EXPLORATORY ANALYSIS of the MovieLens dataset

The provided project performs an exploratory analysis of the MovieLens dataset, focusing on analyzing ratings, genres, and trends over time.

Detailed Explanation:

Step 1:

Load the Dataset

Purpose: Load two datasets, movies.csv and ratings.csv, containing movie metadata and user ratings, respectively.

Code:

```
movies = pd.read_csv('movies.csv')
ratings = pd.read_csv('ratings.csv')
```

This step reads the datasets into Pandas DataFrames for easier manipulation.

Step 2:

Data Exploration

Purpose: Understand the structure of the datasets by printing the first few rows using .head() and identifying the columns.

Insights: This helps confirm that the datasets are loaded correctly and provides a quick glance at available features like movieId, title, genres, and rating.

Step 3:

Data Cleaning and Preparation

Merging Datasets: Combines ratings and movies on the movieId column.

Extracting Year: Extracts the release year from the movie titles using a regex pattern: \((\d{4})\).

Handling Missing Values: Removes rows with missing values in the Year column to ensure clean data for analysis.

Step 4:

Analysis and Visualization

Visualization 1: Average Ratings by Year

Purpose: To observe how average movie ratings have changed over time.

Method: Group the data by Year and compute the mean of rating.

Plot: Line plot showing the trend of ratings over time.

Code:

```
avg_ratings_by_year = data.groupby('Year')['rating'].mean().reset_index()
plt.plot(avg_ratings_by_year['Year'], avg_ratings_by_year['rating'], marker='o')
```

Visualization 2: Top 10 Genres by Average Rating

Purpose: To identify which genres receive the highest average ratings.

Method: Split the genres column into individual genres using .str.split('|') and .explode().

Plot: Bar chart showing the average ratings of the top 10 genres.

Code:

```
data_genres = data.assign(genre=data['genres'].str.split('|')).explode('genre')
avg_rating_by_genre = data_genres.groupby('genre')['rating'].mean().sort_values(ascending=False).head(10)
```

```
avg_rating_by_genre.plot(kind='bar')
```

Visualization 3: Rating Distribution
Purpose: To understand the distribution of user ratings.
Method: Plot a histogram using plt.hist() with appropriate bin sizes.
Code:

```
plt.hist(data['rating'], bins=np.arange(0.5, 5.5, 0.5))
```

Visualization 4: Most Rated Movies
Purpose: To identify movies with the highest number of ratings.
Method: Group by title, count the number of ratings, and sort in descending order.
Plot: Bar chart of the top 10 most rated movies.
Code:

```
most_rated = data.groupby('title')['rating'].count().sort_values(ascending=False).head(10)
most_rated.plot(kind='bar')
```

Step 5:

Insights

Average Ratings Over Time: Ratings may trend upward or downward based on cultural or industry shifts.
Genres: Some genres, like documentaries or classics, may consistently receive higher ratings compared to mainstream genres like action.
Rating Behavior: Peaks in the histogram indicate common rating patterns (e.g., many users rate movies as 4.0 or 5.0).
Popular Movies: High rating counts suggest movies with broad appeal or strong fan bases.

Potential Extensions

Advanced Filtering: Analyze trends for specific genres or decades.
Sentiment Analysis: Use review text (if available) for deeper insights into user preferences.
User Segmentation: Cluster users based on their rating patterns.
Machine Learning: Build a recommendation system using collaborative filtering or matrix factorization.