Movie Recommendation System

Priyadharshini.S

Data Analyst

Python

Dataset-1 Loaded

```
import pandas as pd

dataframe_1=pd.read_csv(r"C:\Users\s.sathishkumar\Downloads\movies (1).csv")
dataframe_2= pd.read_csv(r"C:\Users\s.sathishkumar\Downloads\ratings.csv\ratings.csv")
dataframe_1
# dataframe_2
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
9737	193581	Black Butler: Book of the Atlantic (2017)	Action|Animation|Comedy|Fantasy
9738	193583	No Game No Life: Zero (2017)	Animation|Comedy|Fantasy
9739	193585	Flint (2017)	Drama
9740	193587	Bungo Stray Dogs: Dead Apple (2018)	Action|Animation
9741	193609	Andrew Dice Clay: Dice Rules (1991)	Comedy

9742 rows × 3 columns

Dataset-2 Loaded

[1]:
```
import pandas as pd

dataframe_1=pd.read_csv(r"C:\Users\s.sathishkumar\Downloads\movies (1).csv")
dataframe_2= pd.read_csv(r"C:\Users\s.sathishkumar\Downloads\ratings.csv\ratings.csv")
# dataframe_1
dataframe_2
```

[1]:

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931
...	...	...	...	...
100831	610	166534	4.0	1493848402
100832	610	168248	5.0	1493850091
100833	610	168250	5.0	1494273047
100834	610	168252	5.0	1493846352
100835	610	170875	3.0	1493846415

100836 rows × 4 columns

Cleaning Title

```
#cleaning movie titles with regex
import re

def clean_title(title):
    return re.sub("[^a-zA-Z0-9]"," ",title)
```

```
dataframe_1["clean_title"] = dataframe_1["title"].apply(clean_title)
```

```
dataframe_1
```

	movieId	title	genres	clean_title
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995
1	2	Jumanji (1995)	Adventure|Children|Fantasy	Jumanji 1995
2	3	Grumpier Old Men (1995)	Comedy|Romance	Grumpier Old Men 1995
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	Waiting to Exhale 1995
4	5	Father of the Bride Part II (1995)	Comedy	Father of the Bride Part II 1995
...	...	...	...	...
9737	193581	Black Butler: Book of the Atlantic (2017)	Action|Animation|Comedy|Fantasy	Black Butler Book of the Atlantic 2017
9738	193583	No Game No Life: Zero (2017)	Animation|Comedy|Fantasy	No Game No Life Zero 2017
9739	193585	Flint (2017)	Drama	Flint 2017
9740	193587	Bungo Stray Dogs: Dead Apple (2018)	Action|Animation	Bungo Stray Dogs Dead Apple 2018
9741	193609	Andrew Dice Clay: Dice Rules (1991)	Comedy	Andrew Dice Clay Dice Rules 1991

9742 rows × 4 columns

Merging Two Tables

```
df = dataframe_1.merge(dataframe_2, on= "movieId", how='inner')
```

```
df
```

	movieId	title	genres	clean_title	userId	rating	timestamp
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	1	4.0	964982703
1	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	5	4.0	847434962
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	7	4.5	1106635946
3	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	15	2.5	1510577970
4	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	17	4.5	1305696483
...	...	...	...	...	...	...	...
100831	193581	Black Butler: Book of the Atlantic (2017)	Action|Animation|Comedy|Fantasy	Black Butler Book of the Atlantic 2017	184	4.0	1537109082
100832	193583	No Game No Life: Zero (2017)	Animation|Comedy|Fantasy	No Game No Life Zero 2017	184	3.5	1537109545
100833	193585	Flint (2017)	Drama	Flint 2017	184	3.5	1537109805
100834	193587	Bungo Stray Dogs: Dead Apple (2018)	Action|Animation	Bungo Stray Dogs Dead Apple 2018	184	3.5	1537110021
100835	193609	Andrew Dice Clay: Dice Rules (1991)	Comedy	Andrew Dice Clay Dice Rules 1991	331	4.0	1537157606

100836 rows × 7 columns

Extracting title and year

```
df[['title', 'year']] = df['clean_title'].str.extract(r'(.+)\s(\d{4})')
```

```
df
```

	movieId	title	genres	clean_title	userId	rating	timestamp	year
0	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	1	4.0	964982703	1995
1	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	5	4.0	847434962	1995
2	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	7	4.5	1106635946	1995
3	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	15	2.5	1510577970	1995
4	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995	17	4.5	1305696483	1995
...	...	...	...	...	...	...	...	...
100831	193581	Black Butler Book of the Atlantic	Action|Animation|Comedy|Fantasy	Black Butler Book of the Atlantic 2017	184	4.0	1537109082	2017
100832	193583	No Game No Life Zero	Animation|Comedy|Fantasy	No Game No Life Zero 2017	184	3.5	1537109545	2017
100833	193585	Flint	Drama	Flint 2017	184	3.5	1537109805	2017
100834	193587	Bungo Stray Dogs Dead Apple	Action|Animation	Bungo Stray Dogs Dead Apple 2018	184	3.5	1537110021	2018
100835	193609	Andrew Dice Clay Dice Rules	Comedy	Andrew Dice Clay Dice Rules 1991	331	4.0	1537157606	1991

100836 rows × 8 columns

Extracting title and year

```
dataframe = df.drop(columns=['clean_title'])
```

```
dataframe
```

	movieId	title	genres	userId	rating	timestamp	year
0	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	1	4.0	964982703	1995
1	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	5	4.0	847434962	1995
2	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	7	4.5	1106635946	1995
3	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	15	2.5	1510577970	1995
4	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	17	4.5	1305696483	1995
...	...	...	...	...	...	...	...
100831	193581	Black Butler Book of the Atlantic	Action|Animation|Comedy|Fantasy	184	4.0	1537109082	2017
100832	193583	No Game No Life Zero	Animation|Comedy|Fantasy	184	3.5	1537109545	2017
100833	193585	Flint	Drama	184	3.5	1537109805	2017
100834	193587	Bungo Stray Dogs Dead Apple	Action|Animation	184	3.5	1537110021	2018
100835	193609	Andrew Dice Clay Dice Rules	Comedy	331	4.0	1537157606	1991

100836 rows × 7 columns

Converting datatype for timestamp

```
# convert timestamp to datetime
dataframe['timestamp']=pd.to_datetime(dataframe['timestamp'],unit= 's')
```

```
dataframe
```

	movieId	title	genres	userId	rating	timestamp	year
0	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	1	4.0	2000-07-30 18:45:03	1995
1	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	5	4.0	1996-11-08 06:36:02	1995
2	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	7	4.5	2005-01-25 06:52:26	1995
3	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	15	2.5	2017-11-13 12:59:30	1995
4	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy	17	4.5	2011-05-18 05:28:03	1995
...	...	...	...	...	...	...	...
100831	193581	Black Butler Book of the Atlantic	Action|Animation|Comedy|Fantasy	184	4.0	2018-09-16 14:44:42	2017
100832	193583	No Game No Life Zero	Animation|Comedy|Fantasy	184	3.5	2018-09-16 14:52:25	2017
100833	193585	Flint	Drama	184	3.5	2018-09-16 14:56:45	2017
100834	193587	Bungo Stray Dogs Dead Apple	Action|Animation	184	3.5	2018-09-16 15:00:21	2018
100835	193609	Andrew Dice Clay Dice Rules	Comedy	331	4.0	2018-09-17 04:13:26	1991

100836 rows × 7 columns

Replacing (|) to (,) in genres

```
dataframe['genres'] = dataframe['genres'].str.replace('|', ',')
dataframe
```

	movieId	title	genres	userId	rating	timestamp	year
0	1	Toy Story	Adventure,Animation,Children,Comedy,Fantasy	1	4.0	2000-07-30 18:45:03	1995
1	1	Toy Story	Adventure,Animation,Children,Comedy,Fantasy	5	4.0	1996-11-08 06:36:02	1995
2	1	Toy Story	Adventure,Animation,Children,Comedy,Fantasy	7	4.5	2005-01-25 06:52:26	1995
3	1	Toy Story	Adventure,Animation,Children,Comedy,Fantasy	15	2.5	2017-11-13 12:59:30	1995
4	1	Toy Story	Adventure,Animation,Children,Comedy,Fantasy	17	4.5	2011-05-18 05:28:03	1995
...	...	...	...	...	...	...	...
100831	193581	Black Butler Book of the Atlantic	Action,Animation,Comedy,Fantasy	184	4.0	2018-09-16 14:44:42	2017
100832	193583	No Game No Life Zero	Animation,Comedy,Fantasy	184	3.5	2018-09-16 14:52:25	2017
100833	193585	Flint	Drama	184	3.5	2018-09-16 14:56:45	2017
100834	193587	Bungo Stray Dogs Dead Apple	Action,Animation	184	3.5	2018-09-16 15:00:21	2018
100835	193609	Andrew Dice Clay Dice Rules	Comedy	331	4.0	2018-09-17 04:13:26	1991

100836 rows × 7 columns

Ordered the Columns

```
column_order=['movieId','userId', 'title', 'year','rating', 'genres', 'timestamp']
df_ = dataframe[column_order]
df_
```

	movieId	userId	title	year	rating	genres	timestamp
0	1	1	Toy Story	1995	4.0	Adventure,Animation,Children,Comedy,Fantasy	2000-07-30 18:45:03
1	1	5	Toy Story	1995	4.0	Adventure,Animation,Children,Comedy,Fantasy	1996-11-08 06:36:02
2	1	7	Toy Story	1995	4.5	Adventure,Animation,Children,Comedy,Fantasy	2005-01-25 06:52:26
3	1	15	Toy Story	1995	2.5	Adventure,Animation,Children,Comedy,Fantasy	2017-11-13 12:59:30
4	1	17	Toy Story	1995	4.5	Adventure,Animation,Children,Comedy,Fantasy	2011-05-18 05:28:03
...	...	...	...	...	...	...	...
100831	193581	184	Black Butler Book of the Atlantic	2017	4.0	Action,Animation,Comedy,Fantasy	2018-09-16 14:44:42
100832	193583	184	No Game No Life Zero	2017	3.5	Animation,Comedy,Fantasy	2018-09-16 14:52:25
100833	193585	184	Flint	2017	3.5	Drama	2018-09-16 14:56:45
100834	193587	184	Bungo Stray Dogs Dead Apple	2018	3.5	Action,Animation	2018-09-16 15:00:21
100835	193609	331	Andrew Dice Clay Dice Rules	1991	4.0	Comedy	2018-09-17 04:13:26

100836 rows × 7 columns

Checking for Duplicated, is null and dropping the Null

```
duplicates= df_.duplicated().sum()
print(duplicates)
```

```
0
```

```
null_values= df_.isnull().sum()
dataset=df_.dropna()
dataset
```

	movieId	userId	title	year	rating	genres	timestamp
0	1	1	Toy Story	1995	4.0	Adventure,Animation,Children,Comedy,Fantasy	2000-07-30 18:45:03
1	1	5	Toy Story	1995	4.0	Adventure,Animation,Children,Comedy,Fantasy	1996-11-08 06:36:02
2	1	7	Toy Story	1995	4.5	Adventure,Animation,Children,Comedy,Fantasy	2005-01-25 06:52:26
3	1	15	Toy Story	1995	2.5	Adventure,Animation,Children,Comedy,Fantasy	2017-11-13 12:59:30
4	1	17	Toy Story	1995	4.5	Adventure,Animation,Children,Comedy,Fantasy	2011-05-18 05:28:03
...	...	...	...	...	...	...	...
100831	193581	184	Black Butler Book of the Atlantic	2017	4.0	Action,Animation,Comedy,Fantasy	2018-09-16 14:44:42
100832	193583	184	No Game No Life Zero	2017	3.5	Animation,Comedy,Fantasy	2018-09-16 14:52:25
100833	193585	184	Flint	2017	3.5	Drama	2018-09-16 14:56:45
100834	193587	184	Bungo Stray Dogs Dead Apple	2018	3.5	Action,Animation	2018-09-16 15:00:21
100835	193609	331	Andrew Dice Clay Dice Rules	1991	4.0	Comedy	2018-09-17 04:13:26

100819 rows × 7 columns

Outliers

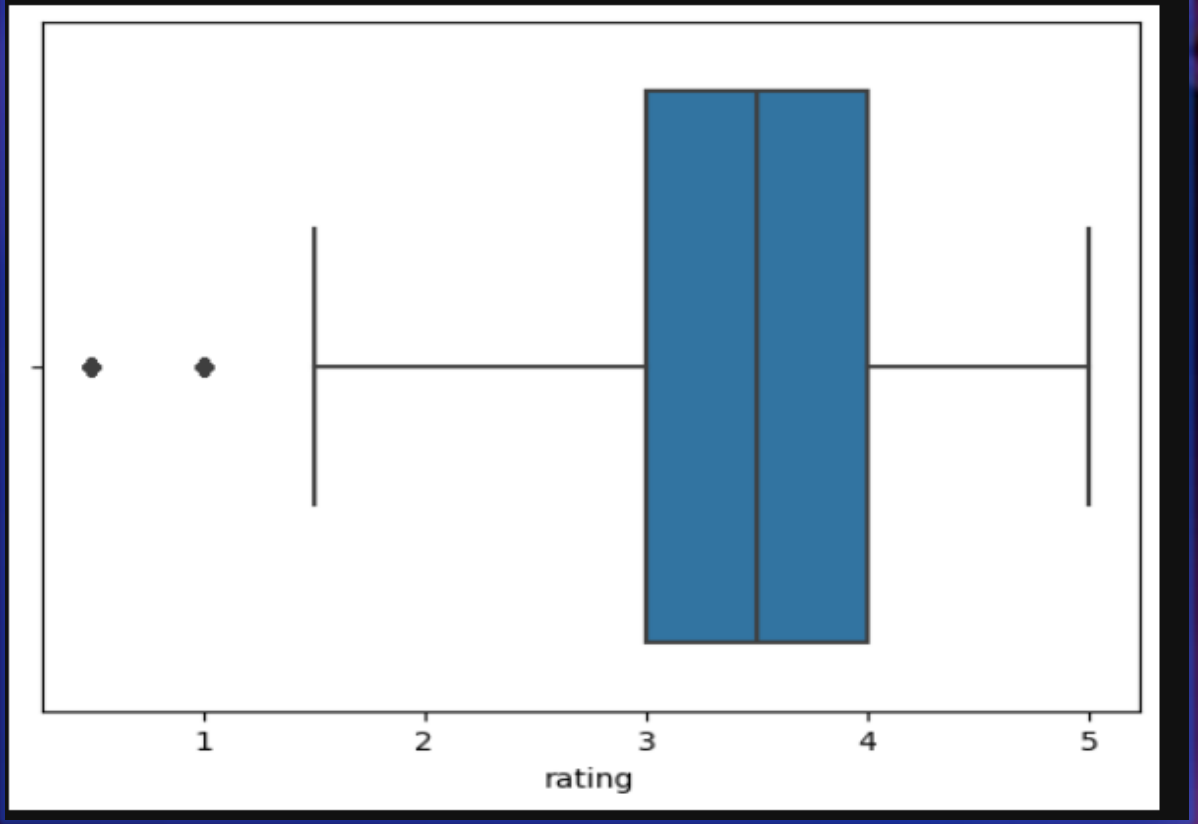
```
clean= dataset.isnull().sum()
clean
```

```
movieId      0
userId       0
title        0
year         0
rating       0
genres       0
timestamp    0
dtype: int64
```

```
dataset.describe()
```

	movieId	userId	rating	timestamp
count	100819.000000	100819.000000	100819.000000	100819
mean	19414.421538	326.126524	3.501547	2008-03-19 02:46:04.222170624
min	1.000000	1.000000	0.500000	1996-03-29 18:36:55
25%	1199.000000	177.000000	3.000000	2002-04-13 16:48:00.500000
50%	2991.000000	325.000000	3.500000	2007-08-02 20:28:46
75%	8044.000000	477.000000	4.000000	2015-07-04 07:10:05.500000
max	193609.000000	610.000000	5.000000	2018-09-24 14:27:30
std	35493.882763	182.620532	1.042474	NaN

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

#visualizing outliers using a boxplot
sns.boxplot(x=dataset['rating'])
plt.show()
```



IQR Calculation

```python
Q1=dataset['rating'].quantile(0.25)
Q3=dataset['rating'].quantile(0.75)

IQR= Q3 - Q1
#define outlier thresholds
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

#filter the outliers
outliers=dataset[(dataset['rating'] < lower_bound) | (dataset['rating'] > upper_bound)]
print(outliers)
```

```
        movieId  userId                                              title  year  \
26            1      76                                          Toy Story  1995
239           2     149                                            Jumanji  1995
261           2     298                                            Jumanji  1995
345           3     217                                   Grumpier Old Men  1995
352           3     294                                   Grumpier Old Men  1995
...         ...     ...                                                ...   ...
100793   187593     338                                         Deadpool 2  2018
100816   189547     210                                       Iron Soldier  2010
100821   190213     338                                          John From  2015
100823   190219     338                                              Bunny  1998
100824   190221     338  Hommage    Zgougou  et salut    Sabine Mamou      2002

        rating                                             genres  \
26         0.5  Adventure,Animation,Children,Comedy,Fantasy
239        1.0                    Adventure,Children,Fantasy
261        0.5                    Adventure,Children,Fantasy
345        1.0                              Comedy,Romance
352        1.0                              Comedy,Romance
...        ...                                            ...
100793     1.0                    Action,Comedy,Sci-Fi
100816     1.0                           Action,Sci-Fi
100821     1.0                                   Drama
100823     1.0                               Animation
100824     1.0                             Documentary

                  timestamp
26      2015-08-10 00:12:28
239     1998-08-02 19:07:54
261     2015-12-18 15:34:57
345     2000-04-17 04:11:53
352     2000-08-18 11:07:34
...                     ...
```

Extracting the rows

```
outliers=dataset[(dataset['rating'] >= lower_bound) & (dataset['rating'] <= upper_bound)]
print(outliers)
```

```
        movieId  userId                                  title  year  rating  \
0             1       1                              Toy Story  1995     4.0
1             1       5                              Toy Story  1995     4.0
2             1       7                              Toy Story  1995     4.5
3             1      15                              Toy Story  1995     2.5
4             1      17                              Toy Story  1995     4.5
...         ...     ...                                    ...   ...     ...
100831   193581     184  Black Butler  Book of the Atlantic   2017     4.0
100832   193583     184                No Game No Life  Zero   2017     3.5
100833   193585     184                                 Flint  2017     3.5
100834   193587     184        Bungo Stray Dogs  Dead Apple    2018     3.5
100835   193609     331        Andrew Dice Clay  Dice Rules    1991     4.0

                                       genres            timestamp
0       Adventure,Animation,Children,Comedy,Fantasy 2000-07-30 18:45:03
1       Adventure,Animation,Children,Comedy,Fantasy 1996-11-08 06:36:02
2       Adventure,Animation,Children,Comedy,Fantasy 2005-01-25 06:52:26
3       Adventure,Animation,Children,Comedy,Fantasy 2017-11-13 12:59:30
4       Adventure,Animation,Children,Comedy,Fantasy 2011-05-18 05:28:03
...                                            ...                 ...
100831              Action,Animation,Comedy,Fantasy 2018-09-16 14:44:42
100832                     Animation,Comedy,Fantasy 2018-09-16 14:52:25
100833                                      Drama 2018-09-16 14:56:45
100834                            Action,Animation 2018-09-16 15:00:21
100835                                     Comedy 2018-09-17 04:13:26

[96640 rows x 7 columns]
```

Exporting the File

```
outliers.to_csv(r'C:\Users\s.sathishkumar\Downloads\movie_recommendation_dataset.csv', index=False)
```

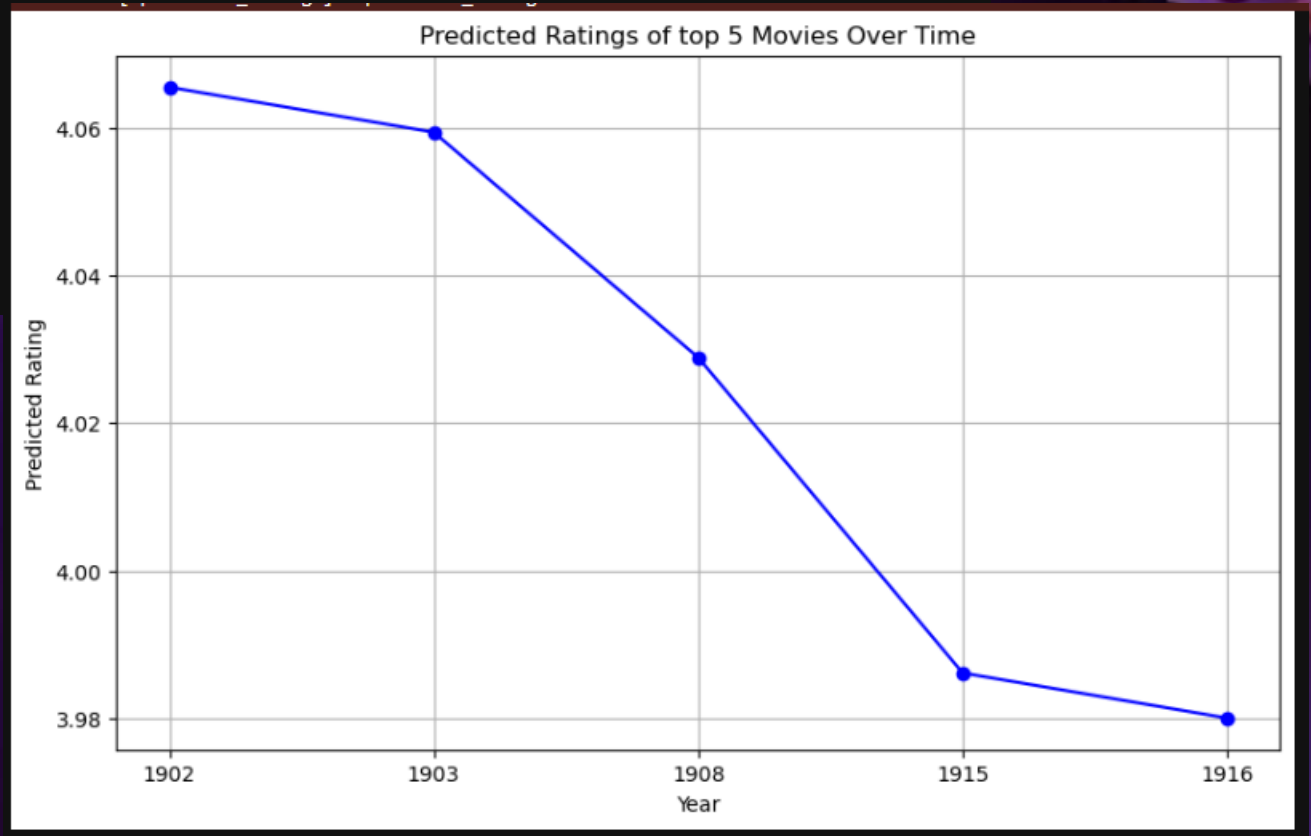
Linear Regression

```python
from sklearn.linear_model import LinearRegression
import numpy as np
import matplotlib.pyplot as plt

# Prepare the data for linear regression
# We will use 'year' as the independent variable and 'rating' as the dependent variable
X = dataset[['year']].values
Y = dataset['rating'].values

# Initialize the linear regression model
model = LinearRegression()

# Fit the model
model.fit(X, Y)

# Get the coefficients
slope = model.coef_[0]
intercept = model.intercept_

# Print the results
print('Slope:', slope)
print('Intercept:', intercept)

# Predict ratings using the linear regression model
predicted_ratings = model.predict(X)

dataset['predicted_rating'] = predicted_ratings

# Sort the dataframe by predicted ratings in descending order
sorted_dataset = dataset.sort_values(by='predicted_rating', ascending=False)

# Select the top five movies
top_five_recommendations = sorted_dataset[['title', 'year', 'predicted_rating']].drop_duplicates().head(5)

# Display the top five recommendations
print(top_five_recommendations)
```

```
Slope: -0.006099845545279404
Intercept: 15.667354474449867
                                                  title  year  predicted_rating
79586  Trip to the Moon  A  Voyage dans la lune  Le    1902          4.065448
84026                        The Great Train Robbery  1903          4.059348
99298                            The Electric Hotel   1908          4.028849
73484                          Birth of a Nation  The  1915          3.986150
88110                  20 000 Leagues Under the Sea   1916          3.980050
```

```python
from sklearn.linear_model import LinearRegression
import numpy as np
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.plot(top_five_recommendations['year'],top_five_recommendations['predicted_rating'], marker='o', linestyle='-', color='b')
plt.xlabel('Year')
plt.ylabel('Predicted Rating')
plt.title('Predicted Ratings of top 5 Movies Over Time')
plt.grid(True)
plt.show()
```

Line Chart



Power BI Dashboard

Page Navigation