Comprehensive Movie Dataset Analysis Report

Introduction

This report presents a detailed analysis of a movie dataset using Python and various data analysis libraries. The project demonstrates advanced skills in data cleaning, manipulation, analysis, grouping, and visualization techniques. By examining multiple aspects of the film industry, this analysis provides valuable insights into movie trends, profitability, and audience preferences.

Data Preprocessing

Data Import and Initial Exploration

1. Imported essential libraries:
   - pandas (as pd): For data manipulation and analysis
   - matplotlib.pyplot (as plt): For creating static visualizations
   - numpy (as np): For numerical operations
   - plotly.express (as px): For interactive visualizations

2. Loaded the dataset:
   python
   df = pd.read_csv('movies.csv')

3. Explored the dataset structure:
   - Used df.shape to determine the number of rows and columns
   - Utilized df.columns to list all column names
   - Employed df.head() to view the first few rows and understand the data format

4. Checked for null values using df.isnull().sum() to identify columns requiring cleaning

Data Cleaning

1. Dropped unnecessary columns to focus on relevant data:
   python
   df = df.drop(columns=['id', 'imdb_id', 'homepage', 'cast', 'tagline', 'overview', 'budget', 'revenue'], inplace=True)

   This step streamlined the dataset, focusing on the most relevant features for analysis.

2. Handled null values:
   - For 'director' and 'genres' columns:
     python
     df.dropna(subset=['director', 'genres'], inplace=True)

     Rows with missing values in these crucial columns were removed to ensure data integrity.

- For 'production_companies' and 'keywords' columns:
  python
  df['production_companies'].fillna(0, inplace=True)
  df['keywords'].fillna(0, inplace=True)

  Filled null values with zero, assuming absence of data in these fields doesn't necessarily invalidate the entry.

3. Rounded 'popularity' column:
  python
  df['popularity'] = df['popularity'].round(2)

  This improved readability and simplified subsequent analyses.

4. Created new columns for financial analysis:
  python
  df['profit'] = df['revenue'] - df['budget']
  df['roi'] = df['profit'] / df['budget']

  These derived columns enable deeper insights into movie financial performance.

Data Type Conversion

1. Converted 'release_date' to datetime:
  python
  df['release_date'] = pd.to_datetime(df['release_date'])

  This allows for more sophisticated time-based analyses.

2. Extracted month from 'release_date':
  python
  df['extracted_month'] = df['release_date'].dt.month

  This new column facilitates monthly trend analysis.

Data Analysis and Visualization

1. Numeric Data Overview

Created histograms for numeric columns to understand data distribution:

python
numeric_columns = ['popularity', 'budget', 'revenue', 'profit', 'roi', 'vote_count', 'vote_average', 'release_year']
df[numeric_columns].hist(bins=30, figsize=(20, 15))
plt.tight_layout()
plt.show()

Key Insights:
- Popularity: Highly skewed distribution with most movies having low popularity scores and a few extremely popular outliers.
- Budget and Revenue: Right-skewed distributions, indicating a few high-budget, high-revenue movies among many lower-budget productions.
- Profit: Wide range, with some movies making significant profits and others incurring losses.
- ROI: Extremely varied, highlighting the unpredictable nature of movie investments.
- Vote Count and Average: Bell-shaped distributions, suggesting a balanced range of audience engagement and ratings.

2. Popularity and Rating Over Time

Analyzed trends in movie popularity and average ratings over the years:

python
```
df_grouped = df.groupby('release_year')[['popularity', 'vote_average']].mean()
df_grouped.plot(kind='line', figsize=(12, 6))
plt.title('Movie Popularity and Average Rating Over Time')
plt.xlabel('Year')
plt.show()
```

Key Insights:
- Popularity Trend: Dramatic increase in movie popularity since 2000, possibly due to improved marketing strategies, wider distribution channels, and the rise of social media.
- Rating Trend: Slight decline in average ratings over time. This could be attributed to:
  1. Increased content saturation leading to more critical audiences.
  2. Shift in viewing habits with the rise of streaming platforms.
  3. Possible rating inflation in earlier years due to limited audience reach.

3. Popularity vs. Rating Correlation

Examined the relationship between popularity and average rating:

python
```
plt.figure(figsize=(10, 6))
plt.scatter(df['popularity'], df['vote_average'], alpha=0.5)
plt.xlabel('Popularity')
plt.ylabel('Average Rating')
plt.title('Movie Popularity vs. Average Rating')
plt.show()
```

Key Insights:
- No strong correlation between popularity and average rating.
- Highly popular movies show a wide range of average ratings, suggesting that popularity doesn't guarantee critical acclaim.

- Some highly-rated movies have low popularity, indicating potential for "hidden gems" in the dataset.
- This lack of correlation highlights the subjective nature of movie reception and the difference between mass appeal and critical appreciation.

4. Genre Analysis

Analyzed popularity by genre:

python
```
df['genres'] = df['genres'].apply(lambda x: x.split('|'))
genre_data =
df.explode('genres').groupby('genres')['popularity'].sum().sort_values(ascending=True)

genre_data.plot(kind='barh', figsize=(10, 8))
plt.title('Movie Popularity by Genre')
plt.xlabel('Total Popularity Score')
plt.ylabel('Genre')
plt.show()
```

Key Insights:
- Most Popular Genres: Drama, Comedy, and Action dominate, reflecting audience preferences and industry focus.
- Mid-Range Genres: Thriller, Romance, and Science Fiction show moderate popularity, indicating steady audience interest.
- Least Popular Genres: Foreign and TV Movie genres have the lowest popularity scores, possibly due to limited distribution or niche appeal.
- Genre Diversity: The wide range of genres suggests a diverse movie industry catering to various audience preferences.

5. Monthly Trends

Analyzed popularity and revenue trends by month:

python
```
monthly_data = df.groupby('extracted_month')[['popularity', 'revenue']].sum()
monthly_data.plot(kind='bar', figsize=(12, 6))
plt.title('Movie Popularity and Revenue by Month')
plt.xlabel('Month')
plt.ylabel('Total Score/Amount')
plt.legend(['Popularity', 'Revenue'])
plt.show()
```

Key Insights:
- Popularity Peak: September shows the highest popularity, possibly due to film festival season and the start of the awards season push.

- Revenue Peak: June generates the highest revenue, likely due to summer blockbuster releases.
- Seasonal Patterns:
  - Summer months (June-August) show high revenue, aligning with school holidays and vacation season.
  - December shows a spike in both popularity and revenue, likely due to holiday season releases and Oscar-contending films.
- Popularity-Revenue Mismatch: The disconnect between peak popularity (September) and peak revenue (June) months suggests that factors beyond mere popularity influence box office performance.

6. Top Movies by Profit

Visualized the top 5 movies by profit:

python
```
top_movies = df.nlargest(5, 'profit')
plt.pie(top_movies['profit'], labels=top_movies['original_title'], autopct='%1.1f%%')
plt.title('Top 5 Movies by Profit')
plt.axis('equal')
plt.show()
```

Key Insights:
- Blockbuster Dominance: Movies like Avatar, Star Wars, and Titanic dominate the profit charts.
- Franchise Power: Presence of franchise films (e.g., Star Wars) highlights the financial success of established intellectual properties.
- High Investment, High Return: These top-grossing films often have large budgets but deliver exponential returns.
- Industry Impact: The outsized success of these few movies can significantly influence overall industry trends and studio strategies.

7. Production Companies Analysis

Analyzed the top 5 production companies:

python
```
top_companies = df['production_companies'].value_counts().head()
plt.pie(top_companies, labels=top_companies.index, autopct='%1.1f%%')
plt.title('Top 5 Production Companies')
plt.axis('equal')
plt.show()
```

Key Insights:
- Market Leaders: Paramount Pictures, Universal Pictures, and Warner Bros. emerge as the most prolific production companies.

- Industry Concentration: The dominance of a few large studios suggests a concentrated industry structure.
- Resource Advantage: These top companies likely have significant resources for marketing and distribution, contributing to their market share.
- Potential for Analysis: Further investigation into these companies' strategies could reveal insights into successful movie production and marketing techniques.

8. Keyword Analysis

Created a treemap to visualize the most commonly used keywords:

python
```
keyword_counts = df['keywords'].str.split('|', expand=True).stack().value_counts()
fig = px.treemap(
    names=keyword_counts.index[:20],
    parents=['']  20,
    values=keyword_counts.values[:20],
    title='Top 20 Movie Keywords'
)
fig.show()
```

Key Insights:
- Diversity in Filmmaking: "Woman director" as a frequent keyword suggests a growing trend in diverse filmmaking.
- Independent Cinema: High frequency of "independent film" indicates a significant presence of non-mainstream productions in the dataset.
- Thematic Trends: Keywords like "dystopia," "based on novel," and "superhero" reflect popular themes and source materials in contemporary cinema.
- Genre Indicators: Presence of keywords like "murder" and "police" suggests a strong representation of crime and thriller genres.

Advanced Analysis Techniques

Correlation Matrix

To further understand relationships between numeric variables:

python
```
correlation_matrix = df[numeric_columns].corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Numeric Variables')
plt.show()
```

Key Insights:

- Strong positive correlation between budget and revenue, suggesting higher budgets often lead to higher revenues.
- Moderate positive correlation between vote_count and popularity, indicating that more voted movies tend to be more popular.
- Weak correlation between vote_average and other variables, reinforcing the earlier finding that quality (as measured by average rating) doesn't strongly predict commercial success.

Time Series Decomposition

To understand underlying trends and seasonality in movie releases:

python
```
from statsmodels.tsa.seasonal import seasonal_decompose

Aggregate data by month
monthly_releases = df.resample('M', on='release_date').size()

Perform time series decomposition
result = seasonal_decompose(monthly_releases, model='additive')

Plot the decomposition
result.plot()
plt.tight_layout()
plt.show()
```

Key Insights:
- Clear upward trend in the number of movie releases over time, indicating industry growth.
- Seasonal pattern in releases, with peaks often occurring during summer and holiday seasons.
- Residual component shows unexplained variations, possibly due to external factors like economic conditions or major world events.

Conclusion

This comprehensive analysis of the movie dataset has revealed several key insights into the film industry:

1. The industry has seen a significant increase in movie popularity over time, particularly since 2000.
2. There's a notable disconnect between a movie's popularity and its average rating, highlighting the complex nature of audience reception.
3. Genre preferences are clear, with Drama, Comedy, and Action leading in popularity.
4. Seasonal trends in both popularity and revenue are evident, with different peaks for each metric.
5. The film industry is dominated by a few top production companies and blockbuster movies.
6. Keywords analysis reveals growing trends in diverse and independent filmmaking.

7. Financial success in movies is influenced by multiple factors, with budget playing a significant role in potential revenue.

This project demonstrates proficiency in various data analysis techniques, including data cleaning, exploratory data analysis, statistical analysis, and data visualization. The use of Python libraries such as pandas, matplotlib, seaborn, and plotly showcases versatility in handling and presenting data.