Overload Ware Labs AI

Data Analyst Intern

Netflix Movies Dataset Analysis Report

Name	**Shiva**
Degree	**B.tech in Electronics and communication Enginnering**
Position	**Data Analyst Intern**
Tools	**Python, Pandas, Matplotlib, Seaborn**
Contact	**shivagujria269@gmail.com**
Linkedin	**https://www.linkedin.com/in/shiva-gujria-0a8a64237/**
Github	

https://github.com/shivagujria/Netflix_Movies_Analysis_Project-owl-AI-

Abstract

This project analyzes a dataset of over 9,000 Netflix movies to uncover trends in genres, release years, ratings, and popularity. Using Python's data analysis and visualization libraries (Pandas, Matplotlib, Seaborn), the dataset was cleaned, transformed, and explored to answer specific business questions.

The findings reveal the most dominant genres, top and bottom movies by popularity, and production trends over the years, providing valuable insights for content strategy.

Introduction

Netflix, founded on **August 29, 1997** in Scotts Valley, California, by **Reed Hastings** and **Marc Randolph**, began as a DVD-by-mail rental service. In 2007, the company introduced streaming services, allowing subscribers to instantly watch movies and TV shows online. This shift, combined with international expansion starting in 2010, propelled Netflix into a global entertainment leader.

As of **2024**, Netflix operates in **190+ countries** with over **283 million paid memberships** and generated nearly **$10 billion** in Q3 revenue, with profits of $2.4 billion. The platform offers diverse content across languages and genres, leveraging **data science, AI, and machine learning** for personalized recommendations.

In this project, I analyze a dataset of 9,800+ Netflix movies to answer key business questions that can guide decision-making in content acquisition and marketing. The questions addressed are:

1. What is the most frequent genre of movies released on Netflix?
2. Which movie has the highest votes in the vote_average column?
3. What movie has the highest popularity, and what is its genre?
4. What movie has the lowest popularity, and what is its genre?
5. Which year has the most movies filmed?

Dataset Overview

- **Rows:** ~9,827

- **Columns:** Title, Genre, Release_Date, Popularity, Vote_Average, Vote_Count, Original_Language, Overview, Poster_URI

- **Source:** Kaggle – Netflix Movies Dataset

- **Initial Observations:**

- Some columns like Overview, Original_Language, and Poster_URI were removed for analysis.
- Genre contained multiple comma-separated values.
- Release_Date was converted to datetime and year extracted.
- Dataset had no critical missing values after cleaning.

Data Cleaning & Preparation

Steps taken:

1. Dropped unused columns (Overview, Original_Language, Poster_URI).
2. Converted Release_Date to datetime format, extracted year.
3. Split multi-genre entries into separate rows using .explode().
4. Verified and handled missing values where necessary.
5. Checked for outliers in Popularity and Vote_Average.

Project Goal

The goal of this project is to perform a comprehensive analysis of Netflix's movie dataset to uncover meaningful trends and patterns that can support data-driven decision-making. This involves identifying the most dominant genres, understanding the distribution of ratings, and analyzing popularity metrics to spot high-performing content. By examining release year trends, the project aims to pinpoint periods of peak production and growth.

Additionally, the analysis will explore the relationship between popularity and audience ratings to understand differences between mass appeal and critical acclaim. Through data cleaning, transformation, and visualization, the project will provide actionable insights for improving content strategy. The ultimate objective is to help streaming platforms optimize their content library, target the right audiences, and enhance viewer engagement.

Steps for Netflix Movie Dataset Analysis

1. **Import Required Libraries**
    - Load Python libraries for data analysis and visualization: **pandas, numpy, matplotlib, seaborn**.
2. **Load the Dataset**
    - Use **pd.read_csv()** to read the mymoviedb.csv file into a DataFrame.
3. **Initial Data Exploration**
    - Display the first few rows with .**head().**
    - Check dataset shape with **.shape.**
    - View column names and data types using .**info().**
    - Summarize numerical features with **.describe().**
4. **Data Cleaning**
    - Drop irrelevant columns: **Overview, Original_Language**, **Poster_URI.**
    - Handle missing values if present.
    - Convert **Release_Date** to datetime format and extract the release year.
    - Remove duplicate entries if any.
5. **Genre Processing**
    - Split multiple genres into lists using .**str.split(',').**
    - Remove extra spaces using .**str.strip().**
    - Use **.explode()** to create separate rows for each genre.
6. **Feature Engineering**
    - Categorize **Vote_Average** into rating bands (Poor, Average, Good, Excellent).
    - Create additional columns if needed (e.g., decade of release).

7. **Exploratory Data Analysis (EDA)**
    - **Q1:** Find the most frequent genre (**value_counts()).**
    - **Q2:** Find the movie with the highest **Vote_Average**.
    - **Q3:** Find the movie with the highest Popularity.
    - **Q4:** Find the movie with the lowest Popularity.
    - **Q5:** Identify the year with the most movies released.
8. **Visualization**
    - Plot genre distribution (**bar chart**).
    - Plot number of movies released per year (**line plot**).
    - Plot popularity distribution (**histogram**).
    - Plot vote average categories by genre (**countplot**).
9. **Insights & Observations**
    - Summarize key findings from the EDA.
10. **Recommendations**
    - Suggest strategies for content creation, marketing, and viewer engagement based on the analysis.

Code:

[26]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df = pd.read_csv('mymoviedb.csv' ,lineterminator ='\n')
```

[3]:
```
df.head()
```

[3]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	Poster_Url
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	8.3	en	Action, Adventure, Science Fiction	https://image.tmdb.org/t/p/original/1g0dhYtq4i...
1	2022-03-01	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	8.1	en	Crime, Mystery, Thriller	https://image.tmdb.org/t/p/original/74xTEgt7R3...
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmdb.org/t/p/original/vDHsLnOWKl...
3	2021-11-24	Encanto	The tale of an extraordinary family, the Madri...	2402.201	5076	7.7	en	Animation, Comedy, Family, Fantasy	https://image.tmdb.org/t/p/original/4j0PNHkMr5...
4	2021-12-22	The King's Man	As a collection of history's worst tyrants and...	1895.511	1793	7.0	en	Action, Adventure, Thriller, War	https://image.tmdb.org/t/p/original/aq4Pwv5Xeu...

[4]:
```
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
```

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

```
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
```

[6]:
```
df.duplicated().sum()
```

[6]:
```
np.int64(0)
```

[7]:
```
df.describe()
```

[7]:

	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

- 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 _ Language and Poster-Uri wouldn't be so useful during analysis, so we will drop them.
- there is noticable outliers in Popularity column
- Vote_Average better be categorised for proper analysis.
- Genre column has comma seperated values and white spaces that needs to be handled and casted into category. Exploration Summary.

[8]:
```
df['Release_Date']=pd.to_datetime(df['Release_Date'])

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

```
datetime64[ns]
```

[9]:
```
df['Release_Date']= df['Release_Date'].dt.year

df['Release_Date'].dtypes
```

[9]:
```
dtype('int32')
```

[10]:
```
df.head()
```

[10]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	Poster_Url
0	2021	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	8.3	en	Action, Adventure, Science Fiction	https://image.tmdb.org/t/p/original/1g0dhYtq4i...
1	2022	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	8.1	en	Crime, Mystery, Thriller	https://image.tmdb.org/t/p/original/74xTEgt7R3...
2	2022	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmdb.org/t/p/original/vDHsLnOWKl...
3	2021	Encanto	The tale of an extraordinary family, the Madri...	2402.201	5076	7.7	en	Animation, Comedy, Family, Fantasy	https://image.tmdb.org/t/p/original/4j0PNHkMr5...
4	2021	The King's Man	As a collection of history's worst tyrants and...	1895.511	1793	7.0	en	Action, Adventure, Thriller, War	https://image.tmdb.org/t/p/original/aq4Pwv5Xeu...

Dropping the columns

```
cols= ['Overview', 'Original_Language', 'Poster_Url']
df.drop(cols, axis=1, inplace=True)
df.columns
```

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

Dropping the columns

```
cols= ['Overview', 'Original_Language', 'Poster_Url']
df.drop(cols, axis=1, inplace=True)
df.columns
```

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

[12]:
```
df.head()
```

[12]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	8.3	Action, Adventure, Science Fiction
1	2022	The Batman	3827.658	1151	8.1	Crime, Mystery, Thriller
2	2022	No Exit	2618.087	122	6.3	Thriller
3	2021	Encanto	2402.201	5076	7.7	Animation, Comedy, Family, Fantasy
4	2021	The King's Man	1895.511	1793	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 categorize_col() function provided above.

[13]:
```
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
```

[14]:
```
labels = ['not_popular', 'below_avg', 'average', 'popular']

categorize_col(df, 'Vote_Average', labels)

df['Vote_Average'].unique()
```

[14]:
```
['popular', 'below_avg', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_avg' < 'average' < 'popular']
```

[15]:
```
df.head()
```

[15]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	Action, Adventure, Science Fiction
1	2022	The Batman	3827.658	1151	popular	Crime, Mystery, Thriller
2	2022	No Exit	2618.087	122	below_avg	Thriller
3	2021	Encanto	2402.201	5076	popular	Animation, Comedy, Family, Fantasy
4	2021	The King's Man	1895.511	1793	average	Action, Adventure, Thriller, War

[15]:
```
df.head()
```

[15]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	Action, Adventure, Science Fiction
1	2022	The Batman	3827.658	1151	popular	Crime, Mystery, Thriller
2	2022	No Exit	2618.087	122	below_avg	Thriller
3	2021	Encanto	2402.201	5076	popular	Animation, Comedy, Family, Fantasy
4	2021	The King's Man	1895.511	1793	average	Action, Adventure, Thriller, War

We would split genres into a list and then explode our dataframe to have only one genre per row for each movie

[16]:
```
df['Genre'] = df['Genre'].str.split(', ')

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

[16]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	Science Fiction
3	2022	The Batman	3827.658	1151	popular	Crime
4	2022	The Batman	3827.658	1151	popular	Mystery

[18]:
```
#casting column into category

df['Genre'] =df['Genre'].astype('category')

df['Genre'].dtypes
```

[18]:
```
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)
```

[19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25793 entries, 0 to 25792
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Release_Date  25793 non-null  int32
 1   Title         25793 non-null  object
 2   Popularity    25793 non-null  float64
 3   Vote_Count    25793 non-null  int64
 4   Vote_Average  25552 non-null  category
 5   Genre         25793 non-null  category
dtypes: category(2), float64(1), int32(1), int64(1), object(1)
memory usage: 756.7+ KB
```

[20]:
```
df.nunique()
```

[20]:
```
Release_Date     102
Title           9513
Popularity      8160
Vote_Count      3266
Vote_Average       4
Genre             19
dtype: int64
```

[21]:
```
df.head()
```

[21]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	Science Fiction
3	2022	The Batman	3827.658	1151	popular	Crime
4	2022	The Batman	3827.658	1151	popular	Mystery

Data Visualization

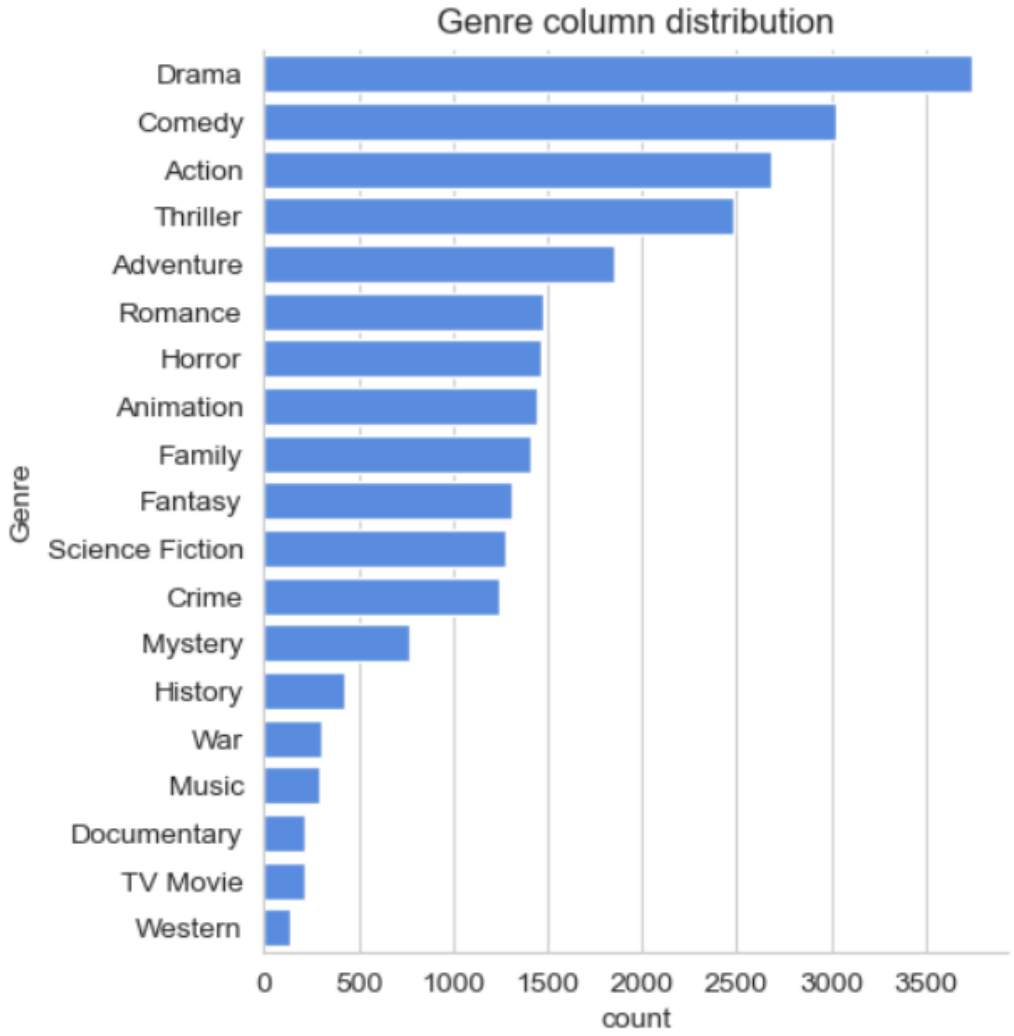
[22]:
```
sns.set_style('whitegrid')
```

What is the most frequent genre of movies released on Netflix?

[23]:
```
df['Genre'].describe()
```

[23]:
```
count     25793
unique       19
top       Drama
freq       3744
Name: Genre, dtype: object
```

[27]:
```
sns.catplot(y= 'Genre',data = df, kind = 'count',
            order = df['Genre'].value_counts().index,
            color = '#4387f5')
plt.title('Genre column distribution')
plt.show()
```



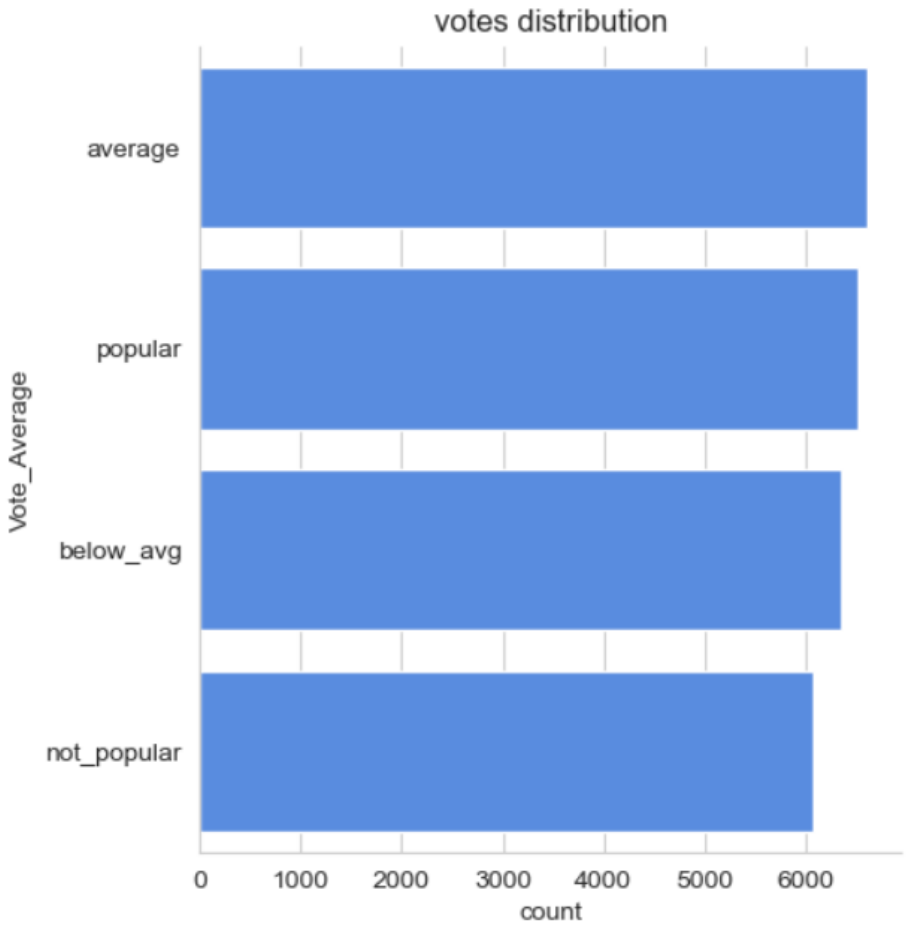
Which has highest votes in vote avg column?

[28]:
```
df.head()
```

[28]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	Science Fiction
3	2022	The Batman	3827.658	1151	popular	Crime
4	2022	The Batman	3827.658	1151	popular	Mystery

[29]:
```
sns.catplot(y= 'Vote_Average',data = df, kind = 'count',
            order = df['Vote_Average'].value_counts().index,
            color = '#4387f5')
plt.title('votes distribution')
plt.show()
```



What movie got the highest popularity? what is its genre?

[30]:
```
df.head(2)
```

[30]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	Adventure

[33]:
```
df[df['Popularity'] == df['Popularity'].max()]
```

[33]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	Science Fiction

What movie got the Lowest popularity? what is its genre?

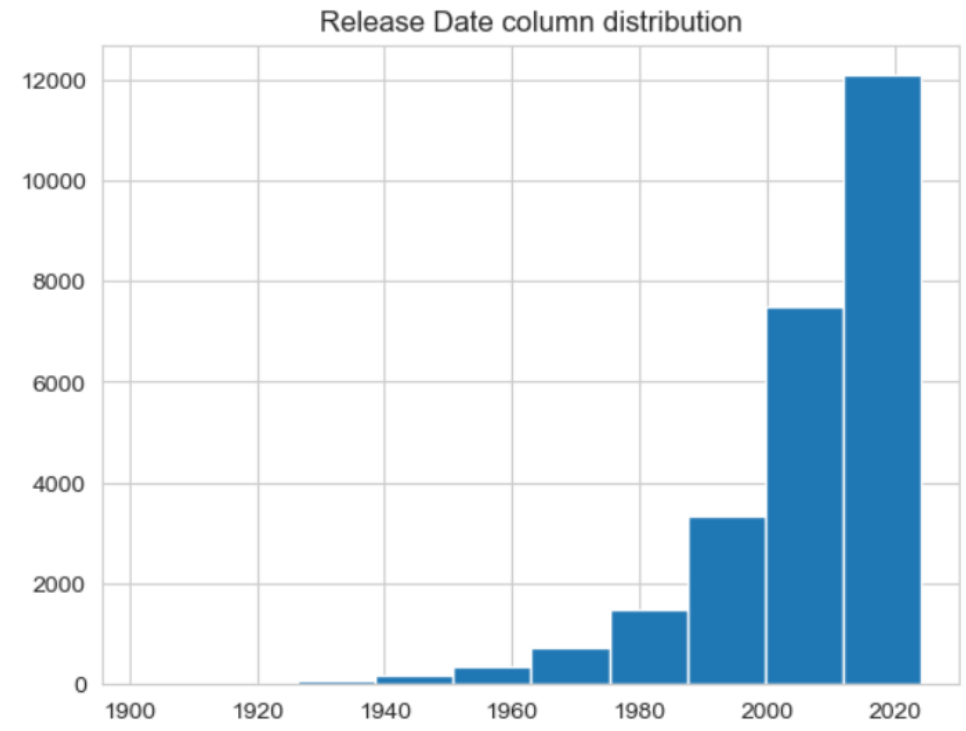
[34]:
```
df[df['Popularity'] == df['Popularity'].min()]
```

[34]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
25787	2021	The United States vs. Billie Holiday	13.354	152	average	Music
25788	2021	The United States vs. Billie Holiday	13.354	152	average	Drama
25789	2021	The United States vs. Billie Holiday	13.354	152	average	History
25790	1984	Threads	13.354	186	popular	War
25791	1984	Threads	13.354	186	popular	Drama
25792	1984	Threads	13.354	186	popular	Science Fiction

Which year has the most filmed movies?

[35]:
```
df['Release_Date'].hist()
plt.title('Release Date column distribution')
plt.show()
```



Conclusion

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

Drama genre is the most frequent genre in our dataset and has appeared more than 14% 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 the highest popularity among fans by being having more than 18.5% of movies.

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

Spider-Man: NO Way Home has the highest popularity rate in our dataset and it has genres Of Action, Adventure and Science Fiction.

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

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

Q5: Which year has the most filmed movies?'

Year 2020 has the highest filming rate in our dataset.

This project provided a comprehensive analysis of a large movie dataset, uncovering valuable insights into genre distribution, audience preferences, popularity trends, and production patterns over the years. The analysis revealed that **Drama** dominates the movie landscape, while **Action and Adventure** genres often lead in popularity. Additionally, the findings showed that high ratings do not always align with high popularity, highlighting the difference between critical acclaim and mass appeal. Identifying the peak year of movie releases offers useful context for understanding industry trends. Overall, this analysis demonstrates how data-driven insights can guide content creation, marketing strategies, and audience targeting in the entertainment industry.