Social Media

Project overview:

This project analyses a movie review dataset consisting of approximately 550 records. The focus is data cleaning, exploratory data analysis and sentiment-based insights to understand public opinion about different movies.

Raw Data:

The raw dataset contains missing values, duplicate records, and inconsistent data types.

Below is a sample of the raw data:

	A	B	C	D	E	F	G	H	I	J
	user_id	movie_na	review_te	release_date	sentiment	duration	platform	timestamp	rating	likes
2		lagaan	Loved the	16-12-2001	positive	120 min	YouTube	20-01-2014 17:21		-25
3	user355	bahubali	EXCELLEN	30-12-2000	NEGATIVE	120 min	YouTube	13-09-2024 02:56	6	
4	user333	RRR	Amazing r	21-12-1991	positive	150 MIN	instagram	14-12-1990 09:58	7.5	0
5		lagaan	It was oka	29-05-2005	very NEGA	180mins	instagram	25-04-2017 20:33	5	
6	user924	Pushpa	Loved the	20-09-2020	very NEGA	90 Min	YouTube	09-05-2013 21:26	10	
7		3 idiots	not worth	15-09-1993	very posi	2h 30m	instagram	09-01-2005 10:35	-2	1200
8	user406	lagaan	It was oka	25-09-1990	NEGATIVE	90 Min	Twitter	09-03-2018 11:00	5	-25
9	user554	3 idiots	not worth	05-09-1992	Neutral	180mins	Netflix	24-04-2022 07:34	0	-25
10	USR_138	lagaan	Loved the	25-12-1991	positive	150 MIN	YouTube	01-09-1998 00:16	5	1200
11	USR_70	dangal	Loved the	06-01-1990	positive	90 Min	Twitter	22-11-2012 11:45	-2	3
12	user824	KGF chapt	Amazing r	09-07-2014	very posi	120 min	FACEBOOI	11-10-2001 15:02	10	0
13		dangal	EXCELLEN	08-05-1997	very posi	120 min	Netflix	17-06-2020 04:23	2.5	1200
14	user854	the DARK	not worth	08-04-2013	very posi	90 Min	FACEBOOI	22-07-1996 20:49	4	3
15	user139	bahubali	bad direct	13-05-1995	NEGATIVE	90 Min	YouTube	21-05-2009 01:47	7.5	1200
16	USR_6	Sholay	EXCELLEN	15-03-2023	very posi	2h 30m	YouTube	25-09-2024 00:10	5	50
17		bahubali	bad direct	13-09-1994	very NEGA	2h 30m	instagram	30-07-2015 10:05	10	-25
18	USR_53	Interstella	Loved the	06-12-1991	very NEGA	120 min	instagram	29-08-2012 06:48		-25
19		Inception	bad direct	22-03-2004	positive	2h 30m	imdb	13-02-2014 04:59	10	-25
20	user514	KGF chapt	not worth	08-04-1991	Neutral	90 Min	YouTube	11-04-2005 12:58	10	-25
21		Pathaan	not worth	20-12-2016	very NEGA	180mins	FACEBOOI	18-03-2021 18:07		0
22	user380	Sholay	Loved the	26-02-2004	Neutral	150 MIN	YouTube	22-05-2005 06:52	6	
23	user394	Interstella	not worth	05-11-1997	very NEGA	180mins	FACEBOOI	07-02-1993 09:07	4	50

Dataset Overview

The dataset contains User Id, Platform, Username, Movie Title, Review Text, Likes, Rating, Sentiment, Release Date, Duration, Timestamp, Weekday and Day columns.

```
import pandas as pd
 import numpy as np
 import regex
 import seaborn as sns
 import matplotlib.pyplot as plt
 import matplotlib matplotlib.use('Agg') import plotly
express as px %matplotlib inline
import pandas as pd
```

Social Media

```
df=pd.read_csv("Movie_review.csv")
print(df.head(10))
print(df.info())
print(df.describe())
```

```
        user_id movie_name                    review_text release_date  \
...  0      NaN    lagaan        Loved the acting and story   2001-12-16
     1  user355    bahubali      EXCELLENT visuals but slow    2000-12-30
     2  user333        RRR      Amazing movie!!  must watch    1991-12-21
     3      NaN    lagaan   It was okay... nothing special    2005-05-29
     4  user924     Pushpa       Loved the acting and story   2020-09-20
     5      NaN   3 idiots            not worth the hype      1993-09-15
     6  user406     lagaan   It was okay... nothing special    1990-09-25
     7  user554    3 idiots            not worth the hype      1992-09-05
     8   USR_138    lagaan        Loved the acting and story   1991-12-25
     9    USR_70     dangal       Loved the acting and story   1990-01-06

          sentiment  duration    platform            timestamp  rating   likes
     0        positive   120 min     YouTube  2014-01-20 17:21:21     NaN   -25.0
     1        NEGATIVE   120 min     YouTube  2024-09-13 02:56:38     6.0     NaN
     2        positive   150 MIN   instagram  1990-12-14 09:58:27     7.5     0.0
     3   very NEGATIVE  180mins    instagram  2017-04-25 20:33:36     5.0     NaN
     4   very NEGATIVE    90 Min     YouTube  2013-05-09 21:26:23    10.0     NaN
     5   very positive    2h 30m   instagram  2005-01-09 10:35:25    -2.0  1200.0
     6        NEGATIVE    90 Min     Twitter  2018-03-09 11:00:57     5.0   -25.0
     7        Neutral   180mins      Netflix  2022-04-24 07:34:30     0.0   -25.0
     8        positive   150 MIN     YouTube  1998-09-01 00:16:51     5.0  1200.0
     9        positive    90 Min     Twitter  2012-11-22 11:45:05    -2.0     3.0
     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 500 entries, 0 to 499
     Data columns (total 10 columns):
```

Cleaning Data:

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

```
...  0      False
     1      False
     2      False
     3      False
     4      False
            ...
     495    False
     496    False
     497    False
     498    False
     499    False
     Length: 500, dtype: bool
```

Social Media

df = df.drop_duplicates()

print(df.isnull())

```
     user_id  movie_name  review_text  release_date  sentiment  duration  \
0       True       False        False         False      False     False
1      False       False        False         False      False     False
2      False       False        False         False      False     False
3       True       False        False         False      False     False
4      False       False        False         False      False     False
..       ...         ...          ...           ...        ...       ...
495     True       False        False         False      False     False
496    False       False        False         False      False     False
497    False       False        False         False      False     False
498    False       False        False         False      False     False
499    False       False        False         False      False     False

     platform  timestamp  rating  likes
0       False      False    True  False
1       False      False   False   True
2       False      False   False  False
3       False      False   False   True
4       False      False   False   True
..        ...        ...     ...    ...
495     False      False   False  False
```

df_cleaned = df.dropna()

df['timestamp'] = pd.to_datetime(df['timestamp'], errors='coerce')

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   user_id       338 non-null    object
 1   movie_name    500 non-null    object
 2   review_text   500 non-null    object
 3   release_date  500 non-null    object
 4   sentiment     500 non-null    object
 5   duration      500 non-null    object
 6   platform      500 non-null    object
 7   timestamp     500 non-null    datetime64[ns]
 8   rating        450 non-null    float64
 9   likes         435 non-null    float64
dtypes: datetime64[ns](1), float64(2), object(7)
memory usage: 39.2+ KB
```

Datatype Change:

print(df.dtypes)

Cleaning and Preparation:

Social Media

Cleaning tasks performed include trimming spaces, converting text to proper format, Capitalizing column and title names

Cleaning and Preparation:

Cleaning tasks performed include trimming spaces, converting text to proper format, Capitalizing column and title names

df = df.apply(lambda col: col.str.strip() if col.dtype == "object" else col)

	user_id	movie_name	review_text	release_date	sentiment	duration	platform	timestamp	rating	likes	Duration_min
0	NaN	Lagaan	Loved the acting and story	2001-12-16	positive	120 Min	Youtube	2014-01-20 17:21:21	4.156667	-25.0	120
1	user355	Bahubali	excellent visuals but slow	2000-12-30	NEGATIVE	120 Min	Youtube	2024-09-13 02:56:38	6.000000	NaN	120
2	user333	Rrr	Amazing movie!! must watch	1991-12-21	positive	150 Min	instagram	1990-12-14 09:58:27	7.500000	0.0	150
3	NaN	Lagaan	It was okay... nothing special	2005-05-29	very NEGATIVE	180 Min	instagram	2017-04-25 20:33:36	5.000000	NaN	180
4	user924	Pushpa	Loved the acting and story	2020-09-20	very NEGATIVE	90 Min	Youtube	2013-05-09 21:26:23	10.000000	NaN	90
...	...	...	...	...	...	...	...	...	...	...	...
495	NaN	Kgf chapter 2	excellent visuals but slow	2017-03-23	positive	180 Min	Facebook	2015-11-21 22:00:25	6.000000	3.0	180
496	USR_147	avatar	Not worth the hype	1990-02-20	positive	90 Min	Facebook	2019-12-08 01:55:43	7.500000	3.0	90

df.columns = df.columns.str.title()

	User_Id	Movie_Name	Review_Text	Release_Date	Sentiment	Duration	Platform	Timestamp	Rating	Likes	Duration_Min
0	NaN	Lagaan	Loved the acting and story	2001-12-16	Positive	120 Min	Youtube	2014-01-20 17:21:21	4.156667	-25.0	120
1	user355	Bahubali	excellent visuals but slow	2000-12-30	Negative	120 Min	Youtube	2024-09-13 02:56:38	6.000000	NaN	120
2	user333	Rrr	Amazing movie!! must watch	1991-12-21	Positive	150 Min	instagram	1990-12-14 09:58:27	7.500000	0.0	150
3	NaN	Lagaan	It was okay... nothing special	2005-05-29	Very negative	180 Min	instagram	2017-04-25 20:33:36	5.000000	NaN	180
4	user924	Pushpa	Loved the acting and story	2020-09-20	Very negative	90 Min	Youtube	2013-05-09 21:26:23	10.000000	NaN	90
...	...	...	...	...	...	...	...	...	...	...	...
495	NaN	Kgf chapter 2	excellent visuals but slow	2017-03-23	Positive	180 Min	Facebook	2015-11-21 22:00:25	6.000000	3.0	180
496	USR_147	avatar	Not worth the hype	1990-02-20	Positive	90 Min	Facebook	2019-12-08 01:55:43	7.500000	3.0	90
497	USR_192	Bahubali	It was okay... nothing special	2011-09-24	Very negative	120 Min	Youtube	2010-07-15 08:51:32	5.000000	-25.0	120
498	USR_82	The dark knight	excellent visuals but slow	2009-07-13	Very negative	90 Min	instagram	2004-08-08 17:47:49	-2.000000	50.0	90

df.groupby("User_Id")[['Rating', 'Likes', 'minutes']].median()

df

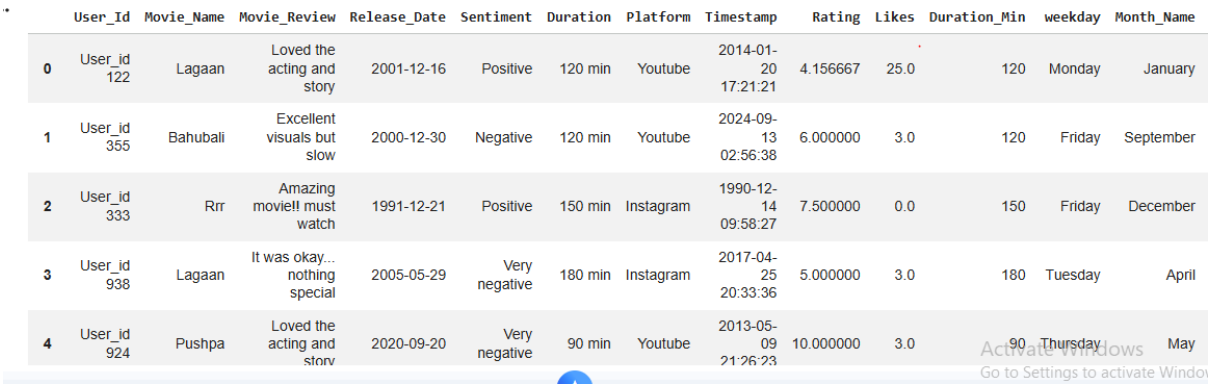
	User_Id	Movie_Name	Movie_Review	Release_Date	Sentiment	Duration	Platform	Timestamp	Rating	Likes	Duration_Min	weekday	Month_Name	Rati
0	NaN	Lagaan	Loved the acting and story	2001-12-16	Positive	120 Min	Youtube	2014-01-20 17:21:21	4.156667	25.0	120	Monday	January	
1	user355	Bahubali	excellent visuals but slow	2000-12-30	Negative	120 Min	Youtube	2024-09-13 02:56:38	6.000000	3.0	120	Friday	September	
2	user333	Rrr	Amazing movie!! must watch	1991-12-21	Positive	150 Min	instagram	1990-12-14 09:58:27	7.500000	0.0	150	Friday	December	
3	NaN	Lagaan	It was okay... nothing special	2005-05-29	Very negative	180 Min	instagram	2017-04-25 20:33:36	5.000000	3.0	180	Tuesday	April	
4	user924	Pushpa	Loved the acting and story	2020-09-20	Very negative	90 Min	Youtube	2013-05-09 21:26:23	10.000000	3.0	90	Thursday	May	

df['Sentiment']=df['Sentiment'].str.capitalize()

df['Movie_Name']=df['Movie_Name'].str.capitalize()

Social Media

```
df['Movie_Review']=df['Movie_Review'].str.capitalize()
df['Platform']=df['Platform'].str.capitalize()
df['Duration']=df['Duration'].str.capitalize()
df
```



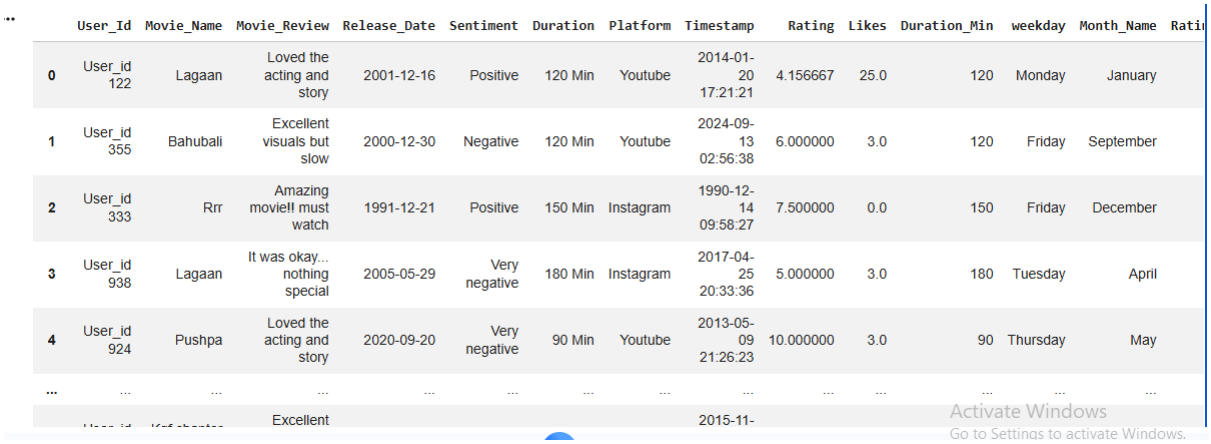
	User_Id	Movie_Name	Movie_Review	Release_Date	Sentiment	Duration	Platform	Timestamp	Rating	Likes	Duration_Min	weekday	Month_Name
0	User_id 122	Lagaan	Loved the acting and story	2001-12-16	Positive	120 min	Youtube	2014-01-20 17:21:21	4.156667	25.0	120	Monday	January
1	User_id 355	Bahubali	Excellent visuals but slow	2000-12-30	Negative	120 min	Youtube	2024-09-13 02:56:38	6.000000	3.0	120	Friday	September
2	User_id 333	Rrr	Amazing movie!! must watch	1991-12-21	Positive	150 min	Instagram	1990-12-14 09:58:27	7.500000	0.0	150	Friday	December
3	User_id 938	Lagaan	It was okay... nothing special	2005-05-29	Very negative	180 min	Instagram	2017-04-25 20:33:36	5.000000	3.0	180	Tuesday	April
4	User_id 924	Pushpa	Loved the acting and story	2020-09-20	Very negative	90 min	Youtube	2013-05-09 21:26:23	10.000000	3.0	90	Thursday	May


```
import re


def convert(x):
    s = str(x).lower().strip()
    numbers = re.findall(r'\d+', s)

    if not numbers:
        return None # Handle cases where no numbers are found

    if len(numbers) == 1:
        # Assume it's a minute value if only one number is found
        total = int(numbers[0])
    elif len(numbers) == 2:
        # Assume it's an 'h m' format if two numbers are found
        h = int(numbers[0])
        m = int(numbers[1])
        total = h * 60 + m
    else:
        return None # Unexpected format with more than two numbers

    return f"{total} Min"
```

Social Media

df['Duration'] = df['Duration'].apply(convert)

df



	User_Id	Movie_Name	Movie_Review	Release_Date	Sentiment	Duration	Platform	Timestamp	Rating	Likes	Duration_Min	weekday	Month_Name	Rati
0	User_id 122	Lagaan	Loved the acting and story	2001-12-16	Positive	120 Min	Youtube	2014-01-20 17:21:21	4.156667	25.0	120	Monday	January	
1	User_id 355	Bahubali	Excellent visuals but slow	2000-12-30	Negative	120 Min	Youtube	2024-09-13 02:56:38	6.000000	3.0	120	Friday	September	
2	User_id 333	Rrr	Amazing movie!! must watch	1991-12-21	Positive	150 Min	Instagram	1990-12-14 09:58:27	7.500000	0.0	150	Friday	December	
3	User_id 938	Lagaan	It was okay... nothing special	2005-05-29	Very negative	180 Min	Instagram	2017-04-25 20:33:36	5.000000	3.0	180	Tuesday	April	
4	User_id 924	Pushpa	Loved the acting and story	2020-09-20	Very negative	90 Min	Youtube	2013-05-09 21:26:23	10.000000	3.0	90	Thursday	May	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
			Excellent					2015-11-						

Handling null Values:

df['Rating']=df['Rating'].fillna(df['Rating'].mean())

df['Likes']=df['Likes'].fillna(df['Likes'].median())

df

	User_Id	Movie_Name	Movie_Review	Release_Date	Sentiment	Duration	Platform	Timestamp	Rating	Likes	weekday	Month_Name	Rating_Review
0	User_id 122	Lagaan	Loved the acting and story	2001-12-16	Positive	120 Min	Youtube	2014-01-20 17:21:21	4.156667	25.0	Monday	January	Average
1	User_id 355	Bahubali	Excellent visuals but slow	2000-12-30	Negative	120 Min	Youtube	2024-09-13 02:56:38	6.000000	3.0	Friday	September	Average
2	User_id 333	Rrr	Amazing movie!! must watch	1991-12-21	Positive	150 Min	Instagram	1990-12-14 09:58:27	7.500000	0.0	Friday	December	Average
3	User_id 938	Lagaan	It was okay... nothing special	2005-05-29	Very negative	180 Min	Instagram	2017-04-25 20:33:36	5.000000	3.0	Tuesday	April	Average
4	User_id 924	Pushpa	Loved the acting and story	2020-09-20	Very negative	90 Min	Youtube	2013-05-09 21:26:23	10.000000	3.0	Thursday	May	Hit
...	...	...	...	...	...	...	...	...	...	...	...	...	...

df.sort_values(by='Rating',ascending=False)

import numpy as np

Extract the numeric part of User_Id. Non-numeric parts or original NaNs will result in NaN here.

numeric_part = df['User_Id'].str.extract(r'(\d+)', expand=False)

Identify the indices where numeric_part is NaN

nan_indices = numeric_part.isnull()

Social Media

```
# Calculate how many random numbers are needed
num_random_ids = nan_indices.sum()

# Generate random numbers for these positions
random_ids = np.random.randint(1, 1000, size=num_random_ids)

# Assign these random numbers (as strings) to the NaN positions in numeric_part
numeric_part.loc[nan_indices] = random_ids.astype(str)

# Convert the entire Series to integer
df['User_Id'] = numeric_part.astype(int)

# Prepend 'User_id ' and convert back to string
df['User_Id'] = 'User_id ' + df['User_Id'].astype(str)
df
```

	User_Id	Movie_Name	Movie_Review	Release_Date	Sentiment	Duration	Platform	Timestamp	Rating	Likes	weekday	Month_Name	Rating_Review
0	User_id 122	Lagaan	Loved the acting and story	2001-12-16	Positive	120 Min	Youtube	2014-01-20 17:21:21	4.156667	25.0	Monday	January	Average
1	User_id 355	Bahubali	Excellent visuals but slow	2000-12-30	Negative	120 Min	Youtube	2024-09-13 02:56:38	6.000000	3.0	Friday	September	Average
2	User_id 333	Rrr	Amazing movie!! must watch	1991-12-21	Positive	150 Min	Instagram	1990-12-14 09:58:27	7.500000	0.0	Friday	December	Average
3	User_id 938	Lagaan	It was okay... nothing special	2005-05-29	Very negative	180 Min	Instagram	2017-04-25 20:33:36	5.000000	3.0	Tuesday	April	Average
4	User_id	Pushpa	Loved the acting and	2020-09-20	Very	90 Min	Youtube	2013-05-09	10.000000	3.0	Thursday	May	Hit

Terminal

Creating new columns

Using Rating _Review Divided rating into three categories below 4 rating Flop,above to 8 average, and above 8 Hit

And created weekday and Month Name by using Time stamp

```
df["weekday"] = df["Timestamp"].dt.day_name()
df["Month_Name"] = df["Timestamp"].dt.month_name()
df["Rating_Review"] = df["Rating"].apply(
    lambda x: "Flop" if x < 4 else "Average" if x <= 8 else "Hit"
)
df
```

Social Media

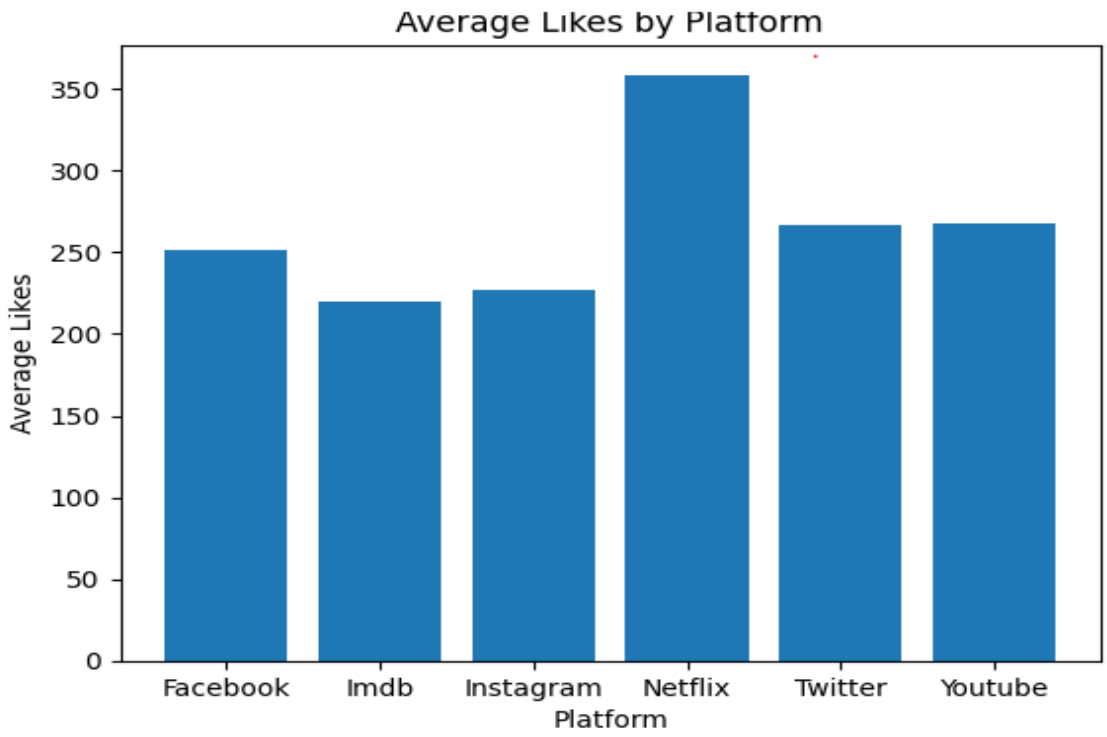
›vie_Review	Release_Date	Sentiment	Duration	Platform	Timestamp	Rating	Likes	Weekday	Month_Name	Rating_Review	Day_Category
Loved the acting and story	2001-12-16	Positive	120 Min	Youtube	2014-01-20 17:21:21	4.156667	25.0	Monday	January	Average	Weekday
Excellent visuals but slow	2000-12-30	Negative	120 Min	Youtube	2024-09-13 02:56:38	6.000000	3.0	Friday	September	Average	Weekday
Amazing movie!! must watch	1991-12-21	Positive	150 Min	Instagram	1990-12-14 09:58:27	7.500000	0.0	Friday	December	Average	Weekday
It was okay... nothing special	2005-05-29	Very negative	180 Min	Instagram	2017-04-25 20:33:36	5.000000	3.0	Tuesday	April	Average	Weekday

Visuvalization Part:

Bar chart:

The bar chart shows the average likes by platform, with platforms on the x-axis and average likes on the y-axis. It highlights differences in engagement, showing which platforms receive more user interaction compared to others.

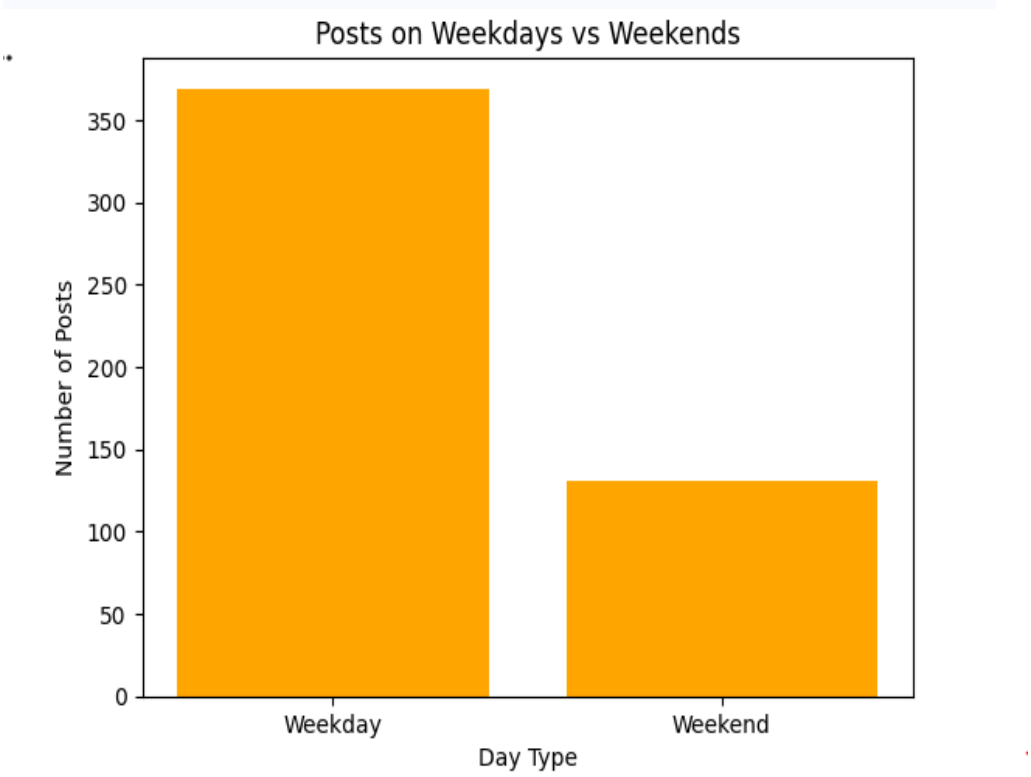
```
import matplotlib.pyplot as plt
avg_likes = df.groupby("Platform")["Likes"].mean()

plt.bar(avg_likes.index, avg_likes.values)
plt.xlabel("Platform")
plt.ylabel("Average Likes")
plt.title("Average Likes by Platform")
plt.show()
```



Social Media

This bar chart comparing the number of posts made on weekdays versus weekends.First, a mapping is defined to classify each day of the week as either a Weekday or Weekend. Using this mapping, a new column called Day_Category is added to the DataFrame. The code then counts how many posts fall into each category.Finally, a bar chart is plotted where the x-axis shows the day type (Weekday or Weekend) and the y-axis shows the number of posts. The chart helps visually compare posting activity between weekdays and weekends.

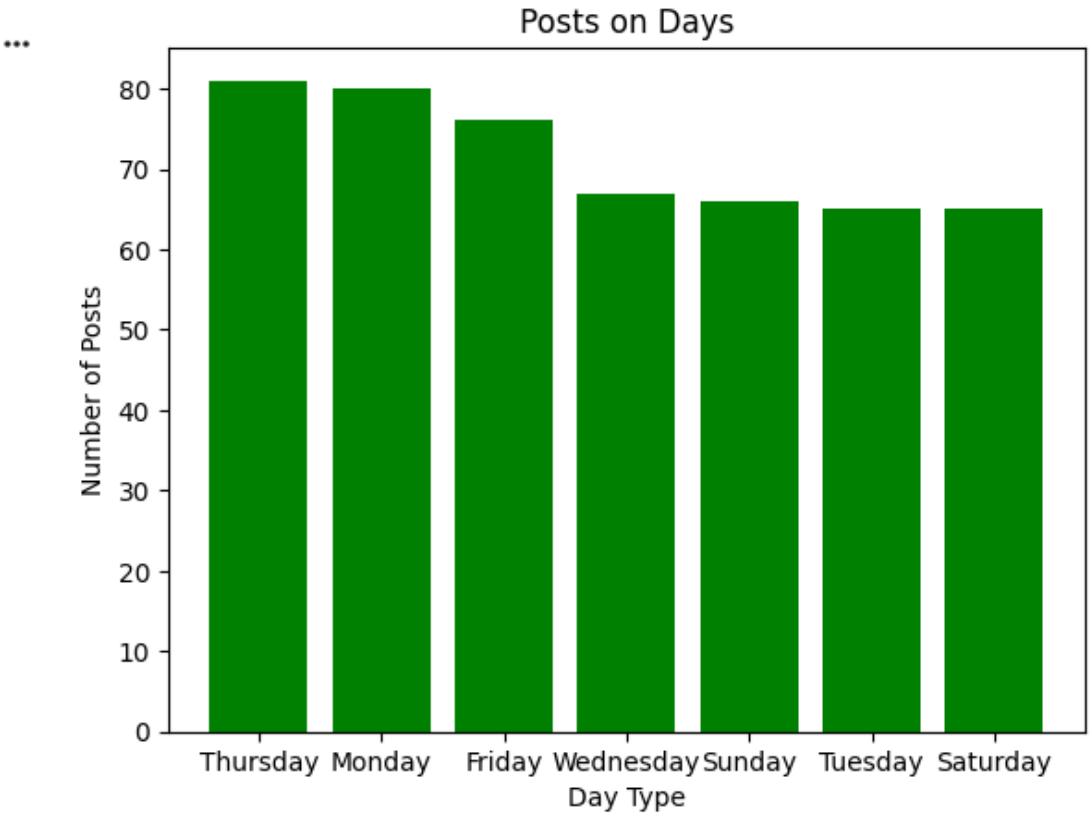
```
import matplotlib.pyplot as plt

weekday_map = {
    'Monday': 'Weekday',
    'Tuesday': 'Weekday',
    'Wednesday': 'Weekday',
    'Thursday': 'Weekday',
    'Friday': 'Weekday',
    'Saturday': 'Weekend',
    'Sunday': 'Weekend'
}

df['Day_Category'] = df['weekday'].map(weekday_map)
Day_counts = df['Day_Category'].value_counts()

plt.figure()
plt.bar(Day_counts.index, Day_counts.values, color="orange")
plt.xlabel("Day Type")
plt.ylabel("Number of Posts")
plt.title("Posts on Weekdays vs Weekends")
plt.show()
```



Social Media

This bar chart showing the number of posts for each day of the week.The data is counted by weekday to see posting frequency.The x-axis represents the days, and the y-axis shows the number of posts.It helps identify which days have higher or lower posting activity.

```
import matplotlib.pyplot as plt
Day_counts = df["weekday"].value_counts()

plt.figure()
plt.bar(Day_counts.index, Day_counts.values,color='green')
plt.xlabel("Day Type")
plt.ylabel("Number of Posts")
plt.title("Posts on Days")
plt.show()
```

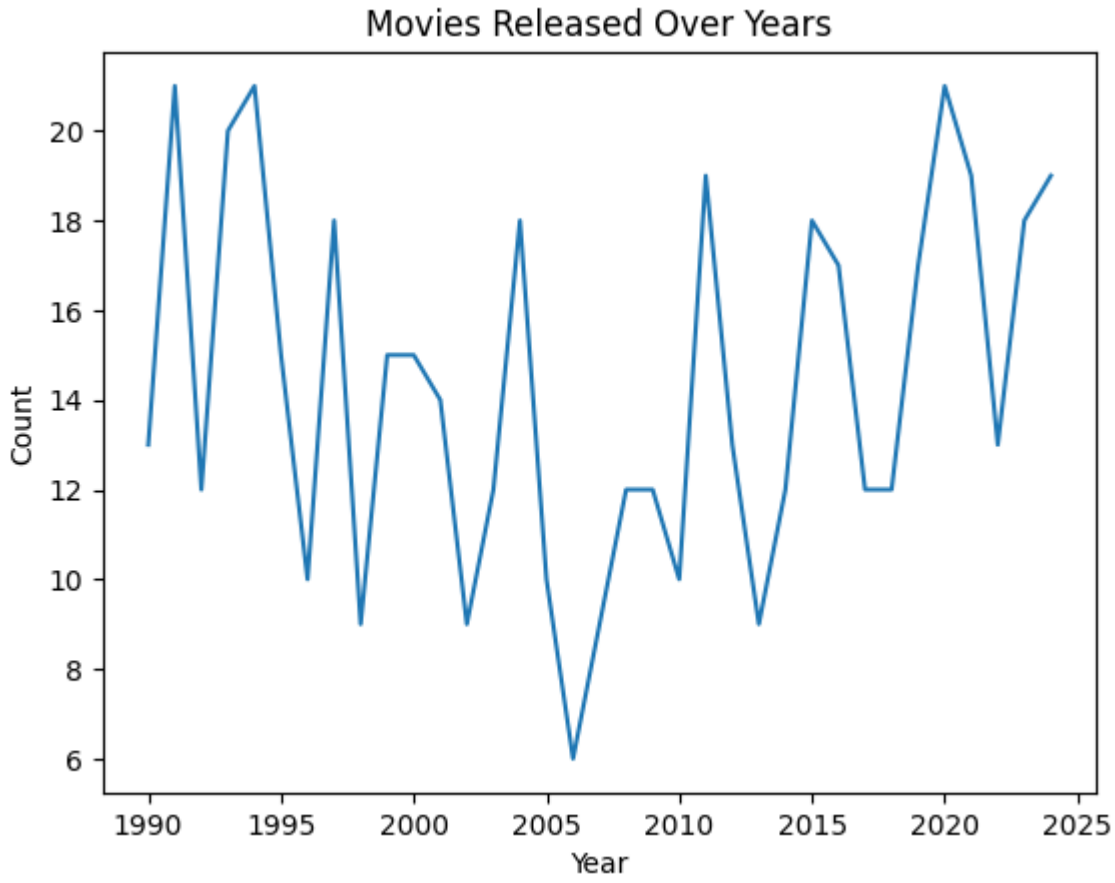


Line Chart:

The line chart is created by first converting the Release_Date column into a datetime format so the year can be extracted accurately. The data is then grouped by year, and the total number of movies released in each year is counted. These yearly counts are plotted on a line graph, with the year on the x-axis and the number of movies on the y-axis. The visualization helps identify growth, decline, or trends in movie production over time.

Social Media

```
df["Release_Date"] = pd.to_datetime(df["Release_Date"])
df.groupby(df["Release_Date"].dt.year).size().plot(kind="line")
plt.title("Movies Released Over Years")
plt.xlabel("Year")
plt.ylabel("Count")
plt.show()
```

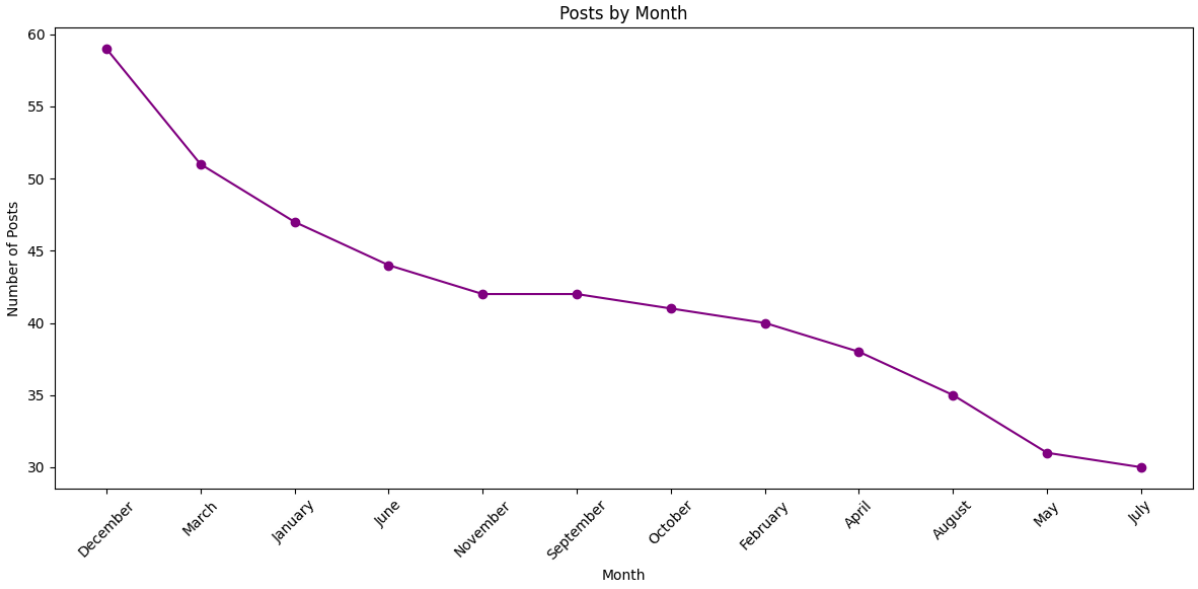


Line Chart :

line chart showing the number of posts for each month.First, the month name is extracted from the timestamp and stored in a new column. The data is then counted for each month to find how many posts occurred. These counts are plotted as a line graph with months on the x-axis and number of posts on the y-axis. The larger figure size and rotated labels improve readability, helping clearly identify monthly posting trends.

Social Media

```
df["Month_name"] = df["Timestamp"].dt.month_name()
Month_counts = df["Month_name"].value_counts()

plt.figure(figsize=(12, 6))   # expand chart size
plt.plot(Month_counts.index, Month_counts.values, color="purple", marker="o")
plt.xlabel("Month")
plt.ylabel("Number of Posts")
plt.title("Posts by Month")
plt.xticks(rotation=45)       # rotate month names
plt.tight_layout()            # adjust spacing
plt.show()
```

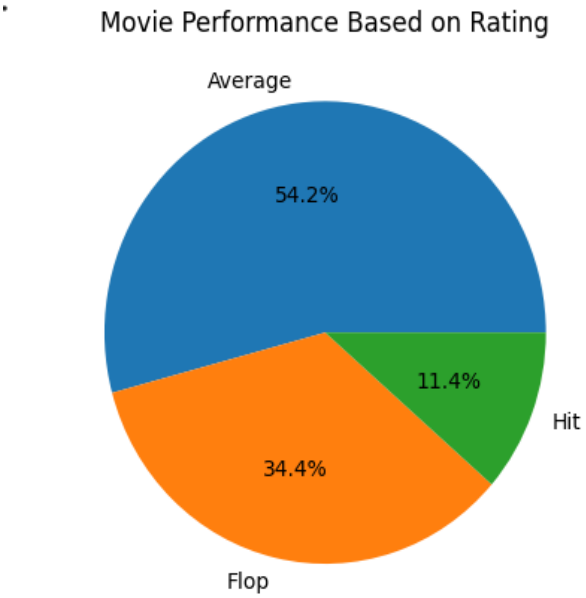


Pie Chart:

This pie chart showing movie performance based on ratings.First, the Rating_Review column is counted to find how many movies fall into each category (such as Hit, Average, or Flop). These counts are then plotted as a pie chart, where each slice represents a category and its percentage of the total. The chart makes it easy to compare the overall distribution of movie performance at a glance.

```
counts = df["Rating_Review"].value_counts()

plt.figure()
plt.pie(counts.values, labels=counts.index, autopct="%1.1f%%")
plt.title("Movie Performance Based on Rating")
plt.show()
```

Social Media

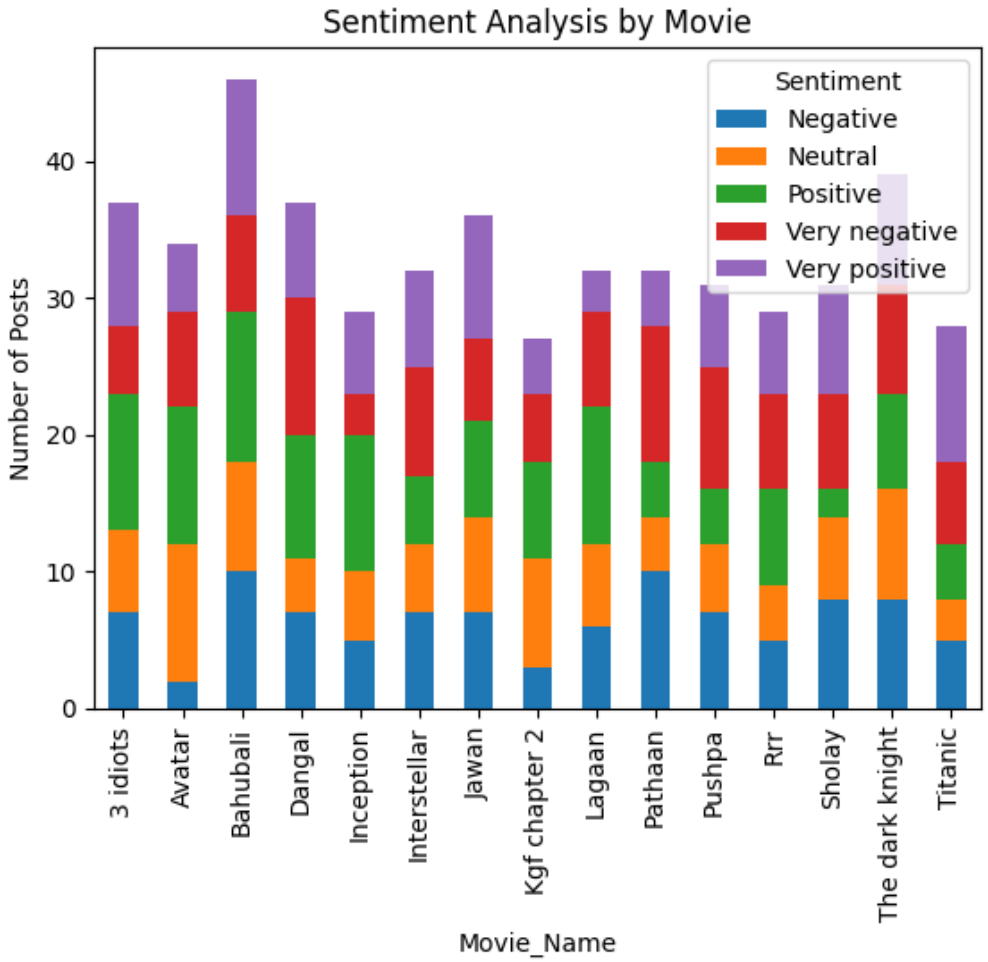


Bar Chart:

This code creates a stacked bar chart showing sentiment distribution for each movie.First, the data is grouped by Movie_Name and Sentiment, and the number of posts for each sentiment is counted. The results are reshaped so each sentiment becomes a separate bar segment. The stacked bars display how positive, negative, and neutral sentiments contribute to the total posts for each movie, making it easy to compare audience sentiment across movies.

```
Sentiment_movie = df.groupby(["Movie_Name", "Sentiment"]).size().unstack()

Sentiment_movie.plot(kind="bar", stacked=True)
plt.xlabel("Movie_Name")
plt.ylabel("Number of Posts")
plt.title("Sentiment Analysis by Movie")
plt.show()
```

Social Media



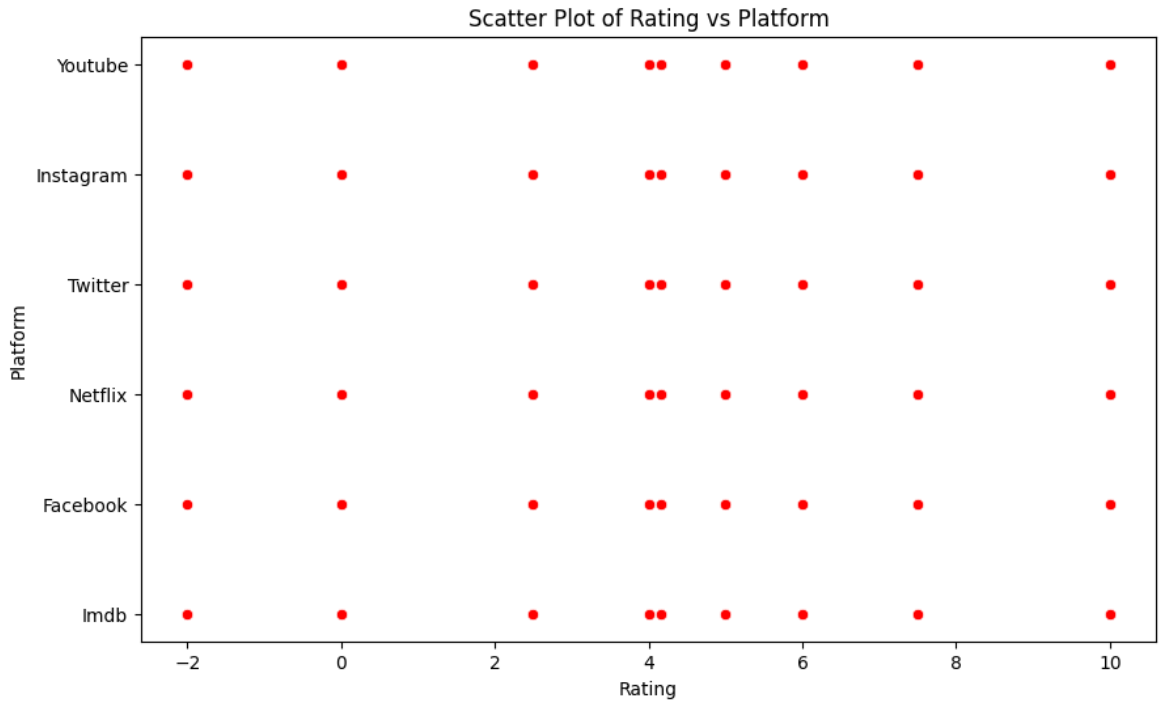
Using Seaborn :

Scatter plot:

This code creates a scatter plot showing the relationship between movie ratings and platforms.The ratings are plotted on the x-axis, while the platforms appear on the y-axis, with each point representing a movie or post. The scatter plot helps visualize how ratings are distributed across different platforms and whether certain platforms tend to have higher or lower ratings.

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))

sns.scatterplot(x=df["Rating"], y=df["Platform"],color="Red")
plt.xlabel("Rating")
plt.ylabel("Platform")
plt.title("Scatter Plot of Rating vs Platform")
plt.show()
```

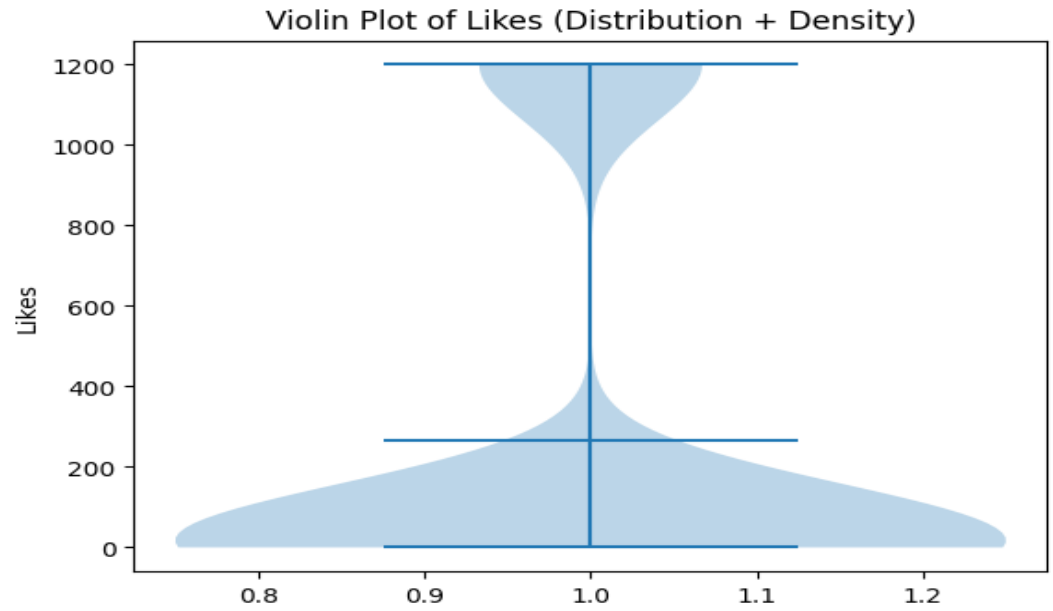
Social Media



Violin plot:

This code creates a violin plot of likes, showing how they are distributed.The width of the violin represents the density of likes at each value.A marker shows the mean, helping visualize the average and spread of likes.

```
plt.violinplot(df["Likes"].dropna(), showmeans=True)
plt.title("Violin Plot of Likes (Distribution + Density)")
plt.ylabel("Likes")
plt.show()
```



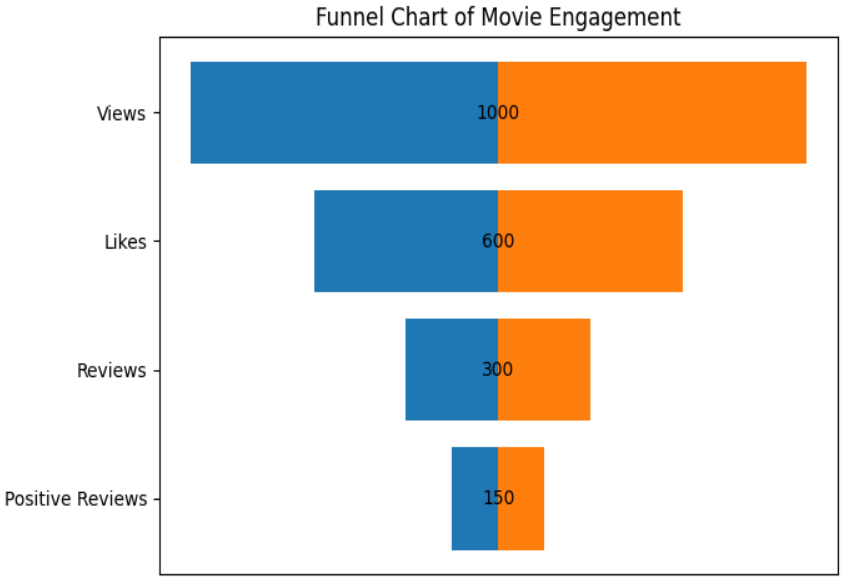
Social Media

Funnel chart:

Funnel chart showing movie engagement at different stages.

The stages—Views, Likes, Reviews, and Positive Reviews—are plotted vertically, with bar widths representing the number of users at each stage.It visualizes how engagement decreases from top to bottom, highlighting the drop-off at each stage

```
import matplotlib.pyplot as plt

stages = ["Views", "Likes", "Reviews", "Positive Reviews"]
values = [1000, 600, 300, 150]

# Reverse for top-down funnel
stages = stages[::-1]
values = values[::-1]

left = [-v for v in values]
right = values

plt.figure(figsize=(7,5))
plt.barh(stages, left)
plt.barh(stages, right)

for i, v in enumerate(values):
    plt.text(0, i, str(v), ha='center', va='center')

plt.title("Funnel Chart of Movie Engagement")
plt.xticks([])
plt.show()
```

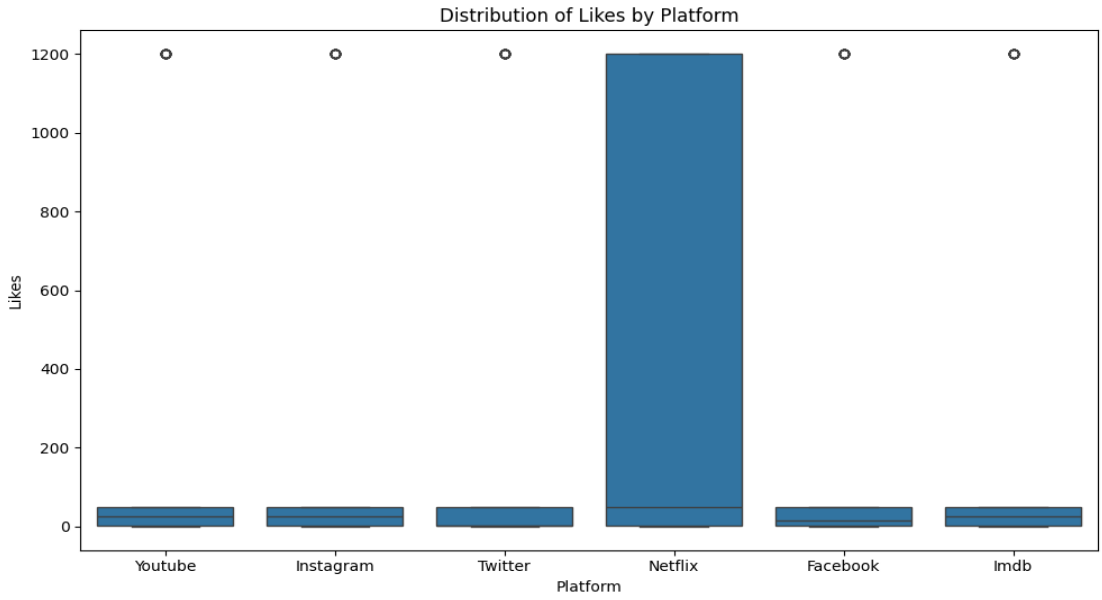


Social Media

Box plot:

Boxplot showing the distribution of likes across different platforms. Each box represents the spread of likes on a platform, including median, quartiles, and potential outliers.It helps compare how likes vary between platforms and identify which platforms get higher or lower engagement

```
import seaborn as sns
import matplotlib.pyplot as plt

clean_df = df[df["Likes"] >= 0]
plt.figure(figsize=(10, 6))
sns.boxplot(data=clean_df, x="Platform", y="Likes")
plt.title("Distribution of Likes by Platform")
plt.xlabel("Platform")
plt.ylabel("Likes")
plt.tight_layout()
plt.show()
```



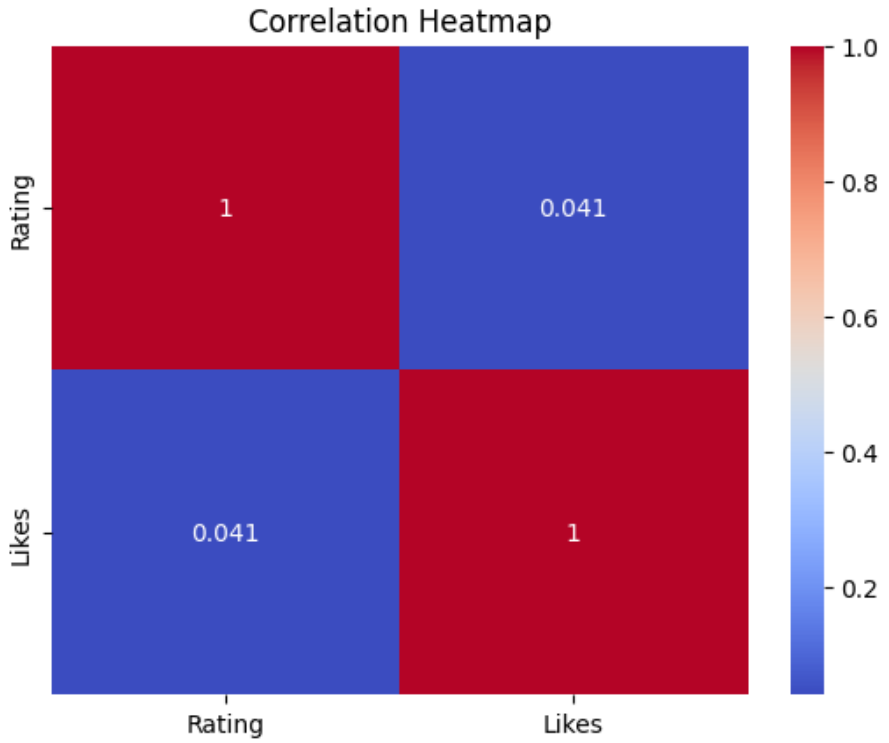
Heat map :

This heatmap showing the correlation between ratings and likes.The colors indicate the strength and direction of the relationship, with positive correlations in warm colors and negative in cool colors.

It helps visualize whether higher ratings are associated with more likes.

Social Media

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(df[["Rating", "Likes"]].corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```



KEY INSIGHTS:

Analysis Performed Insights were generated on sentiment distribution, most liked movies, rating trends, weekday vs weekend viewing behaviour and month-wise like comparison based on release dates.

1. Sentiment Trends

Most movie reviews fall under positive and neutral sentiment, indicating overall audience satisfaction.Movies with higher positive sentiment generally received higher ratings and more likes.Negative sentiment was more common for movies with lower ratings or unmet audience expectations.

2. Movie Popularity Based on Likes

Certain movies received significantly higher likes, showing strong audience engagement.Popular movies tend to attract more reviews, increasing visibility and interaction.Likes are a strong indicator of audience approval and popularity.

3. Ratings vs Sentiment Relationship

Higher ratings (4–5) are closely associated with positive sentiment reviews.Lower ratings (1–2) are mostly linked with negative sentiment.This confirms that sentiment analysis aligns well with user ratings.

Social Media

4.Weekday vs Weekend Engagement

User activity and engagement are higher on weekends compared to weekdays.More reviews and likes are observed during weekends, indicating higher viewership.This suggests that audiences prefer watching and reviewing movies during free time.

5.Month-Wise Performance

Movies released during holiday or festive months gained more likes and reviews.

Early release periods usually show peak engagement, which slowly stabilizes over time.

Release timing plays an important role in audience reach.

6. Duration Impact

Movies with moderate duration received better engagement than very long or very short movies.Extremely long durations may reduce viewer interest, reflected in lower likes.

7.Data Quality & Cleaning Importance

Raw data contained missing values, inconsistent text, and invalid ratings.Data cleaning significantly improved analysis accuracy.This highlights the importance of preprocessing before analysis.

Overall Insight

Combining sentiment analysis, ratings, likes, and time-based analysis provides meaningful insights into audience behavior.Data analytics helps understand what kind of movies audiences prefer and when they are most engaged.