# Analyzing Netflix Dataset: EDA and Data Cleaning

## Importing libraries.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

## Reading Dataset

```
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 information

```
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
```

## Exploring genres column

```
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
```

## Checking for duplicated rows

```
df.duplicated().sum()
```

```
0
```

## Exploring summary statistics

```
df.describe()
```

```
        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
```

# Data Cleaning

Casting Release_Date column and extracing year values

```
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...
```

casting column

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

confirming changes

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

```
datetime64[ns]
```

To see only Year of Realise

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

```
dtype('int32')
```

```
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
```

```
df.head()
```

```
   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 Overview, Original_Language and Poster-Url

making list of column to be dropped

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

dropping columns and confirming changes

```
df.drop(cols, axis = 1, inplace = True)
df.columns
```

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

```
df.head()
```

```
   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

```
# Making a Function For Categorizing Column
def catigorize_col (df, col, labels):
    """
    catigorizes a certain column based on its quartiles

    Args:
    (df) df - dataframe we are proccesing
    (col) str - to be catigorized 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

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

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

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

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

```
df.head()
```

```
   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
```

exploring column

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

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

dropping NaNs

```
df.dropna(inplace = True)
```

confirming

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

```
Release_Date    0
Title           0
Popularity      0
Vote_Count      0
Vote_Average    0
```

```
Genre            0
dtype: int64
```

```
df.head()
```

```
   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 are splitting genres into a list and then explode our dataframe to have only one genre per row for each movie

split the strings into lists

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

explode the lists

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

```
   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
```

casting column into category

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

confirming changes

```
df['Genre'].dtypes
```

```
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)
```

```
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
```

```
df.nunique()
```

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

# Data Visualization

Gaining Visuals and insights of our Data.

```
# setting up seaborn configurations

sns.set_style('whitegrid')
```

Checking which is the most frequent genre in the dataset?

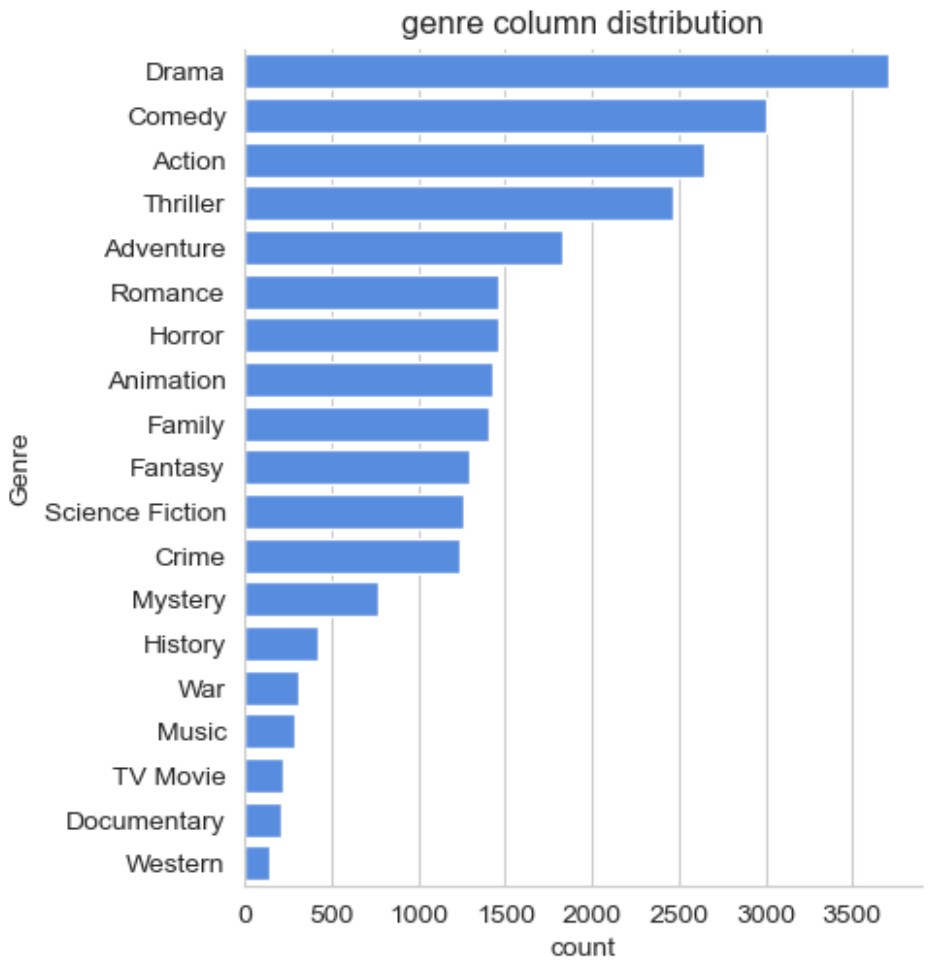
```
# showing stats. on genre column
df['Genre'].describe()
```

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

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()
```

```
D:\Anaconda\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

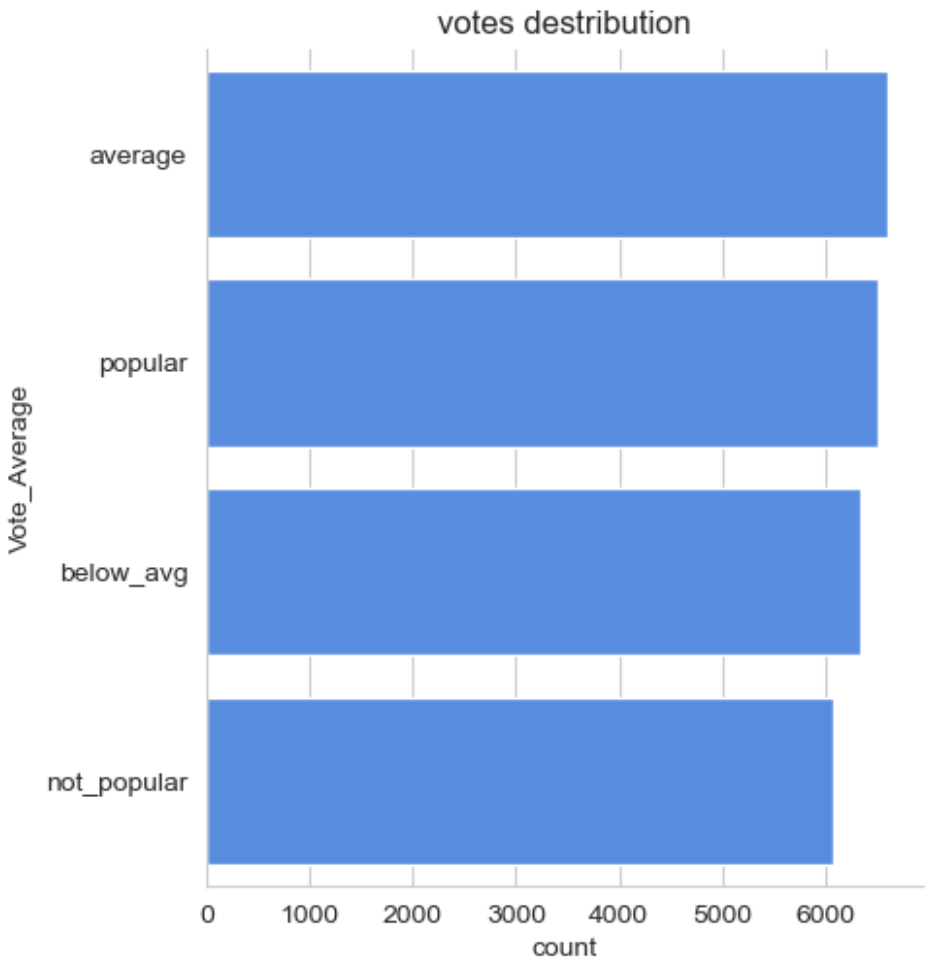



## Checking Which genres has highest votes ?

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 destribution')
plt.show()
```

```
D:\Anaconda\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```




## Checking Which movie got the highest popularity ? what's its genre ?

checking max popularity in dataset

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

```
   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
```

## Checking Which movie got the lowest popularity? what's its genre?

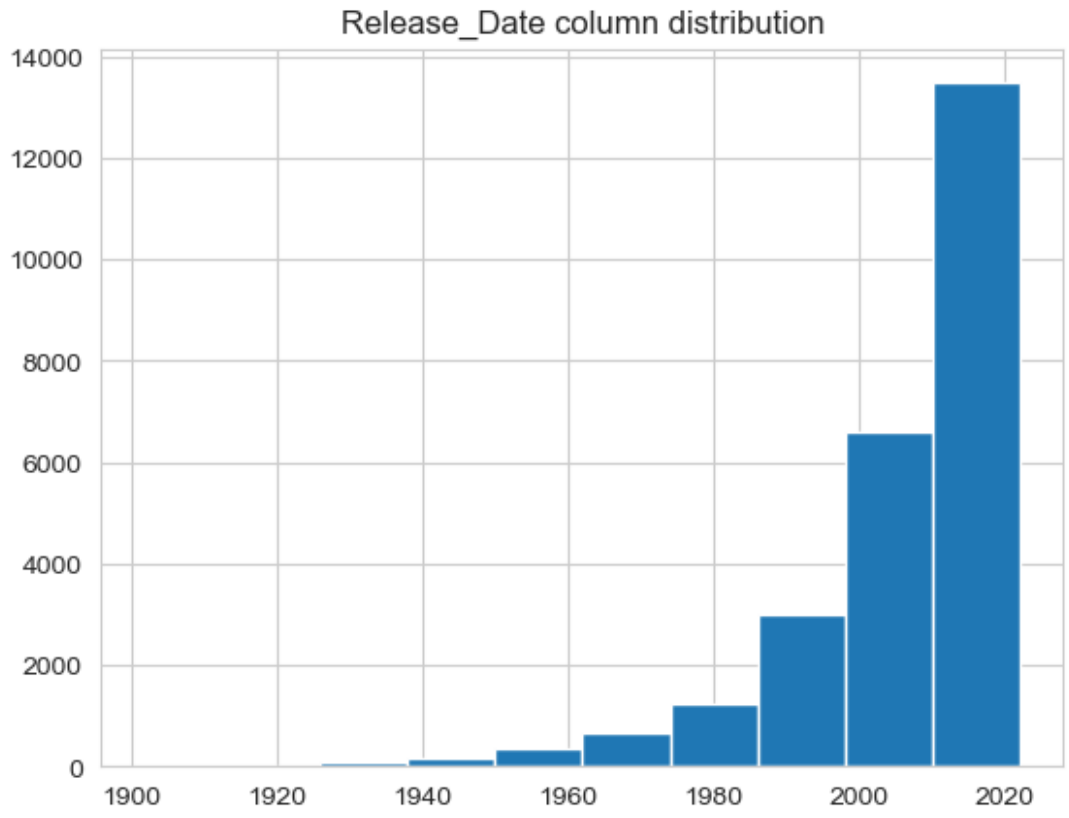
checking min popularity in dataset

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

```
       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
```

## Which year has the most filmmed movies?

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


Release_Date column distribution
14000
12000
10000
8000
6000
4000
2000
0
1900
1920
1940
1960
1980
2000
2020