# CSDS 234
# Structured and Unstructured data

Siyeon Park
sxp1129

# Course Project Final Report

## Abstract

This project explores trends in the movie industry, focusing on the evolution and dominance of genres over time and their relationship with key attributes such as budget, popularity, and ratings. Using a movie metadata dataset from Kaggle, analysis were conducted to discover patterns and correlations. It was found out that Drama genre had consistenely been the most dominant genre, with an observed anomaly in 2017, where all top 5 genres experienced max loss, likely due to the rise of streaming services. A positive correlation between budget and popularity was identified, though higher budgets did not always guarantee success. Additionally, movie ratings were found to be independent of genres, highlighting the importance of storytelling and quality. This project provides valuable insights for industry professionals and the background of future exploration into audience behavior and emerging trends.

## Introduction

The movie industry is a dynamic and ever-evolving field, constantly influenced by any single factor such as streaming trends, pandemic, economic trends. Especially since consumers started to watch movies or dramas at home using online streaming services instead of visiting movie theater to watch a movie. The related work shows how total cinema industry's revenue decreased after COVID pandemic, where it changed the trend completely from watching movies at a traditional style movie theater to the online streaming services. In 2020, it was found out that only 14% of the consumers preferred the movie theater when 36% preferred the streaming services (Raphael, Rudly). Therefore, understanding the trends and patterns within this industry is crucial for filmmakers, studios, and distributors to make informed decisions and stay relevant in a competitive market. This project aims to explore the dominance and evolution of movie genres over time as well as their relationship with key attributes such as budget, popularity, and ratings. I explored the dataset to answer the following questions:

- How movie genres evolved over time, and what factors have influenced the trend?
- Is there any relationship between budget and genre popularity, and does higher budget movies always equals to the greater success?
- How do audience preferences vary across different genres, and what insights we can recognize?

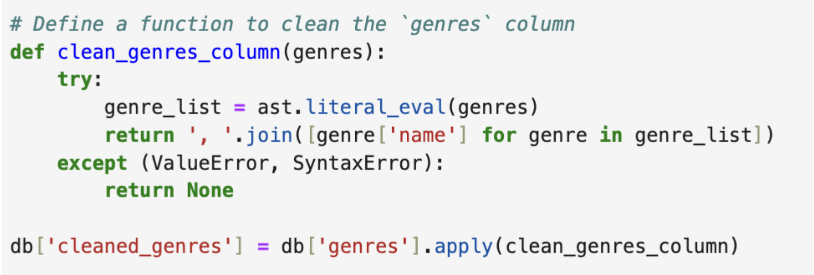
By addressing these question, this project will provide valuable insights into the dynamics of the movie industry and enable a deeper understanding of what factors drive trends in the movie industry and will give the insights for the industry professionals looking to capitalize on trends or cater to audience tastes. It was challenging since finding the good set of dataset was important. I used the dataset from Kaggle, and outputted different types of visualization to show the exploration clearly.

## Related Work

There are plenty of research about the movie industry as well as the analysis of the movie genre. Here are some research papers that previous studies did. One did the empirical investigation on movie industry from 1980 to 2018 that found action movie was the most popular movie genre after 2010(Zhan et al.). Another researcher analyzed about the movie genre and the performance, they found out certain genres work better while others work differently although not all movie genres appeared consistently in all movie performance indicators (Kim et al.). While previous studies have explored aspects of genre evolution and the performance, my project offers a more comprehensive and data-driven approach.

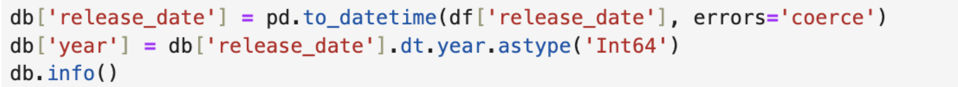
## Method/Solution

The first step, I've done for this project was to download an appropriate dataset from Kaggle. Then, I read the csv file, and print out the information, summary statistics, column names, and the total number of rows and columns. By executing db.head(), I was able to find which part should I manipulate to clean the data. Since, the genre columns were a JSON format, I cleaned the JSON format to a string format of a python list. Below is the code that I created to clean genre columns.

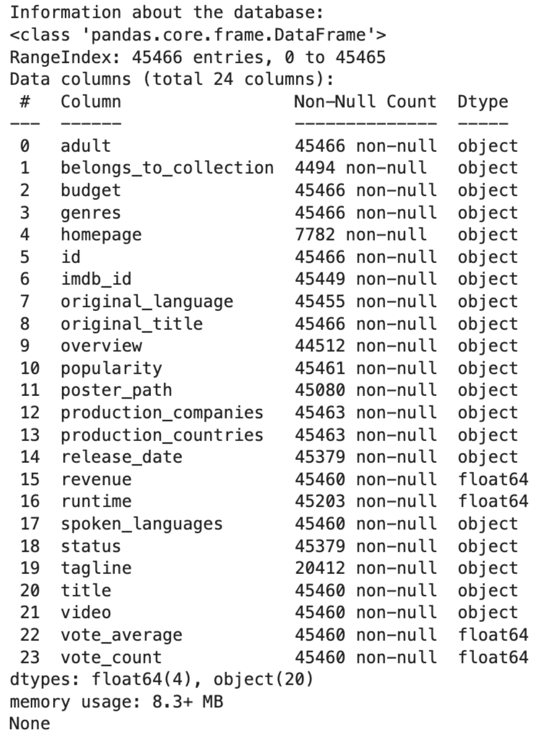


```python
# Define a function to clean the `genres` column
def clean_genres_column(genres):
    try:
        genre_list = ast.literal_eval(genres)
        return ', '.join([genre['name'] for genre in genre_list])
    except (ValueError, SyntaxError):
        return None

db['cleaned_genres'] = db['genres'].apply(clean_genres_column)
```

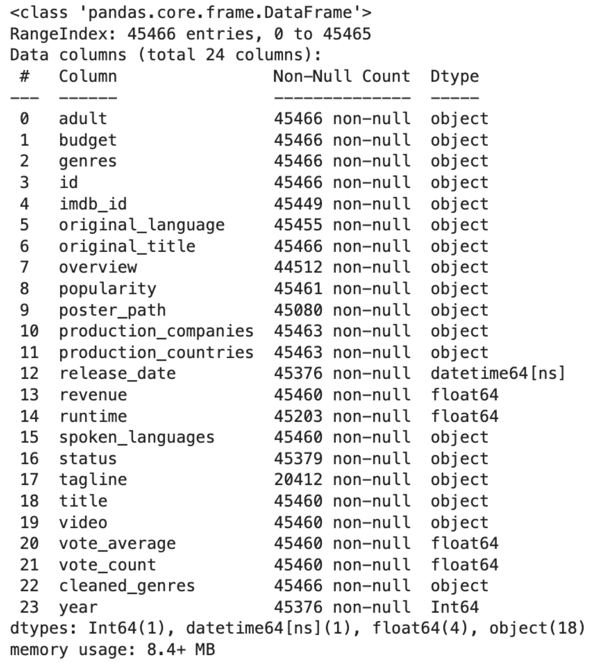
I also dropped unnecessary rows such as ‘belongs_to_collection’, ‘homepage’, as well as the original genre row. Since, the attribute that should be numerical values were a object type of the dataframe, I changed the those attributes such as ‘release date’, and ‘year’ to numerical format. Now we can see that the data type changed to int and datetime.



```python
db['release_date'] = pd.to_datetime(df['release_date'], errors='coerce')
db['year'] = db['release_date'].dt.year.astype('Int64')
db.info()
```



```
Information about the database:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45466 entries, 0 to 45465
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   adult                  45466 non-null  object 
 1   belongs_to_collection  4494 non-null   object 
 2   budget                 45466 non-null  object 
 3   genres                 45466 non-null  object 
 4   homepage               7782 non-null   object 
 5   id                     45466 non-null  object 
 6   imdb_id                45449 non-null  object 
 7   original_language      45455 non-null  object 
 8   original_title         45466 non-null  object 
 9   overview               44512 non-null  object 
 10  popularity             45461 non-null  object 
 11  poster_path            45080 non-null  object 
 12  production_companies   45463 non-null  object 
 13  production_countries   45463 non-null  object 
 14  release_date           45379 non-null  object 
 15  revenue                45460 non-null  float64
 16  runtime                45203 non-null  float64
 17  spoken_languages       45460 non-null  object 
 18  status                 45379 non-null  object 
 19  tagline                20412 non-null  object 
 20  title                  45460 non-null  object 
 21  video                  45460 non-null  object 
 22  vote_average           45460 non-null  float64
 23  vote_count             45460 non-null  float64
dtypes: float64(4), object(20)
memory usage: 8.3+ MB
None
```



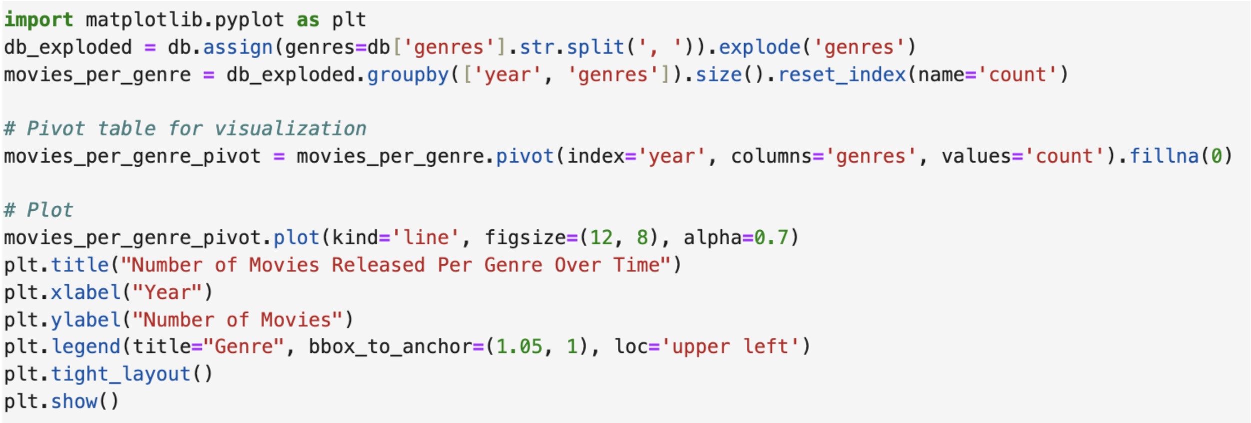
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45466 entries, 0 to 45465
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype         
---  ------                --------------  -----         
 0   adult                 45466 non-null  object        
 1   budget                45466 non-null  object        
 2   genres                45466 non-null  object        
 3   id                    45466 non-null  object        
 4   imdb_id               45449 non-null  object        
 5   original_language     45455 non-null  object        
 6   original_title        45466 non-null  object        
 7   overview              44512 non-null  object        
 8   popularity            45461 non-null  object        
 9   poster_path           45080 non-null  object        
 10  production_companies  45463 non-null  object        
 11  production_countries  45463 non-null  object        
 12  release_date          45376 non-null  datetime64[ns]
 13  revenue               45460 non-null  float64       
 14  runtime               45203 non-null  float64       
 15  spoken_languages      45460 non-null  object        
 16  status                45379 non-null  object        
 17  tagline               20412 non-null  object        
 18  title                 45460 non-null  object        
 19  video                 45460 non-null  object        
 20  vote_average          45460 non-null  float64       
 21  vote_count            45460 non-null  float64       
 22  cleaned_genres        45466 non-null  object        
 23  year                  45376 non-null  Int64         
dtypes: Int64(1), datetime64[ns](1), float64(4), object(18)
memory usage: 8.4+ MB
```

As, my goal of this project is to explore the movies dataset primarily focusing on the genre features, my project doesn't involve implementing significant algorithms but focused on exploring and analyzing the dataset. Instead, I used built in python libraries such as pandas, matplotlib, statsmodels to generate the insights into trends and patterns. Using the built in python libraries I was able to speed up data transformations and avoid slow, iterative row-wise operations. However, due to the dataset's size, comprising 45,466 records, generating visualizations required a notable amount of processing time. The time complexity of preprocessing steps was O(n), and the time complexity for the analysis steps leverage pandas' efficient built-in methods for groupings and aggregations.

Below is the full list for the exploration of this project.

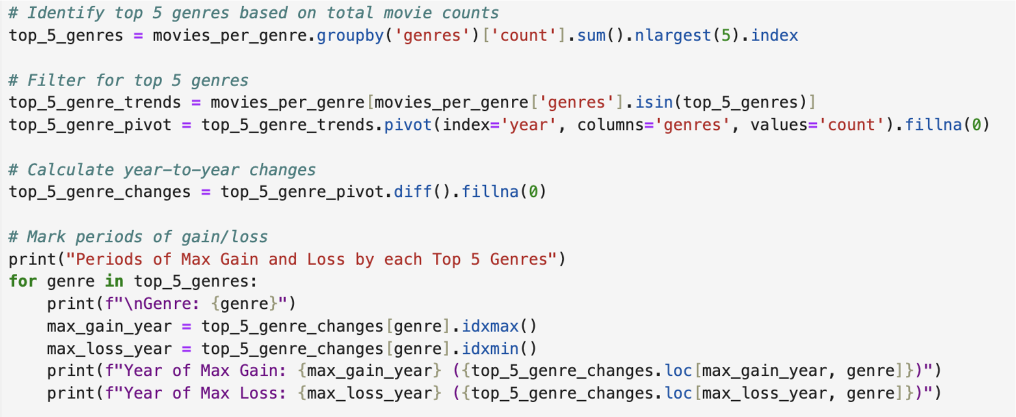
- Movie Released Trends per Genre Over Time
- Periods of Max Gain and Loss by Each Top 5 Genres
- Trends of Top 5 Genres Over Time
- Holt-Winters Forecast for Top 10 Genres
- Correlation Between Budget and Popularity for Top Genres
- Average Ratings and Consistency by Genre
- Correlation Between Runtime and Ratings

For "Movie Released Trends per Genre Over Time", I used matplotlib to plot the visualization. I first splited the genres into a list of individual genres. This allowed each genre in a movie gets its own row which helped visualizing the genre trends. I counted the number of movies for each genre per year, and created a matrix where rows are years, and columns are genres and values are the number of movies. Next, I plotted the trends. Below is the code for this exploration.



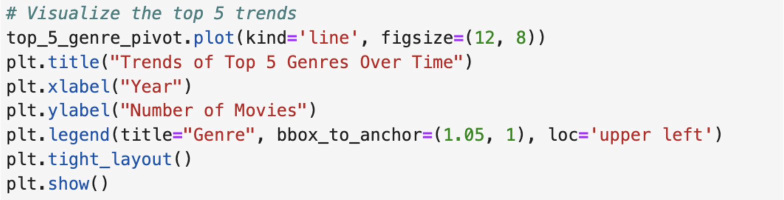
```python
import matplotlib.pyplot as plt
db_exploded = db.assign(genres=db['genres'].str.split(', ')).explode('genres')
movies_per_genre = db_exploded.groupby(['year', 'genres']).size().reset_index(name='count')

# Pivot table for visualization
movies_per_genre_pivot = movies_per_genre.pivot(index='year', columns='genres', values='count').fillna(0)

# Plot
movies_per_genre_pivot.plot(kind='line', figsize=(12, 8), alpha=0.7)
plt.title("Number of Movies Released Per Genre Over Time")
plt.xlabel("Year")
plt.ylabel("Number of Movies")
plt.legend(title="Genre", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

For "Periods of Max Gain and Loss by Each Top 5 Genres", I first identify the top 5 genres based on the total movie counts, and filtered the top 5 genres. Then, I created another attribute 'top_5_genre_changes', and calculated the year to year changes of the top 5 genres and I printed the periods of max gain and loss by each top 5 genres. Max gain year and loss year were calculated based on the top_5_genre_changes column.



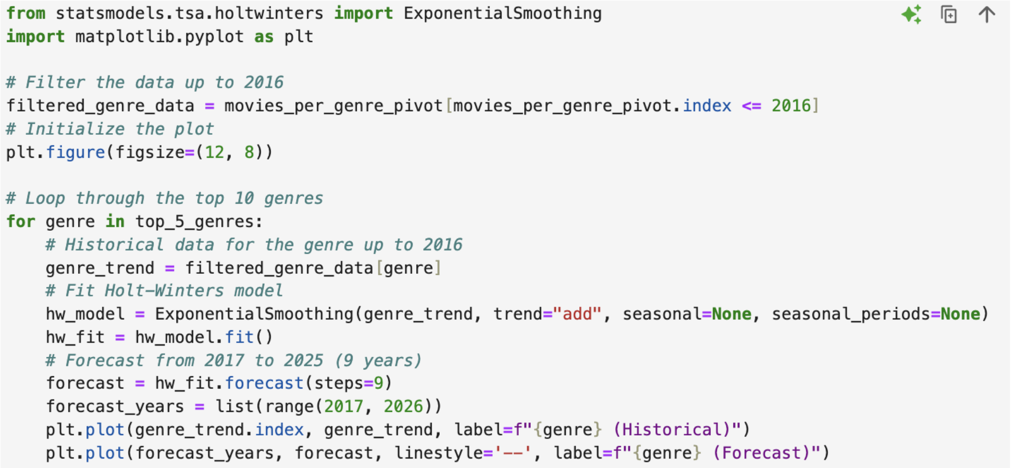
```python
# Identify top 5 genres based on total movie counts
top_5_genres = movies_per_genre.groupby('genres')['count'].sum().nlargest(5).index

# Filter for top 5 genres
top_5_genre_trends = movies_per_genre[movies_per_genre['genres'].isin(top_5_genres)]
top_5_genre_pivot = top_5_genre_trends.pivot(index='year', columns='genres', values='count').fillna(0)

# Calculate year-to-year changes
top_5_genre_changes = top_5_genre_pivot.diff().fillna(0)

# Mark periods of gain/loss
print("Periods of Max Gain and Loss by each Top 5 Genres")
for genre in top_5_genres:
    print(f"\nGenre: {genre}")
    max_gain_year = top_5_genre_changes[genre].idxmax()
    max_loss_year = top_5_genre_changes[genre].idxmin()
    print(f"Year of Max Gain: {max_gain_year} ({top_5_genre_changes.loc[max_gain_year, genre]})")
    print(f"Year of Max Loss: {max_loss_year} ({top_5_genre_changes.loc[max_loss_year, genre]})")
```

For "Trends of Top 5 Genres Over Time", I first identified the top 5 genres by their total movie counts and filtered the dataset accordingly. The data was aggregated by year and genres to calculate the annual count of movies for each genre. This aggregated data was transformed into a pivot table, with rows as years and columns as the top 5 genres. Using matplotlib, I created a line plot to visualize the trends, where each line represents the yearly trend for one genre.

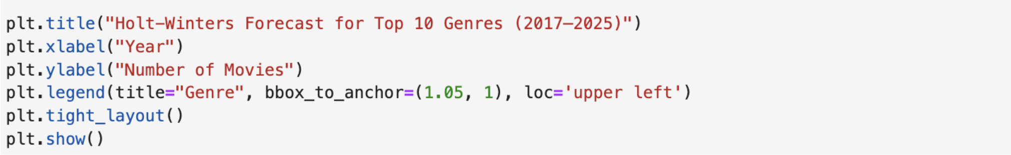


```python
# Visualize the top 5 trends
top_5_genre_pivot.plot(kind='line', figsize=(12, 8))
plt.title("Trends of Top 5 Genres Over Time")
plt.xlabel("Year")
plt.ylabel("Number of Movies")
plt.legend(title="Genre", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

For "Holt-Winters Forecast for Top 10 Genres", I focused on forecasting trends for the top 10 genres from 2017 to 2025 using the Holt-Winters Exponential Smoothing model. Historical data up to 2016 was used to fit the model for each genre. The model was configured with an additive trend and no seasonal component to account for the observed trends. Forecasts for each genre were generated for a nine-year period (2017–2025). Both the historical data and forecasts were visualized on a single plot, with the forecasts displayed using dashed lines for distinction.

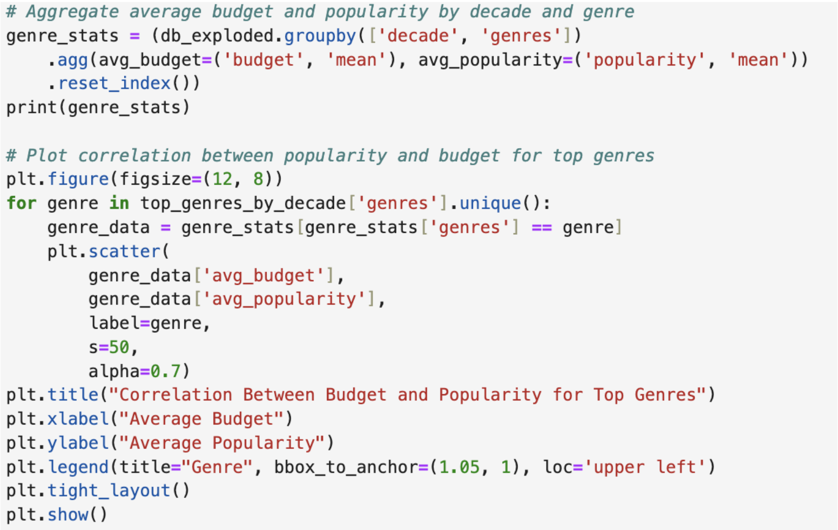


```python
from statsmodels.tsa.holtwinters import ExponentialSmoothing
import matplotlib.pyplot as plt

# Filter the data up to 2016
filtered_genre_data = movies_per_genre_pivot[movies_per_genre_pivot.index <= 2016]
# Initialize the plot
plt.figure(figsize=(12, 8))

# Loop through the top 10 genres
for genre in top_5_genres:
    # Historical data for the genre up to 2016
    genre_trend = filtered_genre_data[genre]
    # Fit Holt-Winters model
    hw_model = ExponentialSmoothing(genre_trend, trend="add", seasonal=None, seasonal_periods=None)
    hw_fit = hw_model.fit()
    # Forecast from 2017 to 2025 (9 years)
    forecast = hw_fit.forecast(steps=9)
    forecast_years = list(range(2017, 2026))
    plt.plot(genre_trend.index, genre_trend, label=f"{genre} (Historical)")
    plt.plot(forecast_years, forecast, linestyle='--', label=f"{genre} (Forecast)")
```



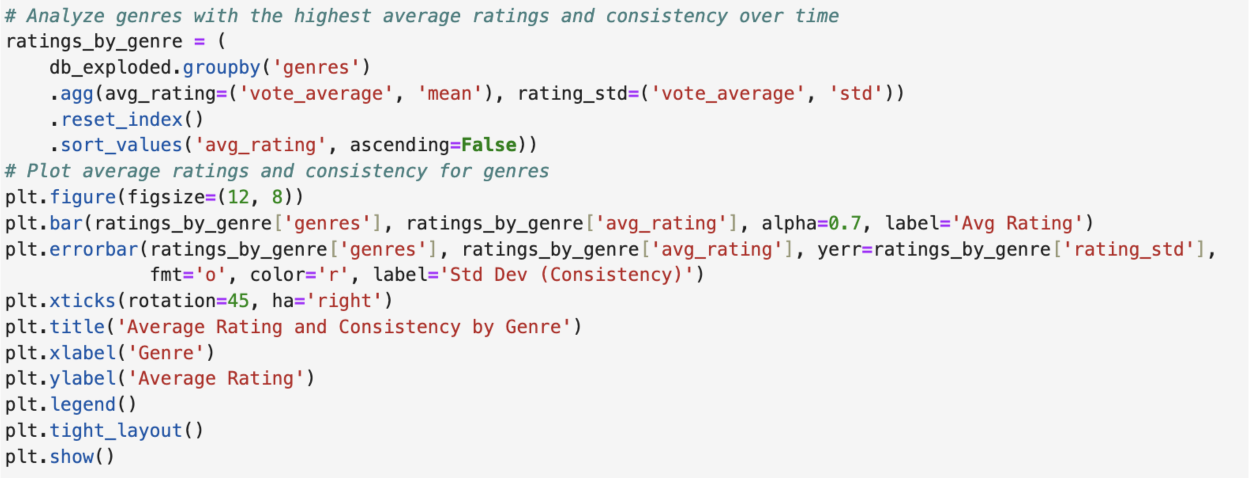
```python
plt.title("Holt-Winters Forecast for Top 10 Genres (2017–2025)")
plt.xlabel("Year")
plt.ylabel("Number of Movies")
plt.legend(title="Genre", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

For "Correlation Between Budget and Popularity for Top Genres", to explore the relationship between budget and popularity, I calculated the average budget and average popularity for each genre by decade. Using scatter plots, I examined the correlation between these two metrics for the top genres. Each genre was plotted with distinct markers to highlight differences, and the visualization revealed trends in how budget allocations might impact a movie's popularity within specific genres.



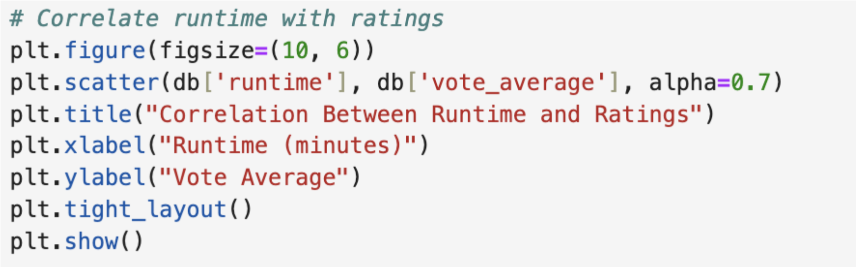
```python
# Aggregate average budget and popularity by decade and genre
genre_stats = (db_exploded.groupby(['decade', 'genres'])
    .agg(avg_budget=('budget', 'mean'), avg_popularity=('popularity', 'mean'))
    .reset_index())
print(genre_stats)

# Plot correlation between popularity and budget for top genres
plt.figure(figsize=(12, 8))
for genre in top_genres_by_decade['genres'].unique():
    genre_data = genre_stats[genre_stats['genres'] == genre]
    plt.scatter(
        genre_data['avg_budget'],
        genre_data['avg_popularity'],
        label=genre,
        s=50,
        alpha=0.7)
plt.title("Correlation Between Budget and Popularity for Top Genres")
plt.xlabel("Average Budget")
plt.ylabel("Average Popularity")
plt.legend(title="Genre", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

For “Average Ratings and Consistency by Genre”, I calculated the average rating (vote_average) and its standard deviation (rating_std) for each genre to assess both quality and consistency. Genres were sorted by their average ratings to identify the highest-rated ones. A bar chart was used to visualize the average ratings for each genre, with error bars representing the standard deviation.



```python
# Analyze genres with the highest average ratings and consistency over time
ratings_by_genre = (
    db_exploded.groupby('genres')
    .agg(avg_rating=('vote_average', 'mean'), rating_std=('vote_average', 'std'))
    .reset_index()
    .sort_values('avg_rating', ascending=False))
# Plot average ratings and consistency for genres
plt.figure(figsize=(12, 8))
plt.bar(ratings_by_genre['genres'], ratings_by_genre['avg_rating'], alpha=0.7, label='Avg Rating')
plt.errorbar(ratings_by_genre['genres'], ratings_by_genre['avg_rating'], yerr=ratings_by_genre['rating_std'],
             fmt='o', color='r', label='Std Dev (Consistency)')
plt.xticks(rotation=45, ha='right')
plt.title('Average Rating and Consistency by Genre')
plt.xlabel('Genre')
plt.ylabel('Average Rating')
plt.legend()
plt.tight_layout()
plt.show()
```

For “Correlation Between Runtime and Ratings”, to examine the relationship between a movie's runtime and its ratings, I created a scatter plot of runtime against vote_average. The plot visually illustrates whether longer movies tend to receive higher or lower ratings.

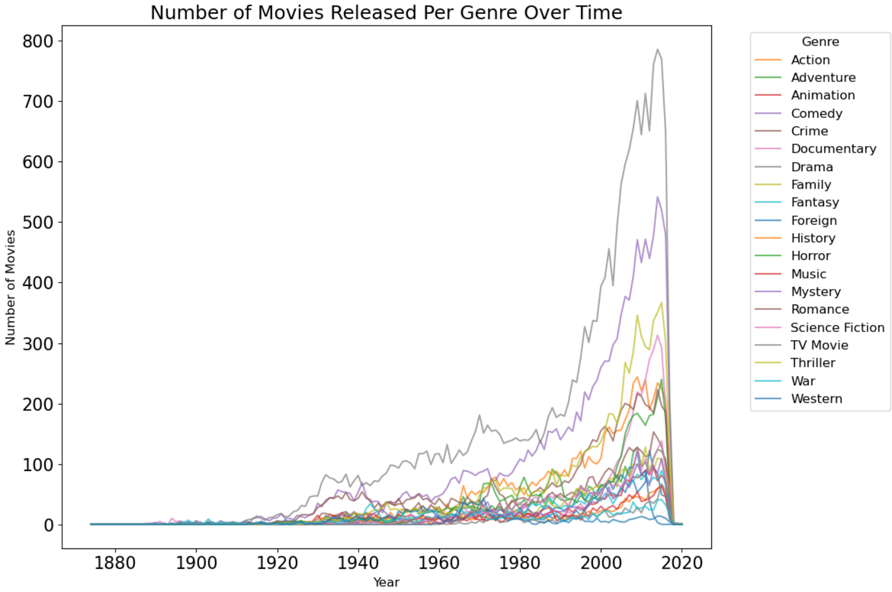


```
# Correlate runtime with ratings
plt.figure(figsize=(10, 6))
plt.scatter(db['runtime'], db['vote_average'], alpha=0.7)
plt.title("Correlation Between Runtime and Ratings")
plt.xlabel("Runtime (minutes)")
plt.ylabel("Vote Average")
plt.tight_layout()
plt.show()
```

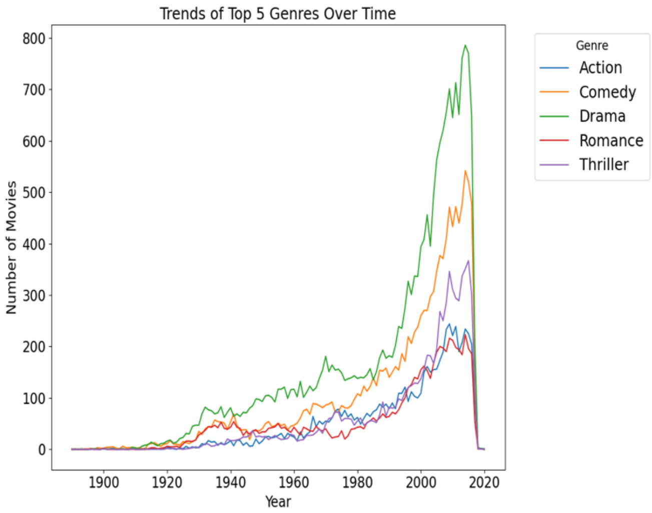
To demonstrate the validity of my project, I did the proof of the runtime analysis for the key parts of my project although no custom algorithms were implemented. For the data cleaning process, the runtime big O is O(n·m) since n is the number of rows and m is the average number of genres per movie. This ensured that transformations were completed in a reasonable amount of time despite the size of the dataset. For generating the visualization, I was able to generate efficiently using pandas and matplotlib although the dataset I used required a significant amount of time to plot especially for multi-line plots.

### Evaluation/Experimental Study

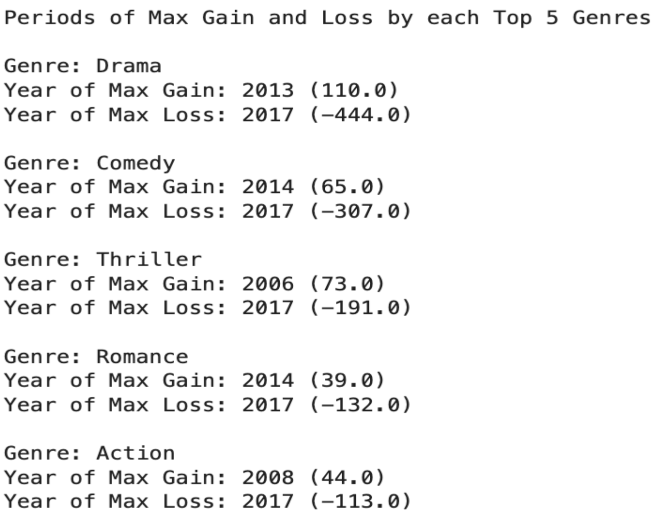
The dataset that I used is a movie metadata dataset sourced from the Kaggle. The dataset contains 45,466 rows and 24 columns, including features like genres, budget, revenue, runtime, popularity, release_date, and vote_average. It was semi-structured data with mixed data types. Data types were both in float type and object type, although columns like genres and spoken_languages were nested JSON formatted values. Revenue, runtime, vote_average, and vote_count were numerical float type. Each row represents a movie with attributes such as genre, budget, revenue, and popularity that enable insights into performance, and trends. I used Python for this entire project, using the data manipulation library pandas, and matplot for the visualization. Data preprocessing was conducted including splitting stringified genres into individual rows for better analysis as well as handling missing values before the analysis of the data.



Above is the observation of the number of movies released per genre over time. Based on the visualization, there has been a steady increase in the number of movies released per genre over the decades, particularly post-1980. This is because that was the time where movie industry being popular to people and massive of people visited cinemas to watch movies for the first time in their life. Some genres like drama, comedy, and action exhibits the highest growth rates which shows their royal popularity and adaptability to audience preferences. By looking at the genre diversification we can see that the consumers demand were varied by each genre. Since, the visualization for every genre was not able to provide a good readability, I generated the visualization that shows the number of movies released for top 5 genre over time.

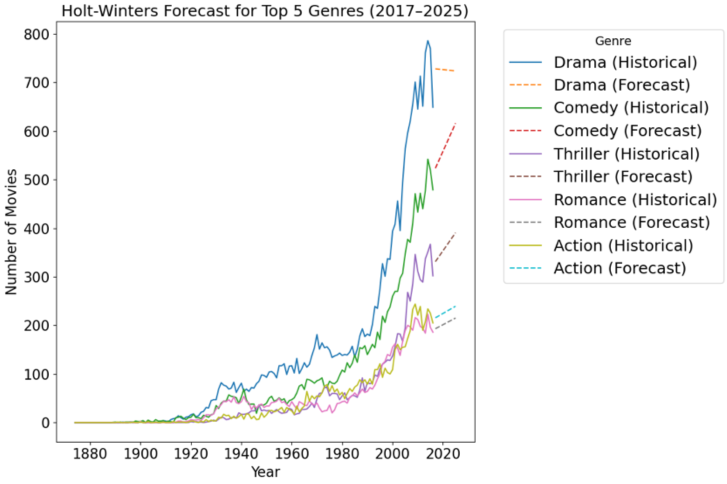


Now, we can clearly see the top 5 genres over time, which followed by drama, comedy, thriller, action and romance. Genre drama had the significant dominant throughout the years, peaking especially in early 2010s. Comedy genres also showed the consistent upward trends indicating the broad appeal. Thriller had a similar growth until mid 2000's it grew massively after mid 2000's resulted the next significant genres within top 5. Action and romance genre showed similar trend but action genre a little bit higher than the romance genre overall.

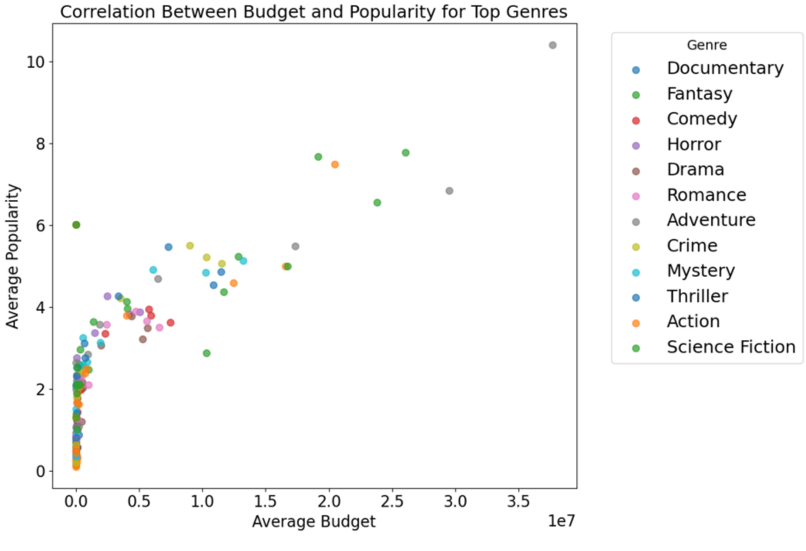


```
Periods of Max Gain and Loss by each Top 5 Genres

Genre: Drama
Year of Max Gain: 2013 (110.0)
Year of Max Loss: 2017 (-444.0)

Genre: Comedy
Year of Max Gain: 2014 (65.0)
Year of Max Loss: 2017 (-307.0)

Genre: Thriller
Year of Max Gain: 2006 (73.0)
Year of Max Loss: 2017 (-191.0)

Genre: Romance
Year of Max Gain: 2014 (39.0)
Year of Max Loss: 2017 (-132.0)

Genre: Action
Year of Max Gain: 2008 (44.0)
Year of Max Loss: 2017 (-113.0)
```

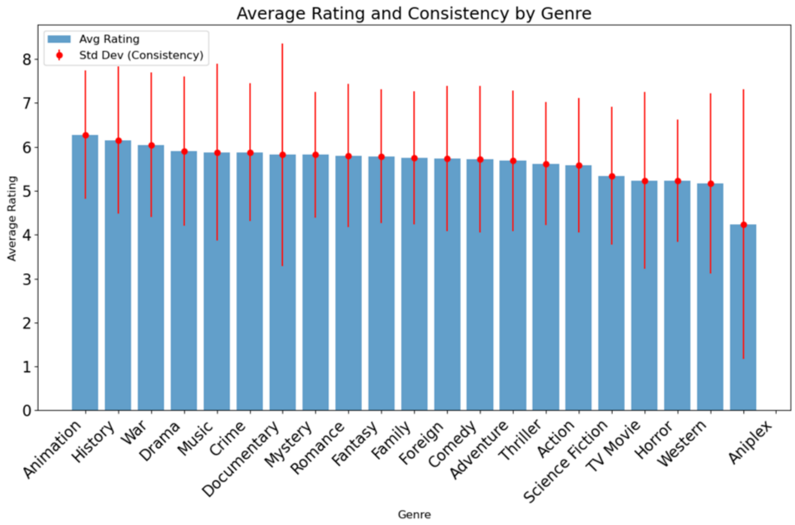
I printed the periods of max gain and loss by each top 5 genres. Surprisingly, we can observe that for every top 5 genre had a year of max loss in 2017. This is a clear abnormal of the data, and this might be due to the lack of data or incorrect information of the datasets. However, considering the dataset were a legit dataset that I downloaded from Kaggle that more than thousands of people used to analyze the movies dataset. Therefore, I researched about the movie trends in 2017, what might have been affected the movie industry and resulted the result. It was found out that there was a massive increase in OTT platforms such as Netflix, and HBO in 2016-2017 (Plaugic). This indicates a huge increase in the online streaming industry impacted negatively on the traditional style movie industry and had a huge decrease in 2017. This is a crucial finding since it shows that the movie industry are actually affected by different types factors, increase in online streaming services in this case.



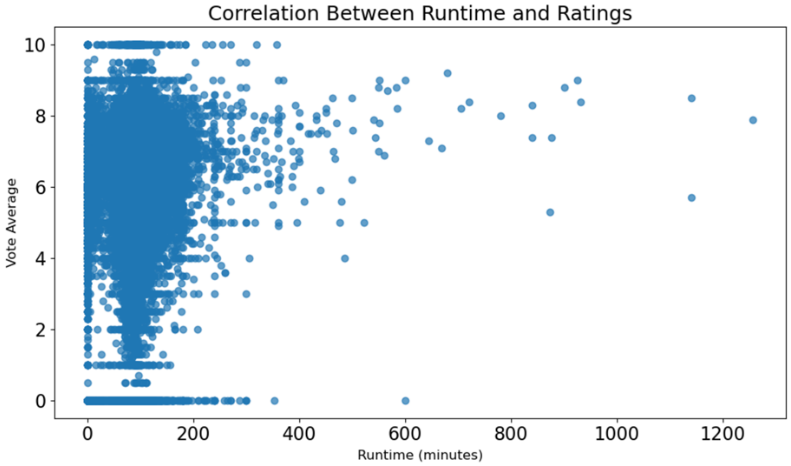
Since, I was looking at the movie genre's trend, I wanted to forecast the trend of top 5 genres from 2017 to 2025. I decided to train the data from post to 2016, since this dataset contains the movie released until 2020, but the amount of movie released shown in the dataset decrease starting from 2019, I wanted to exclude the outlier that might cause by training the entire dataset and forecast beyond. The result showed a steady increase for comedy, thriller, action, and romance genres except drama which showed a slightly decreased in the forecasted trend. Since, we know the actual trend from 2017 to 2020, we know that the forecast of comedy, thriller, action, and romance genres is not true. This indicates even we forecast the data for the next future, we cannot predict about the external factor that would affect the entire trend significantly. As it only trained by the historical trend, it is less likely to be accurate as the actual trend that will happen in the next few years. Still, we can predict which genres will get a dominance based on the holt-winters model as the dominance of the drama genre remained the same.



The whole idea of this project is to do some exploration in movies dataset especially related with genre trends, I looked at the relationship between the budget and popularity by genres. The visualization indicates for some genres such as action, sci-fi, and fantasy it showed some sort of positive correlation between the budget and popularity, but that doesn't always guarantee the popularity. On the other hand, genres like comedy and documentary were often made with smaller budgets still maintained a steady level of popularity highlighting higher budget allocations might not always translate to audience success, suggesting other factors like storytelling or marketing plays a crucial role.



I also created a visualization audience reception through average ratings and consistency by genres. Genres like animation and history had higher average ratings but it also exhibited greater standard deviations meaning, there exist a big interval between the top rated movies and the low rated movies of animation and history genres. Consistency is comparatively higher in genres like Crime, and mystery genres although it still showed pretty high standard deviation meaning that ratings of each movie did not aligned similar suggesting ratings are depend on the movies itself individually based on how well they were made rather than depending on the genre. We can conclude that movie ratings are not related with each genre. In other words, producing a movie of a certain genre does not guarantee a high movie rating.



Finally, I was wondering if ratings and runtime do have a relationship each other. Based on the visualization, there are no strong relationship between runtime an ratings meaning movies with varied runtimes achieve similar ratings. We can also observe that movies that are longer than 3 hours are extremely rare and may polarize audiences. This shows quality and storytelling are more crucial than runtime to have a success for a movie.

**Conclusion/Future Work**

Overall, I observed the movie trends by genres, and the relationship between different features. I found out that drama genre had a significant dominant throughout the trend. The abnormality in 2017 of the movie trend was also observed as all top 5 genres had a max loss in 2017, suggesting a external factor that cause the significant drop on the trend. Additionally, a positive correlation was found between budget and popularity, with higher-budget movies generally achieving greater popularity, though this was not always guaranteed. Finally, it was found out that producing movies in certain genre does not guarantee the ratings.

There were some limitations with this project. Although 45,500 columns of dataset is not a small number of datasets considering how big the movie industry is, the more data could have a better understanding as well as the bias in data collection. The data itself wasn't up to date, the data wasn't a lot starting from 2019. In general, this project had shed light on the dominance and evolution of genres over decades finding key correlations between movie attributes.

For future work, deeper exploration into the 2017 anomaly could help uncover specific causes, such as industry changes or shifts in consumer behavior. Expanding the analysis to include more nuanced features, such as audience reviews or regional data, could provide additional insights. Applying machine learning models for genre-based predictions or recommendation systems could further enhance the value of this analysis.

**Works Cited**

Banik, Rounak. “The Movies Dataset.” Kaggle, 10 Nov. 2017, www.kaggle.com/datasets/rounakbanik/the-movies-dataset/data?select=movies_metadata.csv.

Plaugic, Lizzie. “Domestic Movie Theater Attendance Hit a 25-Year Low in 2017.” The Verge, The Verge, 3 Jan. 2018, www.theverge.com/2018/1/3/16844662/movie-theater-attendance-2017-low-netflix-streaming?utm_source=chatgpt.com.

Kim, Iksuk, and HannEarl Kim. "The More, the Better? Movie Genre and Performance Analysis." Journal of Business and Educational Leadership 7.1 (2018): 105-113.

Raphael, Rudly. “Consumers and the Future of the Film Industry.” Greenbook, www.greenbook.org/insights/executive-insights/consumers-and-the-future-of-the-film-industry. Accessed 11 Dec. 2024.

Zhan, Choujun, Jianjin Li, and Wei Jiang. "An Empirical Investigation on Movie Industry from 1980 to 2018." 2018 IEEE Symposium on Product Compliance Engineering-Asia (ISPCE-CN). IEEE, 2018.