In [2]:
```
#importing necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
df = pd.read_csv("C:\\Users\\skp18\\Downloads\\archive (2)\\IMDb Movies India.csv", encoding="latin1")
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15509 entries, 0 to 15508
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Name      15509 non-null  object 
 1   Year      14981 non-null  object 
 2   Duration  7240 non-null   object 
 3   Genre     13632 non-null  object 
 4   Rating    7919 non-null   float64
 5   Votes     7920 non-null   object 
 6   Director  14984 non-null  object 
 7   Actor 1   13892 non-null  object 
 8   Actor 2   13125 non-null  object 
 9   Actor 3   12365 non-null  object 
dtypes: float64(1), object(9)
memory usage: 1.2+ MB
```

In [5]:
```
df.head()
```

Out[5]:

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		NaN	NaN	Drama	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
2	#Homecoming	(2021)	90 min	Drama, Musical	NaN	NaN	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana
3	#Yaaram	(2019)	110 min	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor
4	...And Once Again	(2010)	105 min	Drama	NaN	NaN	Amol Palekar	Rajat Kapoor	Rituparna Sengupta	Antara Mali

In [6]:
```
df.describe()
```

Out[6]:

	Rating
count	7919.000000
mean	5.841621
std	1.381777
min	1.100000
25%	4.900000
50%	6.000000
75%	6.800000
max	10.000000

In [7]:
```
#checking for null values and cleaning data
df.isna().sum()/len(df)*100
```

Out[7]:
```
Name         0.000000
Year         3.404475
Duration    53.317429
Genre       12.102650
Rating      48.939326
Votes       48.932878
Director     3.385131
Actor 1     10.426204
Actor 2     15.371720
Actor 3     20.272100
dtype: float64
```

In [8]:
```
df.drop(['Actor 2' , 'Actor 3'], axis=1, inplace=True)
df.dropna(subset=['Duration'], inplace = True)
df.dropna(subset=['Rating', 'Votes'], inplace=True)
director_desc = df['Director'].describe()

director_count = df['Director'].value_counts().sort_values(ascending=False)
df['Director'].fillna('rajmouli', inplace=True)

genre_counts = df['Genre'].value_counts().sort_values(ascending=False)
df['Genre'].fillna('Action', inplace=True)

actor1_desc = df['Actor 1'].describe()
df['Actor 1'].fillna('mahesh babu', inplace=True)
```

In [9]:

```
df['Year'] = df['Year'].str.replace(r'[()]', '', regex=True)
df['Duration'] = df['Duration'].str.replace(r' min', '', regex=True)
df.info()

int_columns = ['Year', 'Duration']
df[int_columns] = df[int_columns].astype(int)
df['Votes'] = df['Votes'].str.replace(',', '').astype(int)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5851 entries, 1 to 15508
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Name      5851 non-null   object 
 1   Year      5851 non-null   object 
 2   Duration  5851 non-null   object 
 3   Genre     5851 non-null   object 
 4   Rating    5851 non-null   float64
 5   Votes     5851 non-null   object 
 6   Director  5851 non-null   object 
 7   Actor 1   5851 non-null   object 
dtypes: float64(1), object(7)
memory usage: 411.4+ KB
<class 'pandas.core.frame.DataFrame'>
Index: 5851 entries, 1 to 15508
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Name      5851 non-null   object 
 1   Year      5851 non-null   int32  
 2   Duration  5851 non-null   int32  
 3   Genre     5851 non-null   object 
 4   Rating    5851 non-null   float64
 5   Votes     5851 non-null   int32  
 6   Director  5851 non-null   object 
 7   Actor 1   5851 non-null   object 
dtypes: float64(1), int32(3), object(4)
memory usage: 342.8+ KB
```

In [10]:
```
df.dropna(subset=['Name','Year','Duration','Votes','Rating'],inplace=True)
df.isna().sum()
```

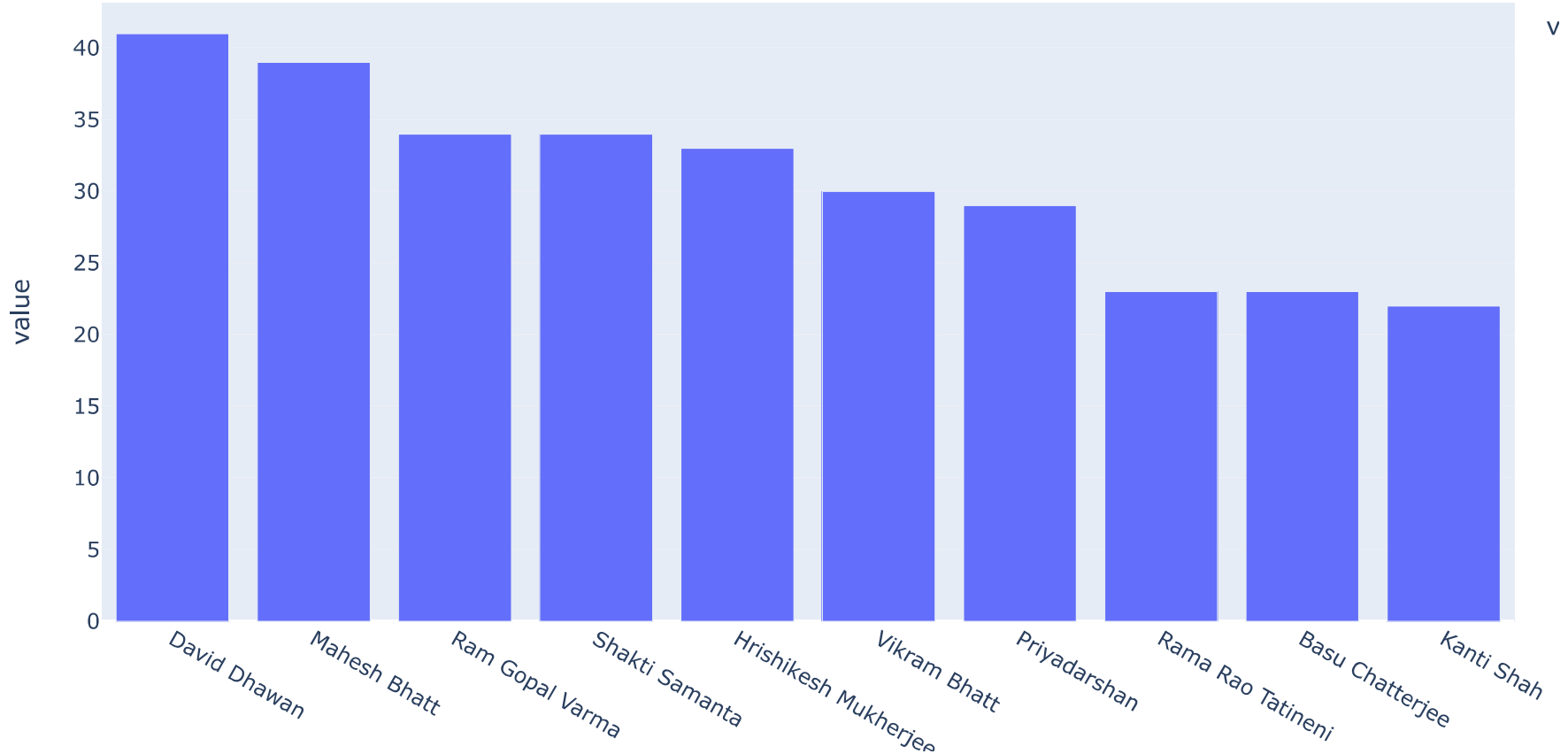
Out[10]:
```
Name        0
Year        0
Duration    0
Genre       0
Rating      0
Votes       0
Director    0
Actor 1     0
dtype: int64
```

In [11]:
```
df.head()
```

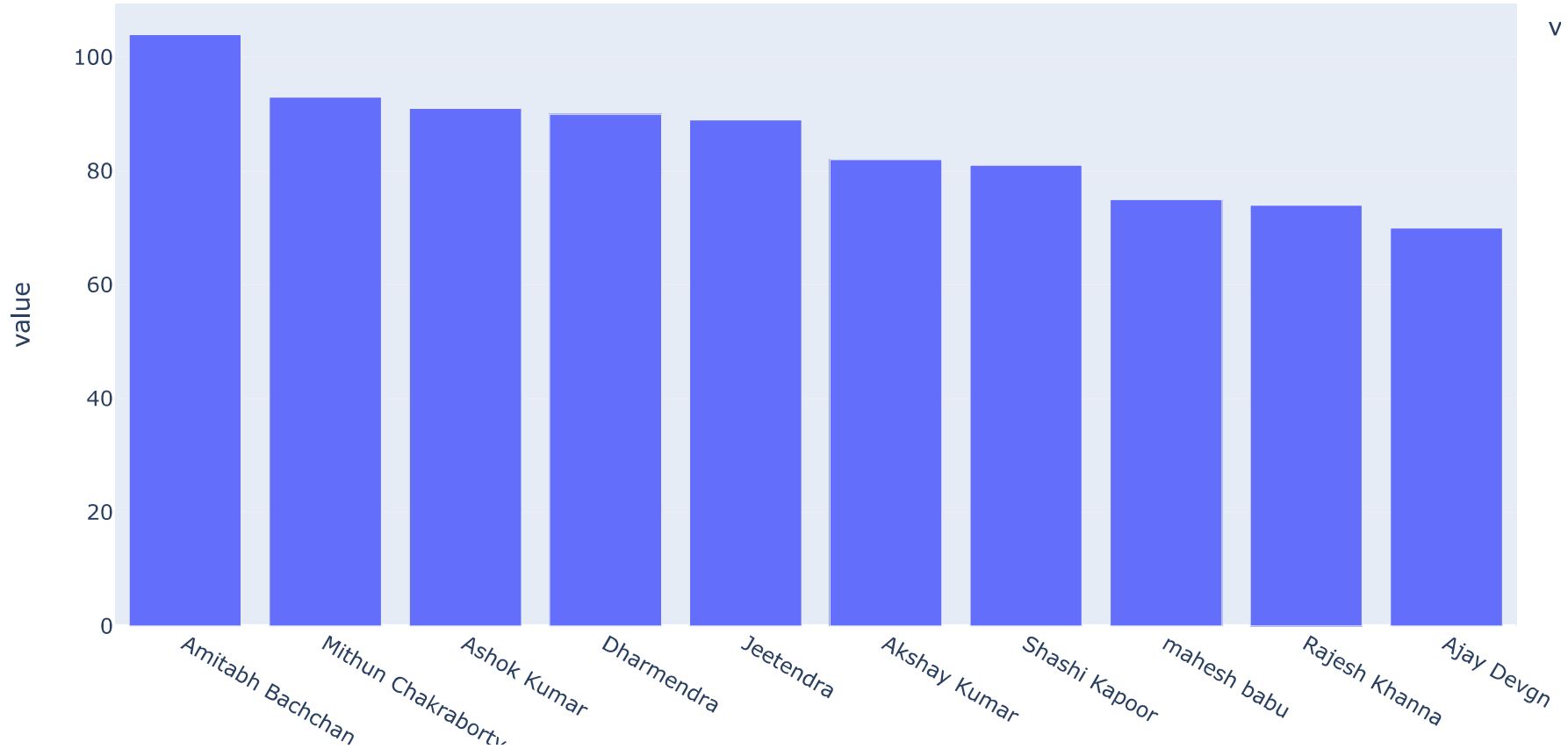
Out[11]:

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1
1	#Gadhvi (He thought he was Gandhi)	2019	109	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal
3	#Yaaram	2019	110	Comedy, Romance	4.4	35	Ovais Khan	Prateik
5	...Aur Pyaar Ho Gaya	1997	147	Comedy, Drama, Musical	4.7	827	Rahul Rawail	Bobby Deol
6	...Yahaan	2005	142	Drama, Romance, War	7.4	1086	Shoojit Sircar	Jimmy Sheirgill
8	?: A Question Mark	2012	82	Horror, Mystery, Thriller	5.6	326	Allyson Patel	Yash Dave

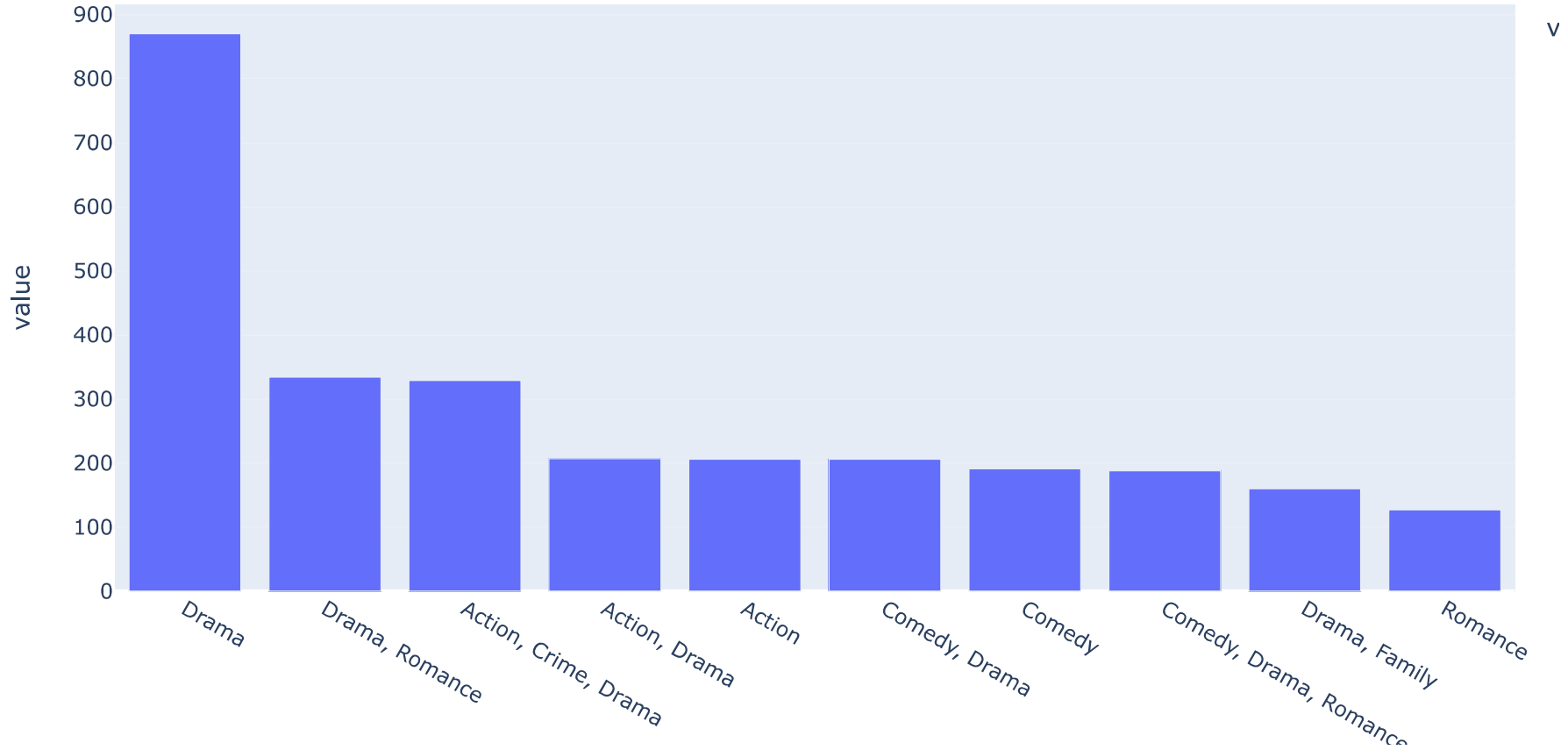
In [12]:
```
#finding out director with most movies.

top_directors = df['Director'].value_counts().head(10)  #top 10 directors
import plotly.express as px

bargraph = px.bar(top_directors)
bargraph.show()
```



In [13]:
```
actor_count = df['Actor 1'].value_counts().head(10)  #top 10 actors
bargraph = px.bar(actor_count)
bargraph.show()
```



In [14]:
```
#top 10 genre
genre_count=df['Genre'].value_counts().head(10)
bargraph = px.bar(genre_count)
bargraph.show()
```

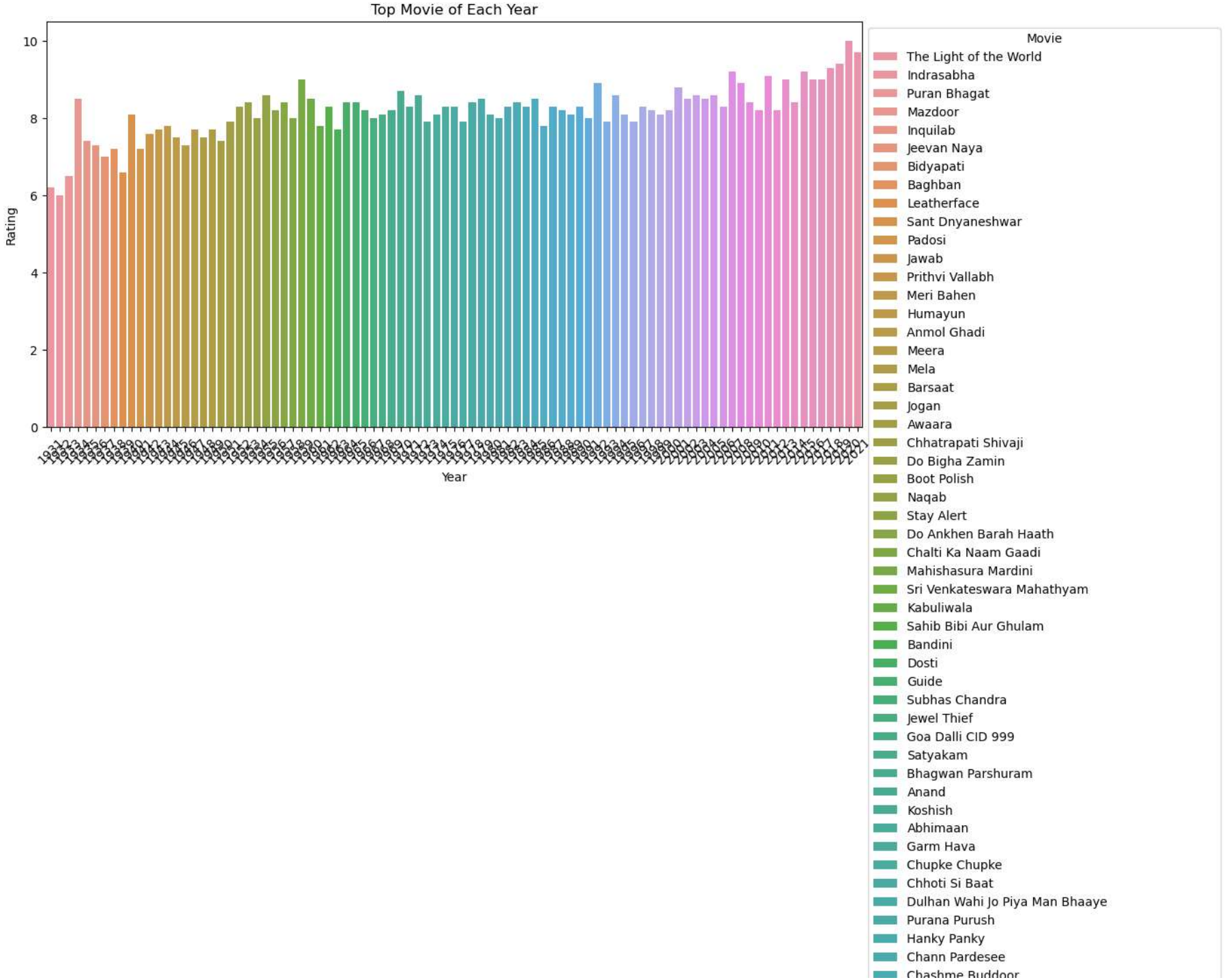


In [15]:
```
#top movies of each year

top_movie_indices = df.groupby('Year')['Rating'].idxmax()
top_movies = df.loc[top_movie_indices]
print(top_movies[['Year', 'Name', 'Rating','Genre','Director']])
```

```
       Year                    Name  Rating             Genre  \
14161  1931  The Light of the World     6.2    Drama, Fantasy
6073   1932              Indrasabha     6.0  Musical, Romance
11138  1933            Puran Bhagat     6.5            Action
9053   1934                 Mazdoor     8.5             Drama
6087   1935                Inquilab     7.4             Drama
...     ...                     ...     ...               ...
11841  2017     Rediscovering India     9.0       Documentary
1314   2018            Ashok Vatika     9.3             Drama
5077   2019            Gho Gho Rani     9.4  History, Romance
8339   2020         Love Qubool Hai    10.0    Drama, Romance
5410   2021              Half Songs     9.7    Music, Romance

                    Director
14161          Ardeshir Irani
6073               J.J. Madan
11138              Debaki Bose
9053   Mohan Dayaram Bhavnani
6087               Debaki Bose
...                        ...
11841            Meenal Dixit
1314            Rahul Mallick
5077             Munni Pankaj
8339          Saif Ali Sayeed
5410              Sriram Raja

[91 rows x 5 columns]
```

In [16]:

```
plt.figure(figsize=(12, 6))
bar_plot = sns.barplot(x='Year', y='Rating', hue='Name', data=top_movies, dodge=False)
bar_plot.set_title('Top Movie of Each Year')
bar_plot.set_xlabel('Year')
plt.xticks(rotation=45)
bar_plot.set_ylabel('Rating')
bar_plot.legend(title='Movie', bbox_to_anchor=(1, 1), loc='upper left')
plt.show()
```


Top Movie of Each Year
Rating
Year
Movie
The Light of the World
Indrasabha
Puran Bhagat
Mazdoor
Inquilab
Jeevan Naya
Bidyapati
Baghban
Leatherface
Sant Dnyaneshwar
Padosi
Jawab
Prithvi Vallabh
Meri Bahen
Humayun
Anmol Ghadi
Meera
Mela
Barsaat
Jogan
Awaara
Chhatrapati Shivaji
Do Bigha Zamin
Boot Polish
Naqab
Stay Alert
Do Ankhen Barah Haath
Chalti Ka Naam Gaadi
Mahishasura Mardini
Sri Venkateswara Mahathyam
Kabuliwala
Sahib Bibi Aur Ghulam
Bandini
Dosti
Guide
Subhas Chandra
Jewel Thief
Goa Dalli CID 999
Satyakam
Bhagwan Parshuram
Anand
Koshish
Abhimaan
Garm Hava
Chupke Chupke
Chhoti Si Baat
Dulhan Wahi Jo Piya Man Bhaaye
Purana Purush
Hanky Panky
Chann Pardesee


Angoor
Jaane Bhi Do Yaaro
Chandrawal
Kuk Doo Koo
Punnagai Mannan
Permission
Libaas
Dhruvanakshatram
Ek Doctor Ki Maut
Prahaar: The Final Attack
Ram Ke Naam
Sardar
Pitra, Putra Aur Dharamyuddha
Dilwale Dulhania Le Jayenge
Halo
The Chalbaaz
Satya
Sarfarosh
Hera Pheri
Laadli
Simhadriya Simha
Chale Chalo: The Lunacy of Film Making
Black Friday
Sunset Bollywood
Khosla Ka Ghosla!
I'm in Love
Leaving Home: The Life and Music of Indian Ocean
3 Idiots
Bohlol Dana - A Sage of Baghdad
Nirvana13
Gangs of Wasseypur
Foresting Life
Goonga Pehelwan
The Flip Side: A Truth That Could Not Reach You
Nashebaaz
Rediscovering India
Ashok Vatika
Gho Gho Rani
Love Qubool Hai
Half Songs

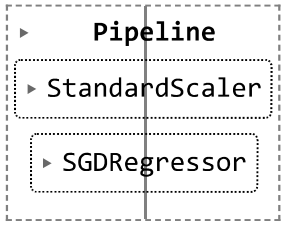
In [24]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.linear_model import SGDRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

X = df[['Year','Duration','Votes']]
y = df['Rating']
```

In [25]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,random_state=1000)
```

In [26]:
```
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('sgd', SGDRegressor(max_iter=10000, random_state=1000))
])
```

In [27]:
```
pipeline.fit(X_train, y_train)
```

Out[27]:



In [28]:
```
# Predict ratings on the test set
y_pred_pipeline = pipeline.predict(X_test)
```

In [29]:
```
# Evaluation Metrics for the Pipeline
mae_pipeline = mean_absolute_error(y_test, y_pred_pipeline)
mse_pipeline = mean_squared_error(y_test, y_pred_pipeline)
r2_pipeline = r2_score(y_test, y_pred_pipeline)
```

In [30]:
```
print("Pipeline Mean Absolute Error:", mae_pipeline)
print("Pipeline Mean Squared Error:", mse_pipeline)
print("Pipeline R-squared:", r2_pipeline)
```

```
Pipeline Mean Absolute Error: 1.040142363499226
Pipeline Mean Squared Error: 1.75589466147756
Pipeline R-squared: 0.037929023872087186
```

In [ ]: