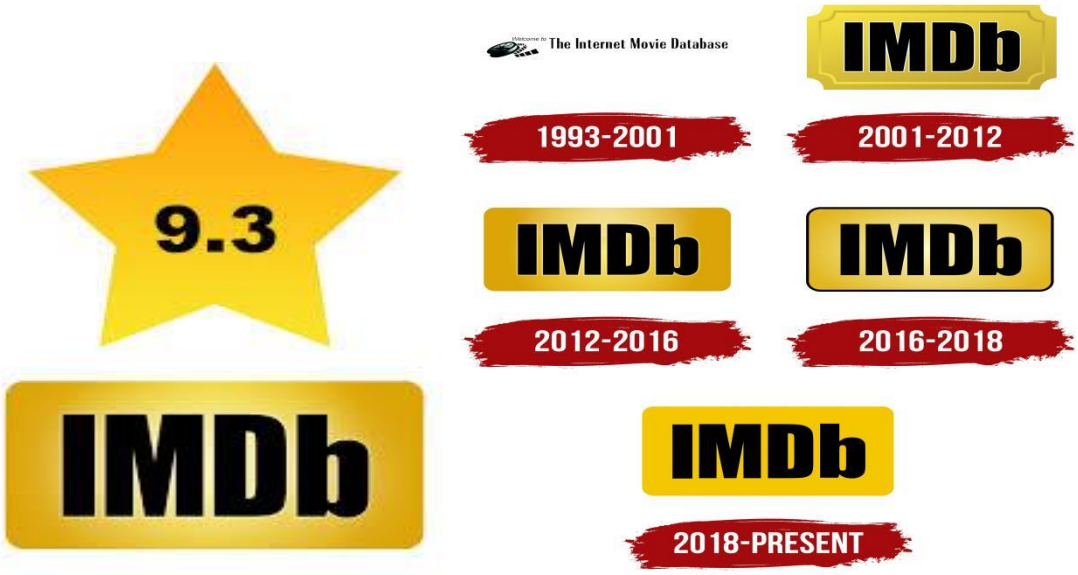
TITLE: IMDb Score Prediction using Data Science

PHASE 3: DEVELOPMENT PART 1

LOADING AND DATA PREPROCESSING

TEAM MEMBER NAME : MONICA HIRIN M

NAAN MUTHALVAN ID : au723721243033



INTRODUCTION:

IMDb scores are determined by user ratings and can change over time as more users rate the movie or show.

The problem is to develop a machine learning model to predict the IMDb scores of movies available on Films based on their genre, premiere date, runtime, and language.

This project involves data collection, data prepossessing, feature engineering, clustering algorithms, visualization, and interpretation of results.

The model aims to accurately estimate the popularity of movies to assist users in discovering highly rated films that align with their preferences.

WORKS DONE IN PREVIOUS PHASES:

DEFINITION PHASE:

Develop a machine learning model to predict the IMDb scores of movies available on Films based on their genre, premiere date, runtime, and language.

INNOVATION PHASE: In this inovation phase of our IMDb score prediction project you can explore advanced techniques and methods to improve the accuracy of prediction.

PHASE 3:

DEVELOPMENT PHASE:

These phases can be executed using three parts

- Loading and Pre-processing data
- Training and Testing data
- Model testing and Displaying Output

IMPORTING LIBRARIES:

We importing the necessary Python libraries, such as

- Pandas for data manipulation
- NumPy for analysis,
- Matplotlib for visualization.

Loading the dataset:

- To load data points from a file (e.g., a CSV file), you can use the **pd.read.csv()** function.

Dataset link : https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores/

The dataset looks like below,

Title	Genre	Premiere	Runtime	IMDB Score	Language
Enter the A	Documenta	5-Aug-19	58	2.5	English/Japanese
Dark Forces	Thriller	21-Aug-20	81	2.6	Spanish
The App	Science fict	26-Dec-19	79	2.6	Italian
The Open H	Horror thril	19-Jan-18	94	3.2	English
Kaali Khuhi	Mystery	30-Oct-20	90	3.4	Hindi
Drive	Action	1-Nov-19	147	3.5	Hindi
Leyla Everla	Comedy	4-Dec-20	112	3.7	Turkish
The Last Da	Heist film/	5-Jun-20	149	3.7	English
Paradox	Musical/We	23-Mar-18	73	3.9	English
Sardar Ka G	Comedy	18-May-21	139	4.1	Hindi
Searching f	Documenta	22-Apr-21	58	4.1	English
The Call	Drama	27-Nov-20	112	4.1	Korean
Whipped	Romantic c	18-Sep-20	97	4.1	Indonesian
All Because	Action com	1-Oct-20	101	4.2	Malay
Mercy	Thriller	22-Nov-16	90	4.2	English
After the R	Documenta	19-Dec-19	25	4.3	Spanish
Ghost Stori	Horror anth	1-Jan-20	144	4.3	Hindi
The Last Th	Political thr	21-Feb-20	115	4.3	English
What Happ	Comedy	1-Jan-21	102	4.3	Korean
Death Note	Horror thril	25-Aug-17	100	4.4	English
Hello Privil	Documenta	13-Sep-19	64	4.4	English

The Girl on	Thriller	26-Feb-21	120	4.4	Hindi
Thunder Fo	Superhero-	9-Apr-21	105	4.4	English
Fatal Affair	Thriller	16-Jul-20	89	4.5	English
Just Say Ye	Romantic c	2-Apr-21	97	4.5	Dutch
Seriously Si	Comedy	31-Jul-20	107	4.5	English
The Misadv	Comedy	10-Feb-21	99	4.5	French
5 Star Chris	Comedy	7-Dec-18	95	4.6	Italian
After Maria	Documenta	24-May-19	37	4.6	English/Spanish
I Am the Pr	Horror	28-Oct-16	89	4.6	English
Paris Is Us	Romance d	22-Feb-19	83	4.6	French
Porta dos F	Comedy	3-Dec-19	46	4.6	Portuguese
Rattlesnake	Horror	25-Oct-19	85	4.6	English
The Players	Comedy	15-Jul-20	88	4.6	Italian
We Are On	Documenta	14-Jul-20	86	4.6	French
Finding Agr	Drama	30-Nov-20	105	4.7	Filipino
IO	Science fict	18-Jan-19	95	4.7	English
Sentinelle	Action	5-Mar-21	80	4.7	French
Sol Levante	Anime / Sh	2-Apr-20	4	4.7	English
The Bindin	Drama	2-Oct-20	93	4.7	Italian
We Can Be	Superhero	25-Dec-20	100	4.7	English
Christmas (	Thriller	4-Dec-20	106	4.8	German
Coin Heist	Heist	6-Jan-17	97	4.8	English

Here’s the code for importing the libraries,

```
import numpy as np
import pandas as pd
import osfor dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as pxfrom datetime
import datetime,timedelta
```

Dataset

```
ds = pd.read_csv("/kaggle/input/netflix-original-films-imdb-scores/NetflixOriginals.csv",encoding = "ISO-8859-1")
```

```
ds_date = ds.copy()
```

```
ds.head(5)
```

	Title	Genre	Premiere	Runtime	IMDB Score	Language
0	Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese
1	Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish
2	The App	Science fiction/Drama	December 26, 2019	79	2.6	Italian

	Title	Genre	Premiere	Runtime	IMDB Score	Language
3	The Open House	Horror thriller	January 19, 2018	94	3.2	English
4	Kaali Khuhi	Mystery	October 30, 2020	90	3.4	Hindi

```
ds.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Runtime	584.0	93.577055	27.761683	4.0	86.0	97.00	108.0	209.0
IMDB Score	584.0	6.271747	0.979256	2.5	5.7	6.35	7.0	9.0

insights: categorical of IMDB Score 5.7 > rendah 6.35 > sedang 7.0 > tinggi 9.0 > sangat tinggi

```
<class 'pandas.core.frame.DataFrame'>

RangeIndex: 584 entries, 0 to 583
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Title       584 non-null    object
 1   Genre       584 non-null    object
 2   Premiere    584 non-null    object
 3   Runtime     584 non-null    int64 
```

```
 4   IMDB Score  584 non-null    float64
 5   Language    584 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 27.5+ KB
```

```
ds.isna().sum()
```

```
Title         0
Genre         0
Premiere      0
Runtime       0
IMDB Score    0
Language      0
dtype: int64
```

```
ds['Title'].value_counts()
```

```
Enter the Anime                                  1
Have a Good Trip: Adventures in Psychedelics     1
Tallulah                                         1
The Old Guard                                    1
Tony Robbins: I Am Not Your Guru                 1
Cam                                              1
Earthquake Bird                                  1
Frankenstein's Monster's Monster, Frankenstein   1
Horse Girl                                       1
David Attenborough: A Life on Our Planet         1
Name: Title, Length: 584, dtype: int64
```

```
ds['Genre'].value_counts()
```

```
Documentary                          159
Drama                                 77
Comedy                                49
Romantic comedy                       39
Thriller                              33
                                     ...
Romantic comedy-drama                  1
Heist film/Thriller                    1
Musical/Western/Fantasy                1
Horror anthology                       1
Animation/Christmas/Comedy/Adventure   1
Name: Genre, Length: 115, dtype: int64
```

```
ds['Premiere'].value_counts()
```

```
October 2, 2020          6
November 1, 2019         5
October 18, 2019         5
November 2, 2018         4
June 19, 2020            4
                        ..
September 20, 2019       1
March 10, 2017           1
March 17, 2017           1
May 29, 2015             1
October 4, 2020          1
Name: Premiere, Length: 390, dtype: int64
```

```
ds_date["Premiere"] = ds_date["Premiere"].apply(lambda x: "".join(x for x in x.replace(".",",")))
ds_date["PremiereDate"] = ds_date["Premiere"].apply(lambda x: datetime.strptime(x, "%B %d, %Y").date())
```

```
ds_date["Year"] = ds_date["Premiere"].apply(lambda x: "".join(x for x in x.replace(",","").split()[-1]))

ds_date["PremiereDate"] = pd.to_datetime(ds_date["PremiereDate"])
ds_date
```

Title	Genre	Premiere	Runtime	IMDB Score	Language	PremiereDate	Year	
0	Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese	2019-08-05	2019
1	Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish	2020-08-21	2020
2	The App	Science fiction/ Drama	December 26, 2019	79	2.6	Italian	2019-12-26	2019
3	The Open House	Horror thriller	January 19, 2018	94	3.2	English	2018-01-19	2018
4	Kaali Khuhi	Mystery	October 30, 2020	90	3.4	Hindi	2020-10-30	2020
...	...	...	...	...	...	...	...	...
57	Taylor	Concert	Dece	125	8.4	English	20	20

Title	Genre	Premiere	Runtime	IMDB Score	Language	PremiereDate	Year	
9	Swift: Reputation Stadium Tour	Film	mber 31, 2018				18-12-31	18
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	October 9, 2015	91	8.4	English/Ukranian/Russian	2015-10-09	2015
581	Springsteen on Broadway	One-man show	December 16, 2018	153	8.5	English	2018-12-16	2018
582	Emicida: AmarElo - It's All For Yesterday	Documentary	December 8, 2020	89	8.6	Portuguese	2020-12-08	2020
583	David Attenborough: A Life on Our Planet	Documentary	October 4, 2020	83	9.0	English	2020-10-04	2020

```
ds_date.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 584 entries, 0 to 583
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Title         584 non-null    object
 1   Genre         584 non-null    object
 2   Premiere      584 non-null    object
 3   Runtime       584 non-null    int64
 4   IMDB Score    584 non-null    float64
 5   Language      584 non-null    object
 6   PremiereDate  584 non-null    datetime64[ns]
 7   Year          584 non-null    object
dtypes: datetime64[ns](1), float64(1), int64(1), object(5)
memory usage: 36.6+ KB
```

```
ds['Language'].value_counts()
```

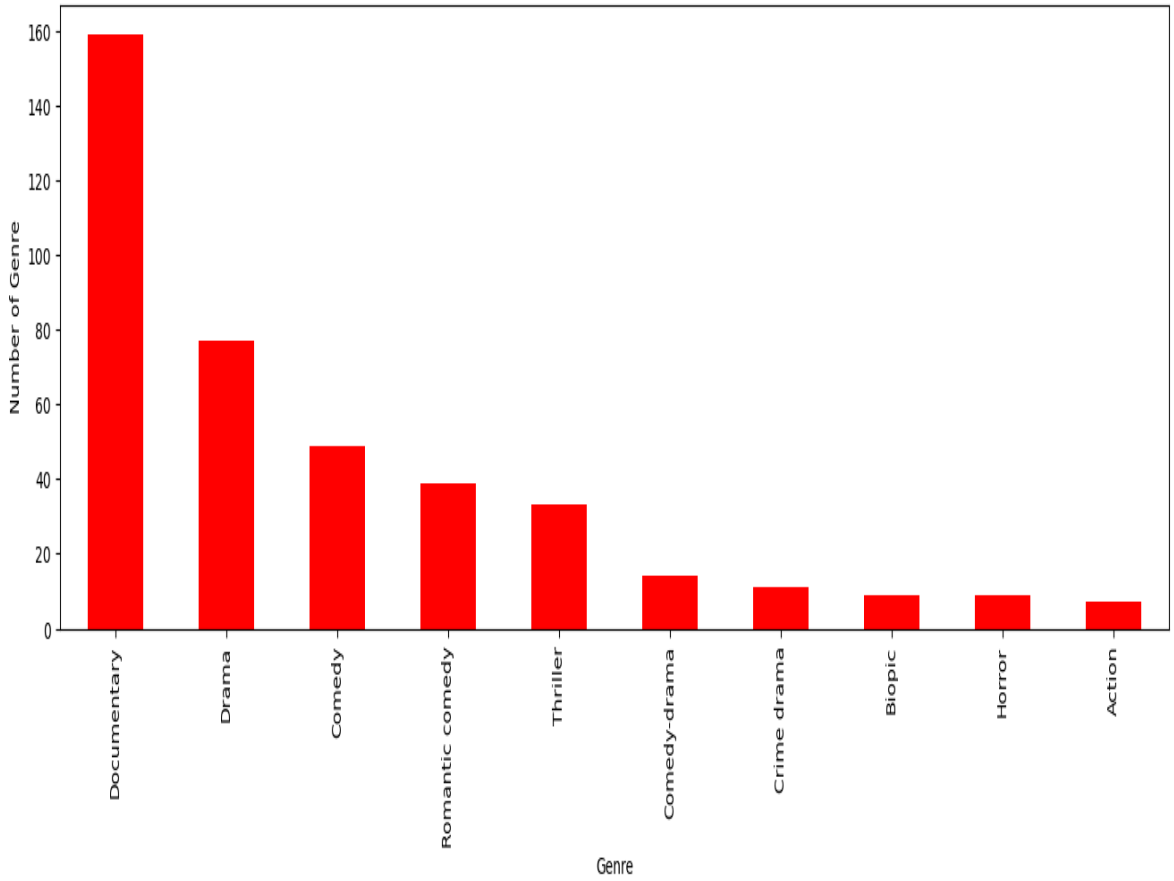
```
English                 401
Hindi                    33
Spanish                  31
French                   20
Italian                  14
Portuguese               12
Indonesian                9
Japanese                  6
Korean                    6
German                    5
Turkish                   5
English/Spanish           5
Polish                    3
Dutch                     3
Marathi                   3
English/Hindi             2
Thai                      2
English/Mandarin          2
English/Japanese          2
```

```
Filipino                    2
English/Russian             1
Bengali                     1
English/Arabic              1
English/Korean              1
Spanish/English             1
Tamil                       1
English/Akan                1
Khmer/English/French        1
Swedish                     1
Georgian                    1
Thia/English                1
English/Taiwanese/Mandarin  1
English/Swedish             1
Spanish/Catalan             1
Spanish/Basque              1
Norwegian                   1
Malay                       1
English/Ukranian/Russian    1
Name: Language, dtype: int64
```

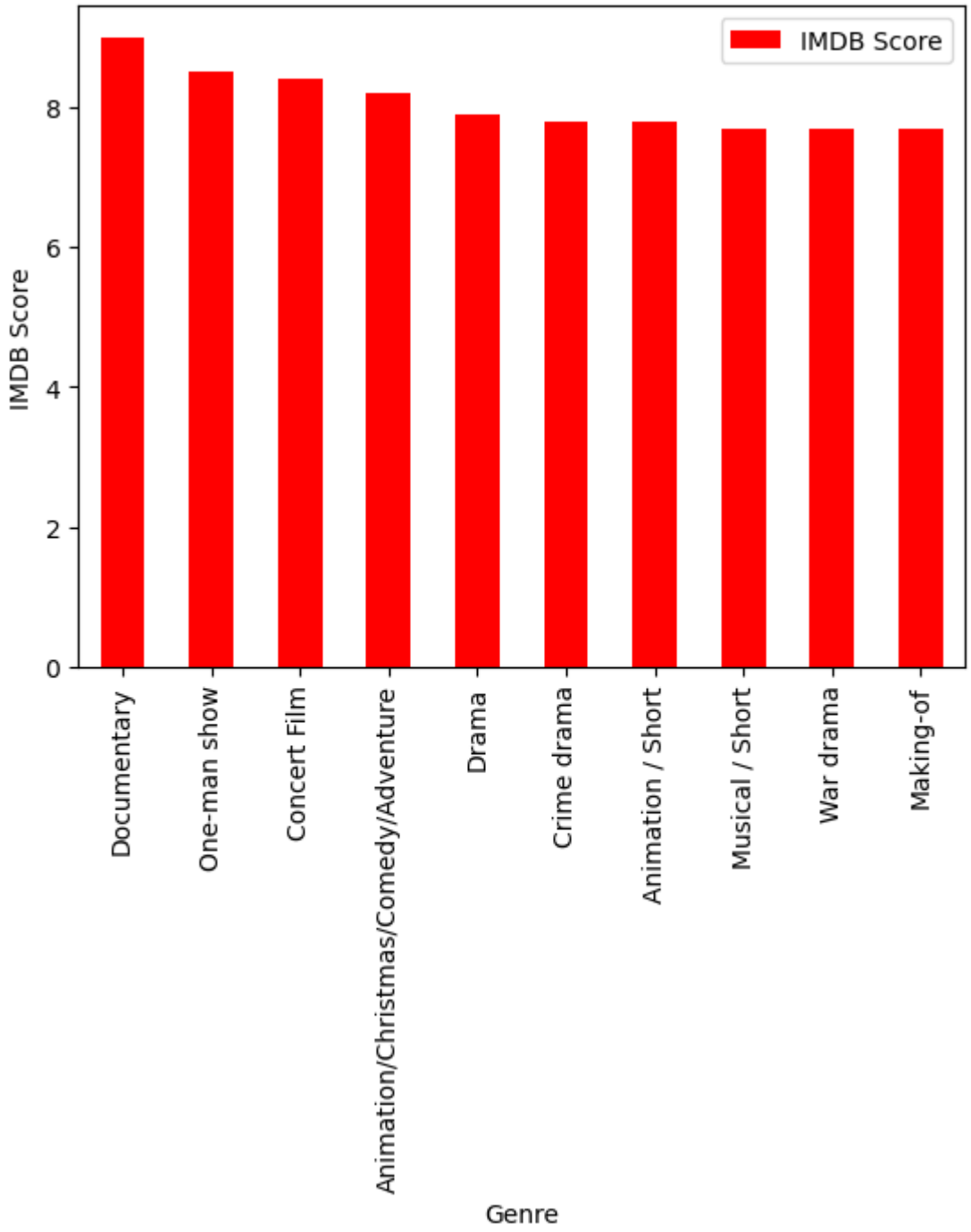
```
ds['Genre'].value_counts()genre = ds['Genre'].value_counts()genre.head()
```

```
Documentary       159
Drama              77
Comedy             49
Romantic comedy    39
Thriller           33
Name: Genre, dtype: int64
```

```
plt.figure(figsize=(16, 5))ds['Genre'].value_counts().head(10).plot(kind='bar', color='red')plt.xlabel('Genre')plt.ylabel('Number of Genre')plt.xticks(rotation=90)plt.show(block=True)
```




```
ds[['Genre', 'IMDB Score']].sort_values('IMDB Score', ascending=False).drop_duplicates('Genre').head(10).plot(x='Genre', y='IMDB Score', kind='bar', color='red')
plt.xlabel('Genre')
plt.ylabel('IMDB Score')
plt.show(block=True)
```



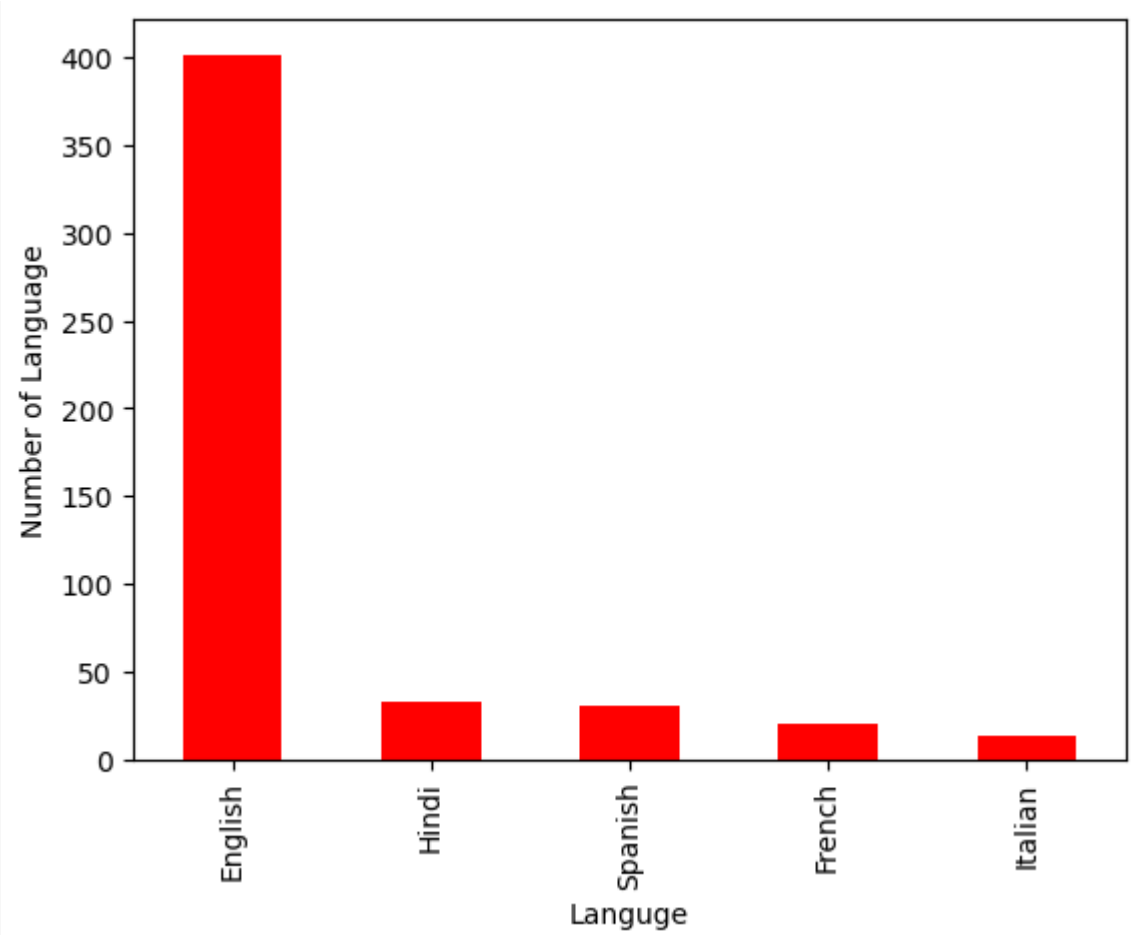

```
ds['Language'].value_counts()
```

English	401
Hindi	33
Spanish	31
French	20
Italian	14
Portuguese	12

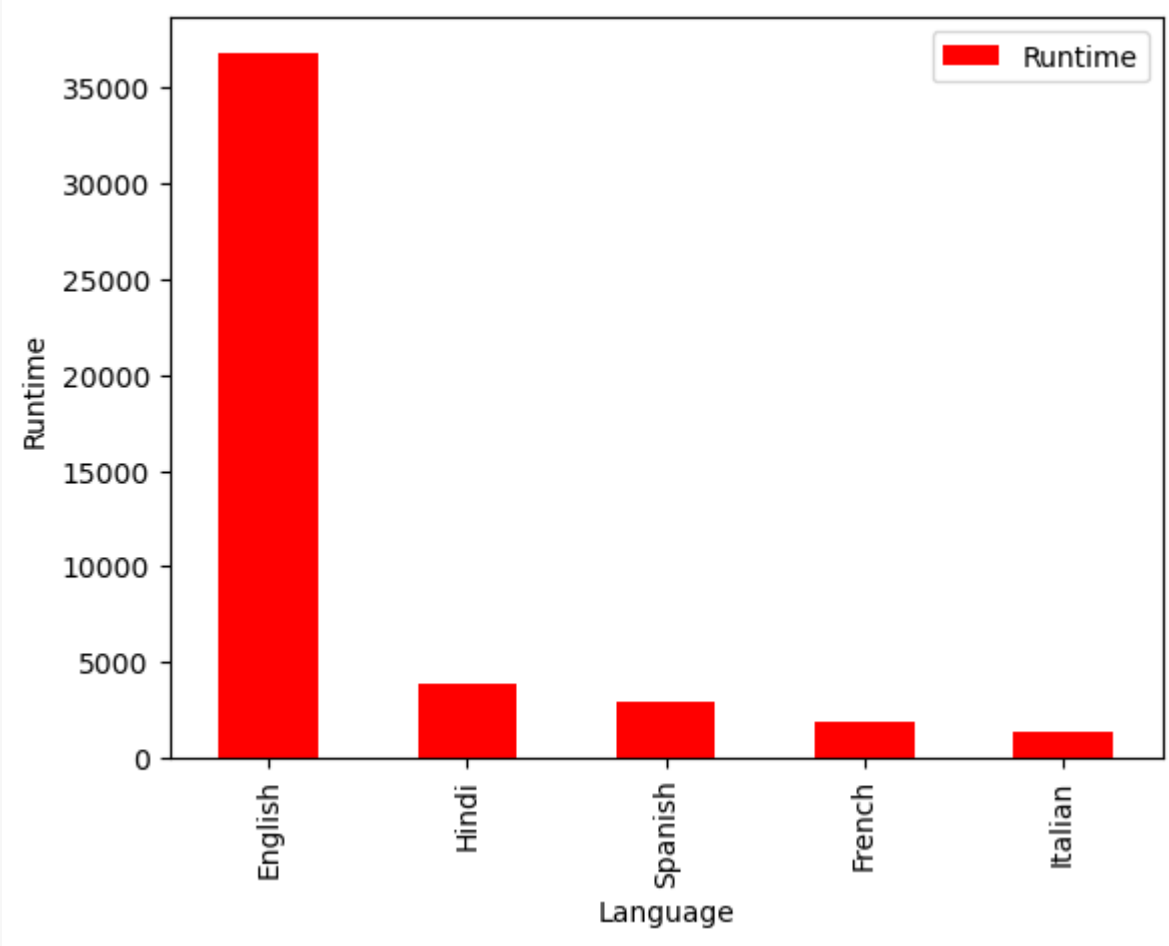
Indonesian	9
Japanese	6
Korean	6
German	5
Turkish	5
English/Spanish	5
Polish	3
Dutch	3
Marathi	3
English/Hindi	2
Thai	2
English/Mandarin	2
English/Japanese	2
Filipino	2
English/Russian	1
Bengali	1
English/Arabic	1
English/Korean	1
Spanish/English	1
Tamil	1
English/Akan	1
Khmer/English/French	1
Swedish	1
Georgian	1
Thia/English	1
English/Taiwanese/Mandarin	1
English/Swedish	1
Spanish/Catalan	1
Spanish/Basque	1
Norwegian	1
Malay	1
English/Ukranian/Russian	1

Name: Language, dtype: int64

```
ds_lang = ds['Language'].value_counts()
ds_lang.head(5).plot(kind='bar', color='red')
plt.xlabel('Languge')
plt.ylabel('Number of Language')
plt.show(block=True)
```




```
ds.groupby('Language').agg({'Runtime': 'sum'}).sort_values('Runtime', ascending=False).head(5).plot(kind='bar', color='red')
plt.xlabel('Language')
plt.ylabel('Runtime')
plt.show(block=True)
```

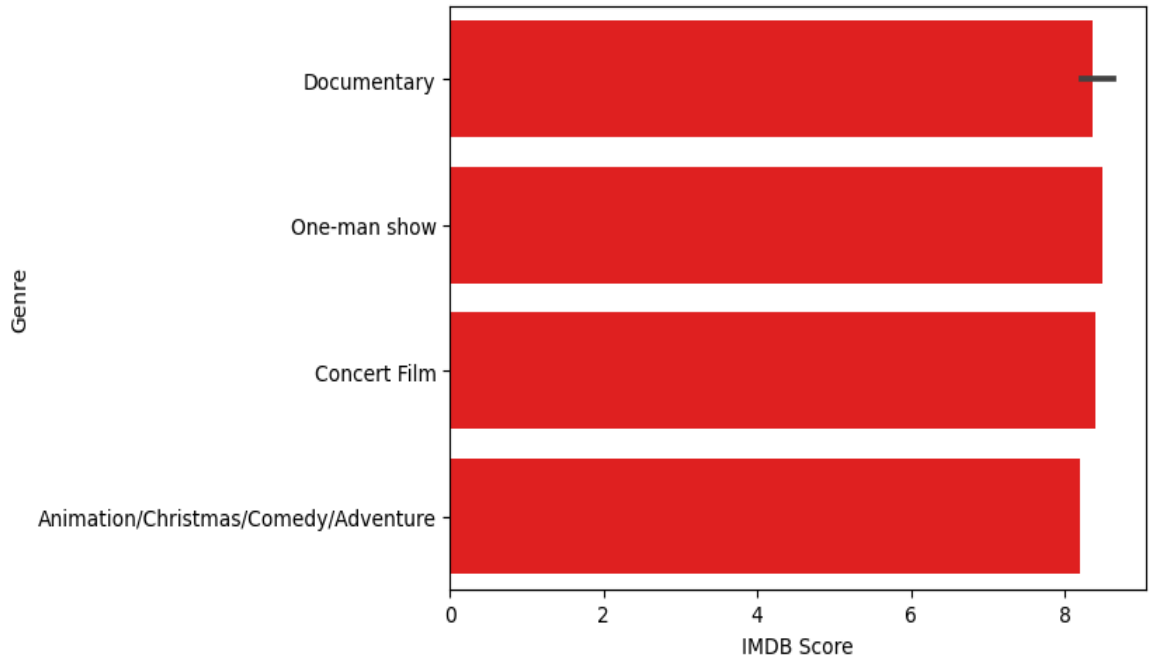



```
ds_english = ds[ds['Language'] == 'English'].sort_values
('IMDB Score', ascending=False)ds_english.head()
```

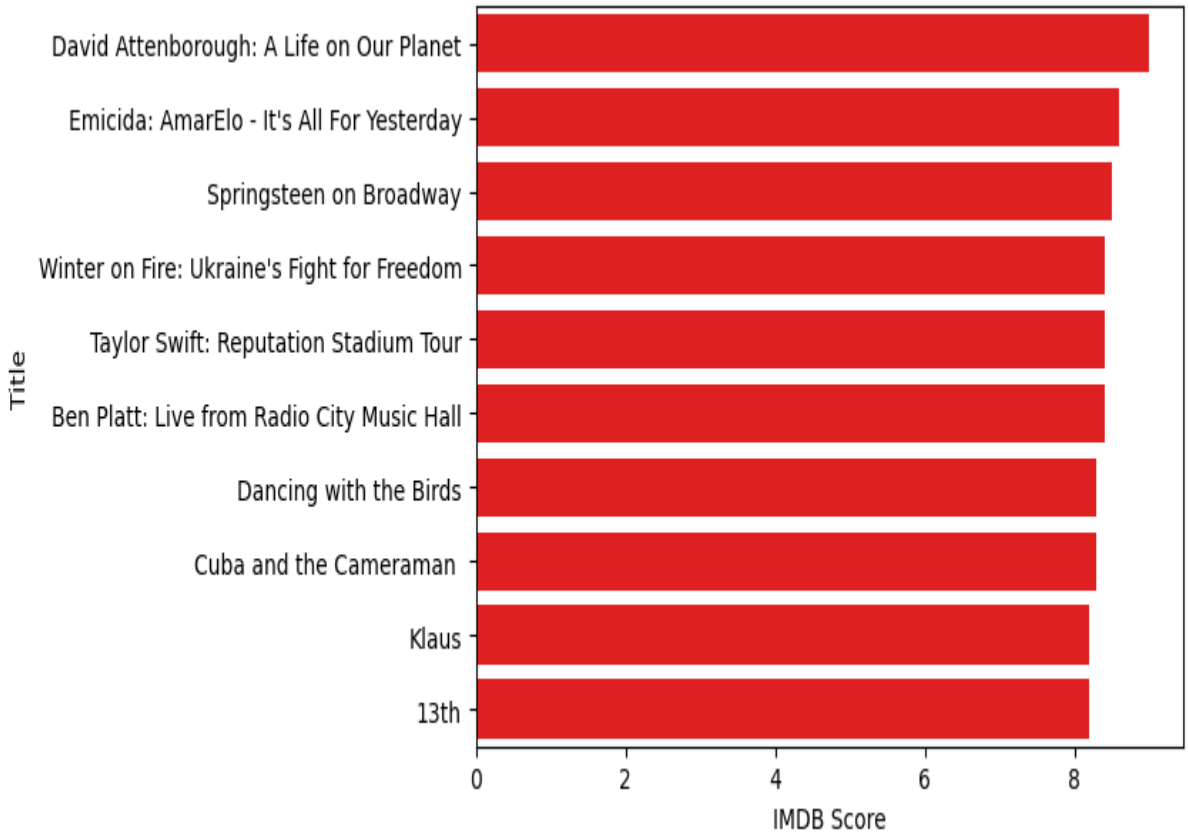
Title	Genre	Premiere	Runtime	IMDB Score	Language	
583	David Attenborough: A Life on Our Planet	Documentary	October 4, 2020	83	9.0	English
581	Springsteen on Broadway	One-man show	December 16, 2018	153	8.5	English
579	Taylor Swift: Reputation Stadium Tour	Concert Film	December 31, 2018	125	8.4	English

Title	Genre	Premiere	Runtime	IMDB Score	Language	
578	Ben Platt: Live from Radio City Music Hall	Concert Film	May 20, 2020	85	8.4	English
577	Dancing with the Birds	Documentary	October 23, 2019	51	8.3	English

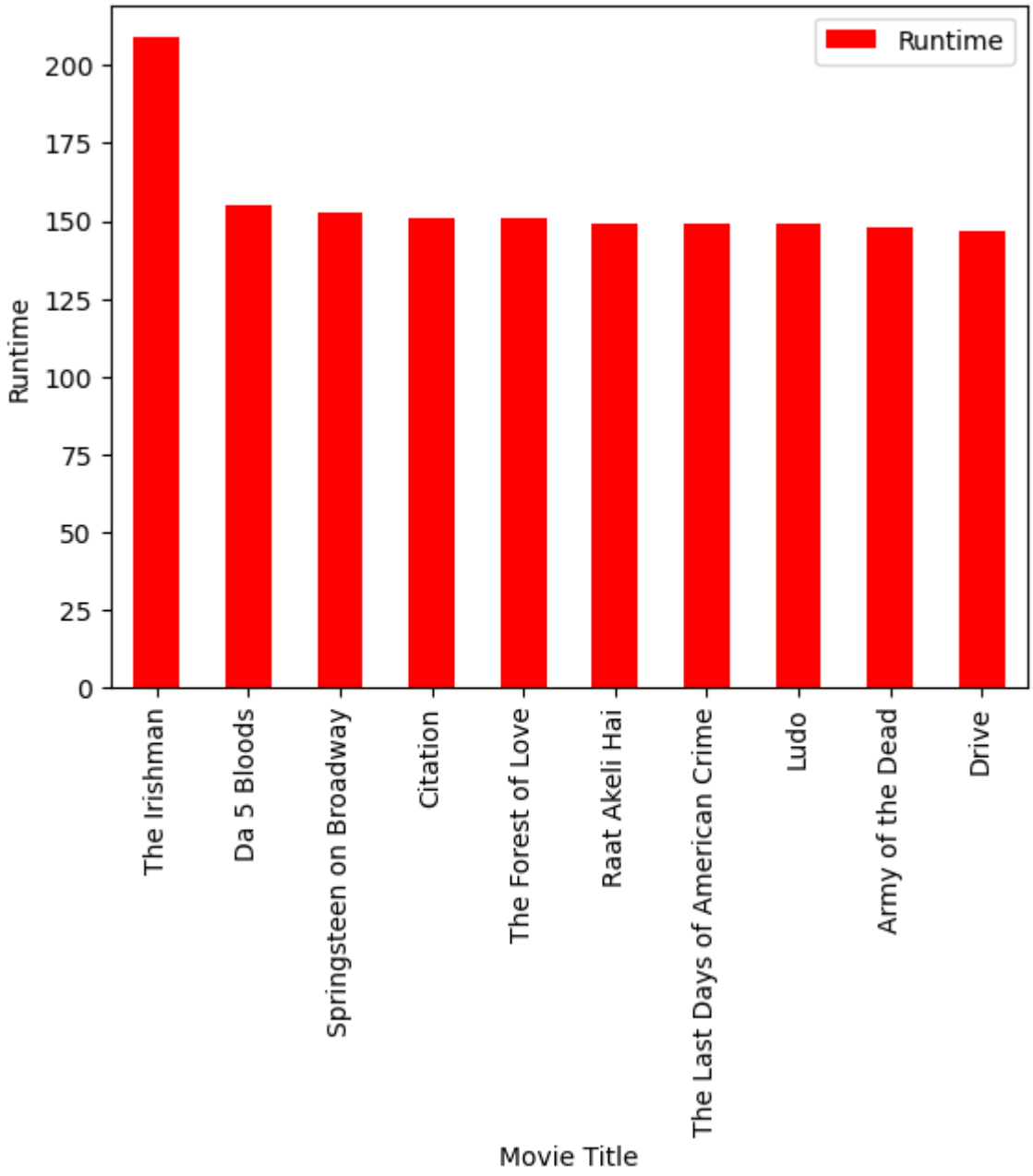
```
sns.barplot(y=ds_english['Genre'].head(10), x=ds_englis
h['IMDB Score'], color='red')plt.show(block=True)
```



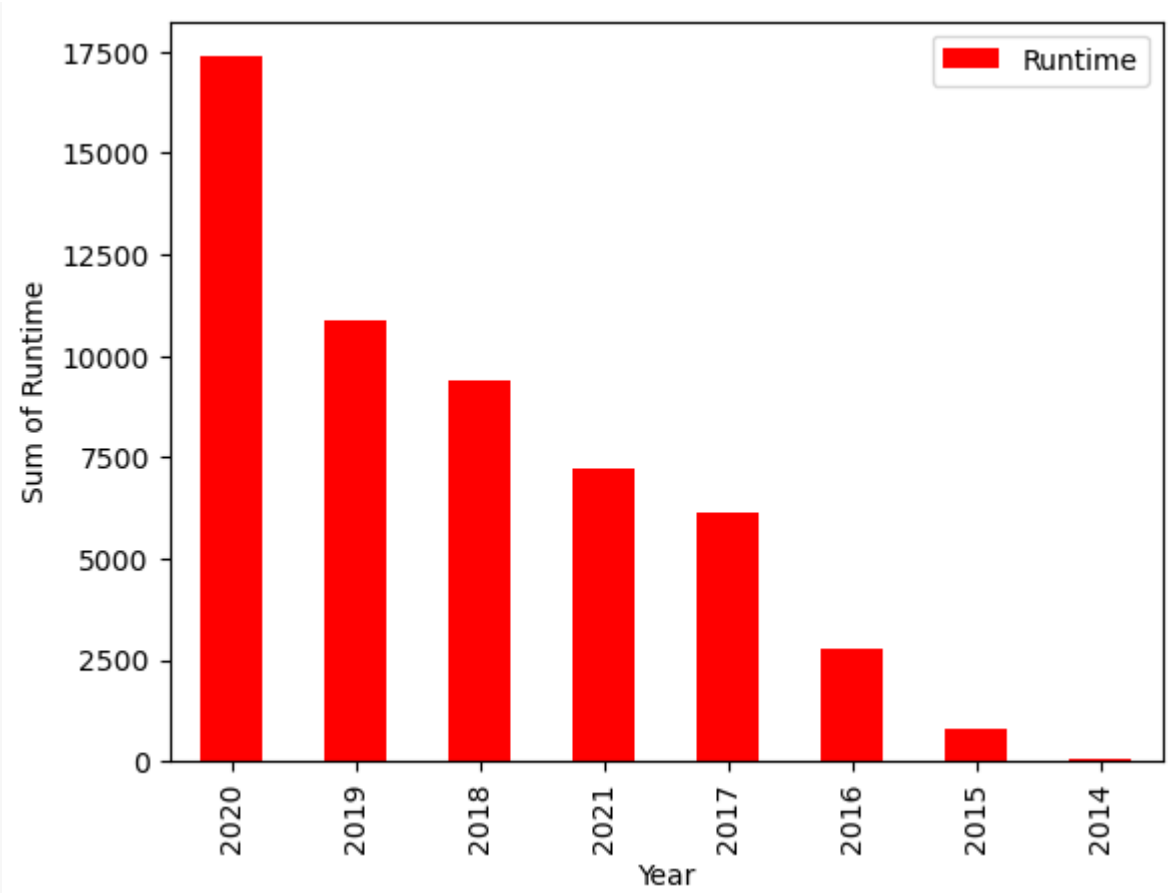

```
ds_movie = ds[['Title', 'IMDB Score']].sort_values('IMD
B Score', ascending=False).head(10)
sns.barplot(y='Title', x='IMDB Score', data=ds_movie, c
olor='red')
plt.show(block=True)
```



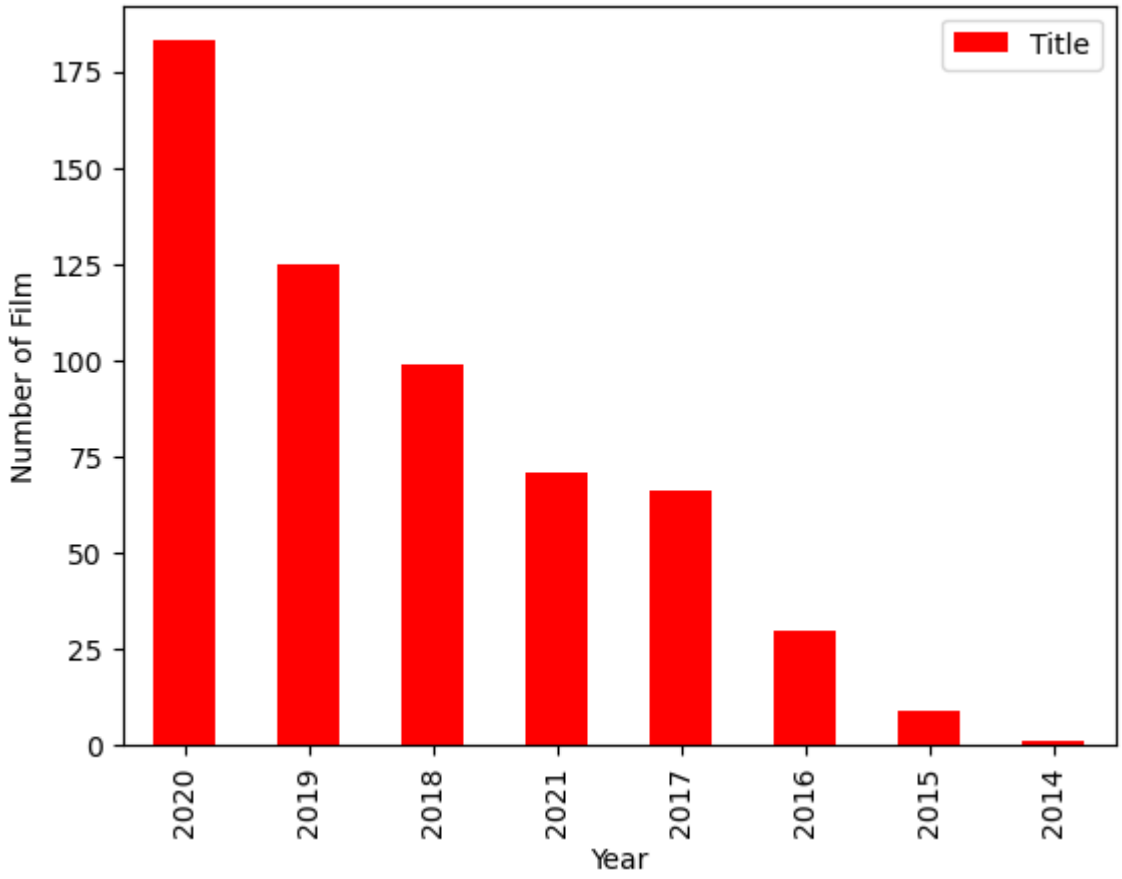

```
ds[['Title', 'Runtime']].sort_values('Runtime', ascending=
False).head(10).plot(x='Title', y='Runtime', kind='bar', co
lor='red')
plt.xlabel('Movie Title')
plt.ylabel('Runtime')
plt.show(block=True)
```




```
ds_date.groupby('Year').agg({'Runtime': 'sum'}).sort_values('Runtime', ascending=False).plot(kind='bar', color='red')
plt.xlabel('Year')
plt.ylabel('Sum of Runtime')
plt.show(block=True)
```




```
ds_date.groupby('Year').agg({'Title': 'count'}).sort_values('Title', ascending=False).plot(kind='bar', color='red')plt.xlabel('Year')
plt.ylabel('Number of Film')
plt.show(block=True)
```



Conclusion :

In conclusion, predicting IMDb scores is a complex task that involves various factors and challenges.IMDb scores are influenced by a multitude of subjective and contextual factors, and no model can perfectly capture all of these nuances.

To improve IMDb score predictions, it's crucial to consider factors such as user reviews, genre, director, actors, and release date, among others. However, it's essential to remember that IMDb scores are ultimately a reflection of audience opinions, and these opinions can change over time. Therefore, any prediction model should be periodically updated and validated against new data.