netflix-movie-data-analysis-1

August 28, 2025

1 Netflix Movie Data Analysis

Import Libraries

[93]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the Dataset

[94]:
```
df = pd.read_csv('mymoviedb.csv', lineterminator='\n')
```

[95]:
```
df.head()
```

[95]:
```
  Release_Date                    Title  \
0   2021-12-15  Spider-Man: No Way Home
1   2022-03-01               The Batman
2   2022-02-25                  No Exit
3   2021-11-24                  Encanto
4   2021-12-22           The King's Man

                                            Overview  Popularity  Vote_Count  \
0  Peter Parker is unmasked and no longer able to…    5083.954        8940
1  In his second year of fighting crime, Batman u…    3827.658        1151
2  Stranded at a rest stop in the mountains durin…    2618.087         122
3  The tale of an extraordinary family, the Madri…    2402.201        5076
4  As a collection of history's worst tyrants and…    1895.511        1793

   Vote_Average Original_Language                               Genre  \
0           8.3                en  Action, Adventure, Science Fiction
1           8.1                en            Crime, Mystery, Thriller
2           6.3                en                            Thriller
3           7.7                en  Animation, Comedy, Family, Fantasy
4           7.0                en    Action, Adventure, Thriller, War

                                         Poster_Url
0  https://image.tmdb.org/t/p/original/1g0dhYtq4i…
```

```
1  https://image.tmdb.org/t/p/original/74xTEgt7R3…
2  https://image.tmdb.org/t/p/original/vDHsLnOWKl…
3  https://image.tmdb.org/t/p/original/4j0PNHkMr5…
4  https://image.tmdb.org/t/p/original/aq4Pwv5Xeu…
```

[96]:

```
df.tail()
```

[96]:

```
     Release_Date                                 Title  \
9822   1973-10-15                              Badlands
9823   2020-10-01                      Violent Delights
9824   2016-05-06                          The Offering
9825   2021-03-31  The United States vs. Billie Holiday
9826   1984-09-23                               Threads

                                               Overview  Popularity  \
9822  A dramatization of the Starkweather-Fugate kil…      13.357
9823  A female vampire falls in love with a man she …      13.356
9824  When young and successful reporter Jamie finds…      13.355
9825  Billie Holiday spent much of her career being …      13.354
9826  Documentary style account of a nuclear holocau…      13.354

      Vote_Count  Vote_Average Original_Language                        Genre  \
9822         896           7.6                en                 Drama, Crime
9823           8           3.5                es                       Horror
9824          94           5.0                en    Mystery, Thriller, Horror
9825         152           6.7                en        Music, Drama, History
9826         186           7.8                en  War, Drama, Science Fiction

                                             Poster_Url
9822  https://image.tmdb.org/t/p/original/z81rBzHNgi…
9823  https://image.tmdb.org/t/p/original/4b6HY7rud6…
9824  https://image.tmdb.org/t/p/original/h4uMM1wOhz…
9825  https://image.tmdb.org/t/p/original/vEzkxuE2sJ…
9826  https://image.tmdb.org/t/p/original/lBhU4U9Eeh…
```

[97]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9827 entries, 0 to 9826
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Release_Date       9827 non-null   object
 1   Title              9827 non-null   object
 2   Overview           9827 non-null   object
 3   Popularity         9827 non-null   float64
 4   Vote_Count         9827 non-null   int64
 5   Vote_Average       9827 non-null   float64
 6   Original_Language  9827 non-null   object
 7   Genre              9827 non-null   object
 8   Poster_Url         9827 non-null   object
dtypes: float64(2), int64(1), object(6)
memory usage: 691.1+ KB
```

```
[98]: df['Genre'].head()
```

```
[98]: 0    Action, Adventure, Science Fiction
      1              Crime, Mystery, Thriller
      2                              Thriller
      3    Animation, Comedy, Family, Fantasy
      4      Action, Adventure, Thriller, War
      Name: Genre, dtype: object
```

```
[99]: df.duplicated()
```

```
[99]: 0       False
      1       False
      2       False
      3       False
      4       False
              …
      9822    False
      9823    False
      9824    False
      9825    False
      9826    False
      Length: 9827, dtype: bool
```

```
[100]: df.duplicated().sum()
```

```
[100]: np.int64(0)
```

```
[101]: df.describe()
```

```
[101]:         Popularity    Vote_Count  Vote_Average
       count  9827.000000   9827.000000   9827.000000
       mean     40.326088   1392.805536      6.439534
       std     108.873998   2611.206907      1.129759
       min      13.354000      0.000000      0.000000
       25%      16.128500    146.000000      5.900000
       50%      21.199000    444.000000      6.500000
       75%      35.191500   1376.000000      7.100000
       max    5083.954000  31077.000000     10.000000
```

Exploration Summary

- we have a dataframe consisting of 9827 rows and 9 columns.

- our dataset looks a bit tidy with no NaNs nor duplicated values.

- Release_Date column needs to be casted into date time and to extract only the year value

- Overview, Original_Languege and Poster-Url wouldn't be so useful during analysis,so we will drop them.

- there is noticable outliers in Popularity column

- Vote_Average bettter be categorised for proper analysis.

- Genre column has comma saperated values and white spaces that needs to be handled and casted into category

Data Cleaning

[102]:
```
df.head()
```

[102]:
```
  Release_Date                    Title  \
0   2021-12-15  Spider-Man: No Way Home   
1   2022-03-01               The Batman   
2   2022-02-25                  No Exit   
3   2021-11-24                  Encanto   
4   2021-12-22           The King's Man   

                                            Overview  Popularity  Vote_Count  \
0  Peter Parker is unmasked and no longer able to…    5083.954        8940   
1  In his second year of fighting crime, Batman u…    3827.658        1151   
2  Stranded at a rest stop in the mountains durin…    2618.087         122   
3  The tale of an extraordinary family, the Madri…    2402.201        5076   
4  As a collection of history's worst tyrants and…    1895.511        1793   

   Vote_Average Original_Language                               Genre  \
0           8.3                en  Action, Adventure, Science Fiction   
1           8.1                en            Crime, Mystery, Thriller   
2           6.3                en                            Thriller   
3           7.7                en  Animation, Comedy, Family, Fantasy   
4           7.0                en    Action, Adventure, Thriller, War   

                                          Poster_Url  
0  https://image.tmdb.org/t/p/original/1g0dhYtq4i…  
1  https://image.tmdb.org/t/p/original/74xTEgt7R3…  
2  https://image.tmdb.org/t/p/original/vDHsLnOWKl…  
3  https://image.tmdb.org/t/p/original/4j0PNHkMr5…  
4  https://image.tmdb.org/t/p/original/aq4Pwv5Xeu…  
```

[103]:
```
df['Release_Date']=pd.to_datetime(df['Release_Date'])
print(df['Release_Date'])
```

```
0       2021-12-15
1       2022-03-01
2       2022-02-25
3       2021-11-24
4       2021-12-22
           …
9822    1973-10-15
9823    2020-10-01
9824    2016-05-06
9825    2021-03-31
9826    1984-09-23
Name: Release_Date, Length: 9827, dtype: datetime64[ns]
```

[104]:
```
print(df['Release_Date'].dtypes)
```

```
datetime64[ns]
```

[105]:
```
df['Release_Date']=df['Release_Date'].dt.year
df['Release_Date'].dtypes
```

[105]:
```
dtype('int32')
```

[106]:
```
df.head()
```

[106]:
```
   Release_Date                    Title  \
0          2021  Spider-Man: No Way Home
1          2022               The Batman
2          2022                  No Exit
3          2021                  Encanto
4          2021           The King's Man

                                            Overview  Popularity  Vote_Count  \
0  Peter Parker is unmasked and no longer able to…    5083.954        8940
1  In his second year of fighting crime, Batman u…    3827.658        1151
2  Stranded at a rest stop in the mountains durin…    2618.087         122
3  The tale of an extraordinary family, the Madri…    2402.201        5076
4  As a collection of history's worst tyrants and…    1895.511        1793

   Vote_Average Original_Language                               Genre  \
0           8.3                en  Action, Adventure, Science Fiction
1           8.1                en            Crime, Mystery, Thriller
2           6.3                en                            Thriller
3           7.7                en  Animation, Comedy, Family, Fantasy
4           7.0                en    Action, Adventure, Thriller, War

                                          Poster_Url
0  https://image.tmdb.org/t/p/original/1g0dhYtq4i…
1  https://image.tmdb.org/t/p/original/74xTEgt7R3…
```

```
2  https://image.tmdb.org/t/p/original/vDHsLnOWKl…
3  https://image.tmdb.org/t/p/original/4j0PNHkMr5…
4  https://image.tmdb.org/t/p/original/aq4Pwv5Xeu…
```

Dropping columns

```
[107]: cols=['Overview','Original_Language','Poster_Url']
```

```
[108]: df.drop(cols, axis=1, inplace=True)
       df.columns
```

```
[108]: Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
              'Genre'],
             dtype='object')
```

```
[109]: df.head()
```

```
[109]:    Release_Date                    Title  Popularity  Vote_Count  \
       0          2021  Spider-Man: No Way Home    5083.954        8940
       1          2022               The Batman    3827.658        1151
       2          2022                  No Exit    2618.087         122
       3          2021                  Encanto    2402.201        5076
       4          2021           The King's Man    1895.511        1793

          Vote_Average                               Genre
       0           8.3  Action, Adventure, Science Fiction
       1           8.1            Crime, Mystery, Thriller
       2           6.3                            Thriller
       3           7.7  Animation, Comedy, Family, Fantasy
       4           7.0    Action, Adventure, Thriller, War
```

categorizing Vote_Average column

We would cut the Vote_Average values and make 4 categories: popular average below_avg not_popular to describe it more using catigorize_col() function provided above.

```
[110]: def categorize_col(df,col,labels):

        edges=[df[col].describe()['min'],
               df[col].describe()['25%'],
               df[col].describe()['50%'],
               df[col].describe()['75%'],
               df[col].describe()['max']]

        df[col]=pd.cut(df[col],edges,labels=labels,duplicates ='drop')
        return df
```

```
[111]: labels=['non_popular','below_avg','average','popular']
```

[112]:
```
categorize_col(df,'Vote_Average',labels)
df['Vote_Average'].unique()
```

[112]:
```
['popular', 'below_avg', 'average', 'non_popular', NaN]
Categories (4, object): ['non_popular' < 'below_avg' < 'average' < 'popular']
```

[113]:
```
df.head()
```

[113]:
```
   Release_Date                    Title  Popularity  Vote_Count Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940      popular
1          2022               The Batman    3827.658        1151      popular
2          2022                  No Exit    2618.087         122    below_avg
3          2021                  Encanto    2402.201        5076      popular
4          2021           The King's Man    1895.511        1793      average

                              Genre
0  Action, Adventure, Science Fiction
1            Crime, Mystery, Thriller
2                            Thriller
3  Animation, Comedy, Family, Fantasy
4    Action, Adventure, Thriller, War
```

[114]:
```
df['Vote_Average'].value_counts()
```

[114]:
```
Vote_Average
non_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

[115]:
```
df.dropna(inplace = True)

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

[115]:
```
Release_Date    0
Title           0
Popularity      0
Vote_Count      0
Vote_Average    0
Genre           0
dtype: int64
```

[116]:
```
df.head()
```

[116]:
```
   Release_Date                    Title  Popularity  Vote_Count Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940      popular
```

```
1          2022                The Batman   3827.658        1151      popular
2          2022                   No Exit   2618.087         122    below_avg
3          2021                   Encanto   2402.201        5076      popular
4          2021              The King's Man  1895.511        1793      average

                              Genre
0  Action, Adventure, Science Fiction
1            Crime, Mystery, Thriller
2                            Thriller
3  Animation, Comedy, Family, Fantasy
4    Action, Adventure, Thriller, War
```

we'd split genres into a list and then explode our dataframe to have only one genre per row for ezch movie

[117]:

```
df['Genre'] = df['Genre'].str.split(', ')

df=df.explode('Genre').reset_index(drop=True)

df.head()
```

[117]:

```
   Release_Date                    Title  Popularity  Vote_Count Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940      popular
1          2021  Spider-Man: No Way Home    5083.954        8940      popular
2          2021  Spider-Man: No Way Home    5083.954        8940      popular
3          2022               The Batman    3827.658        1151      popular
4          2022               The Batman    3827.658        1151      popular

             Genre
0           Action
1        Adventure
2  Science Fiction
3            Crime
4          Mystery
```

[118]:

```
# casting column into category
df['Genre'] = df['Genre'].astype('category')

 # confirming changes
df['Genre'].dtype
```

[118]:

```
CategoricalDtype(categories=['Action', 'Adventure', 'Animation', 'Comedy',
'Crime',
                  'Documentary', 'Drama', 'Family', 'Fantasy', 'History',
                  'Horror', 'Music', 'Mystery', 'Romance', 'Science Fiction',
                  'TV Movie', 'Thriller', 'War', 'Western'],
, ordered=False, categories_dtype=object)
```

[119]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25552 entries, 0 to 25551
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Release_Date  25552 non-null  int32
 1   Title         25552 non-null  object
 2   Popularity    25552 non-null  float64
 3   Vote_Count    25552 non-null  int64
 4   Vote_Average  25552 non-null  category
 5   Genre         25552 non-null  category
dtypes: category(2), float64(1), int32(1), int64(1), object(1)
memory usage: 749.6+ KB
```

[121]:

```
df.nunique()
```

[121]:

```
Release_Date     100
Title           9415
Popularity      8088
Vote_Count      3265
Vote_Average       4
Genre             19
dtype: int64
```

[122]:

```
df.head()
```

[122]:

```
   Release_Date                    Title  Popularity  Vote_Count Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940      popular
1          2021  Spider-Man: No Way Home    5083.954        8940      popular
2          2021  Spider-Man: No Way Home    5083.954        8940      popular
3          2022               The Batman    3827.658        1151      popular
4          2022               The Batman    3827.658        1151      popular

             Genre
0           Action
1        Adventure
2  Science Fiction
3            Crime
4          Mystery
```

2 Data Visualization

[ ]:

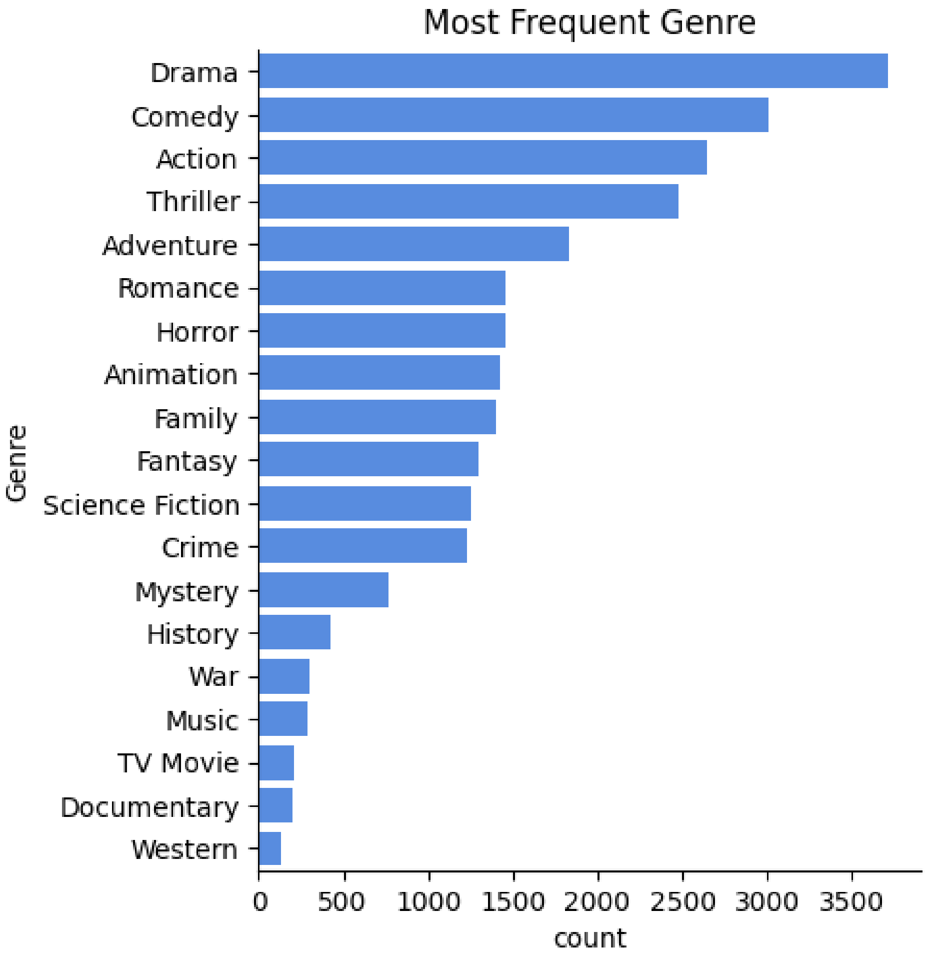
```
sns.set_style('whitegrid')
```

3 What is the most frequent genre in the dataset?

```
[123]: df['Genre'].describe()
```

```
[123]: count      25552
       unique        19
       top        Drama
       freq        3715
       Name: Genre, dtype: object
```

```
[124]: sns.catplot(y='Genre', kind='count',data=df, order=df['Genre'].value_counts().
         ↪index, color ='#4287f5')
       plt.title('Most Frequent Genre')
       plt.show()
```



4 What genres has highest votes

[125]:

```
df.head()
```

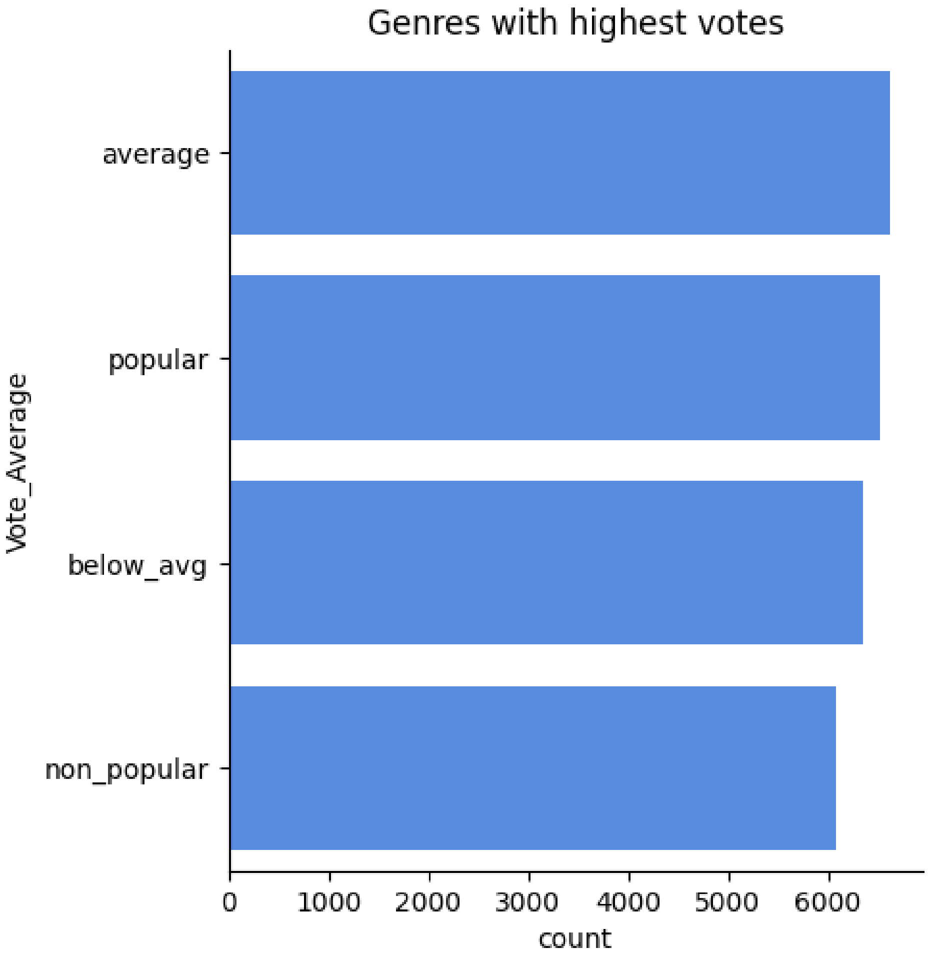
[125]:

```
   Release_Date                    Title  Popularity  Vote_Count Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940      popular
1          2021  Spider-Man: No Way Home    5083.954        8940      popular
2          2021  Spider-Man: No Way Home    5083.954        8940      popular
3          2022               The Batman    3827.658        1151      popular
4          2022               The Batman    3827.658        1151      popular

             Genre
0           Action
1        Adventure
2  Science Fiction
3            Crime
4          Mystery
```

[127]:

```
sns.catplot(y='Vote_Average', kind='count',data=df, order=df['Vote_Average'].
 ↪value_counts().index, color ='#4287f5')
plt.title('Genres with highest votes')
plt.show()
```



5 What movie got the highest popularity ? what's it genre ?

```
[129]: df.head(4)
```

```
[129]:    Release_Date                    Title  Popularity  Vote_Count Vote_Average  \
       0          2021  Spider-Man: No Way Home    5083.954        8940      popular
       1          2021  Spider-Man: No Way Home    5083.954        8940      popular
       2          2021  Spider-Man: No Way Home    5083.954        8940      popular
       3          2022               The Batman    3827.658        1151      popular

                    Genre
       0           Action
       1        Adventure
       2  Science Fiction
       3            Crime
```

```
[132]: df[df['Popularity']==df['Popularity'].max()]
```

```
[132]:    Release_Date                    Title  Popularity  Vote_Count Vote_Average  \
       0          2021  Spider-Man: No Way Home    5083.954        8940      popular
       1          2021  Spider-Man: No Way Home    5083.954        8940      popular
       2          2021  Spider-Man: No Way Home    5083.954        8940      popular

                    Genre
       0           Action
       1        Adventure
       2  Science Fiction
```

6 What movie got the lowest popularity? what's its genre?

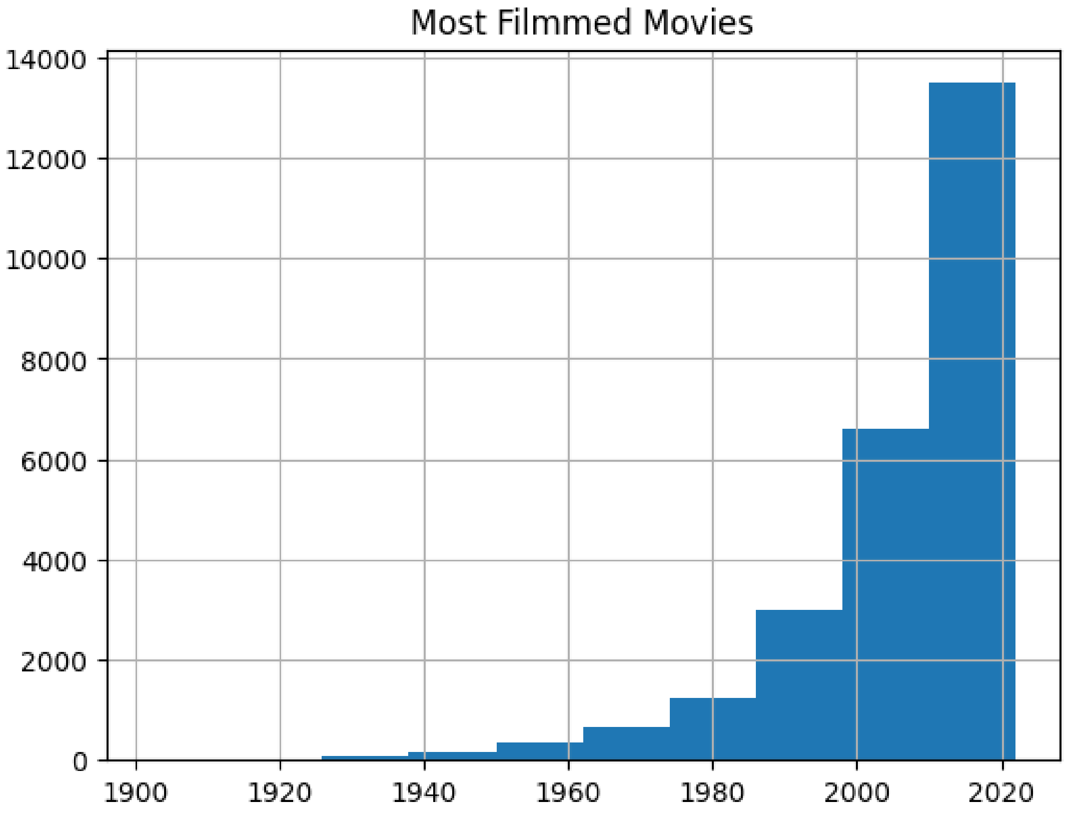
```
[133]: df[df['Popularity']==df['Popularity'].min()]
```

```
[133]:        Release_Date                                    Title  Popularity  \
       25546          2021  The United States vs. Billie Holiday      13.354
       25547          2021  The United States vs. Billie Holiday      13.354
       25548          2021  The United States vs. Billie Holiday      13.354
       25549          1984                                 Threads      13.354
       25550          1984                                 Threads      13.354
       25551          1984                                 Threads      13.354

              Vote_Count Vote_Average            Genre
       25546         152      average            Music
       25547         152      average            Drama
       25548         152      average          History
       25549         186      popular              War
       25550         186      popular            Drama
       25551         186      popular  Science Fiction
```

7 Which year has the most filmmed movies

```
[138]: df['Release_Date'].hist()
       plt.title('Most Filmmed Movies')
       plt.show()
```



Q1: What is the most frequent genre in the dataset?

Drama genre is the most frequent genre in our dataset and has appeared more than14% of the times among 19 other genres.

Q2: What genres has highest votes ?

we have 25.5% of our dataset with popular vote (6520 rows). Drama again gets thehighest popularity among fans by being having more than 18.5% of movies popularities.

Q3: What movie got the highest popularity ? what's its Action , genre ?

Spider-Man: No Way Home has the highest popularity rate in our dataset and it hasgenres of Adventure and Sience Fiction .

Q4: What movie got the lowest popularity ? what's its genre ?

The united states, thread' has the highest lowest rate in our dataset and it has genres of music, drama , 'war', 'sci-fi' and history'.

Q5: Which year has the most filmmed movies?

year -2020 has the highest filmming rate in our dataset.