**Project: Analysis of Movie Ratings and Trends**

**Problem Statement**

The goal of this project is to ana lyze a dataset of movie ratings to uncover insights about trends in movie popularity, genre preferences, and the correlation between various factors (e.g., release year, genre, budget) and movie ratings. The analysis will help in understanding how different attributes influence movie ratings and trends over time.

**Dataset**

You can use the MovieLens dataset. MovieLens is a well-known dataset containing millions of ratings and tags for movies. For this project, we'll use the **MovieLens 20M Dataset**, which contains 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users.

**Dataset Information**

1. **Ratings**: This file contains the user ratings.
   * **userId** - ID of the user
   * **movieId** - ID of the movie
   * **rating** - Rating given by the user
   * **timestamp** - Time of the rating
2. **Movies**: This file contains the information about movies.
   * **movieId** - ID of the movie
   * **title** - Title of the movie
   * **genres** - Genres of the movie (multiple genres separated by **|**)
3. **Tags**: This file contains user-applied tags.
   * **userId** - ID of the user
   * **movieId** - ID of the movie
   * **tag** - Tag applied by the user
   * **timestamp** - Time of the tag
4. **Links**: This file contains identifiers that can be used to link to other movie databases.
   * **movieId** - ID of the movie
   * **imdbId** - IMDb ID
   * **tmdbId** - TMDb ID

**Project Tasks**

1. **Data Cleaning and Preparation**:
   * Load the datasets into Python using pandas.
   * Handle missing values if any.
   * Convert timestamps to readable date formats.
   * Merge datasets to create a comprehensive dataset.
2. **Exploratory Data Analysis (EDA)**:
   * Analyze the distribution of ratings.
   * Identify the most popular movies (by number of ratings).
   * Identify the highest-rated movies.
   * Analyze the distribution of ratings by genre.
   * Determine the trends in movie ratings over time.
3. **Genre Analysis**:
   * Find the most popular genres.
   * Analyze average ratings by genre.
   * Determine how genre popularity and ratings have changed over the years.
4. **User Analysis**:
   * Analyze user rating patterns (e.g., number of ratings per user).
   * Identify the most active users.
   * Explore the correlation between user activity and average rating given.
5. **Correlation Analysis**:
   * Examine the relationship between movie attributes (e.g., year of release, genre) and ratings.
   * Use correlation coefficients to identify significant relationships.
6. **Visualization**:
   * Create visualizations to illustrate key findings (e.g., bar charts, histograms, time series plots, heatmaps).

**Tools and Libraries**

* **Pandas**: For data manipulation and analysis.
* **NumPy**: For numerical operations.
* **Matplotlib** and **Seaborn**: For data visualization.
* **Scikit-learn**: For any machine learning models or advanced data analysis techniques.

**Example Code Snippets**

**Loading Data**

import pandas as pd

# Load the datasets

ratings = pd.read\_csv('ratings.csv')

movies = pd.read\_csv('movies.csv')

tags = pd.read\_csv('tags.csv')

links = pd.read\_csv('links.csv')

**Merging Datasets**

# Merge ratings and movies data

merged\_df = pd.merge(ratings, movies, on='movieId')

**Data Exploration**

# Distribution of ratings

import matplotlib.pyplot as plt

Import seaborn as sns

sns.histplot(merged\_df['rating'], bins=10, kde=True)

plt.xlabel('Rating')

plt.ylabel('Count')

plt.title('Distribution of Movie Ratings')

plt.show()

**Genre Analysis**

# Split genres and explode

merged\_df['genres'] = merged\_df['genres'].str.split('|')

genres\_df = merged\_df.explode('genres')

# Average rating by genre

genre\_ratings = genres\_df.groupby('genres')['rating'].mean().sort\_values(ascending=False)

print(genre\_ratings)

This should give you a solid starting point for your project. Feel free to expand the analysis based on your interests and findings. Good luck with your project!