Name : Ayush Rajendra Andhale

Batch : CS - 5 - 56

PRN : 202401100100

Subject : Essential of Data Science

Topic : Activity 1

Usage of Numpy and Pandas

1. Problem Statement: Find the average rating of all movies in the dataset.

```
import pandas as pd
movie_df = pd.read_csv('movie_reviews.csv')
avg_rating = movie_df['rating'].mean()
print(f"Average Rating: {avg_rating}"
```

2. Problem Statement: Find the movie with the highest rating.

```
highest_rated_movie = df.loc[df['rating'].idxmax()]
print(f"Movie with the Highest Rating: {highest_rated_movie['movie_id']}
with rating {highest_rated_movie['rating']}")
```

3: Count Movies with Rating > 4

```
count_high_rating = len(df[df['rating'] > 4])
print(f"Number of movies with rating > 4: {count_high_rating}")
```

4: Number of Reviews per Movie (Assuming df has a column review_id)

python

```
df = pd.DataFrame({
    'movie_id': [1, 1, 2, 2, 3],
    'review_id': [101, 102, 103, 104, 105]
})
reviews_per_movie = df.groupby('movie_id').size()
print(reviews_per_movie)
```

5: Find the Movie with the Most Reviews

```
df = pd.DataFrame({
    'movie_id': [1, 1, 2, 2, 3, 3, 3],
    'review_id': [101, 102, 103, 104, 105, 106, 107],
})
reviews_count = df.groupby('movie_id').size()
most_reviews_movie = reviews_count.idxmax()
print(f"Movie with the most reviews: {most_reviews_movie}")
```

6: Percentage of Positive Reviews (rating > 4)

```
df = pd.DataFrame({
   'movie_id': [1, 2, 3, 4, 5],
   'rating': [3.5, 4.2, 4.8, 2.9, 4.6]
})
positive_reviews_percentage = (df['rating'] > 4).mean() * 100
print(f"Percentage of positive reviews (rating > 4): {positive_reviews_percentage:.2f}%")
```

7: Identify the Top 5 Movies with the Highest Average Ratings

```
df = pd.DataFrame({
   'movie_id': [1, 1, 2, 2, 3, 3],
   'rating': [3.5, 4.5, 4.8, 4.0, 5.0, 4.6]
})
top_5_movies = df.groupby('movie_id')['rating'].mean().nlargest(5)
print("Top 5 Movies with Highest Average Ratings:")
print(top_5_movies)
```

8: Correlation Between Review Length and Rating

```
df = pd.DataFrame({
   'movie_id': [1, 2, 3, 4, 5],
   'rating': [4.5, 3.8, 5.0, 4.2, 3.9],
   'review': ['Great movie!', 'Not bad', 'Amazing', 'Could be better', 'Good']
})
df['review_length'] = df['review'].apply(len)
correlation = df[['review_length', 'rating']].corr().iloc[0, 1]
print(f"Correlation between review length and rating: {correlation:.2f}")
```

9: Movie with the Most Diverse Ratings (Highest Std Dev)

python

```
df = pd.DataFrame({
   'movie_id': [1, 1, 1, 2, 2, 3, 3, 3],
   'rating': [3.5, 4.0, 5.0, 2.8, 3.9, 4.5, 4.2, 4.8]
})
rating_std = df.groupby('movie_id')['rating'].std()
most_diverse_movie = rating_std.idxmax()
print(f"Movie with the most diverse ratings: {most_diverse_movie}")
```

10: Average Length of Review for Each Movie

```
df = pd.DataFrame({
   'movie_id': [1, 1, 2, 2, 3],
   'review': ['Great movie!', 'Excellent!', 'Not bad', 'Could be better', 'Awesome']
})
df['review_length'] = df['review'].apply(len)
average_review_length = df.groupby('movie_id')['review_length'].mean()
print("Average Review Length for Each Movie:")
print(average_review_length)
```

11: Number of Movies with More Than One Genre

```
df = pd.DataFrame({
   'movie_id': [1, 2, 3, 4, 5],
   'genre': ['Action', 'Action, Drama', 'Comedy', 'Action, Comedy', 'Drama']
})
df['num_genres'] = df['genre'].str.split(',').apply(len)
movies_with_multiple_genres = df[df['num_genres'] > 1].shape[0]
print(f"Number of movies with more than one genre: {movies_with_multiple_genres}")
```

12: Movies with Missing Ratings or Reviews

```
df = pd.DataFrame({
   'movie_id': [1, 2, 3, 4, 5],
   'rating': [4.5, None, 3.9, 4.1, None],
   'review': ['Great movie!', None, 'Good', 'Not bad', 'Amazing']
})
missing_values = df.isna().sum()
print("Missing Values in Dataset:")
print(missing_values)
```

13: Average Rating by Genre

python

```
df = pd.DataFrame({
   'movie_id': [1, 2, 3, 4, 5],
   'rating': [4.5, 3.8, 5.0, 4.2, 3.9],
   'genre': ['Action', 'Drama', 'Comedy', 'Action', 'Drama']
})
average_rating_by_genre = df.groupby('genre')['rating'].mean()
print("Average Rating by Genre:")
print(average_rating_by_genre)
```

14: Top 10 Movies Sorted by Average Rating

```
df = pd.DataFrame({
   'movie_id': [1, 1, 2, 2, 3, 3],
   'rating': [3.5, 4.5, 4.8, 4.0, 5.0, 4.6]
})
top_10_movies = df.groupby('movie_id')['rating'].mean().sort_values(ascending=False).head(10)
print("Top 10 Movies Sorted by Average Rating:")
print(top_10_movies)
```

15: Percentage of Reviews Containing the Word "Great"

```
avg_rating_per_user = df.groupby('user')['rating'].mean()
print(avg_rating_per_user)
```

16. How many unique users have rated each movie?

```
unique_users_per_movie = df.
groupby('movie')['user'].nunique()
print(unique_users_per_movie)
```

17. What is the standard deviation of ratings for each movie?

```
rating_std_per_movie = df.
groupby('movie')['rating'].std()
print(rating_std_per_movie)
```

18. Find the most recent review for each movie.

```
df['review_date'] = pd.
to_datetime(df['review_date'])
most_recent_review_per_movie = df.
groupby('movie')['review_date'].max()
print(most_recent_review_per_movie)
```

19. Which movies have an equal rating (i.e., no variation in ratings)?

```
no_variation_movies = df.groupby('movie')['rating'].std()[df.
groupby('movie')['rating'].std() == 0].index
print(f"Movies with no variation in ratings: {no_variation_movies}")
```

20. What is the correlation between the length of the review and the rating given?

```
df['review_length'] = df['review'].apply(len)
correlation = df[['review_length', 'rating']].corr()
print(correlation)
```


Thank
you