netflix-data-analysis

February 11, 2025

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

[ ]:
```
# Creating the dataframe
df = pd.read_csv("mymoviedb.csv", lineterminator="\n")
df.head()
```

[ ]:
```
  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…
```

[ ]:
```
# viewing dataset info
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
```

Insights: - looks like our dataset has no null values - Overview, Original_Language and Poster-Url wouldn't be so useful during analysis - Release_Date column needs to be casted into date time and to extract only the year value

[105]:
```
# exploring genres column
df['Genre'].head()
```

[105]:
```
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
```

- genres are separated by commas followed by whitespaces

[106]:
```
#checking for duplicated rows
df.duplicated().sum()
```

[106]:
```
np.int64(0)
```

- no duplicated rows are present in this dataset

[107]:
```
# exploring summary statistics
df.describe().T
```

[107]:
```
               count         mean          std     min       25%       50%  \
Popularity    9827.0    40.326088   108.873998  13.354   16.1285   21.199
Vote_Count    9827.0  1392.805536  2611.206907   0.000  146.0000  444.000
```

```
Vote_Average  9827.0     6.439534     1.129759   0.000    5.9000    6.500

                    75%        max
Popularity      35.1915   5083.954
Vote_Count    1376.0000  31077.000
Vote_Average     7.1000     10.000
```

0.0.1 Exploration Summary

- The dataframe comprises 9,827 rows and 9 columns.
- The dataset appears clean with no NaN values or duplicates.
- The Release_Date column should be converted to datetime format.
- The Overview, Original_Language, and Poster_Url columns may not be useful for analysis.
- There are noticeable outliers in the Popularity column.
- The Vote_Average should be categorized for better analysis.
- The Genre column contains comma-separated values and whitespace that need to be addressed.

1 Data Cleaning

[108]:
```
df.head()
```

[108]:
```
  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…
```

[109]:
```
#casting column "Release Date"
df['Release_Date'] = pd.to_datetime(df["Release_Date"])

#confirming changes
print(df['Release_Date'].dtypes)
```

```
datetime64[ns]
```

[110]:
```
df['Release_Date'] = df['Release_Date'].dt.year
df['Release_Date'].dtypes
```

[110]:
```
dtype('int32')
```

[111]:
```
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   int32
 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), int32(1), int64(1), object(5)
memory usage: 652.7+ KB
```

[112]:
```
df.head()
```

[112]:

	Release_Date	Title	Overview	Popularity	Vote_Count	\
0	2021	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to…	5083.954	8940	
1	2022	The Batman				
2	2022	No Exit				
3	2021	Encanto				
4	2021	The King's Man				

```
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…
```

[113]:
```
# making list of column to be dropped
cols = ['Overview','Original_Language','Poster_Url']
```

[114]:
```
# dropping columns and confirming changes
df.drop(cols, axis=1, inplace=True)
df.columns
```

[114]:
```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
       'Genre'],
      dtype='object')
```

[115]:
```
df.head()
```

[115]:
```
   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
```

1.0.1 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()** functionprovided above

[116]:
```
def categorize_col(df, col, labels):
    """
    categories a certain column based on its quartiles

    Args:
        (df)     df   - dataframe we are processing
        (col)    str  - to be categorized column's name
        (labels) list - list of labels from min to max

    Returns:
        (df)     df   - dataframe with the categorized col

    """

    # setting the edges to cut the column accordingly

    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
```

[117]:
```
# define labels for edges
labels = ['not_popular', 'below_avg', 'average', 'popular']

# categorize column based on labels and edges
categorize_col(df, 'Vote_Average', labels)

# confirming changes
df['Vote_Average'].unique()
```

[117]:
```
['popular', 'below_avg', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_avg' < 'average' < 'popular']
```

[118]:
```
df.head()
```

[118]:

	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
```

[119]:
```
# exploring column
df['Vote_Average'].value_counts()
```

[119]:
```
Vote_Average
not_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

[120]:
```
# dropping NaNs
df.dropna(inplace=True)

# confirming
df.isna().sum()
```

[120]:
```
Release_Date    0
Title           0
Popularity      0
Vote_Count      0
Vote_Average    0
Genre           0
dtype: int64
```

[121]:
```
df.head()
```

[121]:
```
   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 thenexplode our dataframe to have only onegenre per row for ezch movie

[122]:
```
# split the strings into lists
df['Genre'] = df['Genre'].str.split(', ')

# explode the lists
df = df.explode('Genre').reset_index(drop=True)
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
```

[123]:
```
# casting column into category
df['Genre'] = df['Genre'].astype('category')

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

[123]:
```
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)
```

[124]:
```
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
```

[125]:
```
df.nunique()
```

[125]:
```
Release_Date     100
Title           9415
Popularity      8088
Vote_Count      3265
Vote_Average       4
Genre             19
dtype: int64
```

Now that our dataset is clean and tidy, we are left with a total of 6 columns and 25551 rows to dig into during our analysis

1.1 Data Visualization

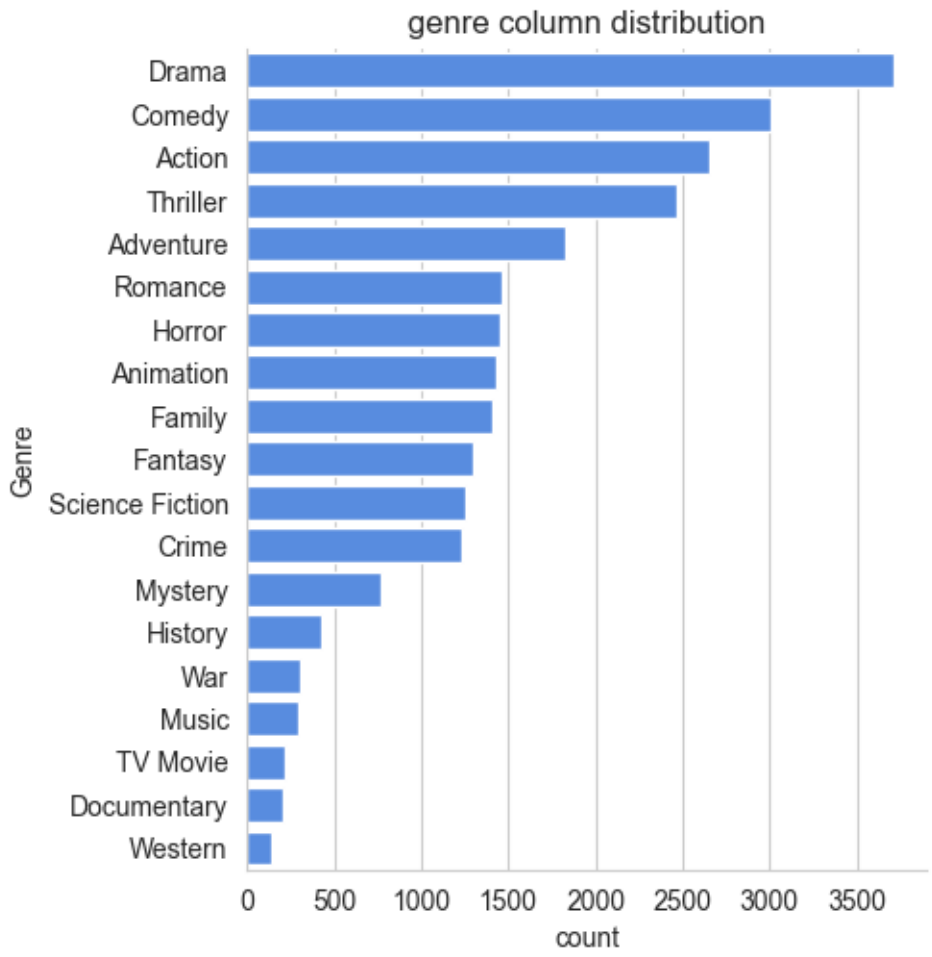
[126]:
```
# setting up seaborn configurations
sns.set_style('whitegrid')
```

1.2 Q1: What is the most frequent genre inthe dataset?

[127]:
```
# showing stats on genre column
df['Genre'].describe()
```

[127]:
```
count     25552
unique       19
top       Drama
freq       3715
Name: Genre, dtype: object
```

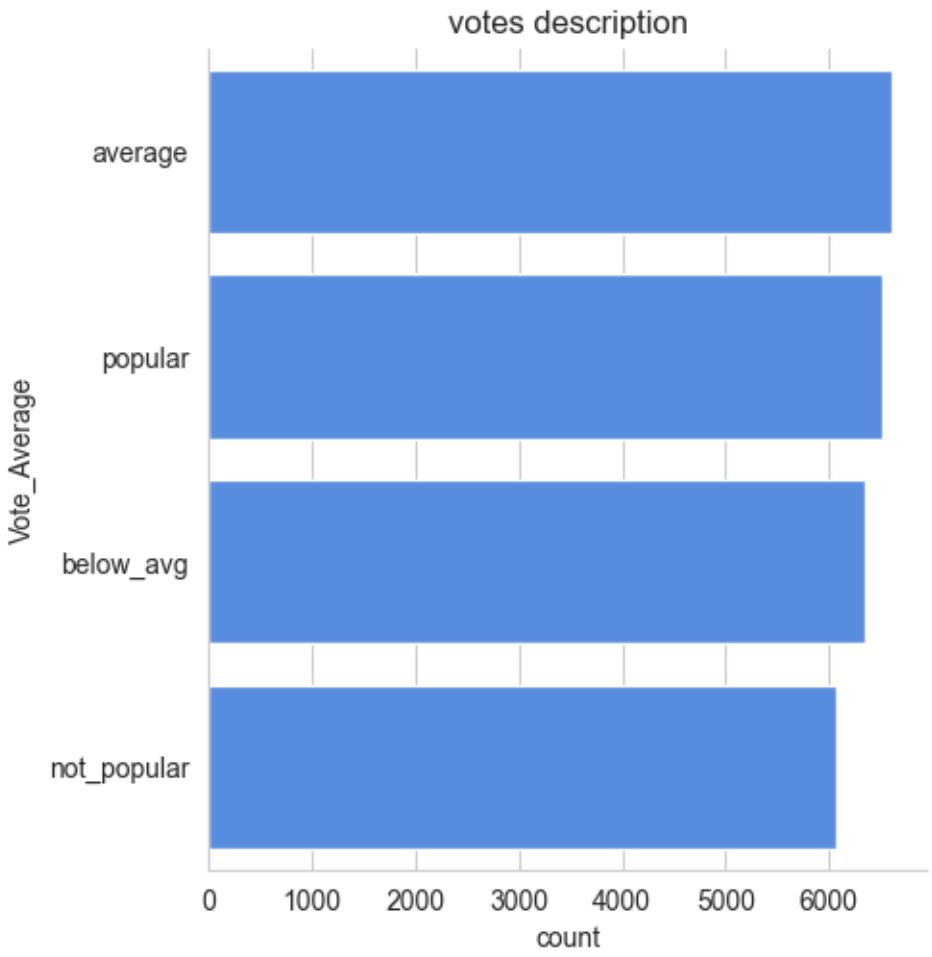
[128]:
```
# visualizing genre column
sns.catplot(y= 'Genre', data=df, kind= 'count',
            order= df['Genre'].value_counts().index,
            color = '#4287f5')
plt.title('genre column distribution')
plt.show()
```



we can notice from the above visual that Drama genre is the most frequent genre in our dataset and has appeared more than 14% of the times among 19 other genres

1.3 Q2: What genres has highest votes

[129]:
```
# visualizing vote_average column
sns.catplot(
    y='Vote_Average', data =df, kind='count',
    order= df['Vote_Average'].value_counts().index,
    color = '#4287f5'
)
plt.title('votes description')
plt.show()
```



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

[130]:
```
# checking max popularity in dataset
df[df['Popularity'] == df['Popularity'].max()]
```

[130]:

	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

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

[131]:
```
# checking max popularity in dataset
df[df['Popularity'] == df['Popularity'].min()]
```

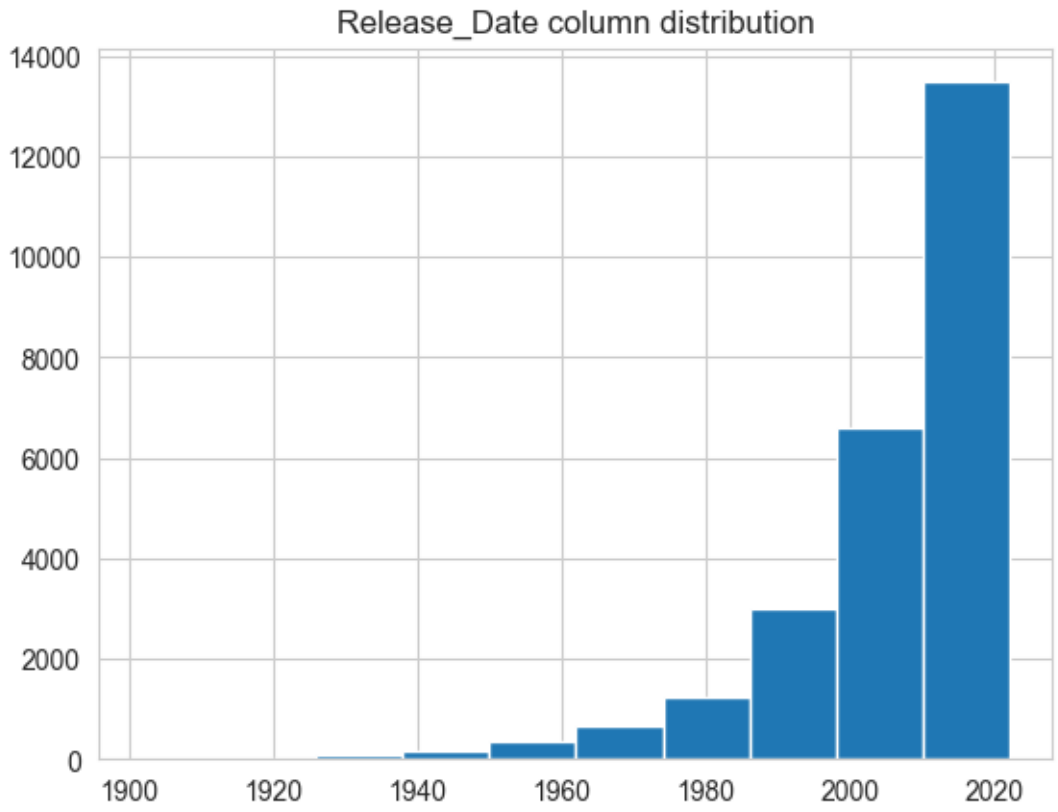
[131]:

	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

1.6 Q5: Which year has the most filmmed movies?

[132]:
```
df['Release_Date'].hist()
plt.title('Release_Date column distribution')
plt.show()
```



2 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 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 genre ? *Spider-Man: No Way Home has the highest popularity rate in our dataset and it hasgenres of Action , 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*

[ ]: