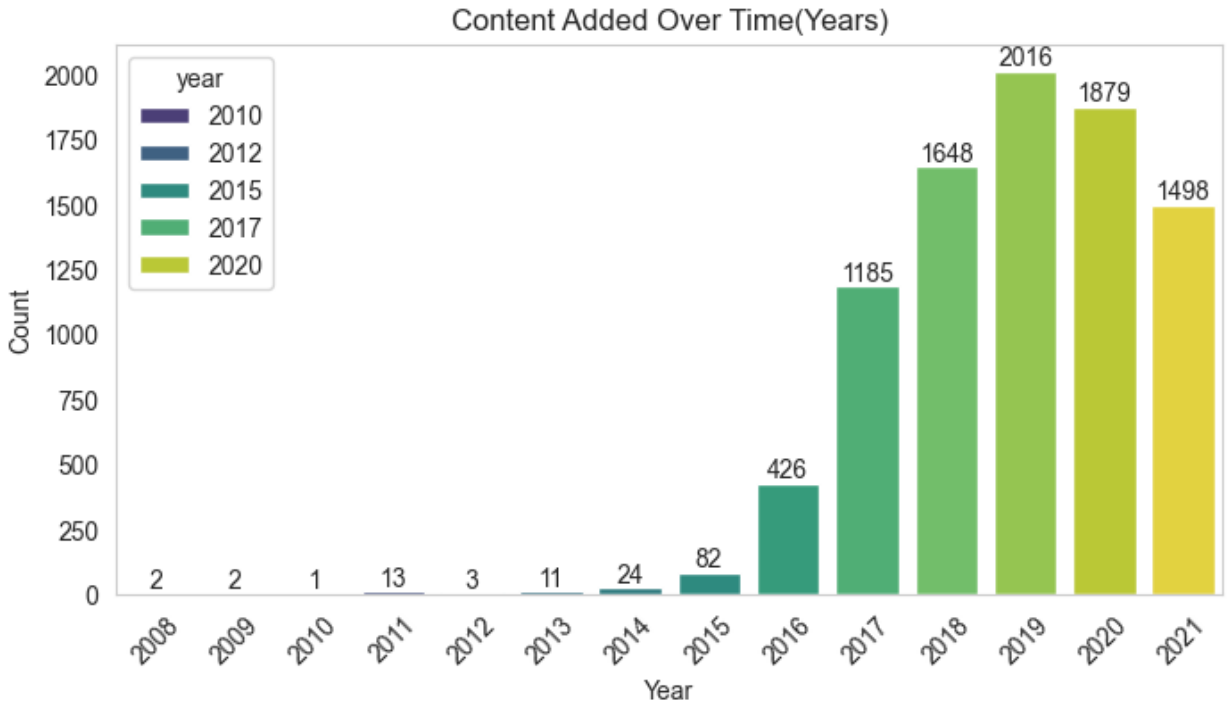
Rating Comparison Between TV Shows and Movies

```
plt.figure(figsize=(12,6))
sns.countplot(x='rating',hue='type',data=df,palette='viridis')
plt.title("Rating Comparison Between TV Shows and Movies")
plt.savefig('Rating_Comparison.jpg')
plt.grid(False)
plt.show()
```



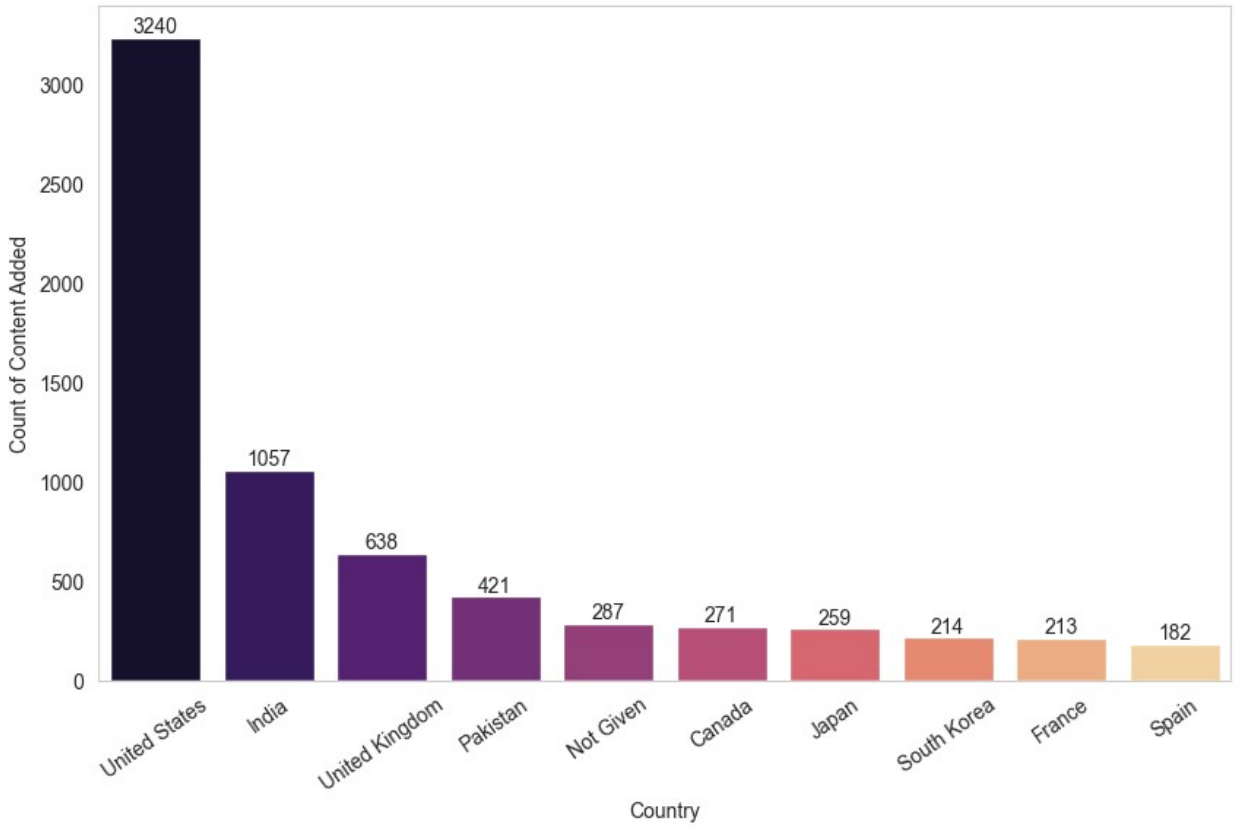
Content Added Over Years

```
plt.figure(figsize=(8, 4))
cnt=sns.countplot(x='year',hue='year',data=df, palette='viridis')
plt.title('Content Added Over Time(Years)')
plt.xlabel('Year')
plt.grid(False)
plt.ylabel('Count')
plt.xticks(rotation=45)
for i in cnt.containers:
    cnt.bar_label(i)
plt.savefig('content_added_over_years.jpg')
plt.show()
```



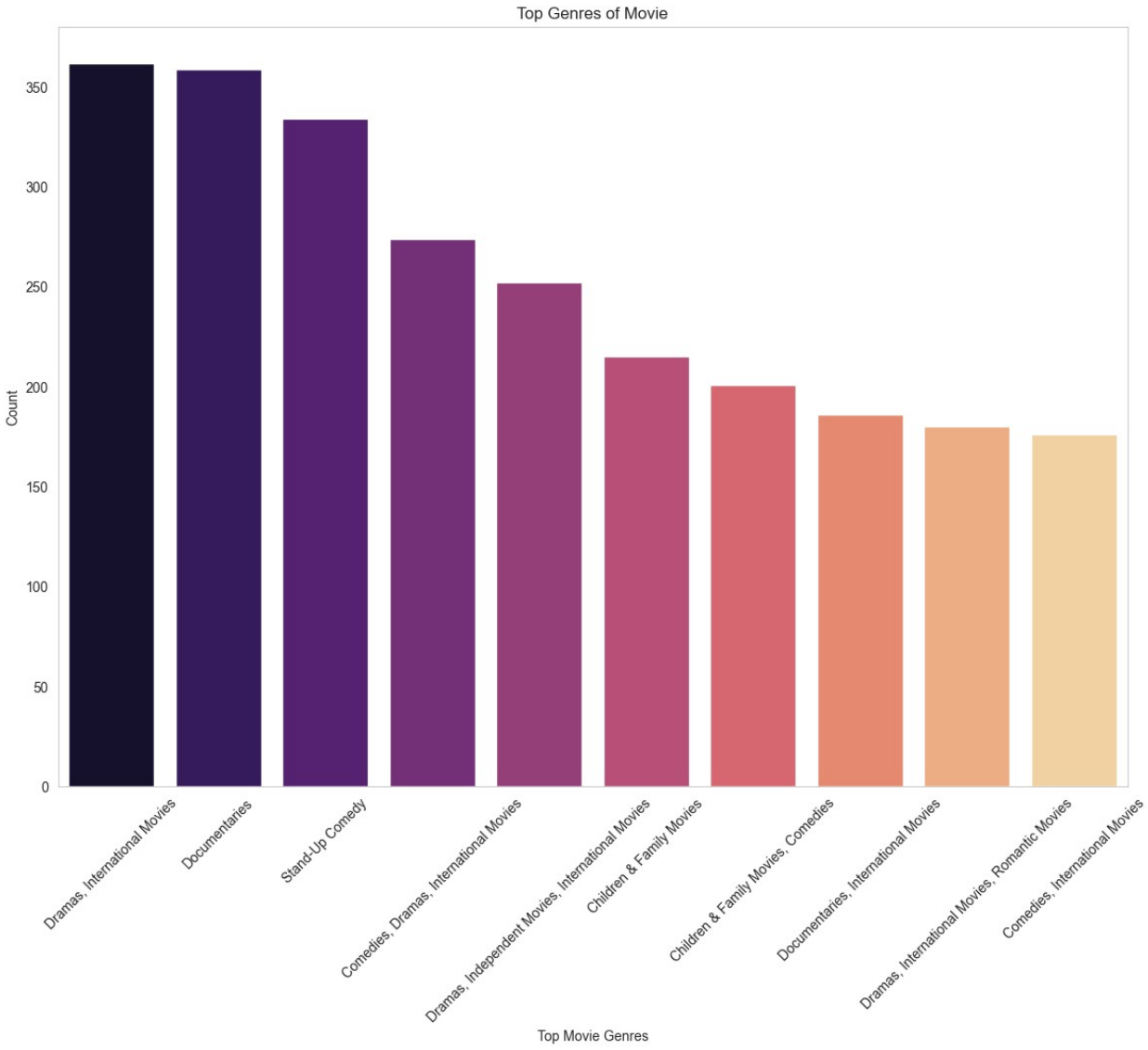
Top 10 Countries

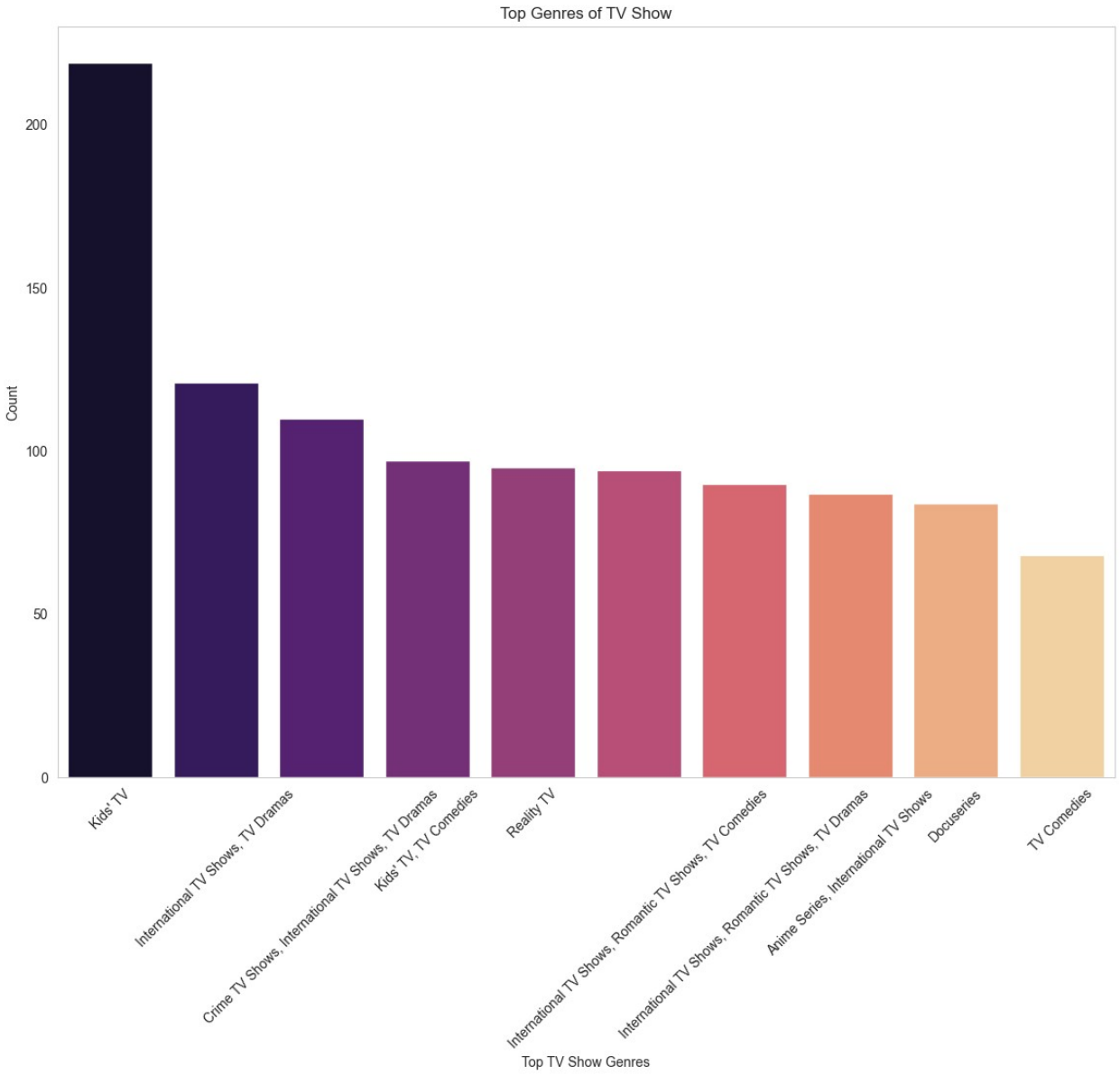
```
plt.figure(figsize=(10,6))
country=sns.barplot(x=top_10_countries.index , y=
top_10_countries.values, palette='magma')
plt.xlabel("Country")
plt.ylabel("Count of Content Added")
plt.xticks(rotation=35)
plt.grid(False)
for i in country.containers:
    country.bar_label(i)
plt.savefig('Top_10_Countries.jpg')
plt.show()
```



Top 10 Movie and TV Show Genres

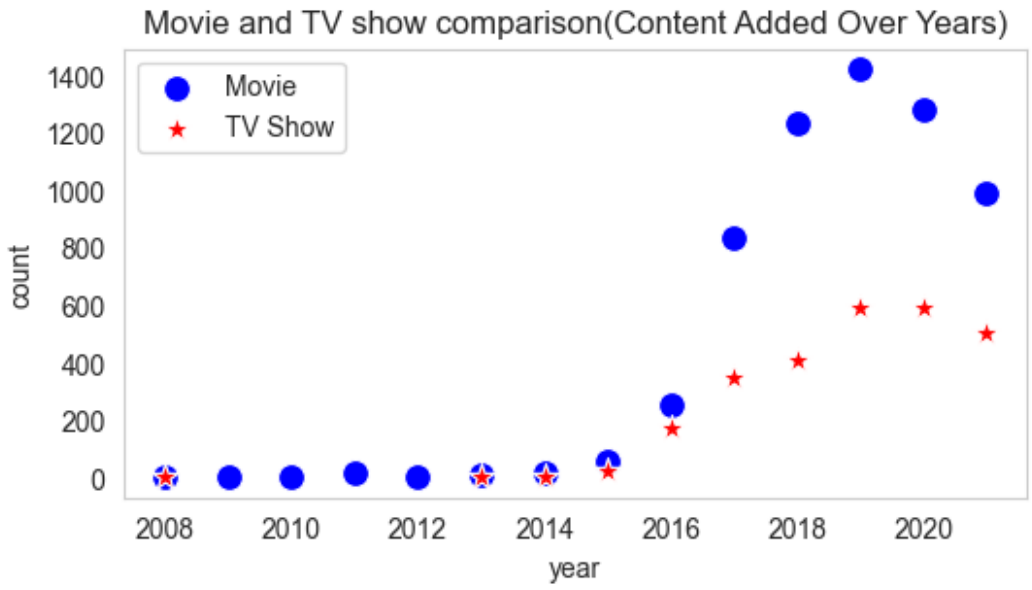
```
plt.figure(figsize=(8,))
plt.title("Top Genres of Movie")
sns.barplot(x=top10_Movie_genres.index , y=top10_Movie_genres.values ,
palette='magma')
plt.xlabel("Top Movie Genres")
plt.xticks(rotation=45)
plt.grid(False)
plt.ylabel("Count")
plt.savefig('top_10_genre_moviess.jpg')
plt.show()


plt.figure(figsize=(14,10))
sns.barplot(x=top10_tv_show_genres.index ,
y=top10_tv_show_genres.values , palette='magma')
plt.title("Top Genres of TV Show")
plt.xlabel("Top TV Show Genres")
plt.ylabel("Count")
plt.grid(False)
plt.xticks(rotation=45)
plt.savefig('top_10_genre_TVs.jpg')
plt.show()
```


Top Genres of Movie
Count
0
50
100
150
200
250
300
350
Dramas, International Movies
Documentaries
Stand-Up Comedy
Comedies, Dramas, International Movies
Dramas, Independent Movies, International Movies
Children & Family Movies
Children & Family Movies, Comedies
Documentaries, International Movies
Dramas, International Movies, Romantic Movies
Comedies, International Movies
Top Movie Genres



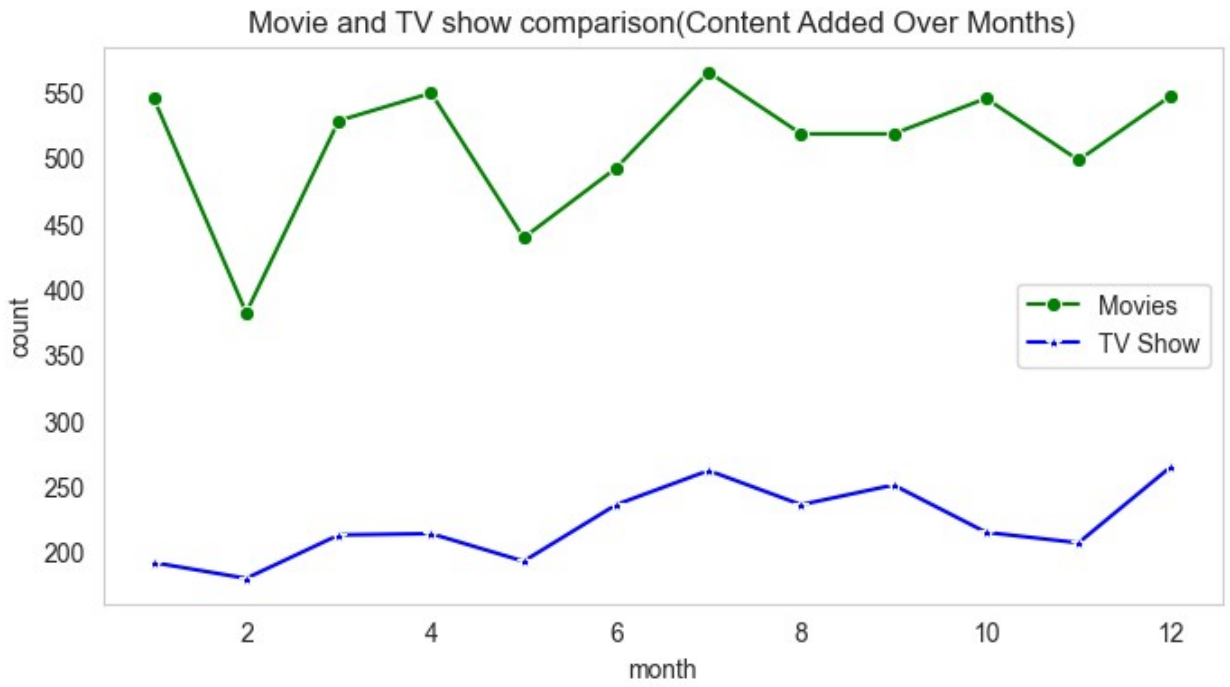
Movie and TV show comparison(Content Added Over Years)

```
plt.figure(figsize=(6,3))
sns.scatterplot(x='year', y='count', data=movies_by_year, marker='o',
color='b',s=100,label='Movie')
sns.scatterplot(x='year' , y='count', data=tv_shows_by_year ,
marker='*' , color='r',s=100, label='TV Show')
plt.title("Movie and TV show comparison(Content Added Over Years)")
plt.grid(False)
plt.savefig('Movie_TV_Comparison_yearly.jpg')
plt.show()
```

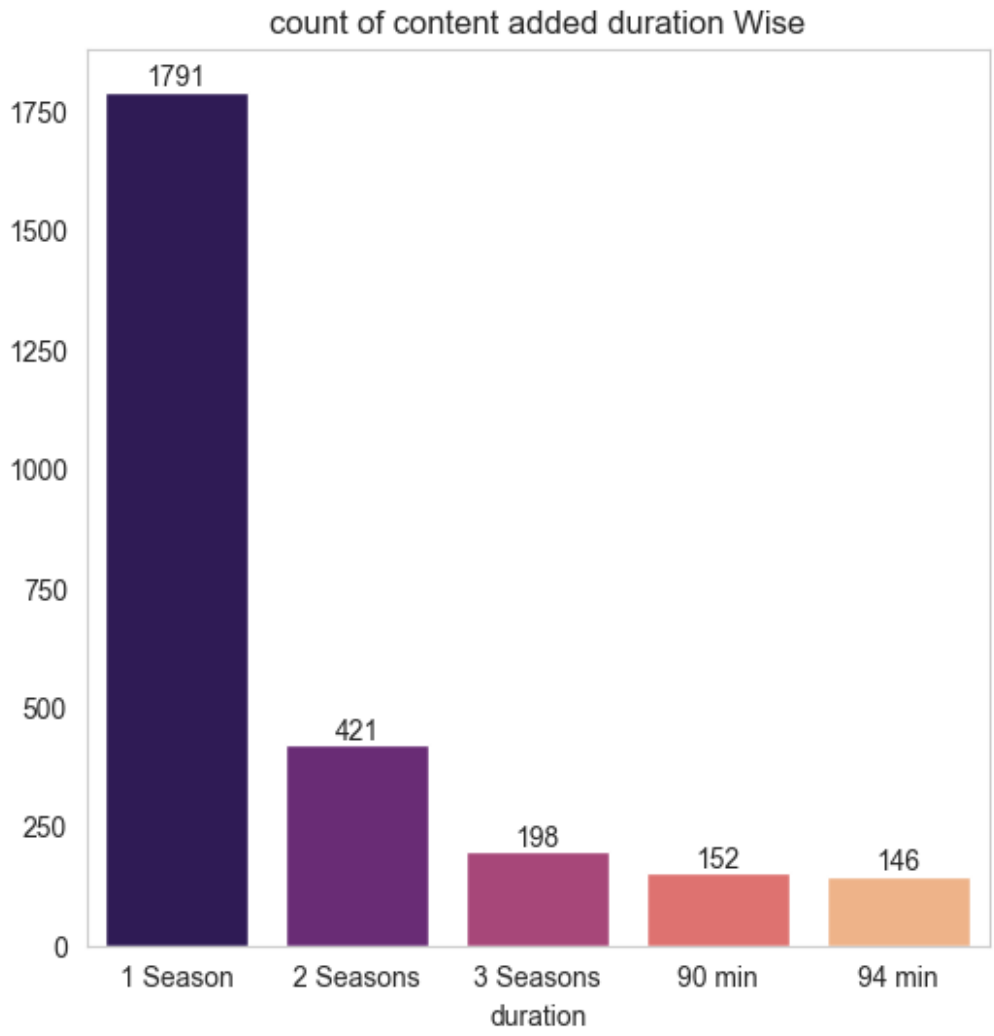


Movie and TV show comparison(Content Added Over Months)

```
plt.figure(figsize=(8,4))
plt.title("Movie and TV show comparison(Content Added Over Months)")
sns.lineplot(x='month' , y='count' , data=Monthly_release_movies  ,
marker='o',label='Movies' ,color='g')
sns.lineplot(x='month' , y='count', data=monthly_release_tv_show,
marker='*' , label='TV Show' , color='b')
plt.grid(False)
plt.savefig('Movie_TV_Comparison_monthly.jpg')
plt.show()
```




```
plt.figure(figsize=(6,6))
plt.title("count of content added duration Wise")
cnt=sns.barplot(x=top5_duration_added.index ,
y=top5_duration_added.values , palette='magma')
plt.grid(False)
for i in cnt.containers:
    cnt.bar_label(i)
plt.savefig('Duration.jpg')
plt.show()
```



In this project we,

1. Cleaned the data by handling missing values, removing duplicates, and converting data types. *Used Feature Engineering for improved model performance
2. Explored the data through various visualizations such as bar plots , countplots , lineplot , scatterplot.
3. Analyzed content trends over time.
4. Identified popular genre
5. analyze top Cities which added most content over years
6. Identified top 5 dirctors
7. compared the Movies , TV Shows which added content over monthly, yearly.
8. compared ratings for movies and tv shows
9. Indentified most content added by duration wise