Name:-Gaurav Vijaykumar Mache

Div:-CS3

Roll.No:-CS3-34

PRN.No:-202401040309

Import necessary libraries

```
import pandas as pd
import numpy as np
```

Load datasets

```
movies = pd.read_csv('/content/drive/MyDrive/Dataset/movies.csv')
ratings = pd.read_csv('/content/drive/MyDrive/Dataset/ratings.csv')
tags = pd.read_csv('/content/drive/MyDrive/Dataset/tags.csv')
links = pd.read_csv('/content/drive/MyDrive/Dataset/links.csv')
```

Problem 1: Find the total number of movies in the dataset.

```
total_movies = movies.shape[0]
print(f"Total number of movies: {total_movies}")
```

```
Total number of movies: 9742
```

Problem 2: Find the number of unique users who have rated movies.

```
unique_users = ratings['userId'].nunique()
print(f"Number of unique users: {unique_users}")
```

```
Number of unique users: 610
```

Problem 3: Find the average rating given by users.

```
average_rating = ratings['rating'].mean()
print(f"Average rating: {average_rating:.2f}")
```

```
Average rating: 3.50
```

Problem 4: Find the movie with the highest average rating (minimum 50 ratings).

```
avg_rating_movie = ratings.groupby('movieId').agg({'rating': ['mean', 'count']})
avg_rating_movie.columns = ['mean_rating', 'rating_count']
high_avg_movie = avg_rating_movie[avg_rating_movie['rating_count'] >= 50].sort_values(by='mean_rating', ascending=False)
highest_rated_movie_id = high_avg_movie.index[0]
highest_rated_movie_title = movies[movies['movieId'] == highest_rated_movie_id]['title'].values[0]
print(f"Highest rated movie (min 50 ratings): {highest_rated_movie_title}")
```

```
Highest rated movie (min 50 ratings): Shawshank Redemption, The (1994)
```

Problem 5: Find the number of movies belonging to the "Comedy" genre.

```
comedy_movies = movies[movies['genres'].str.contains('Comedy', na=False)]
print(f"Number of Comedy movies: {comedy_movies.shape[0]}")
```

```
Number of Comedy movies: 3756
```

Problem 6: Find the most common genre among all movies.

```
all_genres = movies['genres'].str.split('|').explode()
most_common_genre = all_genres.value_counts().idxmax()
print(f"Most common genre: {most_common_genre}")
```

```
Most common genre: Drama
```

Problem 7: Find the number of users who have given a rating of 5.0.

```
users_5_star = ratings[ratings['rating'] == 5.0]['userId'].nunique()
print(f"Number of users who gave 5-star ratings: {users_5_star}")
```

```
Number of users who gave 5-star ratings: 573
```

Problem 8: Find the movie that received the most ratings.

```
most_rated_movie_id = ratings['movieId'].value_counts().idxmax()
most_rated_movie_title = movies[movies['movieId'] == most_rated_movie_id]['title'].values[0]
print(f"Movie with most ratings: {most_rated_movie_title}")
```

```
Movie with most ratings: Forrest Gump (1994)
```

Problem 9: Find the user who has rated the most number of movies.

```
top_user_id = ratings['userId'].value_counts().idxmax()
print(f"User who rated most movies: UserID {top_user_id}")
```

```
User who rated most movies: UserID 414
```

Problem 10: Calculate the standard deviation of ratings.

```
rating_std = ratings['rating'].std()
print(f"Standard deviation of ratings: {rating_std:.2f}")
```

```
Standard deviation of ratings: 1.04
```

Problem 11: List the top 5 movies with the most 5-star ratings.

```
five_star_ratings = ratings[ratings['rating'] == 5.0]
top5_five_star_movies = five_star_ratings['movieId'].value_counts().head(5)
print("Top 5 movies with most 5-star ratings:")
for movie_id in top5_five_star_movies.index: title = movies[movies['movieId'] == movie_id]['title'].values[0]
print(f"{title}")
```

```
Top 5 movies with most 5-star ratings:
Star Wars: Episode IV - A New Hope (1977)
```

Problem 12: Find how many movies have no genre listed.

```
no_genre_movies = movies[movies['genres'] == '(no genres listed)']
print(f"Movies with no genres listed: {no_genre_movies.shape[0]}")
```

```
Movies with no genres listed: 34
```

Problem 13: Find the oldest movie in the dataset.

```
movies['year'] = movies['title'].str.extract(r'\((\d{4})\)').dropna()
movies['year'] = pd.to_numeric(movies['year'], errors='coerce')
oldest_movie = movies.sort_values('year').iloc[0]['title']
print(oldest_movie)
```

```
Trip to the Moon, A (Voyage dans la lune, Le) (1902)
```

Problem 14: Find the newest movie in the dataset.

```
newest_movie = movies.sort_values('year', ascending=False).iloc[0]['title']
print(f"Newest movie: {newest_movie}")
```

```
Newest movie: SuperFly (2018)
```

Problem 15: Find the percentage of movies tagged by users.

```
movies_tagged = tags['movieId'].nunique()
percentage_tagged = (movies_tagged / total_movies) * 100
print(f"Percentage of movies tagged: {percentage_tagged:.2f}%")
```

```
Percentage of movies tagged: 16.14%
```

Problem 16: Find the average number of ratings per movie.

```
avg_ratings_per_movie = ratings.groupby('movieId').size().mean()
print(f"Average number of ratings per movie: {avg_ratings_per_movie:.2f}")
```

```
Average number of ratings per movie: 10.37
```

Problem 17: Find the number of unique tags used.

```
unique_tags = tags['tag'].nunique()
print(f"Number of unique tags used: {unique_tags}")
```

```
Number of unique tags used: 1589
```

Problem 18: Find the top 5 most commonly used tags.

```
top5_tags = tags['tag'].value_counts().head(5)
print("Top 5 most common tags:")
print(top5_tags)
```

```
Top 5 most common tags:
tag
In Netflix queue     131
atmospheric           36
thought-provoking     24
superhero             24
surreal               23
Name: count, dtype: int64
```

```
#Problem 19: Check if there are any missing (null) values in each file.
```

```
print("\nMissing values in movies.csv:")
print(movies.isnull().sum())
print("\nMissing values in ratings.csv:")
print(ratings.isnull().sum())

print("\nMissing values in tags.csv:")
print(tags.isnull().sum())

print("\nMissing values in links.csv:")
print(links.isnull().sum())
```

```
Missing values in movies.csv:
movieId     0
title       0
genres      0
year       13
dtype: int64

Missing values in ratings.csv:
userId       0
movieId      0
rating       0
timestamp    0
dtype: int64

Missing values in tags.csv:
userId       0
movieId      0
tag          0
timestamp    0
dtype: int64

Missing values in links.csv:
movieId    0
imdbId     0
tmdbId     8
dtype: int64
```

```
#Problem 20: Merge movies and ratings datasets and find the average rating per genre.


ratings_movies = pd.merge(ratings, movies, on='movieId')
ratings_movies = ratings_movies.dropna(subset=['genres'])
ratings_movies['genres_split'] = ratings_movies['genres'].str.split('|')
ratings_exploded = ratings_movies.explode('genres_split')
genre_avg_rating = ratings_exploded.groupby('genres_split')['rating'].mean().sort_values(ascending=False)
print("\nAverage rating per genre:")
print(genre_avg_rating)
```

```
Average rating per genre:
genres_split
Film-Noir             3.920115
War                   3.808294
Documentary           3.797785
Crime                 3.658294
Drama                 3.656184
Mystery               3.632460
Animation             3.629937
IMAX                  3.618335
Western               3.583938
Musical               3.563678
Adventure             3.508609
Romance               3.506511
Thriller              3.493706
Fantasy               3.491001
(no genres listed)    3.489362
Sci-Fi                3.455721
Action                3.447984
Children              3.412956
Comedy                3.384721
Horror                3.258195
Name: rating, dtype: float64
```