In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [121…
```
data=pd.read_csv("C:/Users/hp/Downloads/mymoviedb.csv",engine="python")
data.head()
```

Out[121…

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Lan
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	8.3	
1	2022-03-01	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	8.1	
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	
3	2021-11-24	Encanto	The tale of an extraordinary family, the Madri...	2402.201	5076	7.7	
4	2021-12-22	The King's Man	As a collection of history's worst tyrants and...	1895.511	1793	7.0	

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9837 entries, 0 to 9836
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Release_Date       9837 non-null   object 
 1   Title              9828 non-null   object 
 2   Overview           9828 non-null   object 
 3   Popularity         9827 non-null   float64
 4   Vote_Count         9827 non-null   object 
 5   Vote_Average       9827 non-null   object 
 6   Original_Language  9827 non-null   object 
 7   Genre              9826 non-null   object 
 8   Poster_Url         9826 non-null   object 
dtypes: float64(1), object(8)
memory usage: 691.8+ KB
```

Checking Null Entries

In [5]:
```
print(data.isnull().sum())
```

```
Release_Date          0
Title                 9
Overview              9
Popularity           10
Vote_Count           10
Vote_Average         10
Original_Language    10
Genre                11
Poster_Url           11
dtype: int64
```

Checking NaN Entries

In [23]:
```
print(data.isna().sum())
```

```
Release_Date          0
Title                 9
Overview              9
Popularity           10
Vote_Count           10
Vote_Average         10
Original_Language    10
Genre                11
Poster_Url           11
dtype: int64
```

Dropping All the Null Entries

In [38]:
```
data.dropna(inplace=True)
print(data.isna().sum())
```

```
Release_Date         0
Title                0
Overview             0
Popularity           0
Vote_Count           0
Vote_Average         0
Original_Language    0
Genre                0
Poster_Url           0
dtype: int64
```

In [43]:
```
data[1104:1120]
```

Out[43]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Or
1104	2012-08-21	Batman: The Dark Knight Returns, Part 1	Batman has not been seen for ten years. A new ...	61.340	1194	7.8	
1116	2019-10-09	Zombieland: Double Tap	Columbus, Tallahassee, Wichita, and Little Roc...	61.286	4369	7.0	
1117	1976-11-21	Rocky	When world heavyweight boxing champion, Apollo...	61.256	5969	7.8	
1118	2017-04-12	Gifted	Frank, a single man raising his child prodigy ...	61.234	4285	8.1	
1119	2019-01-03	Escape Room	Six strangers find themselves in circumstances...	61.177	3636	6.5	
1120	2021-02-25	A un paso de mí	Tatiana is a journalist with a routine life in...	61.151	48	6.9	
1121	2011-07-27	Colombiana	After witnessing her parents' murder as a chil...	61.116	2073	6.6	
1122	2021-11-19	Boiling Point	A head chef balances multiple personal and pro...	61.107	82	7.3	
1123	1997-11-20	Anastasia	This animated adventure spins a more optimisti...	61.104	4533	7.6	
1124	2021-08-04	Insensate	A woman tries to help her twin sister, with wh...	61.028	65	6.8	
1125	2020-10-28	The Craft: Legacy	An eclectic foursome of	60.977	521	6.4	

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Or
			aspiring teenage witch...				
1126	2011-02-16	Big Mommas: Like Father, Like Son	FBI agent Malcolm Turner and his 17-year-old s...	60.962	981	5.6	
1127	2017-07-21	Descendants 2	When the pressure to be royal becomes too much...	60.917	1329	7.4	
1128	2010-11-12	Death Race 2	In the world's most dangerous prison, a new ga...	60.909	866	5.8	
1129	2015-08-11	Straight Outta Compton	In 1987, five young men, using brutally honest...	60.906	3123	7.8	
1130	2021-09-10	Prey	A hiking trip into the wild turns into a despe...	60.901	215	4.6	

Dropping all the Useless Columns

In [44]:
```
dropcols=['Overview','Poster_Url']
data.drop(dropcols,axis=1,inplace=True)
```

In [48]:
```
data.head()
```

Out[48]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	G
0	2021-12-15	Spider-Man: No Way Home	5083.954	8940	8.3	en	Ac Adven Sci Fi
1	2022-03-01	The Batman	3827.658	1151	8.1	en	Cr Mys Th
2	2022-02-25	No Exit	2618.087	122	6.3	en	Th
3	2021-11-24	Encanto	2402.201	5076	7.7	en	Anima Com Fa Far
4	2021-12-22	The King's Man	1895.511	1793	7.0	en	Ac Adven Thr

1. changing the format of Release_Date to Datetime # 2. Changing Votecount to integer # 3. Changing VoteAverage to Float

In [53]:
```
data['Release_Date']=pd.to_datetime(data["Release_Date"],errors='coerce')
data['Vote_Count']=data['Vote_Count'].astype('int')
data['Vote_Average']=data['Vote_Average'].astype('float')
data['Genre']=data['Genre'].astype("object")
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9826 entries, 0 to 9836
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype         
---  ------             --------------  -----         
 0   Release_Date       9826 non-null   datetime64[ns]
 1   Title              9826 non-null   object        
 2   Popularity         9826 non-null   float64       
 3   Vote_Count         9826 non-null   int64         
 4   Vote_Average       9826 non-null   float64       
 5   Original_Language  9826 non-null   object        
 6   Genre              9826 non-null   object        
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 872.2+ KB
```

Function to add a Column Movie Label ACC to VoteAverage

In [57]:
```
def addcol(Vote_Average):
    if Vote_Average>8:
        return "Hit"
    elif Vote_Average<=8 and Vote_Average>7:
```

```
        return "Good"
    elif Vote_Average<=7 and Vote_Average>6:
        return "Average"
    else:
        return "Flop"
```

In [67]:
```
data["Movie_Label"]=data['Vote_Average'].apply(addcol)
data.head()
```

Out[67]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	G
0	2021-12-15	Spider-Man: No Way Home	5083.954	8940	8.3	en	Ac Adven Sci Fi
1	2022-03-01	The Batman	3827.658	1151	8.1	en	Cr Mys Th
2	2022-02-25	No Exit	2618.087	122	6.3	en	Th
3	2021-11-24	Encanto	2402.201	5076	7.7	en	Anima Com Fa Far
4	2021-12-22	The King's Man	1895.511	1793	7.0	en	Ac Adven Thr

Changing the year-month-date to only Year for ease of Analysis

In [ ]:
```
data["Release_Date"]=data["Release_Date"].dt.year
```

In [76]:
```
data['Genre']=data['Genre'].str.split(',')
data=data.explode('Genre').reset_index(drop=True)
```

Converting all the Genres to Different Rows for Each movie

In [77]:
```
data.head()
```

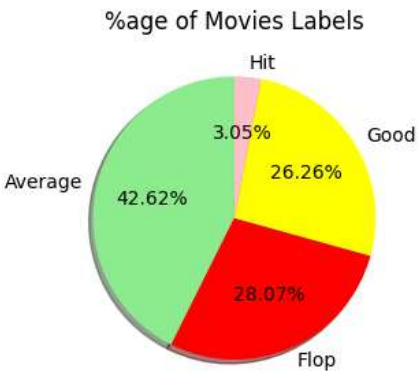
Out[77]:

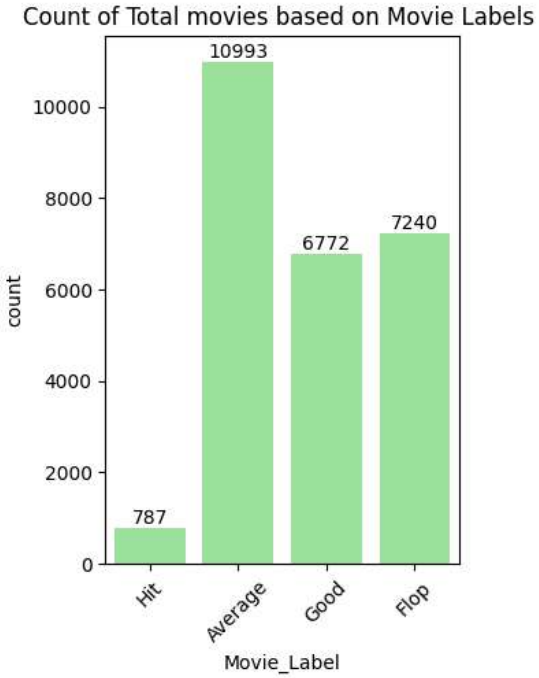
	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Ge
0	2021	Spider-Man: No Way Home	5083.954	8940	8.3	en	Act
1	2021	Spider-Man: No Way Home	5083.954	8940	8.3	en	Advent
2	2021	Spider-Man: No Way Home	5083.954	8940	8.3	en	Scie Fict
3	2022	The Batman	3827.658	1151	8.1	en	Cr
4	2022	The Batman	3827.658	1151	8.1	en	Mys

Data Visualisation

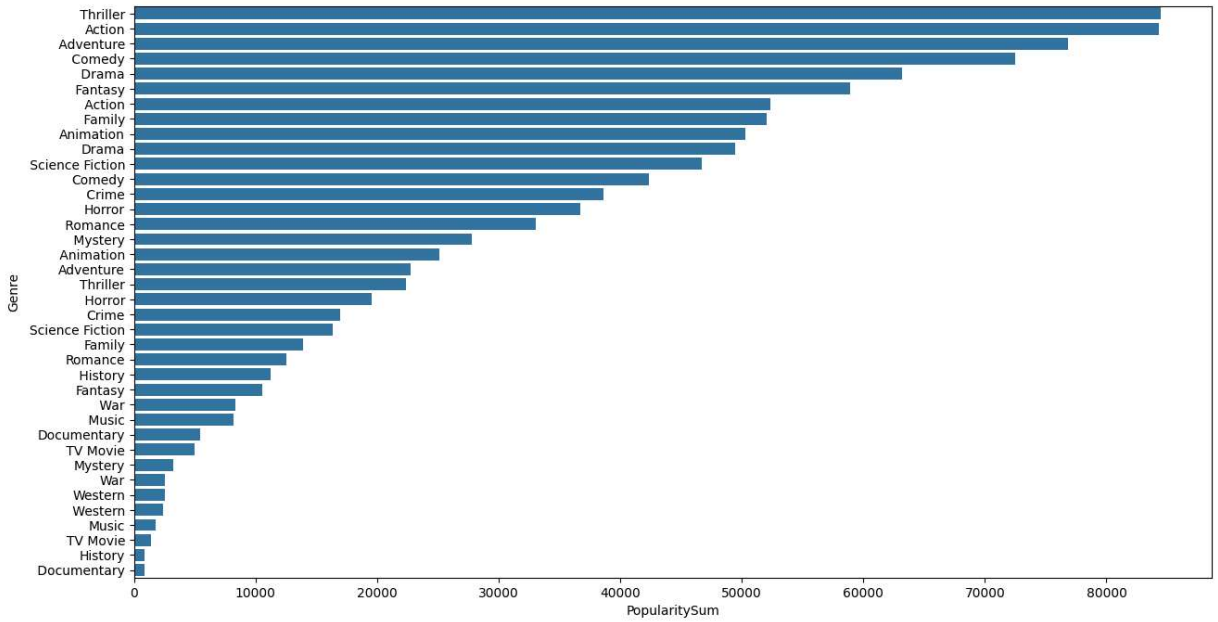
In [90]:
```
plt.figure(figsize=(10,5))
gp1=data.groupby('Movie_Label')['Movie_Label'].count()
plt.subplot(1,2,1)
plt.subplots_adjust(wspace=1)
print(gp1)
plt.pie(gp1.values,labels=gp1.index,autopct='%1.2f%%',startangle=90,colors=['lightg
plt.title('%age of Movies Labels')
plt.subplot(1,2,2)
color=['#90EE90','#FF6B6B','#FFD700']
ax=sns.countplot(data=data,x='Movie_Label',color='lightgreen')
plt.title('Count of Total movies based on Movie Labels ')
ax.bar_label(ax.containers[0])
plt.xticks(rotation=45)
plt.show()
```

```
Movie_Label
Average    10993
Flop        7240
Good        6772
Hit          787
Name: Movie_Label, dtype: int64
```



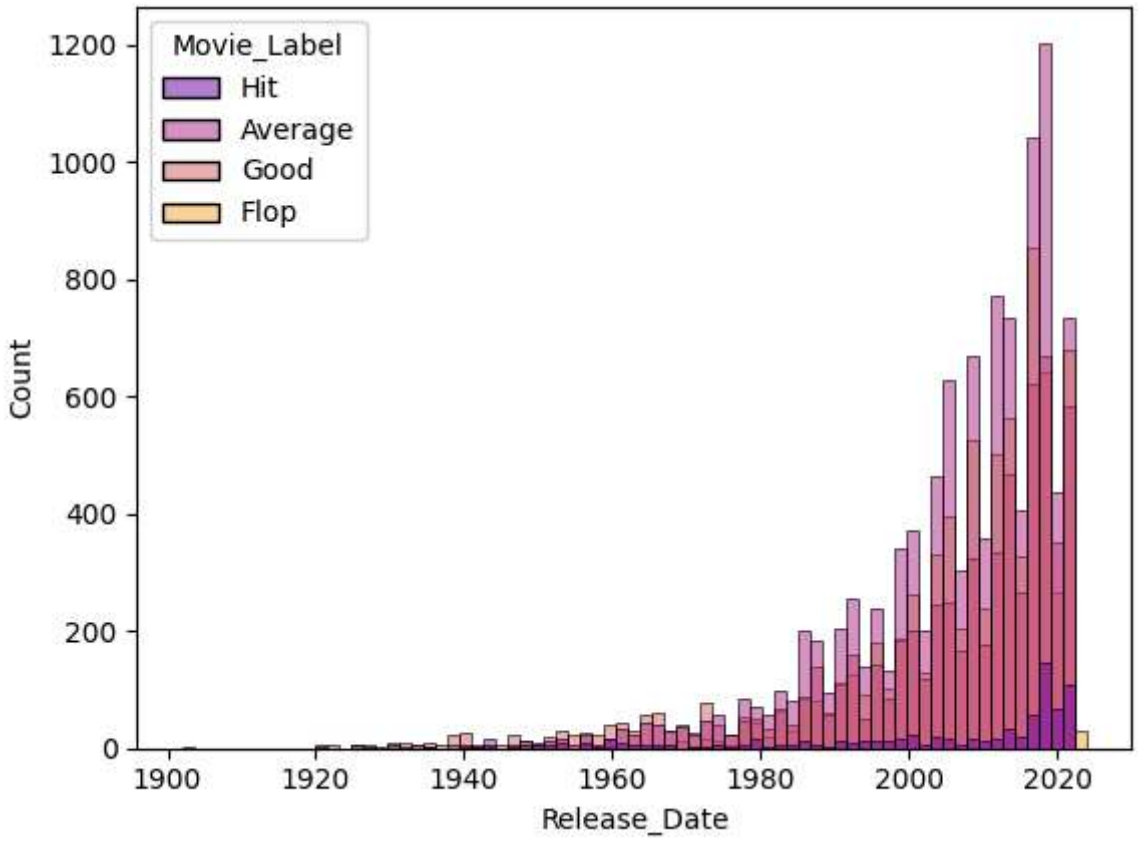


In [97]:
```
plt.figure(figsize=(15,8))
gp=data.groupby('Genre',as_index=False)['Popularity'].sum().sort_values(by="Popular
gp.rename(columns={'Popularity':'PopularitySum'},inplace=True)#from chatgpt
sns.barplot(data=gp,x='PopularitySum',y='Genre',orient='h',color=None)
plt.show()
```



In [120…
```
sns.histplot(data=data,x='Release_Date',bins=75,hue='Movie_Label',palette='plasma')
```

Out[120…
```
<Axes: xlabel='Release_Date', ylabel='Count'>
```



ANALYSED SUMMARY

Here We can See That the Percentage of Hit Movies are Very less as comapred to AVerage and Flop Movies # Also the Thriller Movies and the Action Movies are Having alomost same popularity # Movies having Genres like War,Western,TvMovies,Doucumentaries are least liked by the people # Movies having Genres like Crime,Comedy,Romance,Science fiction Got Average Popularity # From the Histogram it can be Conclude that most of the Hit movies are Telecasted between years 2018 to 2020 but during these years there is Good # Proporion of Flop Movies