


```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES,
# THEN FEEL FREE TO DELETE THIS CELL.
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.
import kagglehub
jocelyndumlao_bengali_movie_dataset_path = kagglehub.dataset_download('jocelyndumlao/bengali-movie-dataset')

print('Data source import complete.')
```

 Data source import complete.

Double-click (or enter) to edit

## ▼ Import Libraries

```
!pip install colorama
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import colorama


import warnings
warnings.filterwarnings('ignore')

import math

rc = {
    "axes.facecolor": "#E6F7FF",
    "figure.facecolor": "#E6F7FF",
    "axes.edgecolor": "#000000",
    "grid.color": "#EBEBE7",
    "font.family": "serif",
    "axes.labelcolor": "#000000",
    "xtick.color": "#000000",
    "ytick.color": "#000000",
    "grid.alpha": 0.4
}

sns.set(rc=rc)


from colorama import Style, Fore
red = Style.BRIGHT + Fore.RED
blu = Style.BRIGHT + Fore.BLUE
mgt = Style.BRIGHT + Fore.MAGENTA
gld = Style.BRIGHT + Fore.YELLOW
res = Style.RESET_ALL
```

 Collecting colorama  
 Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)  
 Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)  
 Installing collected packages: colorama  
 Successfully installed colorama-0.4.6

## ▼ Data Understanding and Exploration

1. Load and inspect the datasets (movies.csv and ratings.csv)
2. Check for missing values, data types, and basic statistics
3. Explore the distribution of ratings
4. Explore the genres and their distribution

```
# Read the CSV file
movies_df = pd.read_csv('/kaggle/input/bengali-movie-dataset/Bengali Movie Dataset/movies.csv')
movies_df.head().style.set_properties(**{'background-color': 'royalblue', 'color': 'white', 'border-color': '#8b8c8c'})
```



	platform_Name	movieId	title	genres	director	starring
0	Chorki	1	SHUKLOPOKKHO	ROMANTIC THRILLER	Vicky Zahed	Khairul Basar, Sunerah Binte Kamal, Ziaul Roshan, Faruk ahmed, Sharif Siraj, Abdullah Shentu
1	Chorki	2	SHILPI	DRAMA	Aragami	Uttam Kumar, Suchitra Sen
2	Chorki	3	SHAREY CHUATTOR	DRAMA	Nirmal Dey	Uttam Kumar, Suchitra Sen, Tulsi Chakraborty
3	Chorki	4	SAGARIKA	DRAMA	Aragami	Uttam Kumar, Suchitra Sen, Jamuna Sinha, Namita Sinha

```
ratings_df = pd.read_csv('/kaggle/input/bengali-movie-dataset/Bengali Movie Dataset/ratings.csv')
ratings_df.head().style.set_properties(**{'background-color':'orange', 'color':'white', 'border-color':'#8b8c8c'})
```



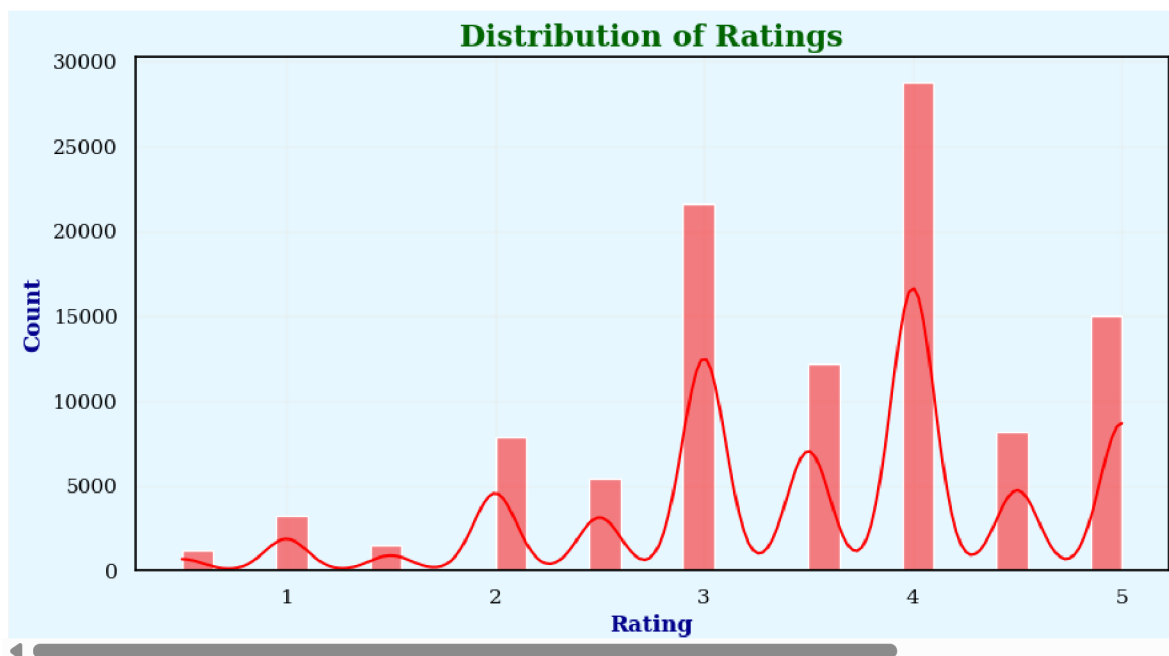
	userId	movieId	rating	timestamp
0	1	1	4.000000	1217897793
1	1	2	1.500000	1217895807
2	1	3	4.000000	1217896246
3	1	4	4.000000	1217896556
4	1	5	4.000000	1217896523

```
# Check for missing values
movies_df.isnull().sum()
ratings_df.isnull().sum()
```



0
---

```
# Explore the distribution of ratings
plt.figure(figsize=(10, 5))
#sns.set_style("whitegrid")
sns.histplot(ratings_df['rating'], bins=30, kde=True, color='red')
plt.title('Distribution of Ratings', fontsize=16, fontweight = 'bold', color = 'darkgreen')
plt.xlabel('Rating', fontsize=12, fontweight = 'bold', color = 'darkblue')
plt.ylabel('Count', fontsize=12, fontweight = 'bold', color = 'darkblue')
plt.savefig('Distribution of Ratings.png')
plt.show()
```



```
# Analyze missing values in movies_df and ratings_df
movies_missing = movies_df.isnull().sum()
ratings_missing = ratings_df.isnull().sum()

# Analyze the popularity of different genres
genres_count = movies_df['genres'].value_counts().head(10)

print("Missing Values in movies_df:")
print(movies_missing)
print("\nMissing Values in ratings_df:")
print(ratings_missing)
print("\nTop 10 Movie Genres:")
print(genres_count)
```



Missing Values in movies\_df:

```
platform_Name    0
movieId          0
title            0
genres           0
director        158
starring         9
dtype: int64
```

Missing Values in ratings\_df:

```
userId    0
movieId   0
rating    0
timestamp 0
dtype: int64
```

Top 10 Movie Genres:

```
genres
DRAMA          96
Thriller       43
Drama          40
Comedy         27
Horror         20
Crime, Thriller 18
COMEDY         10
THRILLER        9
Drama, Romance  9
Romance         9
Name: count, dtype: int64
```

### Observation:

1. There is a missing values in movies datasets and no missing values for ratings dataset.
2. The distribution of ratings is right-skewed, with most ratings around 4.0.
3. The top 10 movie genres and their respective counts are printed.

▼ Data Preprocessing

- 1. Merge datasets on 'movieId' to create a unified dataset
- 2. Handle any missing or erroneous values (if any)
- 3. Perform data type conversions if necessary
- 4. Encode categorical variables (e.g., platform\_Name, genres)

```
# Merge datasets on 'movieId'
merged_df = pd.merge(ratings_df, movies_df, on='movieId')

# Encode categorical variables (platform_Name and genres)
merged_df = pd.get_dummies(merged_df, columns=['platform_Name', 'genres'], prefix=['platform', 'genre'])

# Check the merged dataset
print("Merged Dataset:")
merged_df.head()
```

🔄 Merged Dataset:

	userId	movieId	rating	timestamp	title	director	starring	platform_Chorki	platform_Hoichoi	genre_DRAMA	...	genre_SciF
0	1	1	4.0	1217897793	SHUKLOPOKKHO	Vicky Zahed	Khairul Basar, Sunerah Binte Kamal, Ziaul Rosh...	True	False	False	...	Fals
1	1	2	1.5	1217895807	SHILPI	Aragami	Uttam Kumar, Suchitra Sen	True	False	False	...	Fals
2	1	3	4.0	1217896246	SHAREY CHUATTOR	Nirmal Dey	Uttam Kumar, Suchitra Sen, Tulsi Chakraborty	True	False	False	...	Fals
3	1	4	4.0	1217896556	SAGARIKA	Aragami	Uttam Kumar, Suchitra Sen, Jamuna Sinha, Namit...	True	False	False	...	Fals
4	1	5	4.0	1217896523	DEEP JWELEY JAI	Asit Sen	Suchitra Sen, Tulsi Chakraborty, Ajit Chatterj...	True	False	False	...	Fals

5 rows × 73 columns

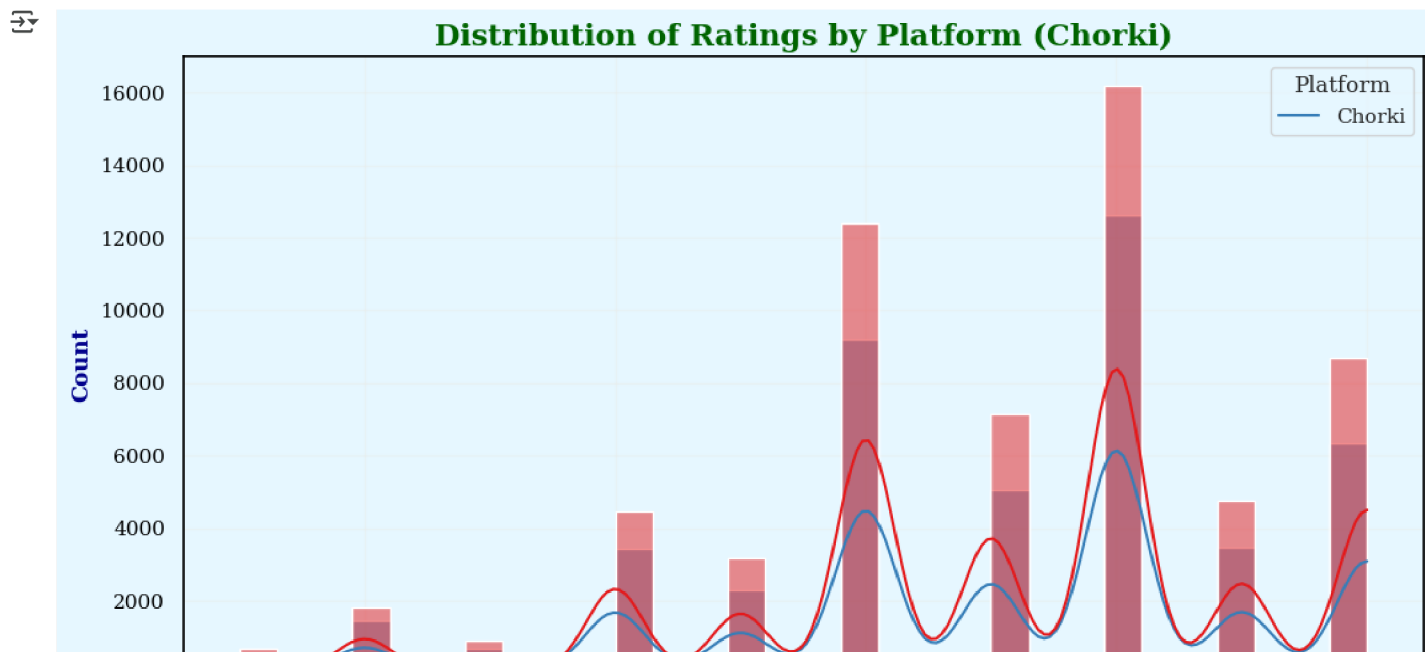
Observation:

- 1. We've merged the ratings and movies datasets on 'movieId' to create a unified dataset.
- 2. We've encoded categorical variables ('platform\_Name' and 'genres') into binary columns.

▼ Exploratory Data Analysis (EDA)

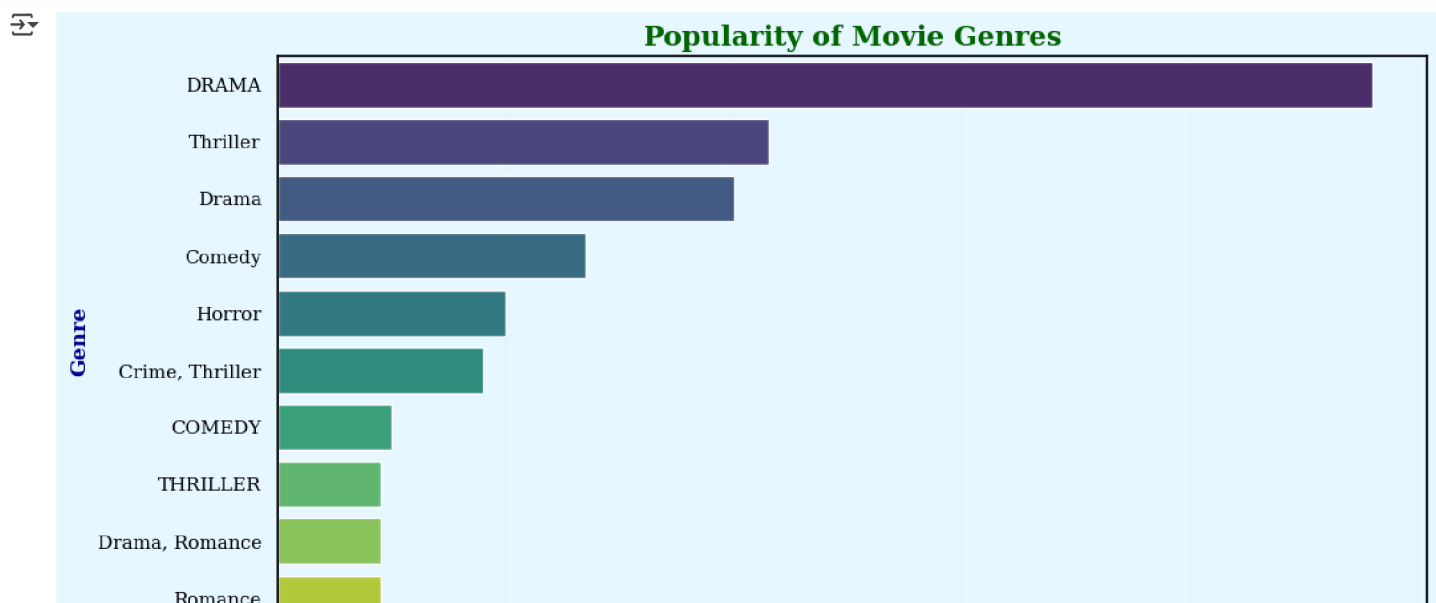
- 1. Visualize the distribution of ratings
- 2. Analyze the popularity of different genres
- 3. Explore the distribution of movies across different platforms
- 4. Identify the most popular directors and starring actors/actresses

```
# Visualize the distribution of ratings with platform comparison
plt.figure(figsize=(12, 6))
#sns.set_style("whitegrid")
sns.histplot(data=merged_df, x='rating', hue='platform_Chorki', bins=30, kde=True, palette='Set1')
plt.title('Distribution of Ratings by Platform (Chorki)', fontsize=16, fontweight = 'bold', color = 'darkgreen')
plt.xlabel('Rating', fontsize=12, fontweight = 'bold', color = 'darkblue')
plt.ylabel('Count', fontsize=12, fontweight = 'bold', color = 'darkblue')
plt.legend(title='Platform', labels=['Chorki'])
plt.savefig('Distribution of Ratings by Platform (Chorki).png')
plt.show()
```



```
# Analyze the popularity of different genres
genre_counts = movies_df['genres'].value_counts().head(10)

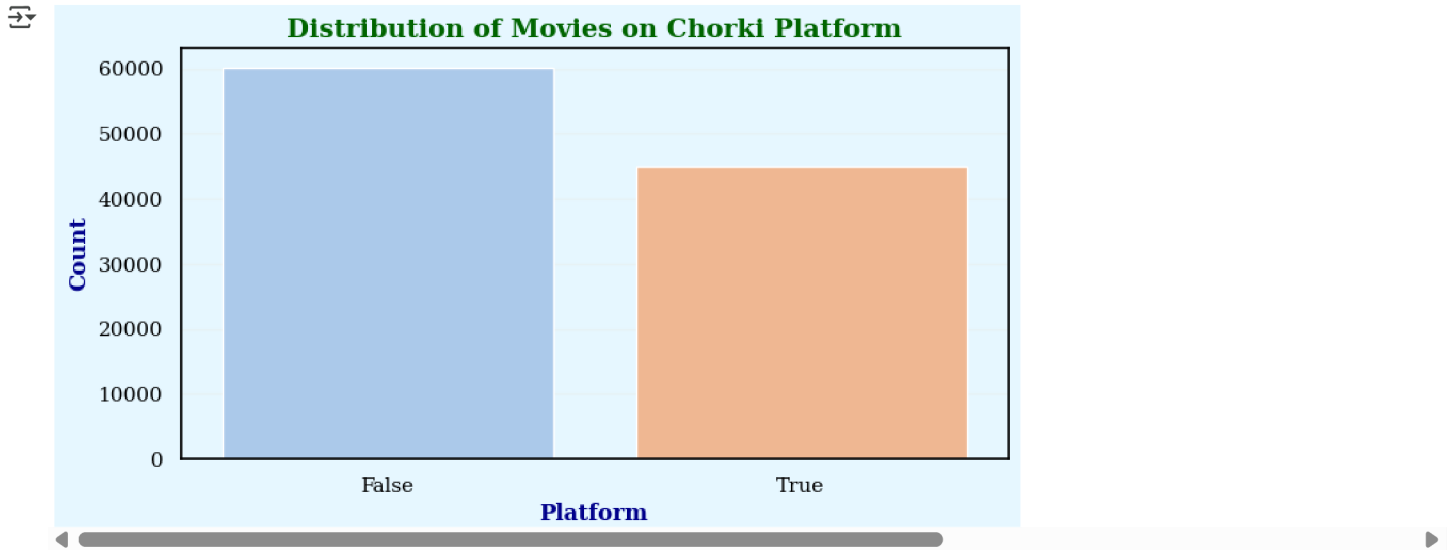
plt.figure(figsize=(12, 6))
#sns.set_style("whitegrid")
sns.barplot(x=genre_counts.index, y=genre_counts.values, palette='viridis')
plt.title('Popularity of Movie Genres', fontsize=16, fontweight = 'bold', color = 'darkgreen')
plt.xlabel('Genre', fontsize=12, fontweight = 'bold', color = 'darkblue')
plt.ylabel('Count', fontsize=12, fontweight = 'bold', color = 'darkblue')
plt.savefig('Popularity of Movie Genres.png')
plt.show()
```



```
# Explore the distribution of movies across different platforms
platform_counts = merged_df['platform_Chorki'].value_counts()

plt.figure(figsize=(8, 4))
```

```
#sns.set_style("whitegrid")
sns.barplot(x=platform_counts.index, y=platform_counts.values, palette='pastel')
plt.title('Distribution of Movies on Chorki Platform', fontsize=14, fontweight = 'bold', color = 'darkgreen')
plt.xlabel('Platform', fontsize=12, fontweight = 'bold', color = 'darkblue')
plt.ylabel('Count', fontsize=12, fontweight = 'bold', color = 'darkblue')
plt.savefig('Distribution of Movies on Chorki Platform.png')
plt.show()
```



#### ▼ Descriptive Statistics

1. Calculate summary statistics for ratings (mean, median, standard deviation, etc.)
2. Identify the highest and lowest rated movies
3. Analyze the frequency of ratings

```
# Calculate summary statistics for ratings
rating_stats = merged_df['rating'].describe()

# Identify the highest and lowest rated movies
top_rated_movies = merged_df.groupby('title')['rating'].mean().nlargest(5)
lowest_rated_movies = merged_df.groupby('title')['rating'].mean().nsmallest(5)

# Combine the results into a single DataFrame
summary_df = pd.DataFrame({
    'Summary Statistics for Ratings': rating_stats,
    'Top 5 Highest Rated Movies': top_rated_movies,
    'Top 5 Lowest Rated Movies': lowest_rated_movies
})

# Define a style function to apply color to the table
def highlight_max(s):
    is_max = s == s.max()
    return ['background-color: lightgreen' if v else '' for v in is_max]

styled_summary = summary_df.style.apply(highlight_max)

# Render the styled table as HTML
styled_summary = styled_summary.set_table_styles([
    {'selector': 'th', 'props': 'background-color: #f2f2f2; font-weight: bold;'},
    {'selector': 'td', 'props': 'font-size: 12px;'},
])

# Display the table
styled_summary
```



Summary Statistics for Ratings				Top 5 Highest Rated Movies	Top 5 Lowest Rated Movies
25%		3.000000		nan	nan
50%		3.500000		nan	nan
75%		4.000000		nan	nan
ARONNAY EKODA		nan		nan	3.391304
BHAT DE		nan		3.637681	nan
BICYCLE		nan		nan	3.367754
Eken Babu		nan		3.663043	nan
Gora		nan		3.634058	nan
JOHIR KARIGOR		nan		nan	3.393116
Karagar		nan		4.764493	nan
SCOOTY		nan		nan	3.373188
SWOPNODANAY		nan		3.637681	nan
Tansener Tanpura,Episode: Abarohon (2020)		nan		nan	3.391304
count	105156.000000			nan	nan
max	5.000000			nan	nan
mean	3.521473			nan	nan
min	0.500000			nan	nan
std	1.045558			nan	nan

▼ User Behavior Analysis

1. Analyze user behavior (e.g., most active users, average number of ratings per user)
2. Identify user preferences based on genres, directors, or actors/actresses

```
# Calculate the average number of ratings per user
average_ratings_per_user = merged_df.groupby('userId')['rating'].count().mean()

# Identify the most active users (top 5)
active_users = merged_df['userId'].value_counts().head(5)

# Combine the results into a single DataFrame
user_stats_df = pd.DataFrame({
    'Average Ratings per User': [average_ratings_per_user],
    'Top 5 Most Active Users': [' ', '.join(active_users.index.astype(str))])
})

# Define a style function to apply color to the table
def highlight_max(s):
    is_max = s == s.max()
    return ['background-color: yellow' if v else '' for v in is_max]

styled_user_stats = user_stats_df.style.apply(highlight_max)

# Render the styled table as HTML
styled_user_stats = styled_user_stats.set_table_styles([
    {'selector': 'th', 'props': 'background-color: #f2f2f2; font-weight: bold;'},
    {'selector': 'td', 'props': 'font-size: 12px;'},
])

# Display the table
styled_user_stats
```



Average Ratings per User			Top 5 Most Active Users
0	157.419162		668, 575, 458, 232, 310

```
# Create visualizations for key findings
```

```
# For example, let's create a heatmap to visualize the correlation between ratings and genres
```

```
plt.figure(figsize=(10, 6))
```

```
#sns.set_style("whitegrid")
```

```
ratings_genres_corr = merged_df[['rating', 'genre_DRAMA', 'genre_ROMANTIC THRILLER']].corr()
```

```
sns.heatmap(ratings_genres_corr, annot=True, cmap='coolwarm', fmt='.2f')
```

```
plt.title('Correlation between Ratings and Genres', fontsize=16, fontweight = 'bold', color = 'darkgreen')
```

```
plt.savefig('Correlation between Ratings and Genres.png')
```

```
plt.show()
```

