Movie Review Dataset - Problem Statements and Solutions

NAME: AYUSH GHOLAP

PRN: 202401070176

DIVISION: ET2

BATCH: E21

ROLL NO : ET2-14

Movie Review Dataset - Problem Statements and Solutions

1. Find the total number of reviews in the dataset.

```
import pandas as pd reviews =

pd.read_csv('movie_reviews.csv')

total_reviews = reviews.shape[0]

print(total_reviews)
```

2. Find the number of unique movies reviewed.

```
unique_movies = reviews['movie_title'].nunique()

print(unique_movies)
```

3. Find the average rating across all movies.

```
average_rating = reviews['rating'].mean()

print(average_rating)
```

4. Find the movie with the highest average rating.

```
highest_rated_movie = reviews.groupby('movie_title')['rating'].mean().idxmax()

print(highest_rated_movie)
```

5. Find the movie with the lowest average rating.

```
lowest_rated_movie = reviews.groupby('movie_title')['rating'].mean().idxmin()

print(lowest_rated_movie)
```

6. How many reviews were posted each year?

```
reviews['review_year'] = pd.to_datetime(reviews['review_date']).dt.year

reviews_per_year = reviews['review_year'].value_counts().sort_index()

print(reviews_per_year)
```

7. List the top 5 reviewers who wrote the most reviews.

```
top_reviewers = reviews['reviewer_name'].value_counts().head(5)

print(top_reviewers)
```

Movie Review Dataset - Problem Statements and Solutions

8. Find the proportion of positive and negative reviews.

```
sentiment_proportion = reviews['sentiment'].value_counts(normalize=True)

print(sentiment_proportion)
```

9. Find the median rating for all movies.

```
median_rating = reviews['rating'].median()

print(median_rating)
```

10. Identify how many reviews have a perfect rating (10/10).

```
perfect_reviews_count = (reviews['rating'] == 10).sum()

print(perfect_reviews_count)
```

11. Find the standard deviation of movie ratings.

```
rating_std_dev = reviews['rating'].std()

print(rating_std_dev)
```

12. List movies that have only positive reviews.

```
positive_movies = reviews.groupby('movie_title')['sentiment'].unique()

only_positive_movies = positive_movies[positive_movies.apply(lambda x: all(s ==

'positive' for s in x))]
print(only_positive_movies.index.tolist())
```

13. Calculate the percentage of reviews that contain the word 'amazing'.

```
                                                                    case=False,


amazing_reviews   =     reviews['review_text'].str.contains('amazing',
na=False).sum() percentage_amazing = (amazing_reviews / total_reviews) *
100 print(percentage_amazing)
```

14. Find the longest review (by character count).

```
reviews['review_length'] = reviews['review_text'].str.len()

longest_review = reviews.loc[reviews['review_length'].idxmax()]

print(longest_review)
```

Movie Review Dataset - Problem Statements and Solutions

15. Find the shortest review (by character count).

```
shortest_review = reviews.loc[reviews['review_length'].idxmin()]

print(shortest_review)
```

16. List the movies reviewed after 2020.

```
recent_reviews = reviews[reviews['review_year'] > 2020]['movie_title'].unique()

print(recent_reviews)
```

17. Find the number of duplicate reviews based on review text.

```
duplicate_reviews_count = reviews['review_text'].duplicated().sum()

print(duplicate_reviews_count)
```

18. Find the correlation between review length and rating.

```
correlation = reviews['review_length'].corr(reviews['rating'])

print(correlation)
```

19. Group the reviews into bins based on rating (Low, Medium, High).

```
rating_bins = pd.cut(reviews['rating'], bins=[0,3,7,10], labels=['Low', 'Medium',

'High']) rating_bin_counts =
rating_bins.value_counts()
print(rating_bin_counts)
```

20. Find the average rating given by each reviewer and list the top 5.

```
                                                                                =

top_reviewers_by_rating
reviews.groupby('reviewer_name')['rating'].mean().sort_values(ascending=False).head(5)
print(top_reviewers_by_rating)
```