🎬 Executive Summary: Movie Data Analysis

📁 Dataset Overview

The dataset includes various features for each movie:

- `name`: Movie title
- `imdb_rating`: IMDb score (0–10 scale)
- `budget`: Total production budget
- `revenue`: Total earnings from the movie
- `release_year`: Year of release
- `industry`: Movie industry (e.g., Bollywood, Hollywood)
- `language`: Language of the movie

The dataset was processed to:

- Filter by time periods

- Classify by industry
- Analyze rating vs. budget
- Derive profit metrics
- Create visualizations for better decision-making

📌 Step-by-Step Analysis

1. Data Loading and Basic Info

- The dataset was loaded using `pandas` and its structure verified using `.info()` and `.shape()`.
- It contains a well-structured format suitable for exploratory data analysis (EDA).

---

2. Year Classification

A new column `year_classify` was created using:

df['year_classify'] = df.apply(lambda x: "Before_Year" if x['release_year'] < 2000 else "after_year", axis=1)

This classified each movie into:

- **Before_Year** (pre-2000)
- **after_year** (2000 onwards)

🔍 **Observation**:

- Majority of the data likely belongs to **modern era movies**, making the analysis more relevant for current market trends.

---

3. Industry-Based Filtering

Filtered the dataset by:

- **Bollywood**
- **Hollywood**
- Other possible industries (like Tollywood)

Example:

df[df['industry'] == 'Bollywood']

🔍 **Insights**:

- Enables a focused view of performance and trends within a particular industry.

- Useful for industry-specific recommendations (e.g., Bollywood production strategies).

---

4. Language Distribution

Checked unique languages using:

df['language'].unique()

🔍 **Insights**:

- Helps in identifying which languages dominate the dataset.
- Enables potential multilingual market strategies.

---

5. Filtering by Release Year

Filtered for movies released **after 2010** and optionally by industry (e.g., Bollywood):

df[df['release_year'] >= 2010]

🔍 **Insights**:

- Allows analysis of modern production trends, budgets, and returns.

- Useful for identifying changes in audience preferences post-2010.

---

6. IMDb Rating Analysis

Sorted data by `imdb_rating`:

df.sort_values(by='imdb_rating', ascending=False)

🔍 **Insights**:

- Highlights top-rated films.
- Can be used to understand what elements contribute to high ratings (genre, language, budget range, etc.).

---

7. Visualization: IMDb Rating vs Budget

Used a bar chart:

plt.bar(df['imdb_rating'], df['budget'])

🔍 **Insights**:

- Gives a rough sense of whether **higher ratings are associated with bigger budgets**.

- However, **bar charts are not ideal** for continuous data like ratings — scatter plots are better for seeing correlations.

---

8. Profit Calculation

Added a new column:

df['Profit'] = df['revenue'] - df['budget']

🔍 **Insights**:

- Enables calculation of profitability.
- Important for investment decision-making.

You can further calculate:

- **Profit Margin**: `(Profit / Budget) * 100`
- **Return on Investment (ROI)**

---

9. GroupBy Analysis

Grouped data by `industry`:

g = df.groupby('industry')

Used to:

- Get size of each group
- Extract industry-specific datasets
- Compare metrics like **average budget**, **average profit**, **mean rating**

🔍 **Insights**:

- Industry-wise breakdown allows you to see which industries perform better overall.

📝 Conclusion

My analysis provides a solid foundation for understanding movie performance across different dimensions. With further enhancements like profitability ratios, better visualizations, and deeper statistical metrics, this can evolve into a **powerful tool for producers, marketers, and investors in the film industry**.