Loading Libraries and Dataset

In [1]:

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

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprec
ated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [3]:
```
dfx = pd.read_excel('/Movie300 Revised V2.xlsx')
dfx.head()
```

Out[3]:

	Movie_name	Movie_Genre	Movie_Genre_Num	Movie_Certification	Movie_Certification_Num	Release_Date	Release_Month	Release_Month_Nur
0	Kannum Kannum Kollaiyadithaal	Thriller	15	U	2	28 Feb 2020	Feb	
1	Oh My Kadavule	Comedy	14	UA	1	14 Feb 2020	Feb	
2	Psycho	Thriller	15	A	3	24 Jan 2020	Jan	
3	Dharala Prabhu	Comedy	14	UA	1	13 Mar 2020	Mar	
4	Gypsy	Drama	17	UA	1	06 Mar 2020	Mar	

Dataset Cleaning and some preliminary steps

In [4]:
```
dfx['Release_Month'] = dfx['Release_Date'].apply(lambda x: x.split(' ')[1])
dfx['Release_Month'].head()
```

Out[4]:
```
0    Feb
1    Feb
2    Jan
3    Mar
4    Mar
Name: Release_Month, dtype: object
```

In [5]:

```
dfx.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 18 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Movie_name                300 non-null    object 
 1   Movie_Genre               300 non-null    object 
 2   Movie_Genre_Num           300 non-null    int64  
 3   Movie_Certification       300 non-null    object 
 4   Movie_Certification_Num   300 non-null    int64  
 5   Release_Date              300 non-null    object 
 6   Release_Month             300 non-null    object 
 7   Release_Month_Num         300 non-null    int64  
 8   Runtime_Duration          300 non-null    object 
 9   Runtime_Minutes           300 non-null    int64  
 10  Movie_Directors           300 non-null    object 
 11  Music_Directors           298 non-null    object 
 12  Lead_Actor                300 non-null    object 
 13  Lead_Actress              300 non-null    object 
 14  Movie_Critic_Rating       300 non-null    float64
 15  Movie_User_Rating         300 non-null    float64
 16  Movie_Synopsis            300 non-null    object 
 17  Movie_Full_Cast           300 non-null    object 
dtypes: float64(2), int64(4), object(12)
memory usage: 42.3+ KB
```

In [6]:

```
dfx.fillna
```

Out[6]:

```
<bound method DataFrame.fillna of                          Movie_name  ...                                    Movie_Fu
ll_Cast
0    Kannum Kannum Kollaiyadithaal  ...  Dulquer Salmaan, Ritu Varma, Gautham Vasudev M...
1                   Oh My Kadavule  ...            Ashok Selvan, Ritika Singh, Vani Bhojan
2                           Psycho  ...  Udhayanidhi Stalin, Aditi Rao Hydari, Nithya M...
3                   Dharala Prabhu  ...                   Harish Kalyan, Tanya Hope, Vivek
4                            Gypsy  ...        Jiiva, Natasha Singh, Lal Jose, Sunny Wayne
..                             ...  ...                                                ...
295                   Nootrenbadhu  ...  Siddharth, Priya Anand, Nithya Menen, Mouli, G...
296                 Ponnar Shankar  ...  Prashanth, Divya Parameswaran, Pooja Chopra, S...
297               Nadunisi Naaygal  ...         Veera, Sameera Reddy, Deva, Swapna Abraham
298                       Ilaignan  ...  Pa Vijay, Kushboo, Meera Jasmine, Ramya Nambee...
299                      Mappillai  ...  Dhanush, Hansika Motwani, Manisha Koirala, Viv...

[300 rows x 18 columns]>
```

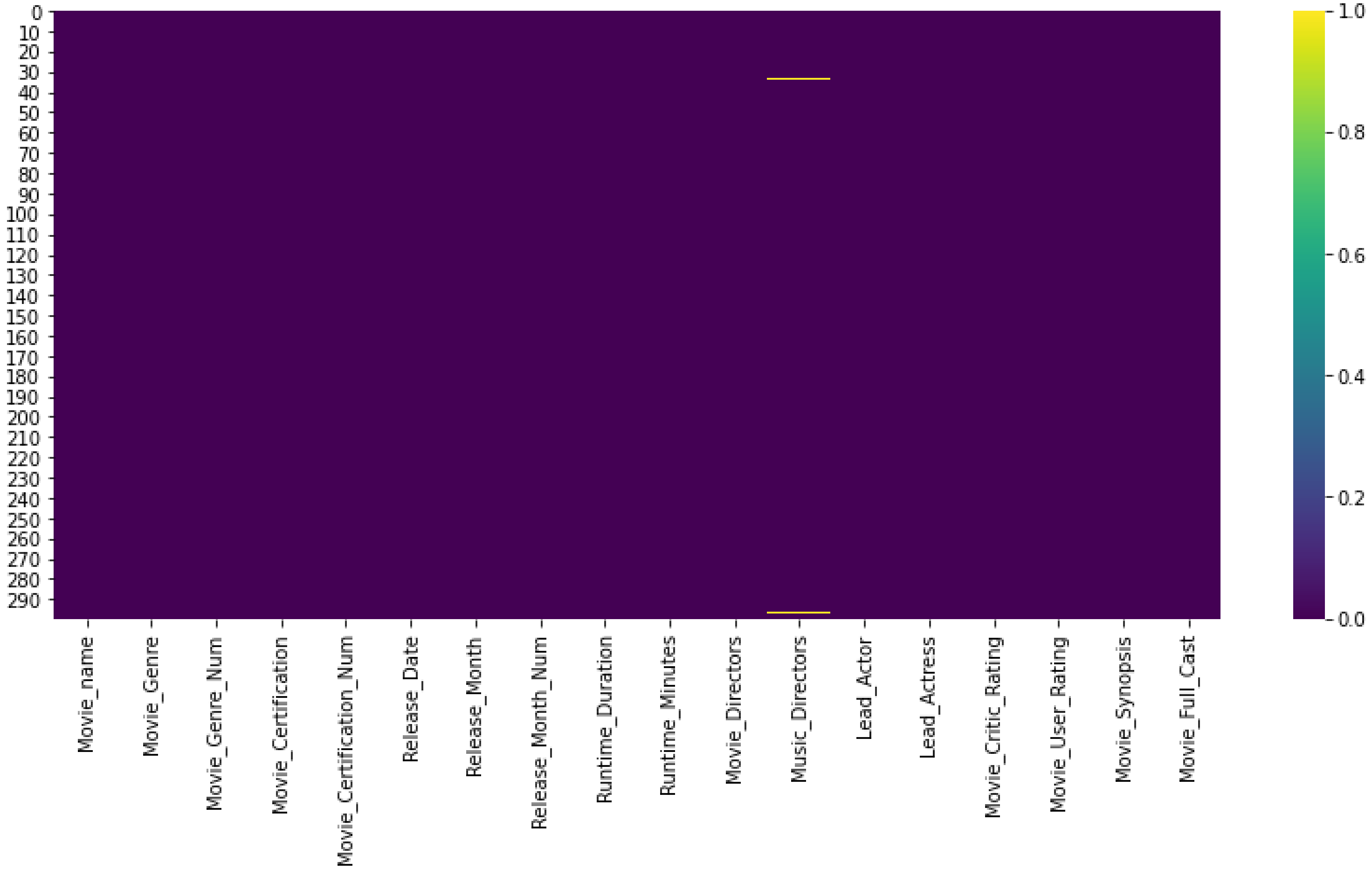
In [7]:

```
dfx.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 18 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Movie_name                300 non-null    object 
 1   Movie_Genre               300 non-null    object 
 2   Movie_Genre_Num           300 non-null    int64  
 3   Movie_Certification       300 non-null    object 
 4   Movie_Certification_Num   300 non-null    int64  
 5   Release_Date              300 non-null    object 
 6   Release_Month             300 non-null    object 
 7   Release_Month_Num         300 non-null    int64  
 8   Runtime_Duration          300 non-null    object 
 9   Runtime_Minutes           300 non-null    int64  
 10  Movie_Directors           300 non-null    object 
 11  Music_Directors           298 non-null    object 
 12  Lead_Actor                300 non-null    object 
 13  Lead_Actress              300 non-null    object 
 14  Movie_Critic_Rating       300 non-null    float64
 15  Movie_User_Rating         300 non-null    float64
 16  Movie_Synopsis            300 non-null    object 
 17  Movie_Full_Cast           300 non-null    object 
dtypes: float64(2), int64(4), object(12)
memory usage: 42.3+ KB
```

In [8]:
```
plt.figure(figsize=(14,6))
sb.heatmap(dfx.isnull(), cmap="viridis")
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa46077ae80>
```



In [9]:
```
dfx['Lead_Actor'].isnull().sum()
```

Out[9]:
```
0
```

In [10]:
```
dfx['Lead_Actress'].isnull().sum()
```

Out[10]:
```
0
```

In [11]:
```
dfy = dfx.copy()
```

In [12]:
```
dfy.head()
```

Out[12]:

	Movie_name	Movie_Genre	Movie_Genre_Num	Movie_Certification	Movie_Certification_Num	Release_Date	Release_Month	Release_Month_Nur
0	Kannum Kannum Kollaiyadithaal	Thriller	15	U	2	28 Feb 2020	Feb	
1	Oh My Kadavule	Comedy	14	UA	1	14 Feb 2020	Feb	
2	Psycho	Thriller	15	A	3	24 Jan 2020	Jan	
3	Dharala Prabhu	Comedy	14	UA	1	13 Mar 2020	Mar	
4	Gypsy	Drama	17	UA	1	06 Mar 2020	Mar	

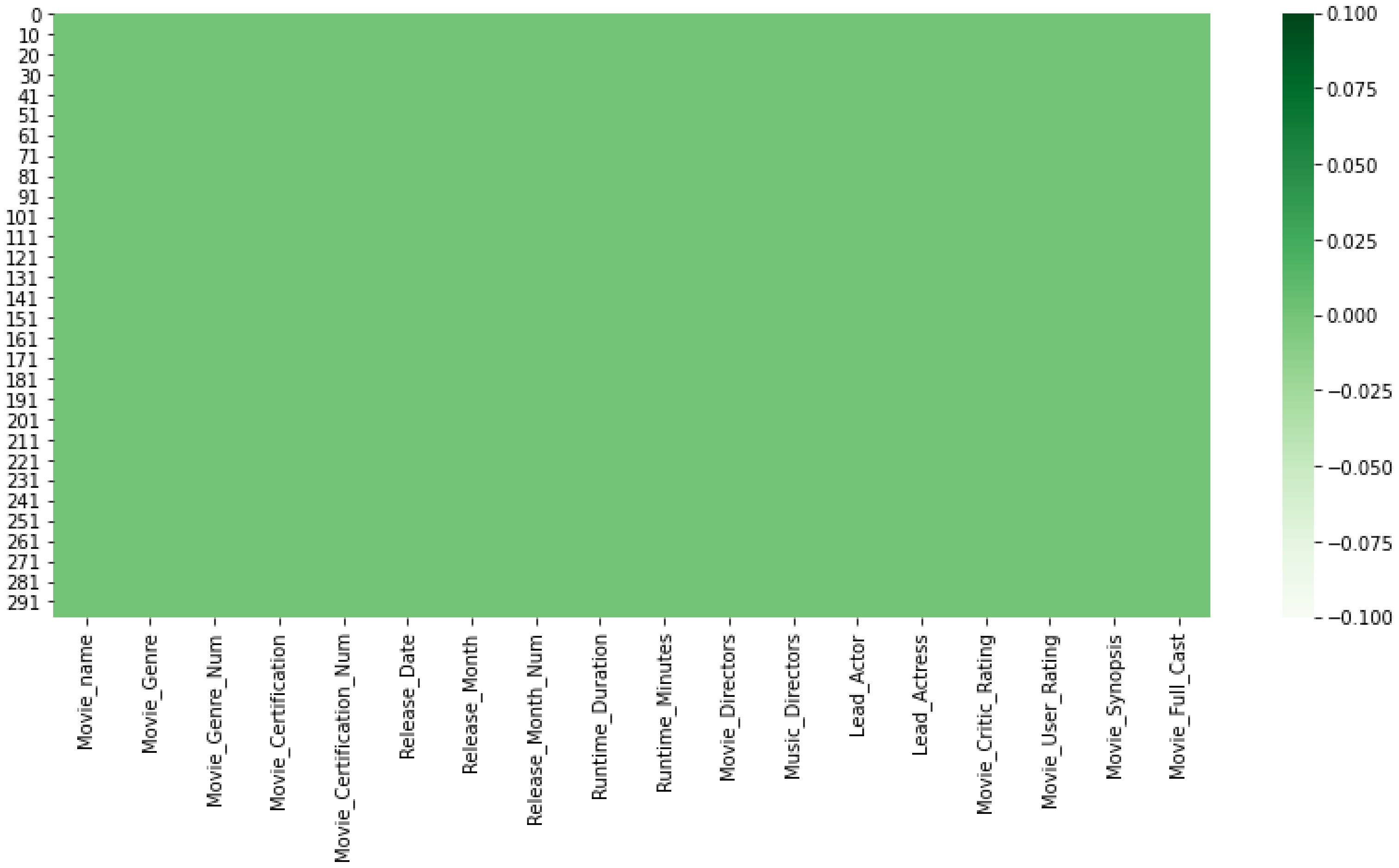
In [13]:
```
dfy.dropna(inplace=True)
```

In [14]:

```
plt.figure(figsize=(14,6))
sb.heatmap(dfy.isnull(), cmap="Greens")

#'Accent', 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGn', 'BuGn_r', 'BuPu', 'BuPu_r', 'CMRmap',
#'CMRmap_r', 'Dark2', 'Dark2_r', 'GnBu', 'GnBu_r', 'Greens', 'Greens_r', 'Greys', 'Greys_r', 'OrRd', 'OrRd_r',
#'Oranges', 'Oranges_r', 'PRGn', 'PRGn_r', 'Paired', 'Paired_r', 'Pastel1', 'Pastel1_r', 'Pastel2', 'Pastel2_r',
#'PiYG', 'PiYG_r', 'PuBu', 'PuBuGn', 'PuBuGn_r', 'PuBu_r', 'PuOr', 'PuOr_r', 'PuRd', 'PuRd_r', 'Purples', 'Purples_r',
#'RdBu', 'RdBu_r', 'RdGy', 'RdGy_r', 'RdPu', 'RdPu_r', 'RdYlBu', 'RdYlBu_r', 'RdYlGn', 'RdYlGn_r', 'Reds', 'Reds_r',
#'Set1', 'Set1_r', 'Set2', 'Set2_r', 'Set3', 'Set3_r', 'Spectral', 'Spectral_r', 'Wistia', 'Wistia_r', 'YlGn', 'YlGnB
u',
#'YlGnBu_r', 'YlGn_r', 'YlOrBr', 'YlOrBr_r', 'YlOrRd', 'YlOrRd_r', 'afmhot', 'afmhot_r', 'autumn', 'autumn_r', 'binar
y',
#'binary_r', 'bone', 'bone_r', 'brg', 'brg_r', 'bwr', 'bwr_r', 'cividis', 'cividis_r', 'cool', 'cool_r', 'coolwarm',
#'coolwarm_r', 'copper', 'copper_r', 'cubehelix', 'cubehelix_r', 'flag', 'flag_r', 'gist_earth', 'gist_earth_r',
#'gist_gray', 'gist_gray_r', 'gist_heat', 'gist_heat_r', 'gist_ncar', 'gist_ncar_r', 'gist_rainbow', 'gist_rainbow_r',
#'gist_stern', 'gist_stern_r', 'gist_yarg', 'gist_yarg_r', 'gnuplot', 'gnuplot2', 'gnuplot2_r', 'gnuplot_r', 'gray',
#'gray_r', 'hot', 'hot_r', 'hsv', 'hsv_r', 'icefire', 'icefire_r', 'inferno', 'inferno_r', 'jet', 'jet_r', 'magma',
#'magma_r', 'mako', 'mako_r', 'n...
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa438fc9f98>
```


0
10
20
30
41
51
61
71
81
91
101
111
121
131
141
151
161
171
181
191
201
211
221
231
241
251
261
271
281
291
Movie_name
Movie_Genre
Movie_Genre_Num
Movie_Certification
Movie_Certification_Num
Release_Date
Release_Month
Release_Month_Num
Runtime_Duration
Runtime_Minutes
Movie_Directors
Music_Directors
Lead_Actor
Lead_Actress
Movie_Critic_Rating
Movie_User_Rating
Movie_Synopsis
Movie_Full_Cast
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100

In [15]:

```
dfy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 298 entries, 0 to 299
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Movie_name              298 non-null    object 
 1   Movie_Genre             298 non-null    object 
 2   Movie_Genre_Num         298 non-null    int64  
 3   Movie_Certification     298 non-null    object 
 4   Movie_Certification_Num 298 non-null    int64  
 5   Release_Date            298 non-null    object 
 6   Release_Month           298 non-null    object 
 7   Release_Month_Num       298 non-null    int64  
 8   Runtime_Duration        298 non-null    object 
 9   Runtime_Minutes         298 non-null    int64  
 10  Movie_Directors         298 non-null    object 
 11  Music_Directors         298 non-null    object 
 12  Lead_Actor              298 non-null    object 
 13  Lead_Actress            298 non-null    object 
 14  Movie_Critic_Rating     298 non-null    float64
 15  Movie_User_Rating       298 non-null    float64
 16  Movie_Synopsis          298 non-null    object 
 17  Movie_Full_Cast         298 non-null    object 
dtypes: float64(2), int64(4), object(12)
memory usage: 44.2+ KB
```

Genre Analysis

In [16]:

```
dfx['Movie_Genre'].unique()
```

Out[16]:

```
array(['Thriller', 'Comedy', 'Drama', 'Action', 'Family', 'Crime',
       'Adventure', 'Musical', 'Biography', 'Sports', 'Romance', 'Sci-Fi',
       'Mystery', 'Horror', 'Documentary', 'History', 'Fantasy'],
      dtype=object)
```

In [17]:

```
genre = dfx.groupby('Movie_Genre')['Movie_Genre'].count()
genre
```

Out[17]:

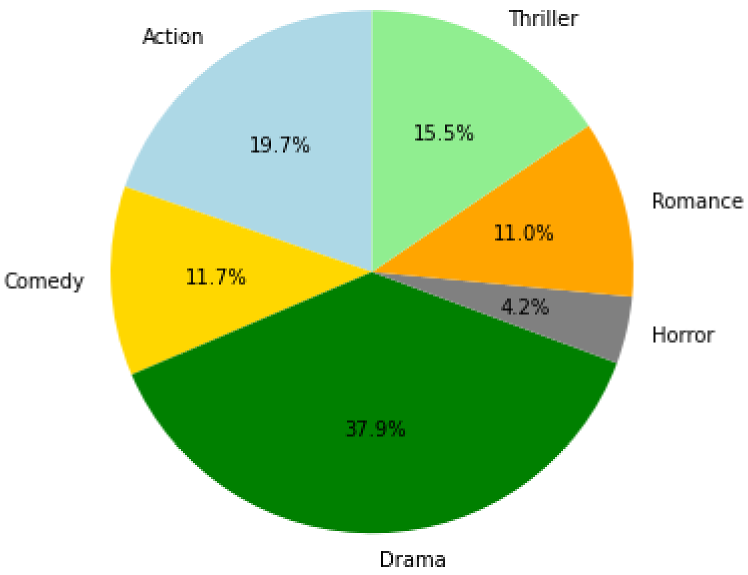
```
Movie_Genre
Action          52
Adventure        2
Biography        2
Comedy          31
Crime           10
Documentary      1
Drama          100
Family           2
Fantasy          2
History          3
Horror          11
Musical          2
Mystery          3
Romance         29
Sci-Fi           5
Sports           4
Thriller        41
Name: Movie_Genre, dtype: int64
```

Pie Chart Representation of basic genre

In [18]:

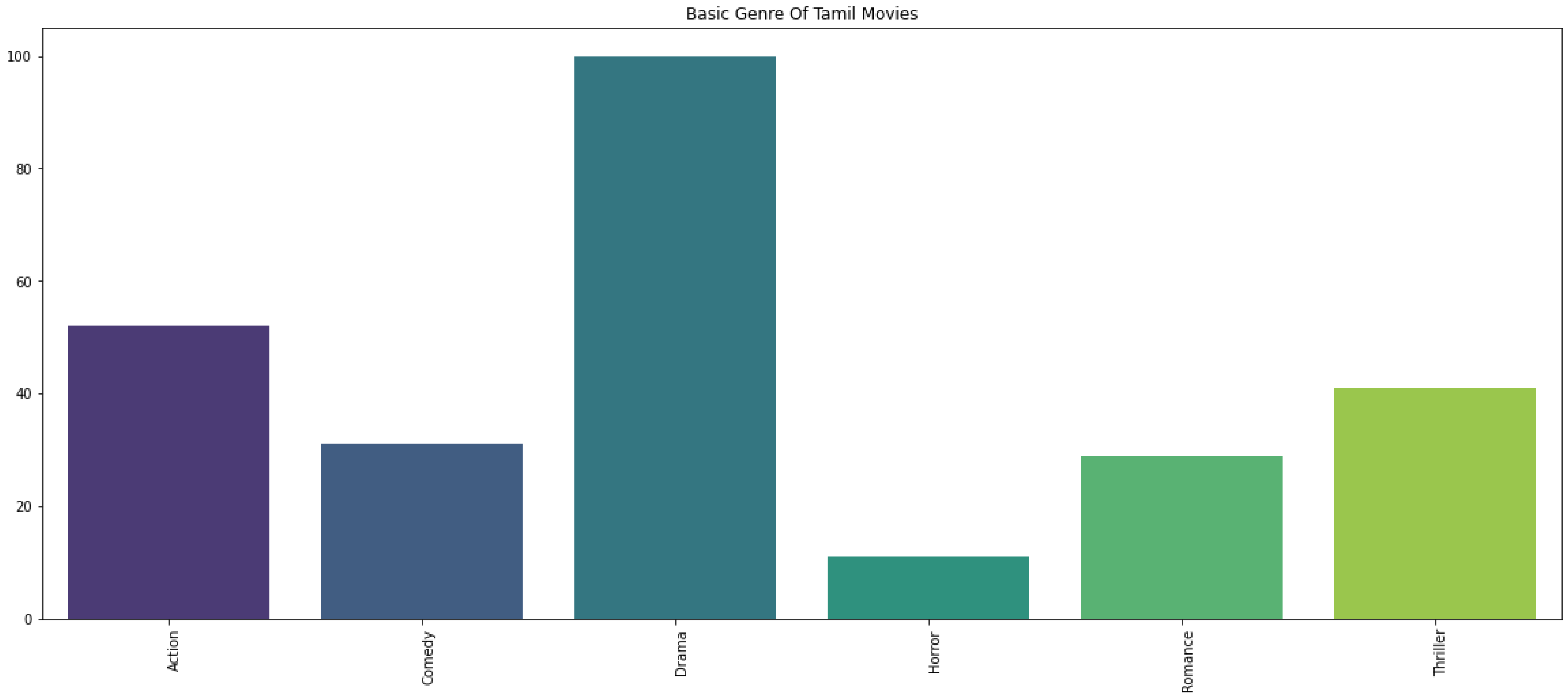
```
genre_general = ['Action', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']
genre_general_values = [52, 31, 100, 11, 29, 41]

#pie chart
colors = ['lightblue', 'gold', 'green','grey', 'orange', 'lightgreen' ]
plt.subplots(figsize=(14,6))
plt.pie(genre_general_values,labels=genre_general, colors=colors, startangle = 90, autopct='%.1f%%')
plt.show()
```



Bar Chart Representation of basic genre

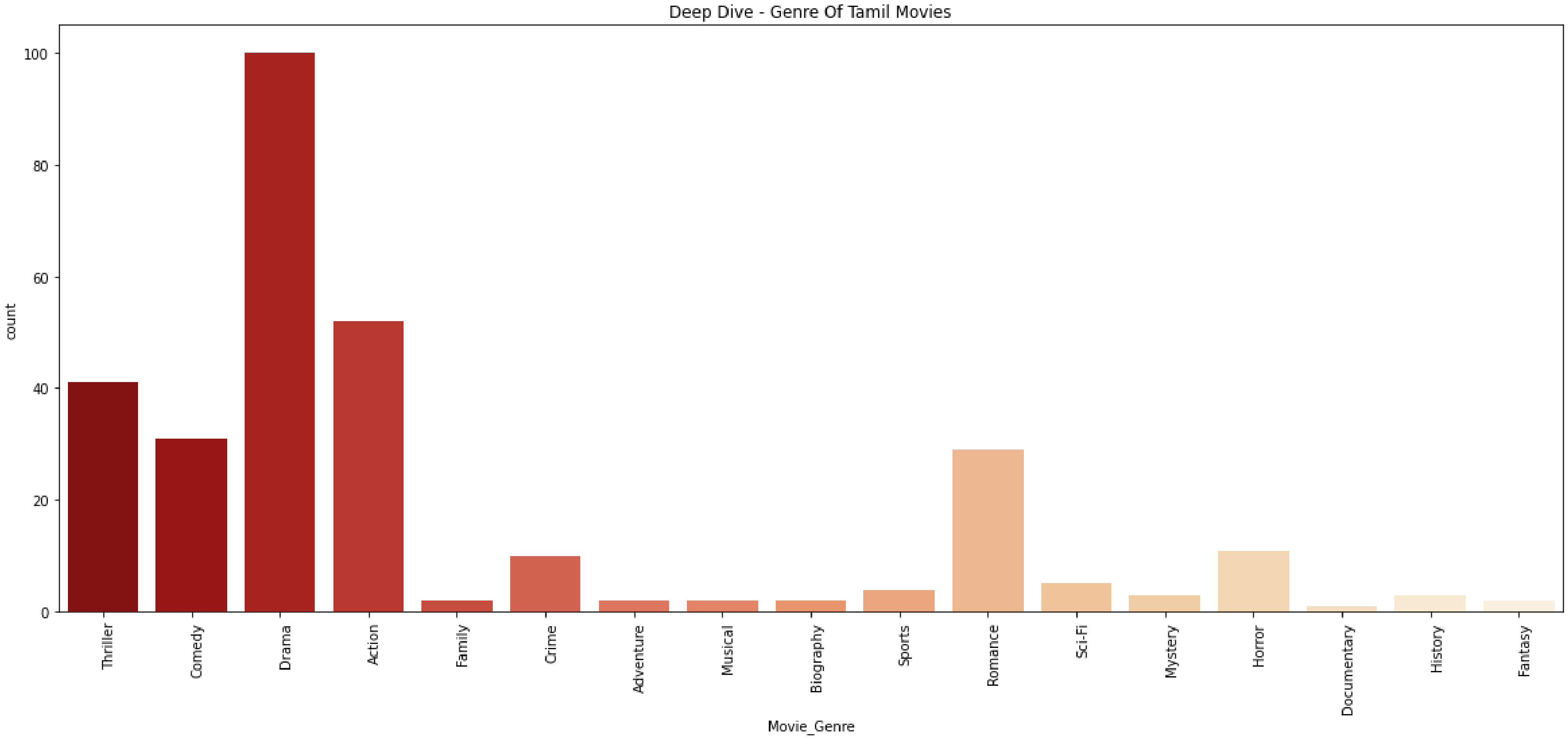
```
In [19]: plt.figure(figsize=(20,8))
         x = genre_general
         y = genre_general_values
         chart1 = sb.barplot(x, y, palette='viridis')
         chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
         chart1.set_title('Basic Genre Of Tamil Movies')
         plt.show()
```



Depth Analysis of Genre

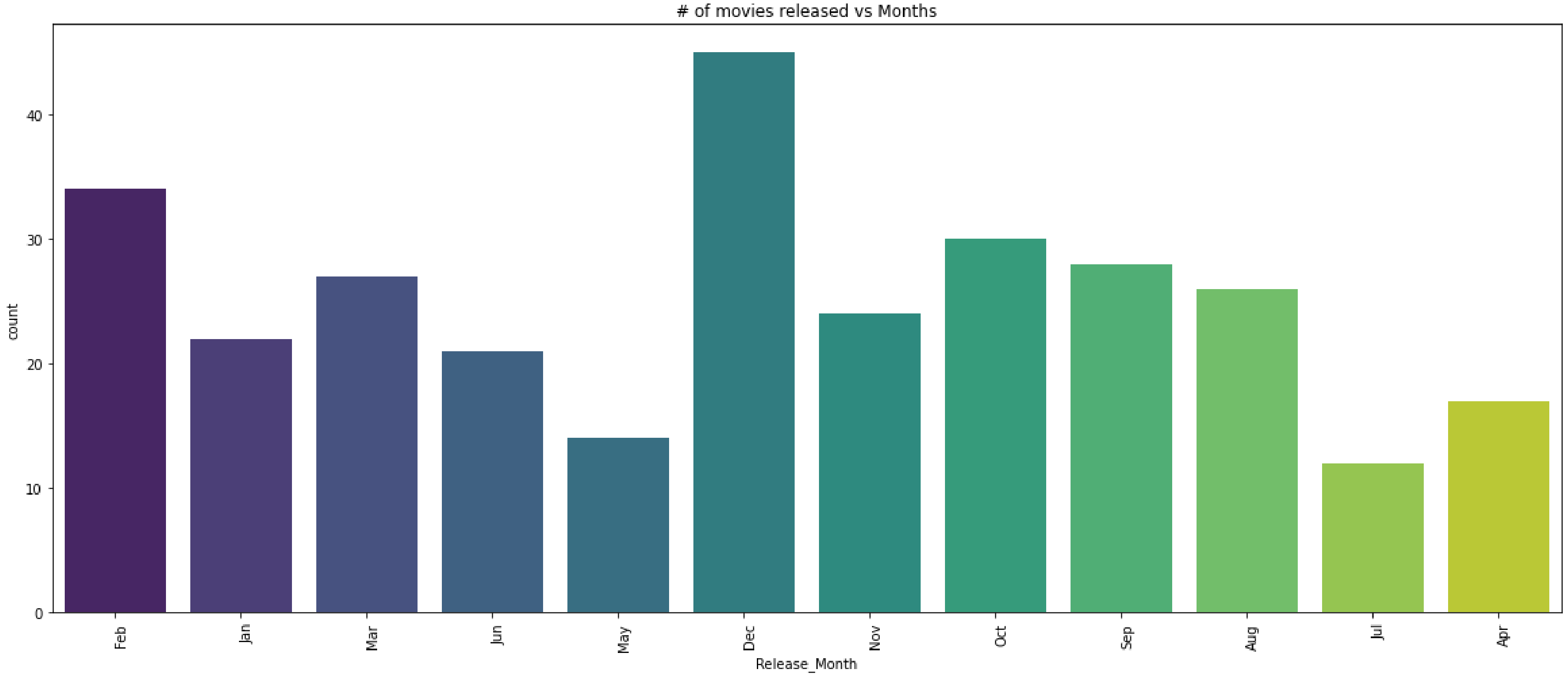
In [20]:

```
plt.figure(figsize=(20,8))
x = dfx['Movie_Genre']
chart1 = sb.countplot(x, data=dfx, palette='OrRd_r')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('Deep Dive - Genre Of Tamil Movies')
plt.show()
```



Month-wise Visualization of Movie releases

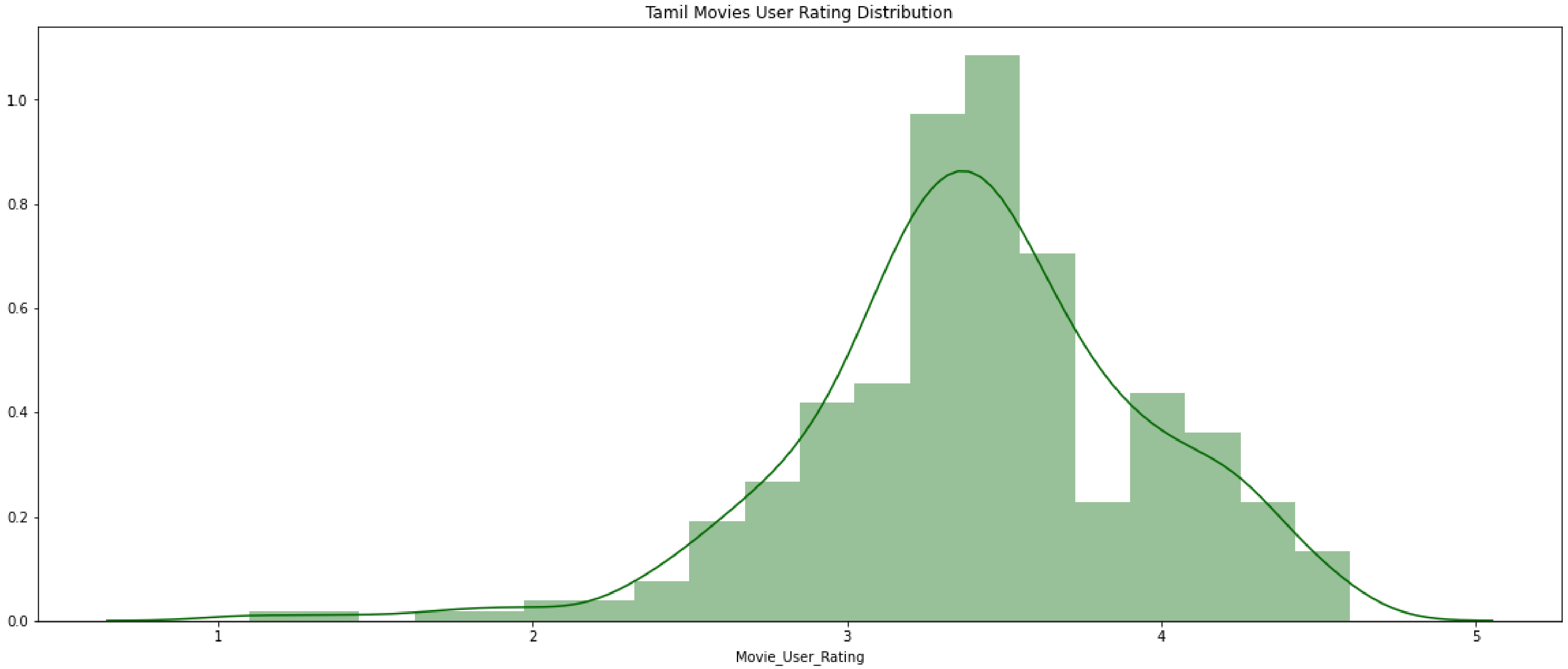
```
In [21]: plt.figure(figsize=(20,8))
         chart1 = sb.countplot(x=dfx['Release_Month'], data=dfx, palette='viridis')
         chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
         chart1.set_title('# of movies released vs Months')
         plt.show()
```



User Rating Distribution Plot

In [22]:

```
plt.figure(figsize=(20,8))
chart2 = sb.distplot(dfx['Movie_User_Rating'],  color="#006600")
chart2.set_title('Tamil Movies User Rating Distribution')
plt.show()
```



Movie Certifications Analysis

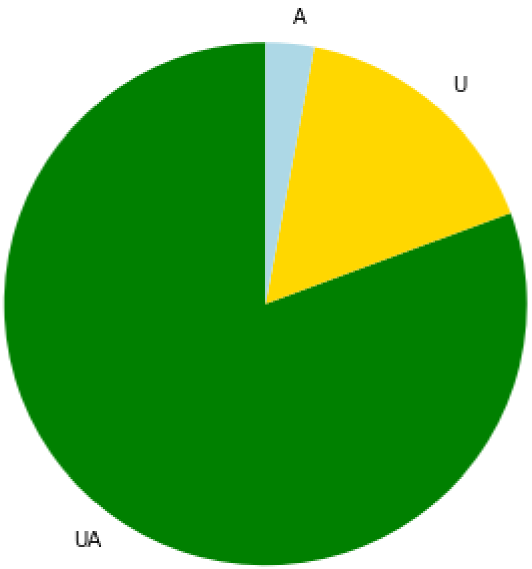
Pie Chart Representation of Movie Censorboard Certification

In [23]:

```
mc = dfx.groupby('Movie_Certification')['Movie_Certification'].count()
mc_entities = ['UA', 'U', 'A']
mc_values = [242,49,9]
```

In [24]:

```
plt.subplots(figsize=(14,6))
colors = ['green', 'gold', 'lightblue']
plt.pie(mc_values, labels = mc_entities, colors=colors, startangle = 90)
plt.show()
```



Analysis of Lead Actors and Lead Actresses of the Decade

Dominance comparison of Lead Actors

In [25]:

```
dfx['Lead_Actor'].nunique()
```

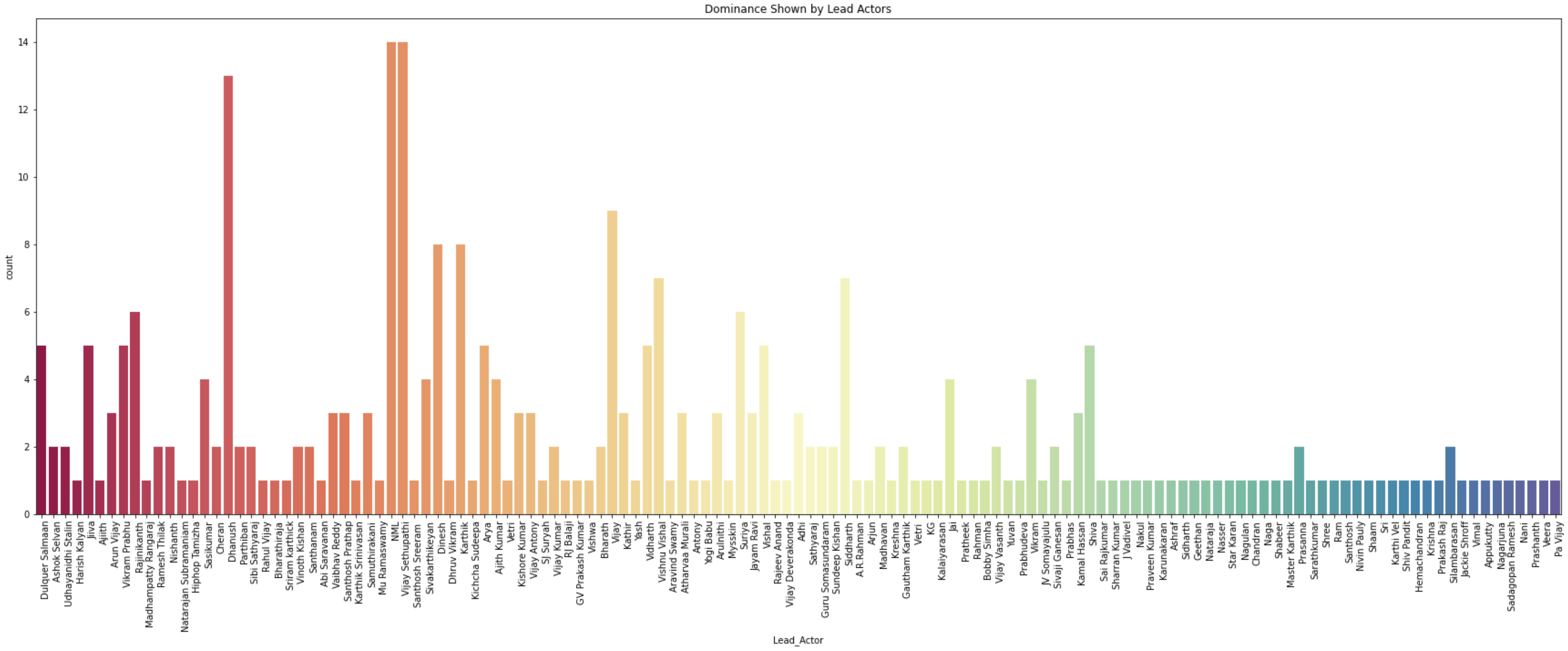
Out[25]:

```
131
```

NML - No specific male lead. Indicates the presence of successful movies with only female lead.

NFL - No specific female lead.

In [26]:
```
plt.figure(figsize=(30,10))
chart1 = sb.countplot(x=dfx['Lead_Actor'], data=dfx, palette='Spectral')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('Dominance Shown by Lead Actors')
plt.show()
```

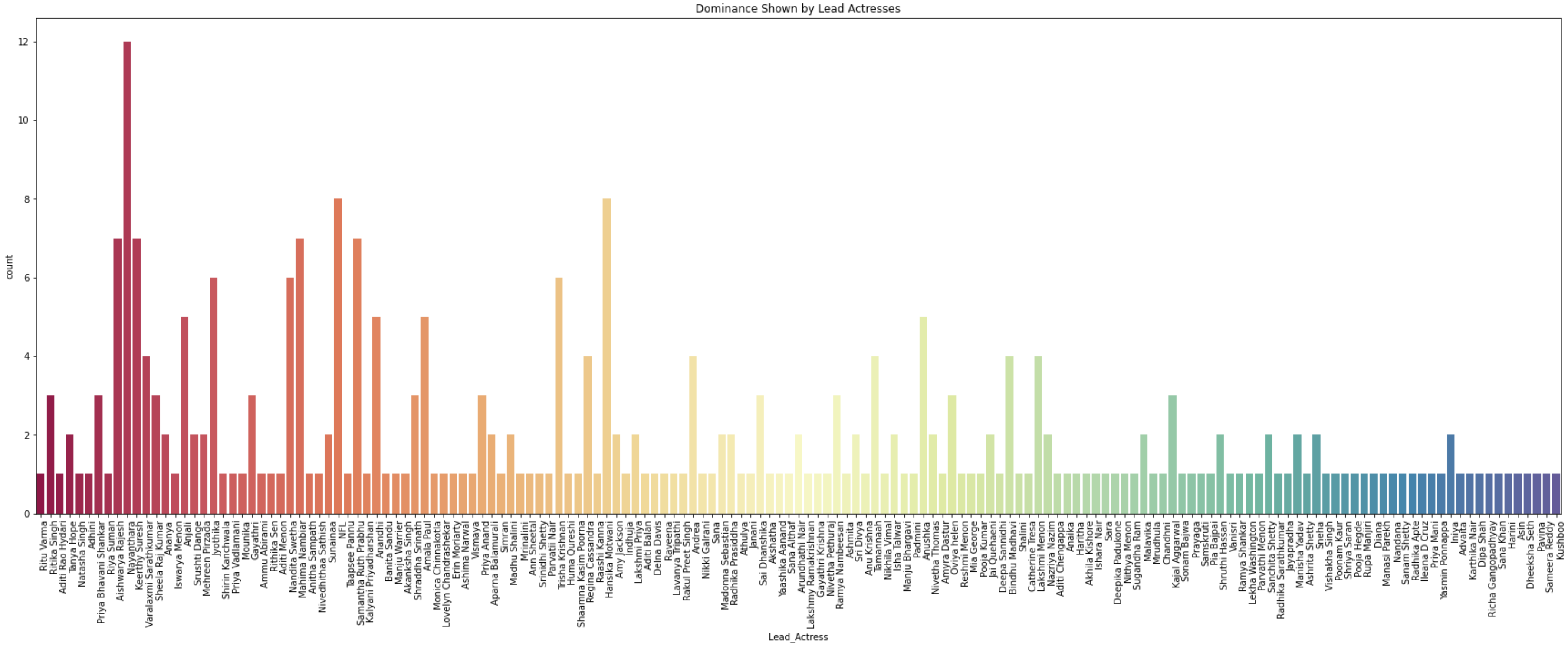


Dominance comparison of Lead Actresses

```
In [27]: dfx['Lead_Actress'].nunique()
```

```
Out[27]: 159
```

```
In [28]: plt.figure(figsize=(30,10))
         chart1 = sb.countplot(x=dfx['Lead_Actress'], data=dfx, palette='Spectral')
         chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
         chart1.set_title('Dominance Shown by Lead Actresses')
         plt.show()
```



Analysis of Movie Directors

In [29]:

```
dfx['Movie_Directors'].nunique()
```

Out[29]:

```
211
```

In [30]:

```
'''plt.figure(figsize=(20,42))
chart1 = sb.countplot(y=dfx['Movie_Directors'], data=dfx, palette='viridis')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('Graph representing Movie Directors')
plt.show()'''
```

Out[30]:

```
"plt.figure(figsize=(20,42))\nchart1 = sb.countplot(y=dfx['Movie_Directors'], data=dfx, palette='viridis')\nchart1.se
t_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')\nchart1.set_title('Graph representin
g Movie Directors')\nplt.show()"
```

In [31]:

```
dirgroup = dfx.groupby('Movie_Directors')['Movie_Directors'].count()
d = dirgroup.to_frame()
d
```

Out[31]:

```
                    Movie_Directors
   Movie_Directors
          A G Amid                1
     A L Abanindran               1
          A L Vijay               2
        A Raajdheep               1
         A Sarkunam               1
                ...             ...
       Vijay Milton               2
       Vikram Kumar               1
  Vikram Sugumaran                1
  Yuvaraj Dhayalan                1
 Yuvaraj Subramani                1

211 rows × 1 columns
```

In [32]:
```
print(d.rename(columns={'Movie_Directors': 'Director', 'Movie_Directors': 'num_movies'}))
```

```
                   num_movies
Movie_Directors              
A G Amid                    1
A L Abanindran              1
A L Vijay                   2
A Raajdheep                 1
A Sarkunam                  1
...                       ...
Vijay Milton                2
Vikram Kumar                1
Vikram Sugumaran            1
Yuvaraj Dhayalan            1
Yuvaraj Subramani           1

[211 rows x 1 columns]
```

```
In [33]: dx = d.rename(columns={'Movie_Directors': 'num_movies'}, index={'Movie_Directors': 'Directors'})
         dx
```

```
Out[33]:
                          num_movies
         Movie_Directors
                A G Amid           1
          A L Abanindran           1
               A L Vijay           2
            A Raajdheep            1
              A Sarkunam           1
                     ...         ...
            Vijay Milton           2
            Vikram Kumar           1
       Vikram Sugumaran            1
        Yuvaraj Dhayalan           1
      Yuvaraj Subramani            1

211 rows × 1 columns
```

In [34]:
```
dx.reset_index(level=0, inplace=True)
dx
```

Out[34]:

	Movie_Directors	num_movies
0	A G Amid	1
1	A L Abanindran	1
2	A L Vijay	2
3	A Raajdheep	1
4	A Sarkunam	1
...	...	...
206	Vijay Milton	2
207	Vikram Kumar	1
208	Vikram Sugumaran	1
209	Yuvaraj Dhayalan	1
210	Yuvaraj Subramani	1

211 rows × 2 columns

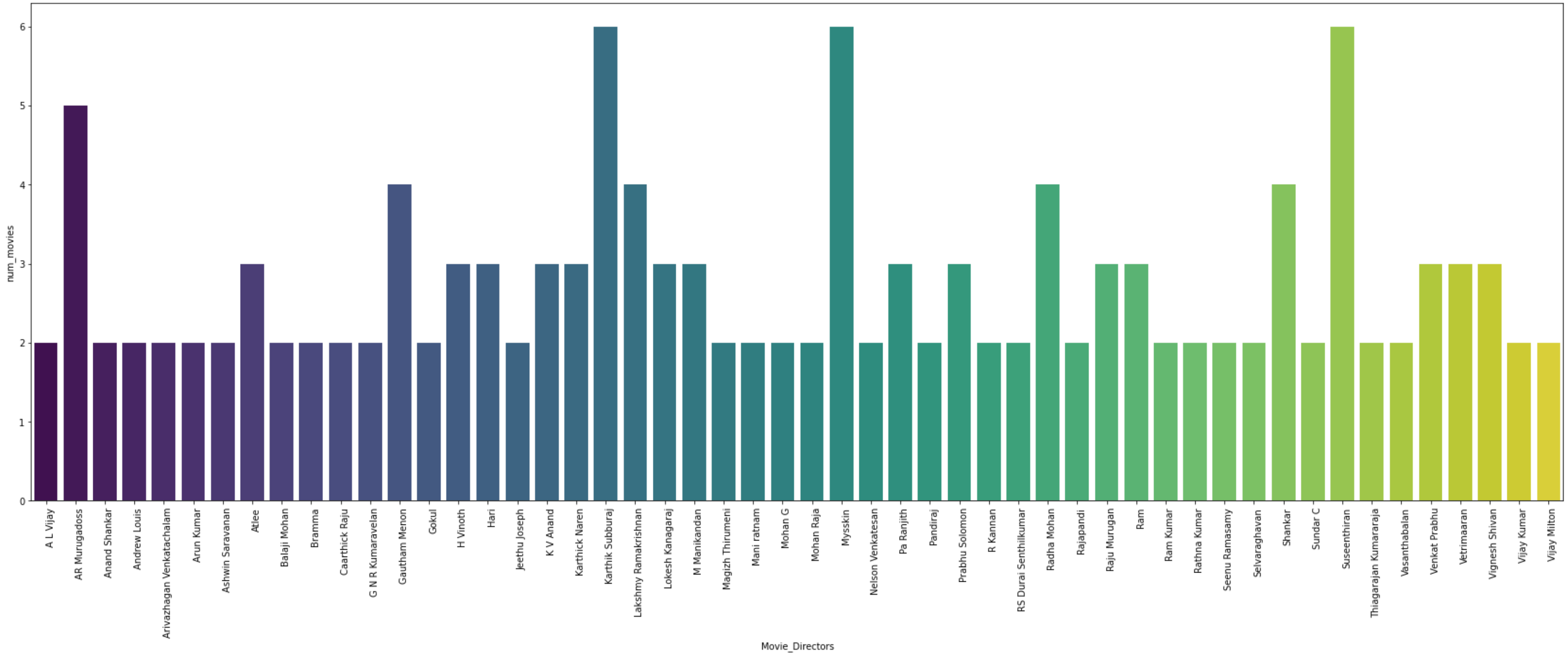
In [35]:
```
dy = dx[dx.num_movies != 1]
```

In [36]:
```
dy.head()
```

Out[36]:

	Movie_Directors	num_movies
2	A L Vijay	2
6	AR Murugadoss	5
10	Anand Shankar	2
12	Andrew Louis	2
15	Arivazhagan Venkatachalam	2

In [37]:
```
plt.figure(figsize=(30,10))
chart1 = sb.barplot(x=dy['Movie_Directors'], y= dy['num_movies'], palette='viridis')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
plt.show()
```

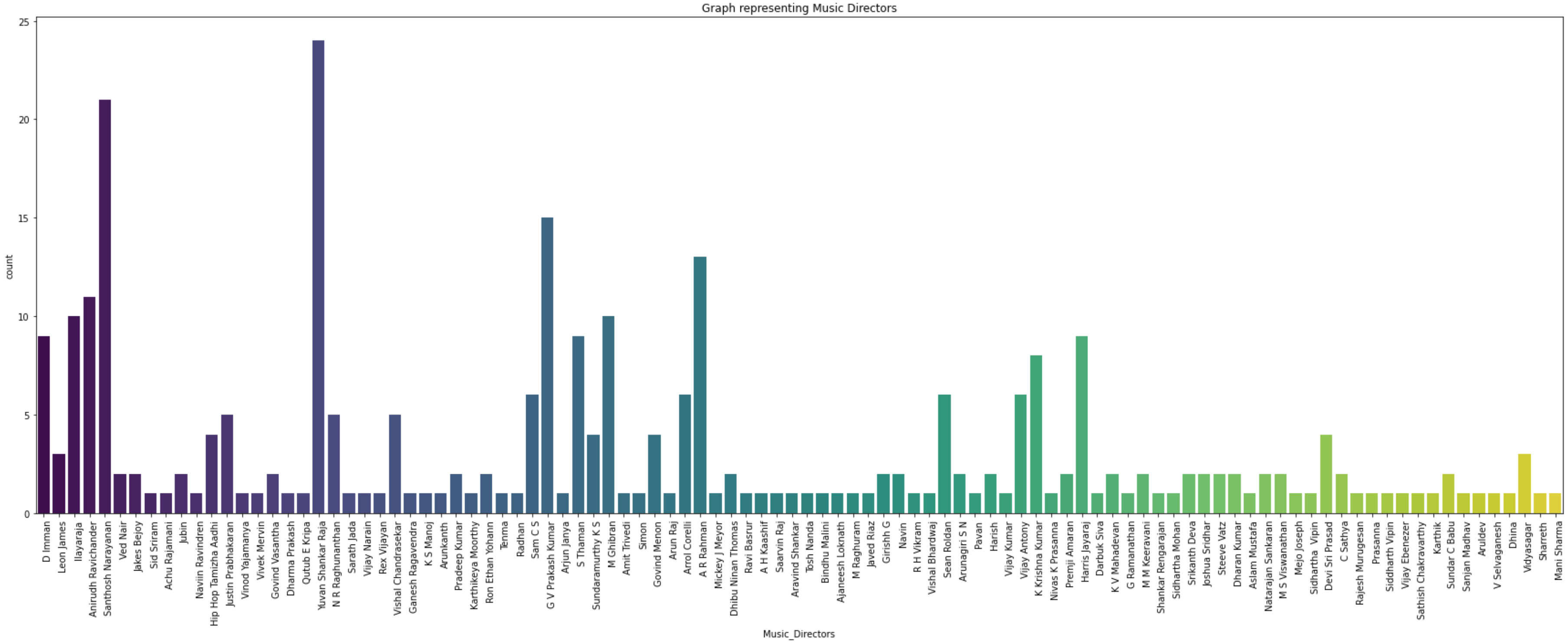


Analysis of Music Directors

In [38]:
```
dfx['Music_Directors'].nunique()
```

Out[38]:
```
100
```

In [39]:
```
plt.figure(figsize=(30,10))
chart1 = sb.countplot(x=dfx['Music_Directors'], data=dfx, palette='viridis')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('Graph representing Music Directors')
plt.show()
```



Derived Conclusions

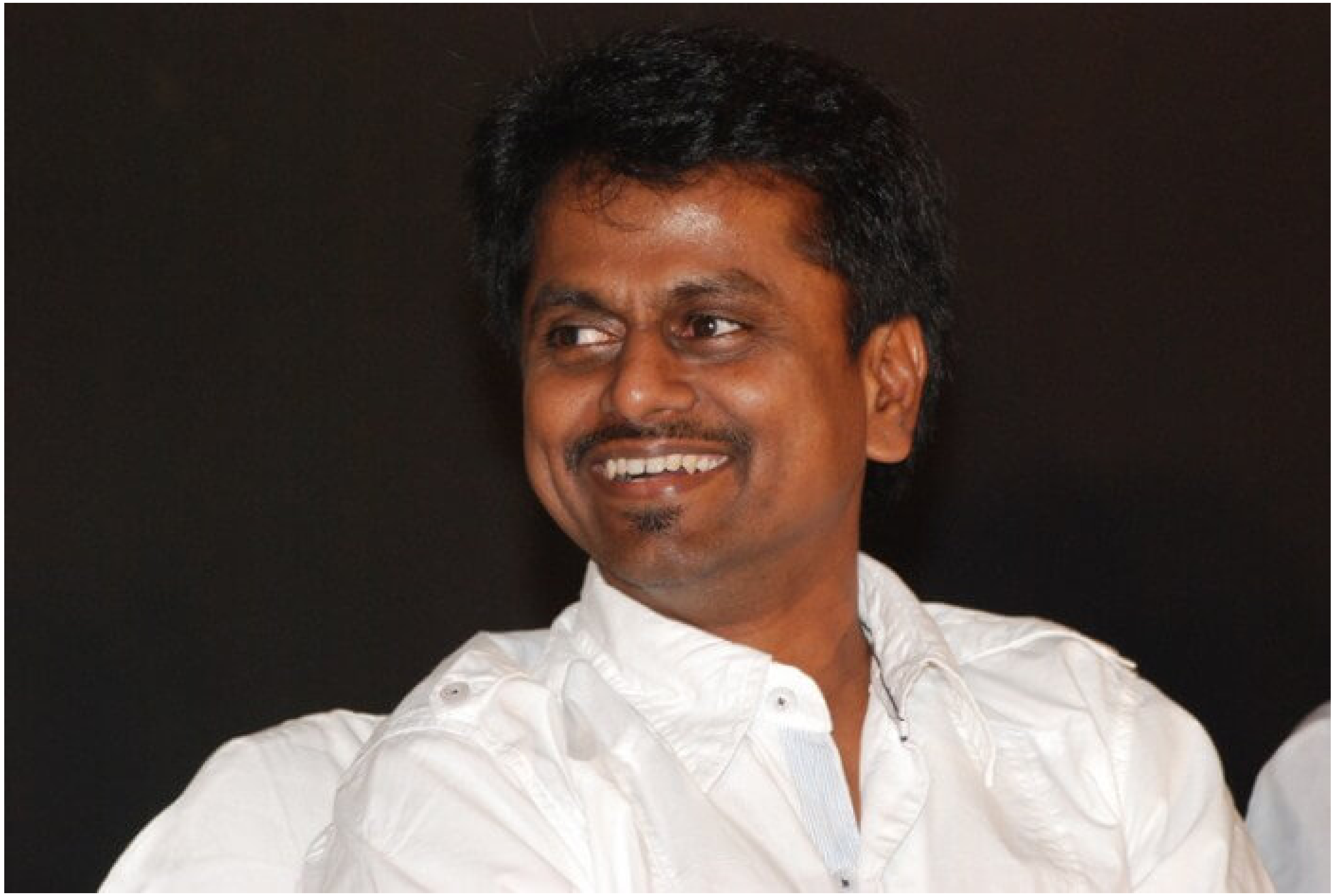
Presenting the Rockstars of Kollywood in the decade 2011-2020

Most Dominating Lead Actor: Vijay Sethupathi

Most Dominating Lead Actress: Nayanthara

Best Director in Tamil Cinema - Genre: Thriller - Mysskin

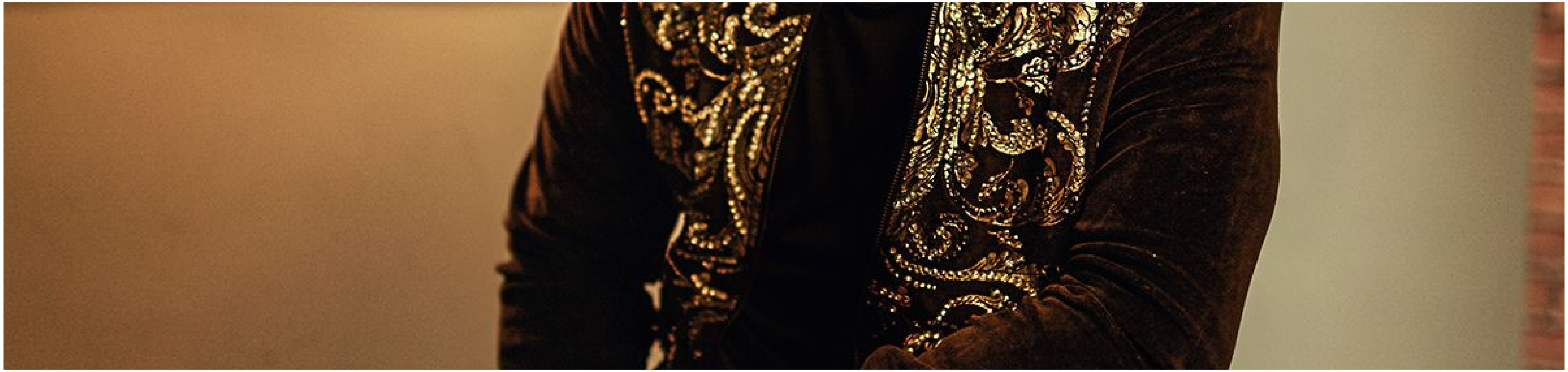
Best Director in Tamil Cinema - Genre: Social Awarness - AR Murugadoss



Best Director in Tamil Cinema - Genre: Drama - Suseenthiran

Best Music Director in Tamil Cinema - Yuvan Shankar Raja

U1
RECORD



Machine Learning Implementation

In [40]:

```
dfx.head(10)
```

Out[40]:

	Movie_name	Movie_Genre	Movie_Genre_Num	Movie_Certification	Movie_Certification_Num	Release_Date	Release_Month	Release_Month_Nur
0	Kannum Kannum Kollaiyadithaal	Thriller	15	U	2	28 Feb 2020	Feb	
1	Oh My Kadavule	Comedy	14	UA	1	14 Feb 2020	Feb	
2	Psycho	Thriller	15	A	3	24 Jan 2020	Jan	
3	Dharala Prabhu	Comedy	14	UA	1	13 Mar 2020	Mar	
4	Gypsy	Drama	17	UA	1	06 Mar 2020	Mar	
5	Baaram	Drama	17	A	3	21 Feb 2020	Feb	
6	Mafia: Chapter 1	Drama	17	UA	1	21 Feb 2020	Feb	
7	Seeru	Action	16	UA	1	07 Feb 2020	Feb	
8	Vaanam Kottattum	Drama	17	U	2	07 Feb 2020	Feb	
9	Darbar	Action	16	UA	1	09 Jan 2020	Jan	

In [41]:
```
'''path = 'C:/Users/gkish/Jupyter Notebooks/BDB/DAY - 3/Movie300 Revised V1.xlsx'
dfx.to_excel(path)'''
```

Out[41]:
```
"path = 'C:/Users/gkish/Jupyter Notebooks/BDB/DAY - 3/Movie300 Revised V1.xlsx'\ndfx.to_excel(path)"
```

In [42]:
```
dfx.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 18 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Movie_name                300 non-null    object 
 1   Movie_Genre               300 non-null    object 
 2   Movie_Genre_Num           300 non-null    int64  
 3   Movie_Certification       300 non-null    object 
 4   Movie_Certification_Num   300 non-null    int64  
 5   Release_Date              300 non-null    object 
 6   Release_Month             300 non-null    object 
 7   Release_Month_Num         300 non-null    int64  
 8   Runtime_Duration          300 non-null    object 
 9   Runtime_Minutes           300 non-null    int64  
 10  Movie_Directors           300 non-null    object 
 11  Music_Directors           298 non-null    object 
 12  Lead_Actor                300 non-null    object 
 13  Lead_Actress              300 non-null    object 
 14  Movie_Critic_Rating       300 non-null    float64
 15  Movie_User_Rating         300 non-null    float64
 16  Movie_Synopsis            300 non-null    object 
 17  Movie_Full_Cast           300 non-null    object 
dtypes: float64(2), int64(4), object(12)
memory usage: 42.3+ KB
```

In [43]:

```
dfx.columns
```

Out[43]:

```
Index(['Movie_name', 'Movie_Genre', 'Movie_Genre_Num', 'Movie_Certification',
       'Movie_Certification_Num', 'Release_Date', 'Release_Month',
       'Release_Month_Num', 'Runtime_Duration', 'Runtime_Minutes',
       'Movie_Directors', 'Music_Directors', 'Lead_Actor', 'Lead_Actress',
       'Movie_Critic_Rating', 'Movie_User_Rating', 'Movie_Synopsis',
       'Movie_Full_Cast'],
      dtype='object')
```

In [44]:

```
dfxml = dfx[['Movie_name','Movie_Genre_Num','Movie_Certification_Num','Release_Month_Num','Runtime_Minutes','Movie_Critic_Rating','Movie_User_Rating']]
dfxml.head()
```

Out[44]:

	Movie_name	Movie_Genre_Num	Movie_Certification_Num	Release_Month_Num	Runtime_Minutes	Movie_Critic_Rating	Movie_User_Rating
0	Kannum Kannum Kollaiyadithaal	15	2	2	122	3.5	3.4
1	Oh My Kadavule	14	1	2	151	3.5	3.4
2	Psycho	15	3	1	134	3.5	3.3
3	Dharala Prabhu	14	1	3	122	3.0	3.3
4	Gypsy	17	1	3	145	3.0	3.2

In [45]:

```
#target variable
y = dfxml['Movie_User_Rating']

#input dataframe
x = dfxml[['Movie_Genre_Num','Movie_Certification_Num','Release_Month_Num','Runtime_Minutes','Movie_Critic_Rating']]
```

In [46]:

```
from sklearn.model_selection import train_test_split
```

In [47]:

```
from sklearn.linear_model import LinearRegression
```

In [48]:

```
x1,x2,y1,y2 = train_test_split(x,y,test_size = 0.1)
```

In [49]:

```
lr = LinearRegression()
```

In [50]:

```
lr.fit(x1,y1)
```

Out[50]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [51]:

```
lr.coef_
```

Out[51]:

```
array([-4.14677747e-04,  4.32383459e-02, -9.52638684e-03,  1.30081750e-03,
        7.92938274e-01])
```

In [52]:

```
lr.intercept_
```

Out[52]:

```
0.6024097505927175
```

In [53]:

```
pd.DataFrame(lr.coef_, index=x.columns, columns=['myval'])
```

Out[53]:

	myval
Movie_Genre_Num	-0.000415
Movie_Certification_Num	0.043238
Release_Month_Num	-0.009526
Runtime_Minutes	0.001301
Movie_Critic_Rating	0.792938

In [54]:

```
t = np.array(dfxml.loc[8][['Movie_Genre_Num','Movie_Certification_Num','Release_Month_Num','Runtime_Minutes','Movie_Critic_Rating']])
```

In [55]:

```
lr.predict([t])
```

Out[55]:

```
array([3.22111178])
```

In [56]:

```
f = []
k = []
for i in range(0,300):
    b = np.array(dfxml.loc[i][['Movie_Genre_Num','Movie_Certification_Num','Release_Month_Num','Runtime_Minutes','Movie_Critic_Rating']])
    f.append(lr.predict([b]))
    k.append(np.array(dfxml.loc[i][['Movie_name']]))
```

In [57]:

```
K = pd.DataFrame(k,columns=['Movie_name'])
K
```

Out[57]:

	Movie_name
0	Kannum Kannum Kollaiyadithaal
1	Oh My Kadavule
2	Psycho
3	Dharala Prabhu
4	Gypsy
...	...
295	Nootrenbadhu
296	Ponnar Shankar
297	Nadunisi Naaygal
298	Ilaignan
299	Mappillai

300 rows × 1 columns

In [58]:
```
F=pd.DataFrame(f,columns=['Machine_Predicted_Rating'])
F
```

Out[58]:
```
     Machine_Predicted_Rating
  0                  3.597597
  1                  3.592497
  2                  3.665972
  3                  3.148778
  4                  3.177453
...                       ...
295                  2.720242
296                  2.790970
297                  1.952872
298                  2.051326
299                  1.601562

300 rows × 1 columns
```

In [59]:
```
J = dfxml[['Movie_Critic_Rating','Movie_User_Rating']]
```

In [60]:

```
final = pd.concat([K,J,F],axis=1)
final
```

Out[60]:

	Movie_name	Movie_Critic_Rating	Movie_User_Rating	Machine_Predicted_Rating
0	Kannum Kannum Kollaiyadithaal	3.5	3.4	3.597597
1	Oh My Kadavule	3.5	3.4	3.592497
2	Psycho	3.5	3.3	3.665972
3	Dharala Prabhu	3.0	3.3	3.148778
4	Gypsy	3.0	3.2	3.177453
...	...	...	...	...
295	Nootrenbadhu	2.5	2.4	2.720242
296	Ponnar Shankar	2.5	2.4	2.790970
297	Nadunisi Naaygal	1.5	1.7	1.952872
298	Ilaignan	1.5	1.4	2.051326
299	Mappillai	1.0	1.1	1.601562

300 rows × 4 columns

In [61]:

```
'''path = 'C:/Users/gkish/Jupyter Notebooks/BDB/DAY - 3/Machine_Predictions.xlsx'
final.to_excel(path)'''
```

Out[61]:

```
"path = 'C:/Users/gkish/Jupyter Notebooks/BDB/DAY - 3/Machine_Predictions.xlsx'\nfinal.to_excel(path)"
```

In [62]:

```
dfx = pd.concat([dfx,F],axis=1)
dfx.columns
```

Out[62]:

```
Index(['Movie_name', 'Movie_Genre', 'Movie_Genre_Num', 'Movie_Certification',
       'Movie_Certification_Num', 'Release_Date', 'Release_Month',
       'Release_Month_Num', 'Runtime_Duration', 'Runtime_Minutes',
       'Movie_Directors', 'Music_Directors', 'Lead_Actor', 'Lead_Actress',
       'Movie_Critic_Rating', 'Movie_User_Rating', 'Movie_Synopsis',
       'Movie_Full_Cast', 'Machine_Predicted_Rating'],
      dtype='object')
```

In [63]:

```
dfx = dfx[['Movie_name', 'Movie_Genre', 'Movie_Genre_Num', 'Movie_Certification',
        'Movie_Certification_Num', 'Release_Date', 'Release_Month',
        'Release_Month_Num', 'Runtime_Duration', 'Runtime_Minutes',
        'Movie_Directors', 'Music_Directors', 'Lead_Actor', 'Lead_Actress',
        'Movie_Critic_Rating', 'Movie_User_Rating','Machine_Predicted_Rating', 'Movie_Synopsis','Movie_Full_Cast']]

dfx.head()
```

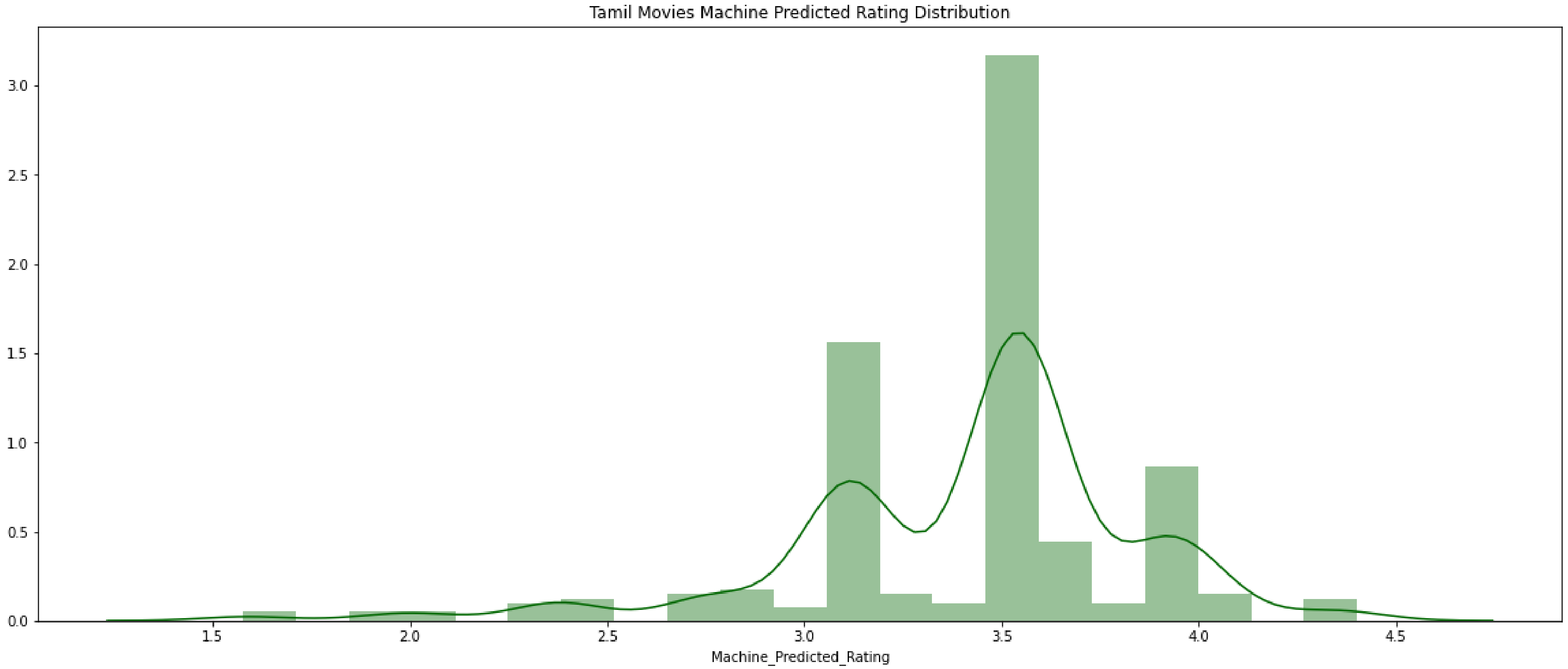
Out[63]:

	Movie_name	Movie_Genre	Movie_Genre_Num	Movie_Certification	Movie_Certification_Num	Release_Date	Release_Month	Release_Month_Nur
0	Kannum Kannum Kollaiyadithaal	Thriller	15	U	2	28 Feb 2020	Feb	
1	Oh My Kadavule	Comedy	14	UA	1	14 Feb 2020	Feb	
2	Psycho	Thriller	15	A	3	24 Jan 2020	Jan	
3	Dharala Prabhu	Comedy	14	UA	1	13 Mar 2020	Mar	
4	Gypsy	Drama	17	UA	1	06 Mar 2020	Mar	

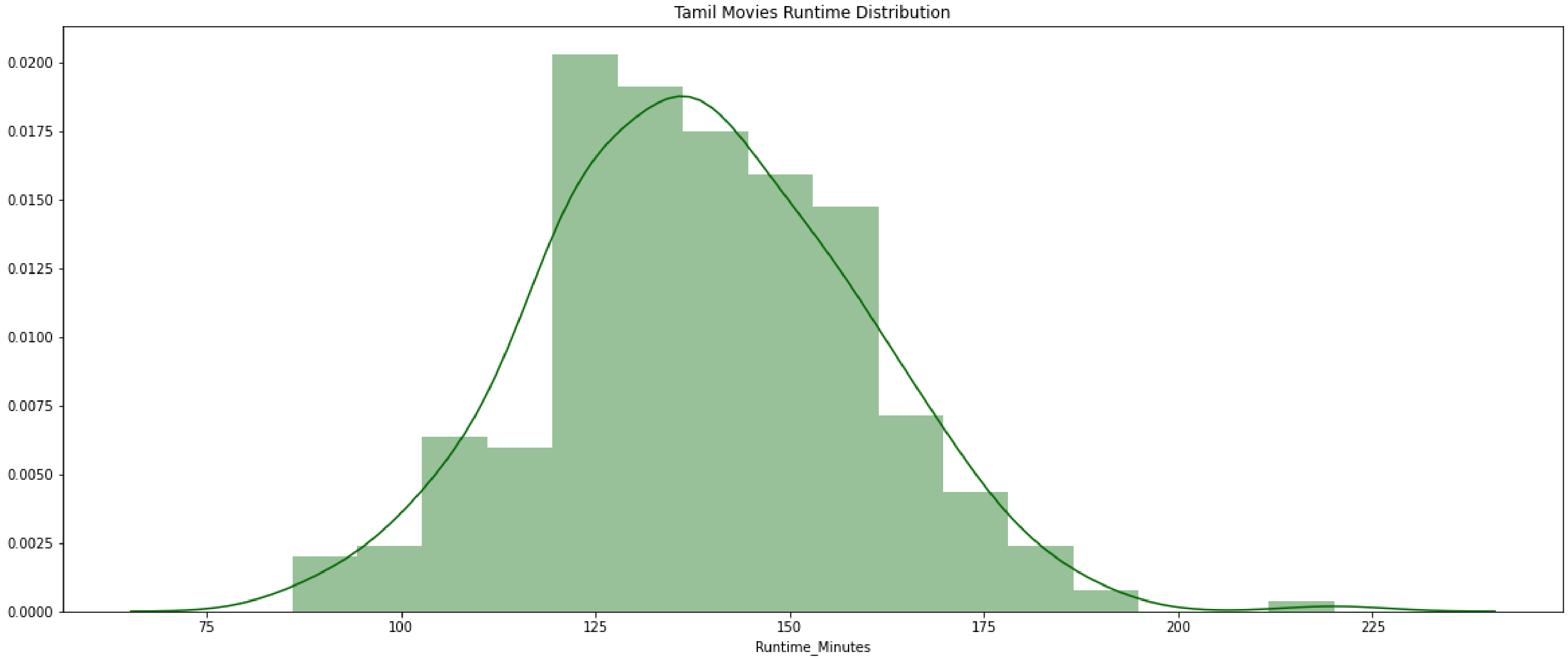
Advanced Visualizations

In [64]:

```
plt.figure(figsize=(20,8))
chart3 = sb.distplot(final['Machine_Predicted_Rating'],  color="#006600")
chart3.set_title('Tamil Movies Machine Predicted Rating Distribution')
plt.show()
```



In [65]:
```
plt.figure(figsize=(20,8))
chart4 = sb.distplot(dfx['Runtime_Minutes'],  color="#006600")
chart4.set_title('Tamil Movies Runtime Distribution')
plt.show()
```



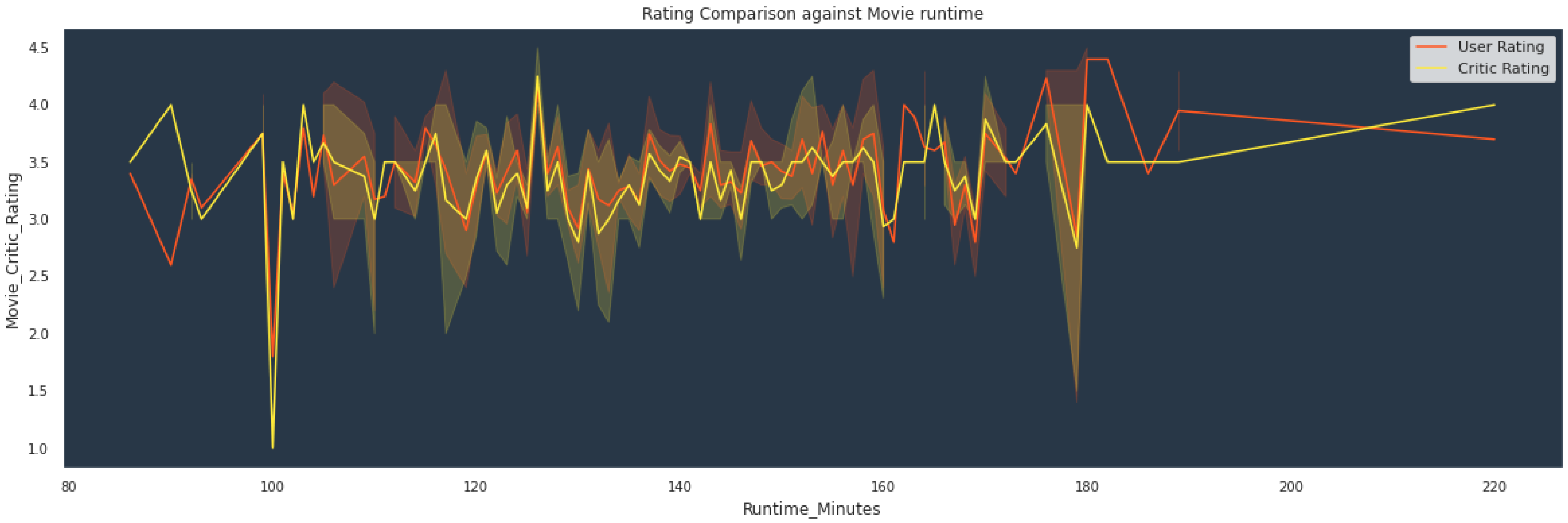
In [66]:
```
dfx['Runtime_Minutes'].mean()
```

Out[66]:
```
138.54666666666665
```

In [67]:
```
import matplotlib as mpl
```

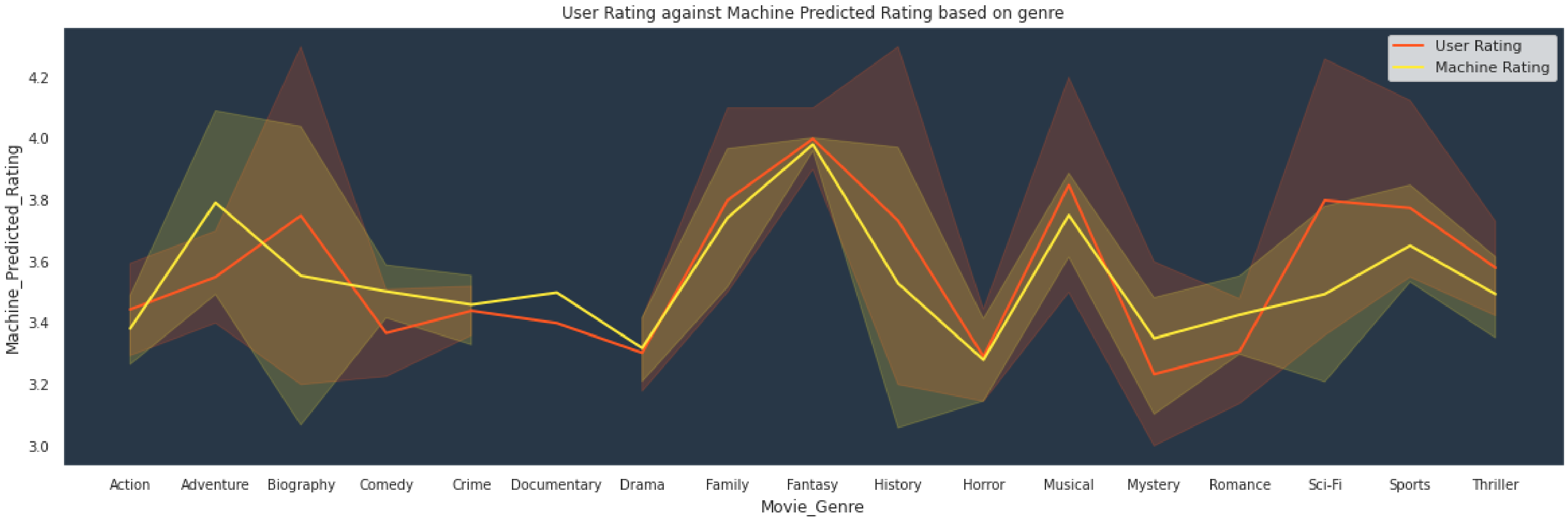
In [68]:

```
plt.figure(figsize=(20,6))
sb.set(rc={"axes.facecolor":"#283747", "axes.grid":False,'xtick.labelsize':10,'ytick.labelsize':10})
chart5 = sb.lineplot(x=dfx.Runtime_Minutes,y=dfx.Movie_User_Rating,data=dfx, color="#FF5722", label='User Rating')
chart5 = sb.lineplot(x=dfx.Runtime_Minutes,y=dfx.Movie_Critic_Rating,data=dfx, color="#FFEB3B", label='Critic Rating')
chart5.set_title('Rating Comparison against Movie runtime')
legend = plt.legend()
frame = legend.get_frame()
frame.set_facecolor('white')
plt.show()
```



In [69]:

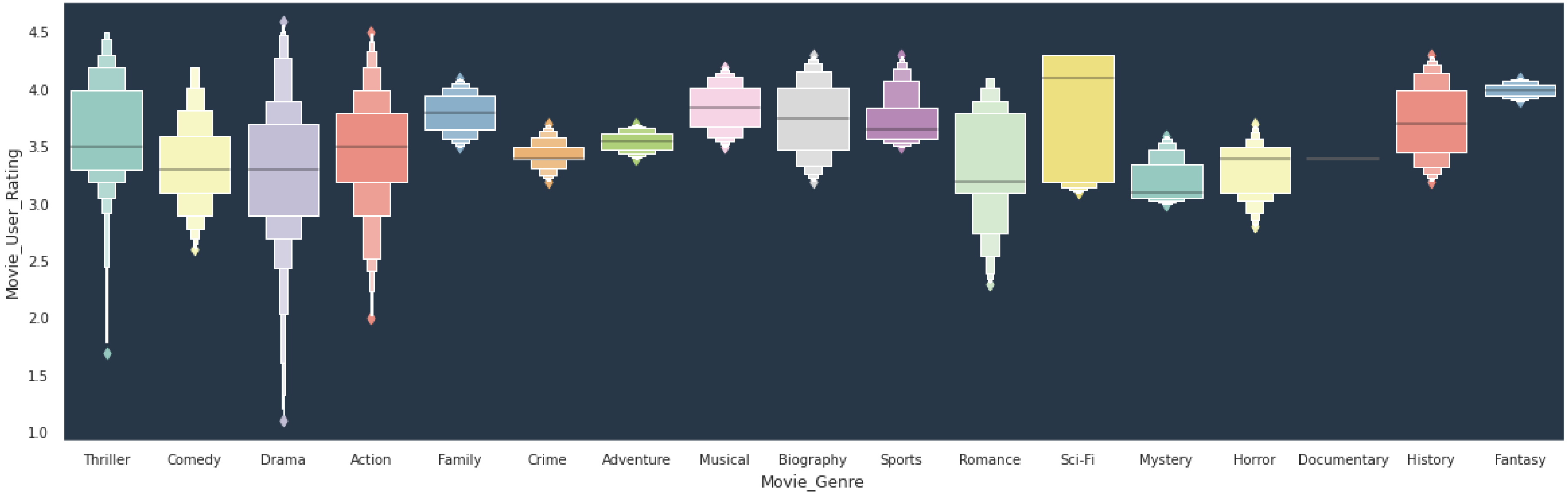
```
plt.figure(figsize=(20,6))
chart6 = sb.lineplot(x=dfx.Movie_Genre,y=dfx.Movie_User_Rating, data=dfx, color = "#FF5722", linewidth=2, label='User Rating')
chart6 = sb.lineplot(x=dfx.Movie_Genre,y=dfx.Machine_Predicted_Rating, data=dfx, color = "#FFEB3B", linewidth=2, label='Machine Rating')
chart6.set_title('User Rating against Machine Predicted Rating based on genre')
legend = plt.legend()
frame = legend.get_frame()
frame.set_facecolor('white')
plt.show()
```



In [110]:

```
mpl.rcParams.update(mpl.rcParamsDefault)
%matplotlib inline
sb.set_style("darkgrid")
```

In [119]:
```
plt.figure(figsize=(20,6))
sb.set(rc={"axes.facecolor":"#283747", "axes.grid":False,'xtick.labelsize':10,'ytick.labelsize':10})
sb.boxenplot(x=dfx.Movie_Genre, y=dfx.Movie_User_Rating, palette="Set3")
plt.show()
```



In [132]:
```
'''plt.figure(figsize=(20,6))
sb.set(rc={"axes.facecolor":"white", "axes.grid":False,'xtick.labelsize':10,'ytick.labelsize':10})
sb.boxenplot(x=dfx.Movie_Genre, y=dfx.Movie_User_Rating, palette="Set2")
plt.show()'''
```

Out[132]:
```
'plt.figure(figsize=(20,6))\nsb.set(rc={"axes.facecolor":"white", "axes.grid":False,\'xtick.labelsize\':10,\'ytick.la
belsize\':10})\nsb.boxenplot(x=dfx.Movie_Genre, y=dfx.Movie_User_Rating, palette="Set2")\nplt.show()'
```

Survey Analysis

We conducted a survey among our circles and it can be analysed as follows.

In [85]:
```
survey = pd.read_excel('/Survey.xlsx')
survey.head()
```

Out[85]:

	S.0	Name	Horror	Romance	Action-Thriller	Comedy	Sports-based	Drama	Impact of Music	Social Awareness	Reviewer Influence
0	1	Kishan	4	2	5	5	3	5	1	0	1
1	2	Abhirami	5	3	5	5	2	3	1	0	0
2	3	Jaffar	5	1	5	5	3	3	1	1	0
3	4	Premalatha	4	5	4	4	3	2	1	1	1
4	5	K.Hariharan	4	4	5	5	4	4	0	0	0

In [86]:
```
survey = survey.drop(['S.0','Name'], axis = 'columns')
survey.head()
```

Out[86]:

	Horror	Romance	Action-Thriller	Comedy	Sports-based	Drama	Impact of Music	Social Awareness	Reviewer Influence
0	4	2	5	5	3	5	1	0	1
1	5	3	5	5	2	3	1	0	0
2	5	1	5	5	3	3	1	1	0
3	4	5	4	4	3	2	1	1	1
4	4	4	5	5	4	4	0	0	0

In [87]:
```
survey['Horror'].sum()
```

Out[87]:
```
197
```

In [88]:
```
survey['Action-Thriller'].sum()
```

Out[88]:
```
211
```

In [89]:
```
survey['Comedy'].sum()
```

Out[89]:
```
193
```

In [90]:

```
survey['Sports-based'].sum()
```

Out[90]:

```
188
```

In [91]:

```
survey['Drama'].sum()
```

Out[91]:

```
200
```

In [129]:

```
survey['Romance'].sum()
```

Out[129]:

```
175
```

In [93]:

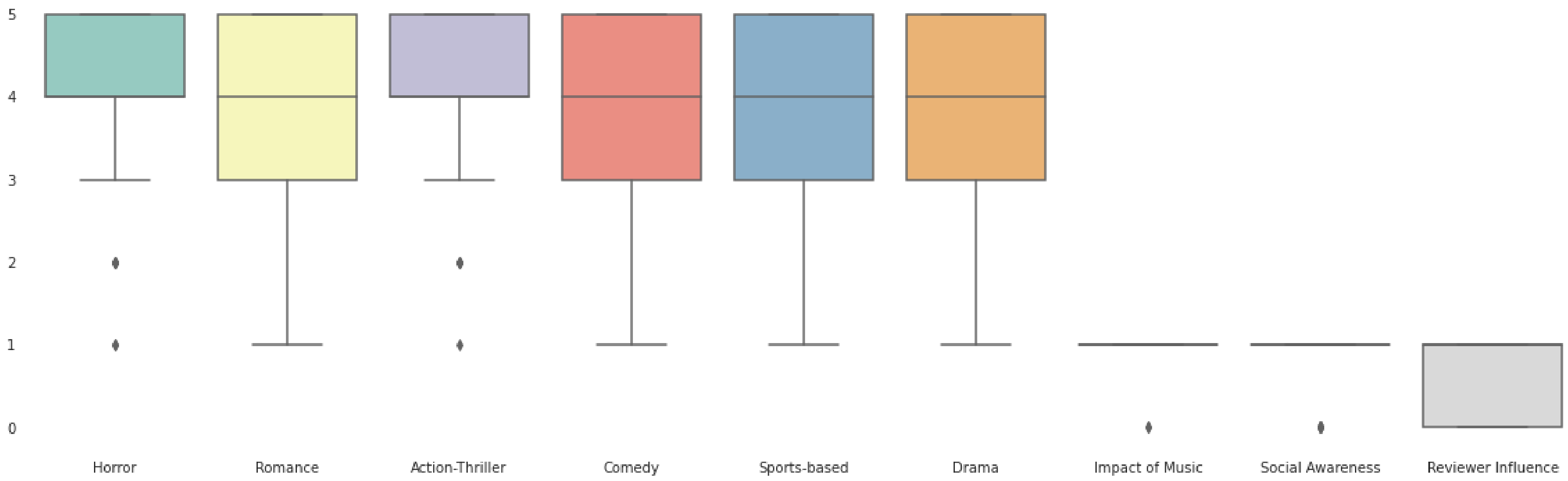
```
survey_genre = ['Horror','Action-Thriller','Romance','Comedy','Sports-based','Drama']
survey_values = [197,211,175,193,188,200]
```

In [123]:

```
plt.figure(figsize=(20,6))
sb.set(rc={"axes.facecolor":"white", "axes.grid":False,'xtick.labelsize':10,'ytick.labelsize':10})
sb.boxplot(data=survey, palette="Set3")
```

Out[123]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa42db8a048>
```



Survey Conclusion

The sample that we selected had very similar interests. It explains that they prefer the presence of every aspect of every genre in general.

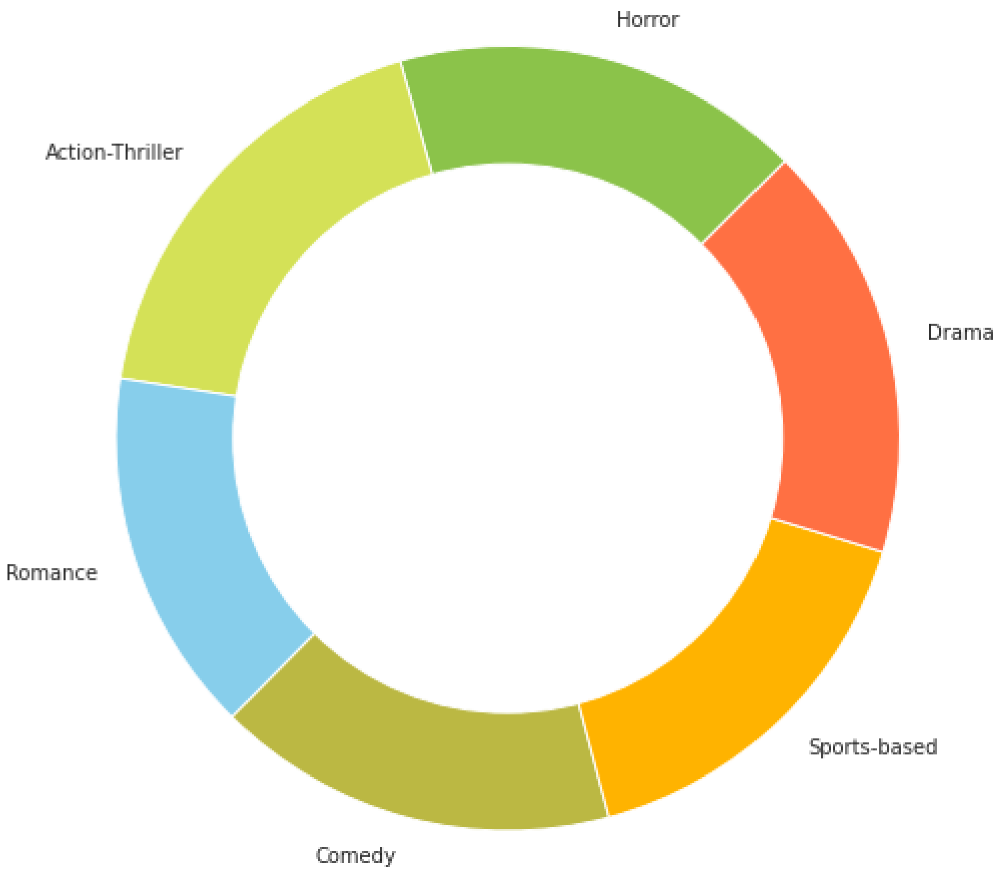
So our audience would like to watch a movie with every element of art in it like a combo of flavours. They would like to have a touch in every emotion they prefer.

Also, our audience feel like the presence of a strong social message in a film would be very much influential.

Majority of the audience also positively agree to the impact of music on the success of the movie.

In [130]:

```
plt.figure(figsize=(9,9))
survey_values = [197,211,175,193,188,200]
labels = ['Horror','Action-Thriller','Romance','Comedy','Sports-based','Drama']
colors = ['#8BC34A','#D4E157', 'skyblue','#BBB843','#FFB300','#FF7043']
plt.pie (survey_values , labels= labels , colors= colors , startangle=45)
my_circle=plt.Circle( (0,0), 0.7, color='white') # Adding circle at the centre
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.show()
```


Horror
Action-Thriller
Drama
Romance
Sports-based
Comedy