

EDS ASSIGNMENT

IMBD DATASET

GOOGLE COLAB LINK -

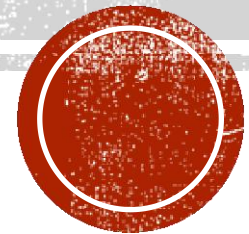
[HTTPS://1DRV.MS/B/C/E16635625F14A4FC/EDNKJQHKEFZPNPOCHVODWL6YBDINAS
D7W2-Z_Q05V74UXJG](https://1drv.ms/B/C/E16635625F14A4FC/EDNKJQHKEFZPNPOCHVODWL6YBDINASD7W2-Z_Q05V74UXJG)

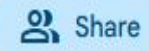
Name – SAKSHI PATIL

Roll no.- ET1-74

PRN – 202401070084

Division – ET1





Q Commands

+ Code

+ Text

Connect ▼



1. Average Rating of All Movies

```
import pandas as pd
data = {'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]}
df = pd.DataFrame(data)
avg_rating = df['Rating'].mean()
print("1. Average Rating of All Movies:", avg_rating)
```

```
1. Average Rating of All Movies: 7.76
```

2. Highest Rating

```
[ ] import pandas as pd
data = {'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]}
df = pd.DataFrame(data)
highest_rating = df['Rating'].max()
print("2. Highest Rating:", highest_rating)
```

```
2. Highest Rating: 9.0
```





Q Commands

+ Code + Text

Connect



3. Standard Deviation of Ratings

```
[ ] import pandas as pd
    data = {'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]}
    df = pd.DataFrame(data)
    std_rating = df['Rating'].std()
    print("3. Standard Deviation of Ratings:", std_rating)
```

```
↕ 3. Standard Deviation of Ratings: 0.9396807968666806
```

4. Count of Movies by Genre

```
[ ] import pandas as pd
    data = {'Genre': ['Drama', 'Action', 'Comedy', 'Action', 'Drama']}
    df = pd.DataFrame(data)
    genre_counts = df['Genre'].value_counts()
    print("4. Count of Movies by Genre:\n", genre_counts)
```

```
↕ 4. Count of Movies by Genre:
```

```
Genre
Drama    2
Action   2
Comedy    1
Name: count, dtype: int64
```





Untitled5.ipynb ☆ ☁

File Edit View Insert Runtime Tools Help



Share

◆ Gemini



Q Commands

+ Code + Text

Connect ▾



5. Average Rating for Each Movie

```
[ ] import pandas as pd
    data = {
        'Rating': [7.8, 8.2, 6.5, 7.3, 9.0],
        'Votes': [1000, 2000, 1500, 1800, 2200]
    }
    df = pd.DataFrame(data)
    df['Average_Rating'] = df[['Rating', 'Votes']].mean(axis=1)
    print("5. Average Rating for First 5 Movies:\n", df.head())
```

5. Average Rating for First 5 Movies:

| | Rating | Votes | Average_Rating |
|---|--------|-------|----------------|
| 0 | 7.8 | 1000 | 503.90 |
| 1 | 8.2 | 2000 | 1004.10 |
| 2 | 6.5 | 1500 | 753.25 |
| 3 | 7.3 | 1800 | 903.65 |
| 4 | 9.0 | 2200 | 1104.50 |





6. Movies with Rating Greater Than 8



```
import pandas as pd
data = {'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]}
df = pd.DataFrame(data)
high_rated_movies = df[df['Rating'] > 8]
print("6. Movies with Rating > 8:\n", high_rated_movies)
```



```
6. Movies with Rating > 8:
  Rating
1    8.2
4    9.0
```



7. Count of Movies by Director

```
[ ] import pandas as pd
data = {'Director': ['John', 'Jane', 'John', 'Paul', 'Jane']}
df = pd.DataFrame(data)
director_counts = df['Director'].value_counts()
print("7. Count of Movies by Director:\n", director_counts)
```





Q Commands | + Code + Text

Connect ▾ ^



7. Count of Movies by Director:

```
Director
John    2
Jane    2
Paul    1
Name: count, dtype: int64
```

8. Average Rating by Genre

```
[ ] import pandas as pd
data = {
    'Genre': ['Drama', 'Action', 'Comedy', 'Action', 'Drama'],
    'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]
}
df = pd.DataFrame(data)
avg_rating_by_genre = df.groupby('Genre')['Rating'].mean()
print("8. Average Rating by Genre:\n", avg_rating_by_genre)
```

```
8. Average Rating by Genre:
Genre
Action    7.75
Comedy    6.50
Drama     8.40
Name: Rating, dtype: float64
```





Q Commands

+ Code + Text

Connect ▾



9. Correlation Between Rating and Number of Votes

```
[ ] import pandas as pd
    data = {
        'Rating': [7.8, 8.2, 6.5, 7.3, 9.0],
        'Votes': [1000, 2000, 1500, 1800, 2200]
    }
    df = pd.DataFrame(data)
    correlation_rating_votes = df['Rating'].corr(df['Votes'])
    print("9. Correlation Between Rating and Number of Votes:", correlation_rating_votes)
```

↔ 9. Correlation Between Rating and Number of Votes: 0.5275106899913818

10. Movies with Rating of 10

```
[ ] import pandas as pd
    data = {'Rating': [7.8, 8.2, 6.5, 7.3, 10.0]}
    df = pd.DataFrame(data)
    topRated_movies = df[df['Rating'] == 10]
    print("10. Movies with Rating of 10:\n", topRated_movies)
```





Untitled5.ipynb ☆ ☁

File Edit View Insert Runtime Tools Help



Share



Gemini



Q Commands

+ Code + Text

Connect ▼



10. Movies with Rating of 10:

| | Rating |
|---|--------|
| 4 | 10.0 |



11. Movies Released in 2020



```
[ ] import pandas as pd
    data = {'Year': [2020, 2021, 2020, 2019, 2020]}
    df = pd.DataFrame(data)
    movies_2020 = df[df['Year'] == 2020]
    print("11. Movies Released in 2020:\n", movies_2020)
```



11. Movies Released in 2020:

| | Year |
|---|------|
| 0 | 2020 |
| 2 | 2020 |
| 4 | 2020 |





Untitled5.ipynb ☆ ☁

File Edit View Insert Runtime Tools Help



Share

◆ Gemini



Q Commands

+ Code + Text

Connect ▼



12. Count of Movies by Country



```
import pandas as pd
data = {'Country': ['USA', 'India', 'USA', 'UK', 'India']}
df = pd.DataFrame(data)
country_counts = df['Country'].value_counts()
print("12. Count of Movies by Country:\n", country_counts)
```



12. Count of Movies by Country:

Country

USA 2

India 2

UK 1

Name: count, dtype: int64





Q Commands

+ Code + Text

Connect ▾



13. Movie with Highest Rating

```
import pandas as pd
data = {
    'Rating': [7.8, 8.2, 6.5, 7.3, 9.0],
    'Title': ['Movie A', 'Movie B', 'Movie C', 'Movie D', 'Movie E']
}
df = pd.DataFrame(data)
highest Rated movie = df.loc[df['Rating'].idxmax(), 'Title']
print("13. Movie with Highest Rating:", highest Rated movie)
```

13. Movie with Highest Rating: Movie E

14. Add a 'Pass/Fail' Column Based on Rating

```
[ ] import pandas as pd
data = {
    'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]
}
df = pd.DataFrame(data)
df['Pass/Fail'] = np.where(df['Rating'] > 7, 'Pass', 'Fail')
print("14. Pass/Fail Status:\n", df[['Rating', 'Pass/Fail']].head())
```





Q Commands

+ Code + Text

Connect ▾



14. Pass/Fail Status:



Rating Pass/Fail

| | | |
|---|-----|------|
| 0 | 7.8 | Pass |
| 1 | 8.2 | Pass |
| 2 | 6.5 | Fail |
| 3 | 7.3 | Pass |
| 4 | 9.0 | Pass |



15. Movie with the Lowest Rating

```
[ ] import pandas as pd
data = {
    'Rating': [7.8, 8.2, 6.5, 7.3, 9.0],
    'Title': ['Movie A', 'Movie B', 'Movie C', 'Movie D', 'Movie E']
}
df = pd.DataFrame(data)
lowest Rated movie = df.loc[df['Rating'].idxmin()]
print("15. Movie with the Lowest Rating:\n", lowest Rated movie)
```



15. Movie with the Lowest Rating:

| | |
|------------------------|---------|
| Rating | 6.5 |
| Title | Movie C |
| Name: 2, dtype: object | |





16. Percentage of Movies with Rating Below 7

```
[ ] import pandas as pd
    data = {'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]}
    df = pd.DataFrame(data)
    below_seven_percentage = (df[df['Rating'] < 7].shape[0] / df.shape[0]) * 100
    print("16. Percentage of Movies with Rating Below 7:", below_seven_percentage, "%")
```

➦ 16. Percentage of Movies with Rating Below 7: 20.0 %

17. Average Rating for Movies from USA

```
▶ import pandas as pd
   data = {
       'Rating': [7.8, 8.2, 6.5, 7.3, 9.0],
       'Country': ['USA', 'India', 'USA', 'UK', 'India']
   }
   df = pd.DataFrame(data)
   avg_rating_usa = df[df['Country'] == 'USA']['Rating'].mean()
   print("17. Average Rating for Movies from USA:", avg_rating_usa)
```

➦ 17. Average Rating for Movies from USA: 7.15





18. Median Rating



```
[ ] import pandas as pd
    data = {'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]}
    df = pd.DataFrame(data)
    median_rating = df['Rating'].median()
    print("18. Median Rating:", median_rating)
```



18. Median Rating: 7.8



19. Compare Ratings by Genre



```
import pandas as pd
data = {
    'Genre': ['Drama', 'Action', 'Comedy', 'Action', 'Drama'],
    'Rating': [7.8, 8.2, 6.5, 7.3, 9.0]
}
df = pd.DataFrame(data)
rating_by_genre = df.groupby('Genre')['Rating'].mean()
print("19. Ratings by Genre:\n", rating_by_genre)
```





Q Commands

+ Code + Text

Connect ▾



19. Ratings by Genre:

| Genre | |
|--------|------|
| Action | 7.75 |
| Comedy | 6.50 |
| Drama | 8.40 |

Name: Rating, dtype: float64

20. NumPy Array of Total Votes and Mean Rating

```
[ ] import pandas as pd
data = {
    'Rating': [7.8, 8.2, 6.5, 7.3, 9.0],
    'Votes': [1000, 2000, 1500, 1800, 2200]
}
df = pd.DataFrame(data)
total_votes = df['Votes'].values
mean_rating = df['Rating'].mean()
print("20. Mean Rating:", mean_rating)
```

20. Mean Rating: 7.76

