A

PROJECT DOCUMENTATION

ON

BOX OFFICE SCRAPING, ANALYSIS AND AUTOMATION

SUBMITTED BY

Amit Duwal

SUBMITTED TO

CodeRush, Nepal

May 29, 2023

Introduction

In the dynamic landscape of the entertainment industry, understanding box office performance is crucial for filmmakers, distributors, and movie enthusiasts. The project "Box Office Scraping, Analysis, and Automation" tackles this need by leveraging data scraping, analysis, and automation techniques to collect, analyze, and provide valuable insights from box office data.

Our project focuses on daily data collection from *boxofficemojo.com*, a reputable and reliable source for box office information. Using advanced web scraping techniques, we extract key details such as movie titles, release dates, theaters, ticket sales, and box office revenues. This data is then stored in a structured manner, such as a CSV file, ensuring easy access and compatibility for further analysis.

To ensure efficiency and regular updates, we employ automation through Airflow, a powerful open-source platform for workflow scheduling and orchestration. By utilizing Airflow, the data collection process is automated to run daily, providing a seamless and reliable stream of box office data. This automation ensures that our analysis is consistently up-to-date and reflective of the latest industry trends.

Once the data is collected, we embark on an in-depth analysis of box office trends. Through sophisticated data analytics techniques, we explore various factors that contribute to a movie's success. Our analysis encompasses identifying top-earning movies, recognizing high-performing distributors, and uncovering influential genres and directors. By extracting patterns and correlations from the data, we offer valuable insights to industry stakeholders, enabling them to make informed decisions.

While the project does not specifically include user-friendly dashboards, the collected data is readily available for further analysis and visualization.

Stakeholders can utilize popular data analysis tools, such as Excel, Python libraries, or business intelligence platforms, to explore and interpret the data. These tools empower users to navigate through different movies, compare earnings, and gain a comprehensive understanding of box office performance.

The "Box Office Scraping, Analysis, and Automation" project revolutionizes the way box office insights are obtained and utilized in the entertainment industry. By combining web scraping, data analysis, and automation, we provide a robust foundation for accessing, analyzing, and leveraging box office data. This project aims to empower industry professionals, researchers, and movie enthusiasts with the necessary tools and knowledge to navigate the dynamic world of box office success and contribute to the growth and sustainability of the film industry.

Requirements

To run the Stock Price Checker, you will need the following:

- Python
- Scrapy
- Airflow
- Pandas
- Scikit-Learn

Features

1. Data Scraping:

The project utilizes advanced web scraping techniques to collect box office data from *boxfficemojo.com*, a reliable source for industry-specific information. It retrieves essential details such as movie titles, release dates, theaters, and box office revenues.

2. Daily Data Updates:

The project is designed to run daily, ensuring the collection of up-to-date box office data. Through automation using Airflow, the data scraping process is scheduled and executed automatically, providing a continuous stream of fresh data for analysis.

3. Data Storage:

The collected box office data is stored in a structured format, such as a CSV file, making it easily accessible and compatible with various data analysis tools. The structured storage facilitates seamless integration with different software and enables efficient data retrieval.

4. Data Analysis:

The project performs comprehensive analysis on the collected box office data, exploring trends, patterns, and correlations. It examines factors such as top-earning movies, successful distributors, influential genres, and impactful directors, providing valuable insights for decision-making.

5. Automation with Airflow: Airflow, an open-source platform, is leveraged to automate the data collection process. By scheduling and orchestrating workflows, Airflow ensures that the data scraping occurs daily without the need for manual intervention, maintaining a consistent and reliable data stream.

6. Compatibility with Data Analysis Tools:

The collected data can be seamlessly integrated with popular data analysis tools such as Excel, Python libraries, or business intelligence platforms. This compatibility enables users to leverage their preferred tools for further exploration, visualization, and interpretation of the box office data.

7. Industry Empowerment:

The project aims to empower stakeholders in the entertainment industry, including filmmakers, distributors, and researchers, by providing them with valuable insights and data-driven decision-making capabilities. The project's features enable industry professionals to gain a comprehensive understanding of box office performance and make informed strategic choices.

Technologies used

The "Box Office Scraping, Analysis, and Automation" project employs a range of technologies to collect, analyze, and automate the processing of box office data. Each technology plays a vital role in different stages of the project.

1. Python:

Python is a versatile and widely-used programming language known for its simplicity and readability. It serves as the primary language for the project, providing a robust ecosystem of libraries and tools. Python's extensive support for web scraping, data analysis, and automation makes it an ideal choice for this project.

2. Scrapy:

Scrapy is a powerful web scraping framework in Python. It offers a high-level and efficient approach to extract data from websites. In the project, Scrapy is utilized to crawl through BoxOfficeMojo.com and extract relevant box office data. It handles the complexities of handling website structure, asynchronous requests, and data parsing, allowing for efficient and reliable data extraction.

3. Airflow:

Airflow is an open-source platform used for workflow management and automation. It provides a flexible and scalable solution to schedule, monitor, and

execute workflows. In this project, Airflow is leveraged to automate the daily data scraping process. It allows the scheduling of scraping tasks, manages dependencies, and handles task execution, ensuring a consistent and reliable stream of box office data.

4. Pandas:

Pandas is a popular data manipulation and analysis library in Python. It offers powerful data structures, such as DataFrames, and functions for data cleaning, manipulation, and exploration. In the project, Pandas is utilized to process and manage the collected box office data. It enables tasks such as filtering, aggregating, and transforming the data, facilitating efficient data analysis.

5. Scikit-Learn:

Scikit-Learn is a widely-used machine learning library in Python. It provides a rich collection of algorithms and tools for data analysis, modeling, and predictive analytics. In the project, Scikit-Learn is employed for advanced data analysis tasks. It enables the identification of patterns, correlations, and trends within the box office data. Machine learning algorithms offered by Scikit-Learn can be utilized for tasks such as classification and regression to gain deeper insights from the data.

Project Description

Scraping Process

The scraping process in the "Box Office Scraping, Analysis, and Automation" project is facilitated by the provided script.

1. Spider Initialization:

- The script defines a class named `BoxSpider`, which inherits from the Scrapy `Spider` class.
- The spider is named `"boxoffice"` using the `name` attribute.

- The `base_url` is set to "https://www.boxofficemojo.com".
- A list of `years` is specified, representing the years for which box office data will be scraped.

2. Start Requests:

- The `start_requests()` method is responsible for generating scrapy.Request objects for each URL based on the provided years.
- It constructs the URLs using the `base_url` and the specified years.
- For each URL, a scrapy.Request object is yielded with the specified callback function, `parse`, for further processing.

3. Parsing Response:

- The `parse()` method serves as the callback function for processing the response from the URLs generated in `start_requests`.
- It first extracts the movie elements from the response using CSS selectors.
- The number of movies found on the page is printed to the console.
- For each movie, the script follows the link to the movie's detail page using `response.follow()` and specifies the callback function `parse_movie1` for further processing.

4. Parsing Movie Details:

- The `parse_movie1()` method is the callback function for processing the response from the movie detail pages.
- It extracts various movie details such as title, description, box office earnings (domestic, international, worldwide), distributor, opening amount, release date, MPAA rating, running time, genres, in-release status, and widest release.

- The extracted data is then yielded as a dictionary object, representing a single movie.

5. Handling Various Movie Formats:

- The script handles different formats of movie pages encountered on boxofficemojo.com using conditional statements and CSS selectors.
- It accounts for variations in the order and availability of movie details such as MPAA rating, budget, opening amount, and running time.
- The script adapts the data extraction process accordingly to ensure accurate collection of relevant movie information.

Scraped Data

The data scraped by the "Box Office Scraping, Analysis, and Automation" project includes various details related to movies obtained from BoxOfficeMojo.com. Here are the key data points that have been collected:

1. Movie Information:

- Title: The title of the movie.
- Description: A brief description or summary of the movie.

2. Box Office Earnings:

- Domestic Box Office (Domestic_BO): The earnings of the movie in the domestic market.
- International Box Office (International_BO): The earnings of the movie in the international market.
- Worldwide Box Office (Worldwide_BO): The overall earnings of the movie worldwide.

3. Distribution Information:

- Distributor: The company or entity responsible for distributing the movie.

4. Release Details:

- Opening Amount: The amount earned during the opening weekend or initial release period.

- Release Date: The date when the movie was released.

- MPAA Rating: The rating provided by the Motion Picture Association of America, indicating the appropriate audience for the movie.

- Running Time: The duration of the movie in minutes.

5. Genre Information:

- Genres: The genres or categories to which the movie belongs. Multiple genres may be listed.

6. Movie Status:

- In Release: The status of the movie, indicating how long movie was in theaters

- Widest Release: The maximum number of theaters in which the movie was simultaneously released.

7. Additional Details:

- Budget: The estimated or reported budget for producing the movie.

Analysis

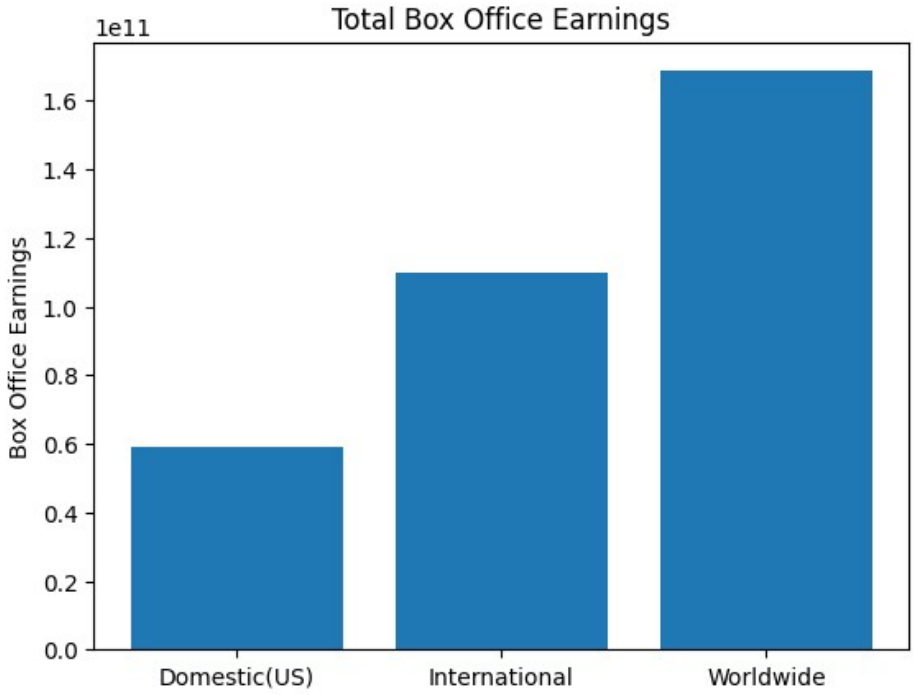
In the analysis segment of the project, a more thorough analysis of the scraped data was performed, particularly by incorporating historical box office data from 2016. The focus was on exploring correlations and conducting comprehensive visualizations to gain deeper insights into the box office trends and patterns.

While the daily analysis primarily involved simpler analyses, such as identifying top earners or distributors, the inclusion of historical data allowed for more extensive exploration and examination of relationships between various factors. Through statistical techniques, such as correlation analysis, it was possible to uncover connections between variables like release date, budget, genre, and box office earnings.

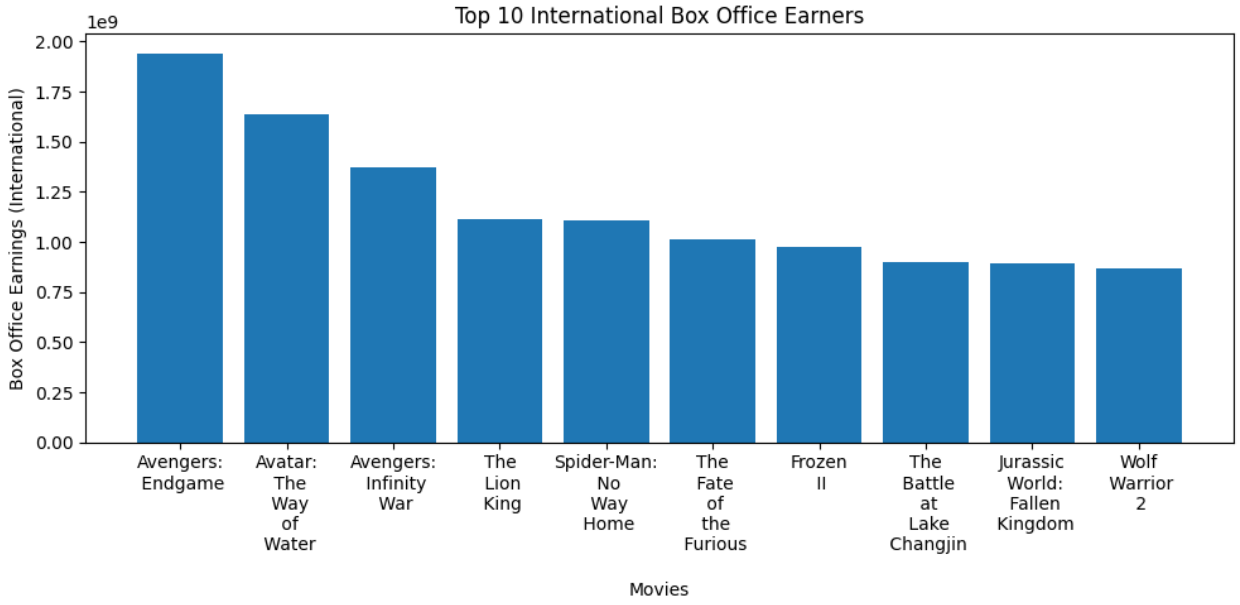
Additionally, visualization techniques were employed to create intuitive and informative charts, graphs, and plots, enabling a clear representation of the analyzed data and facilitating the identification of key trends and patterns. By incorporating historical data and conducting more in-depth analysis, the project aimed to provide a comprehensive understanding of box office dynamics, aiding decision-making and strategic planning in the film industry.

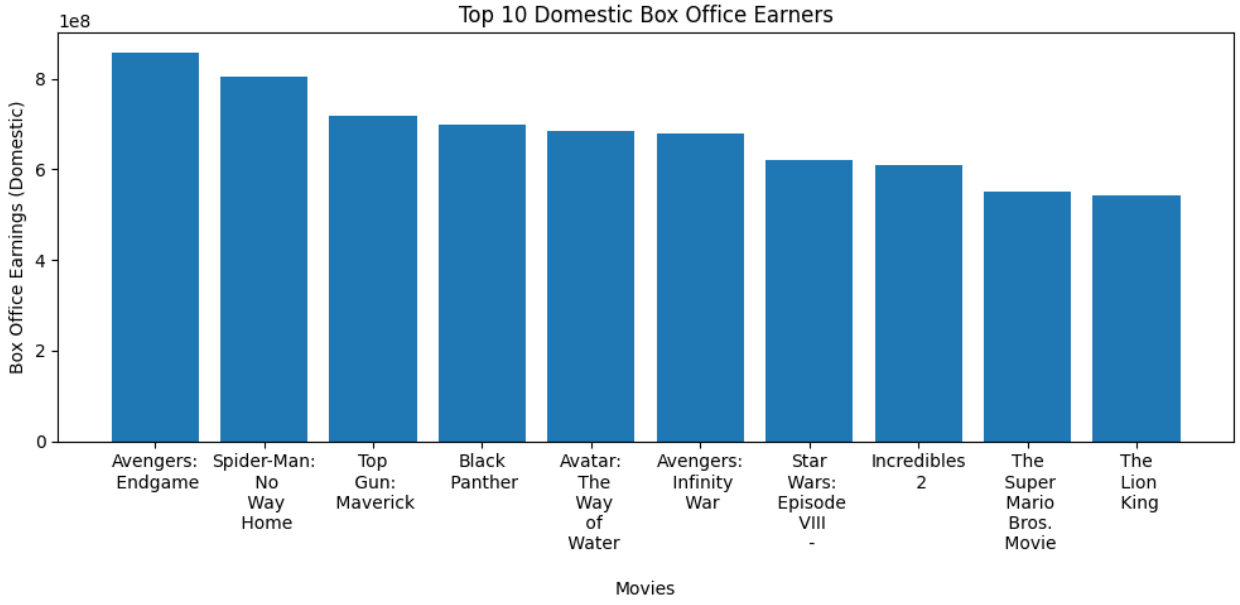
Some of the visualizations are presented below:

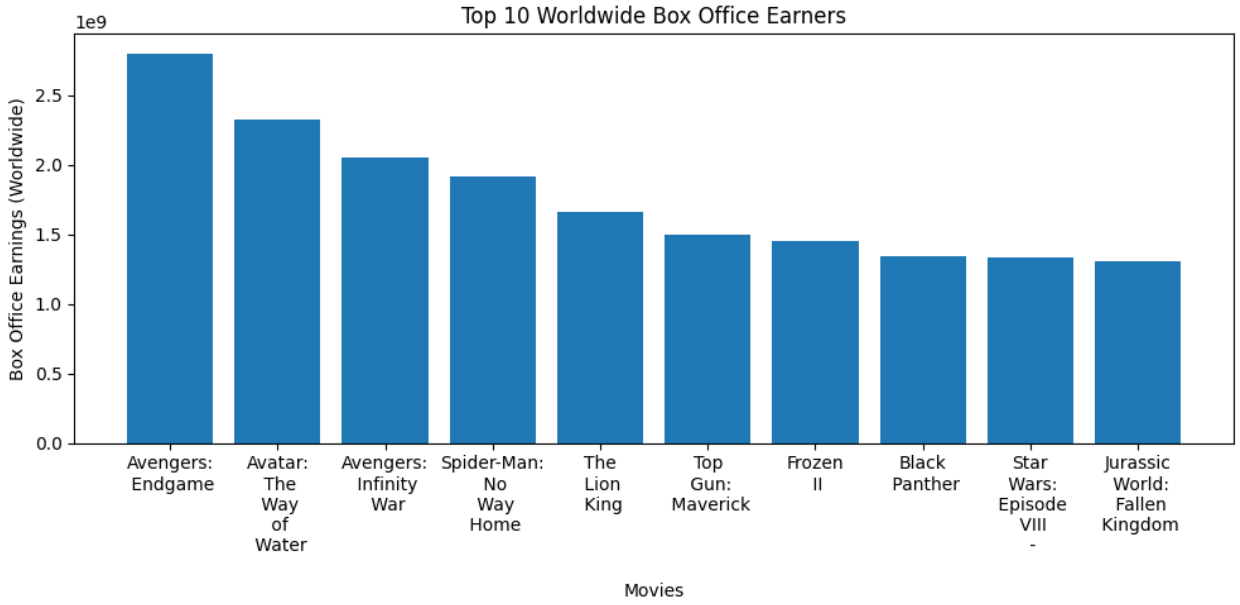
A comparision between Domestic(US), and International Box Office.



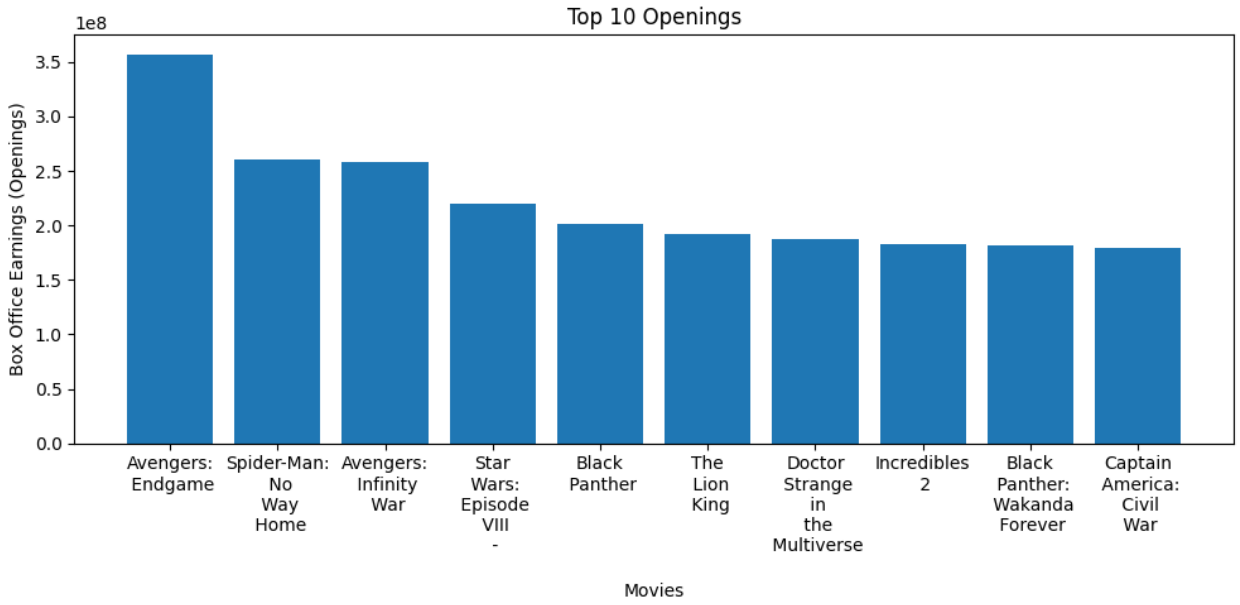
Top 10 movies after 2016



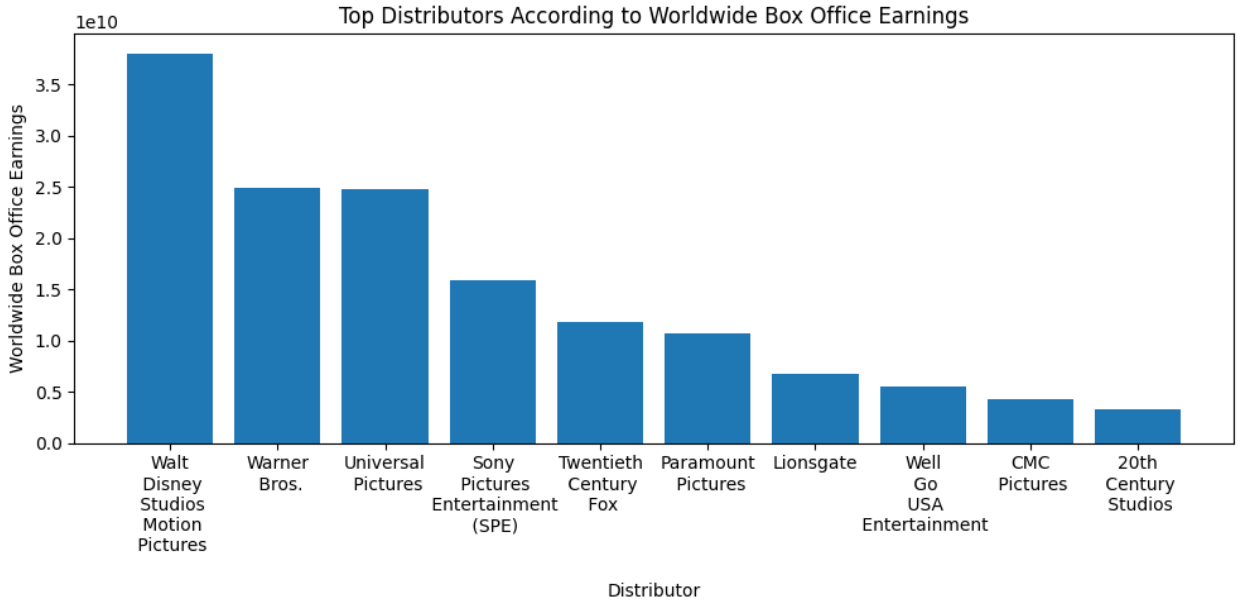


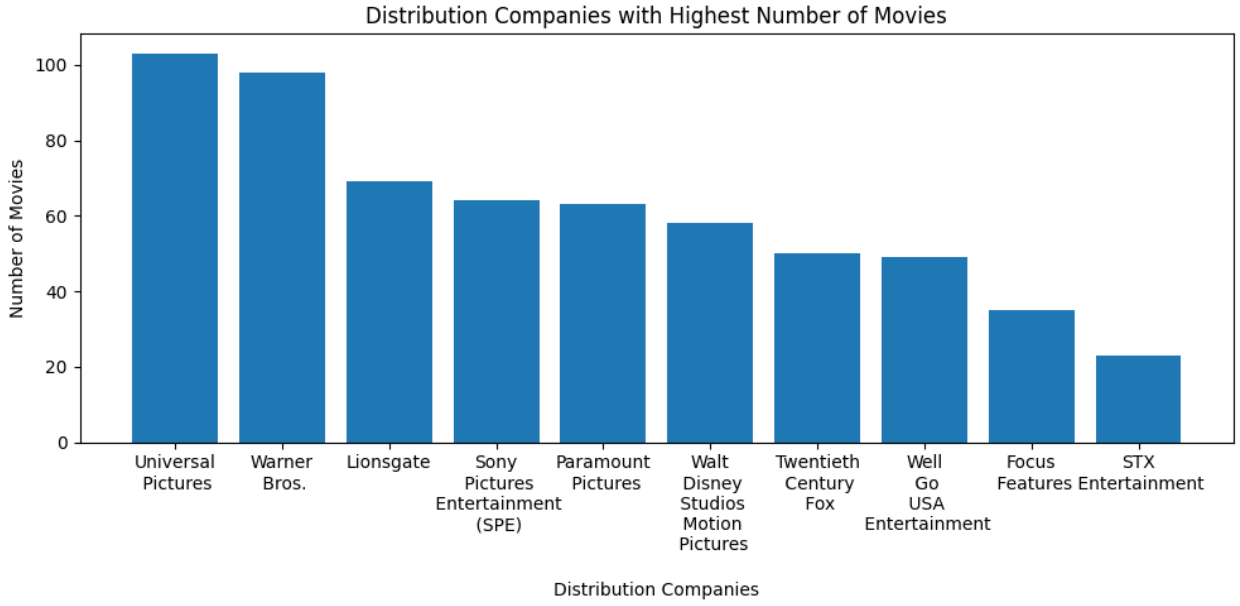


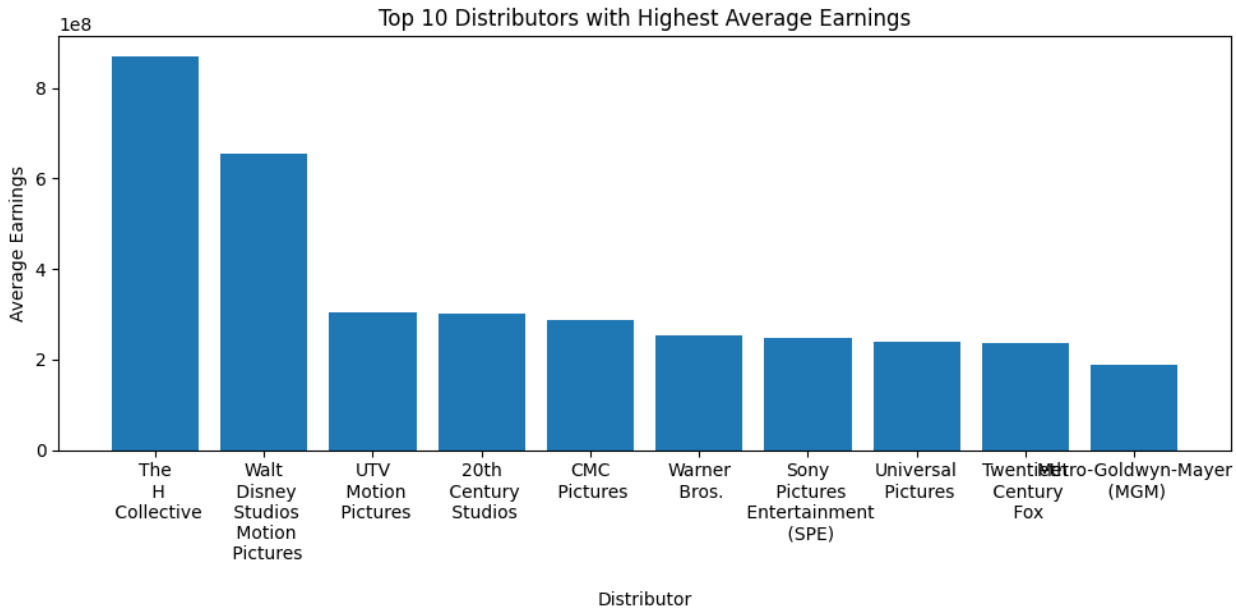
Top 10 movie openings



Comparision among distributors







MPAA ratings analyses

The new ratings system began with four categories:

- G (general audiences)

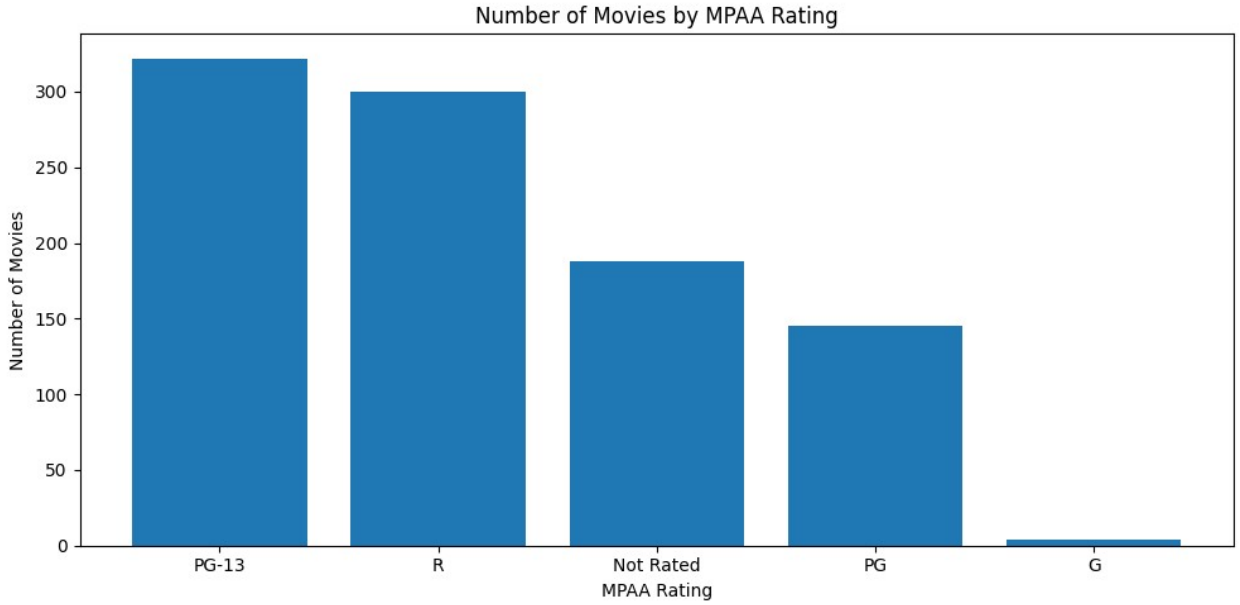
- M (mature audiences, changed in 1969 to PG parental guidance suggested)

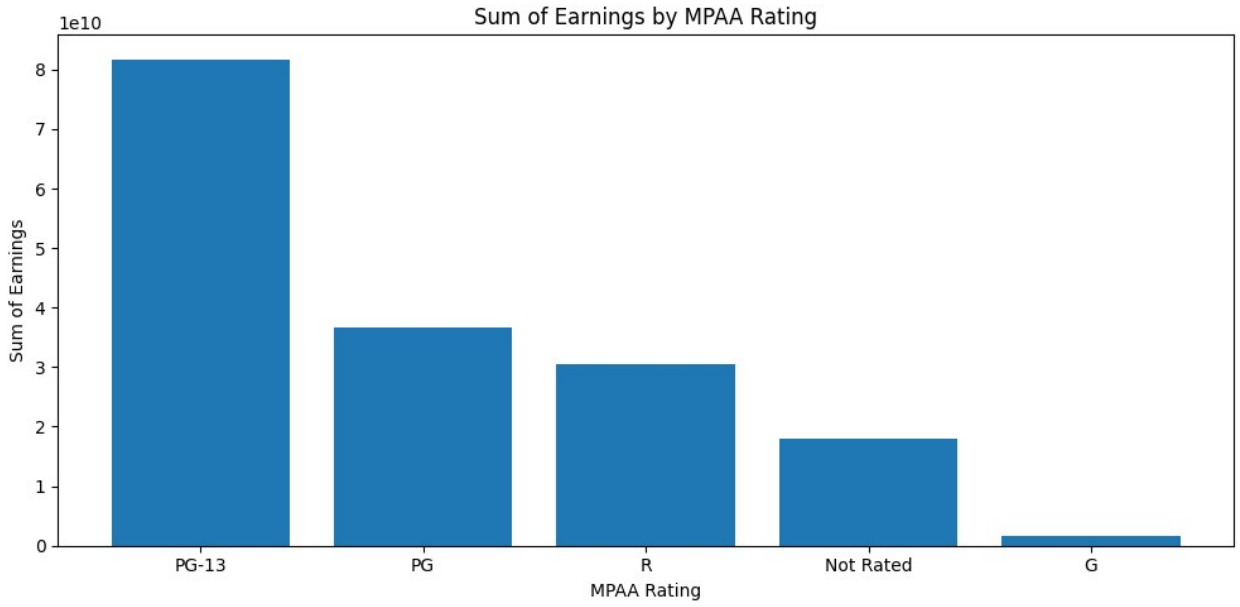
- R (restricted, no children under 17 allowed without parents or adult guardians) and

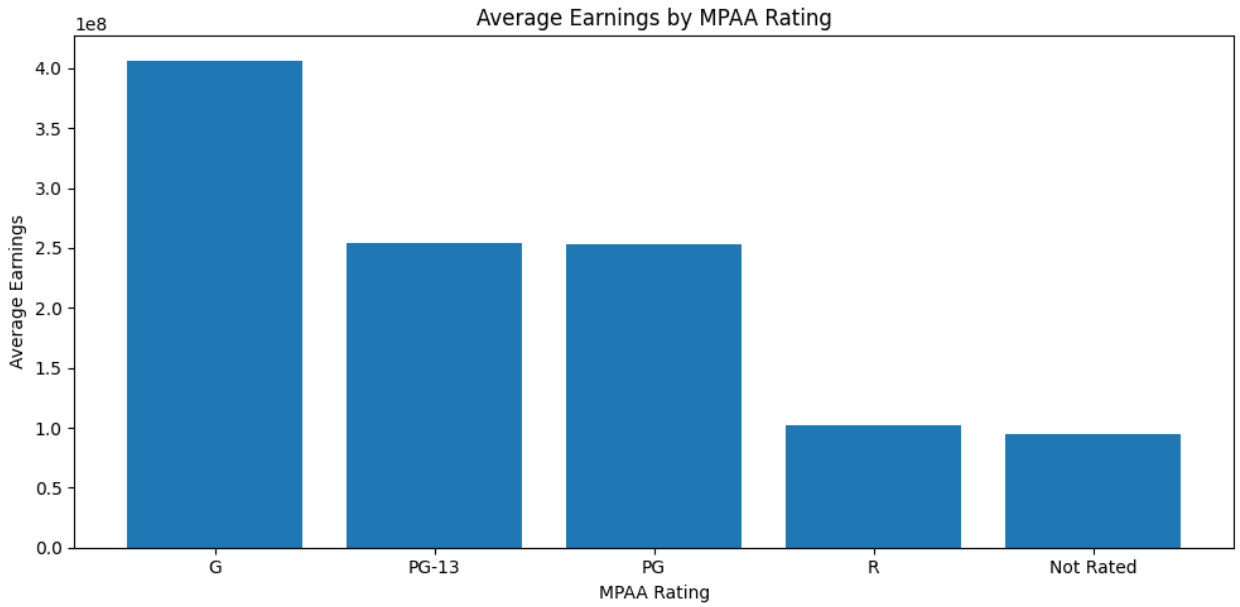
- X (no one under 17 admitted)

The ratings were revised several times over the years, to include in 1984 a new

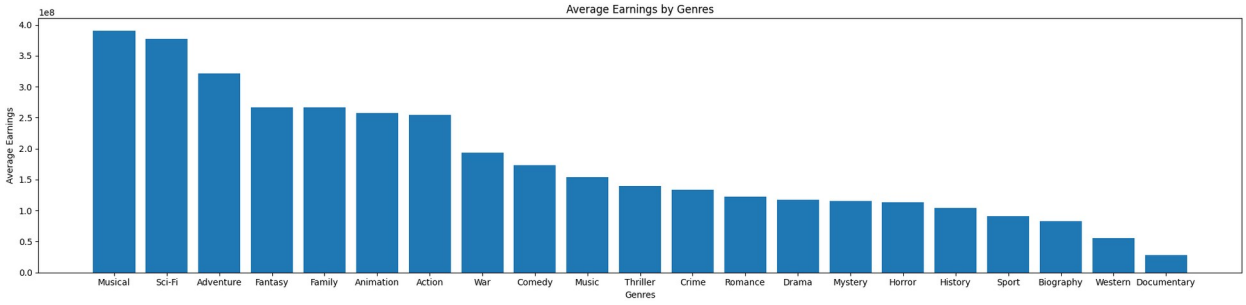
- PG-13 label, and in 1990 a new NC-17 rating (which stands for no one 17 and under admitted). The NC-17 rating replaced the X rating, which came to signify pornography.

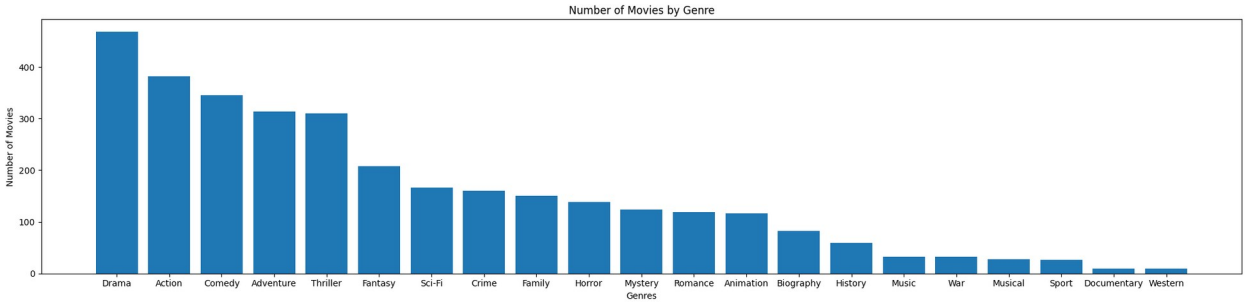

Number of Movies by MPAA Rating
Number of Movies
0
50
100
150
200
250
300
PG-13
R
Not Rated
PG
G
MPAA Rating


Sum of Earnings by MPAA Rating
1e10
Sum of Earnings
0
1
2
3
4
5
6
7
8
PG-13
PG
R
Not Rated
G
MPAA Rating

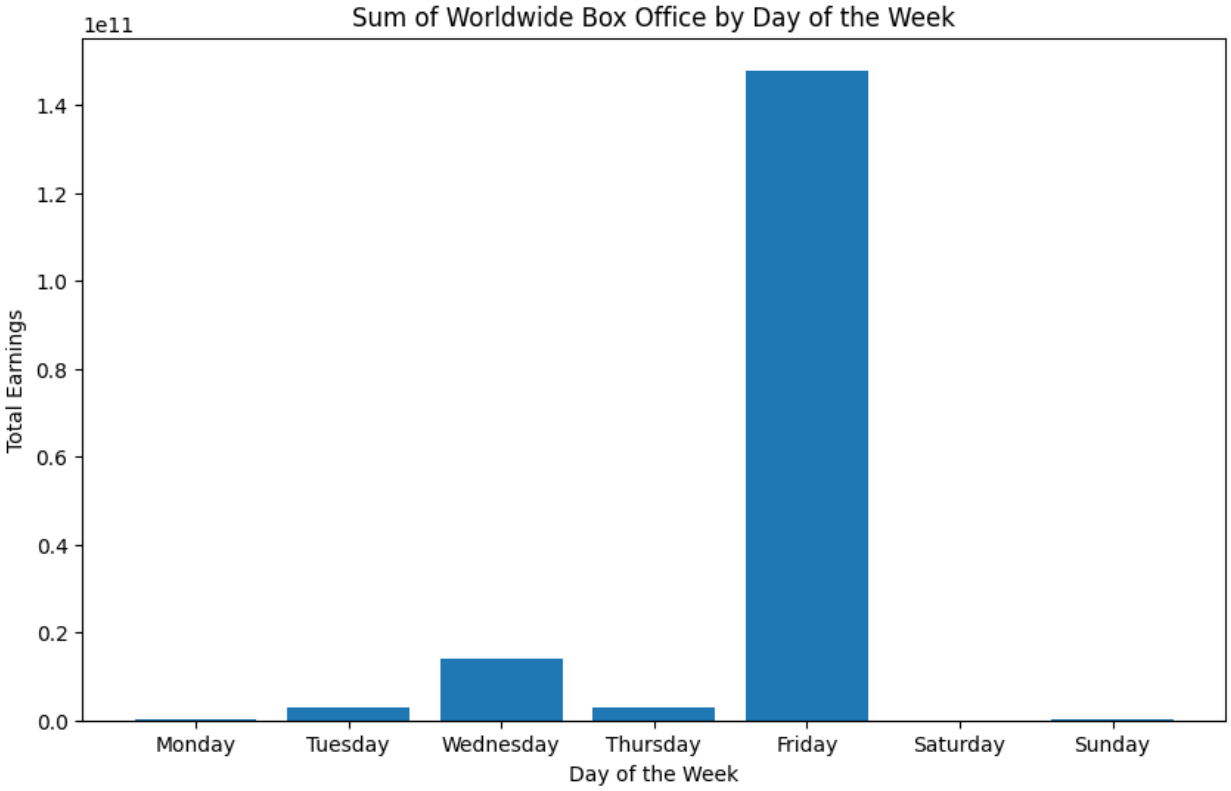


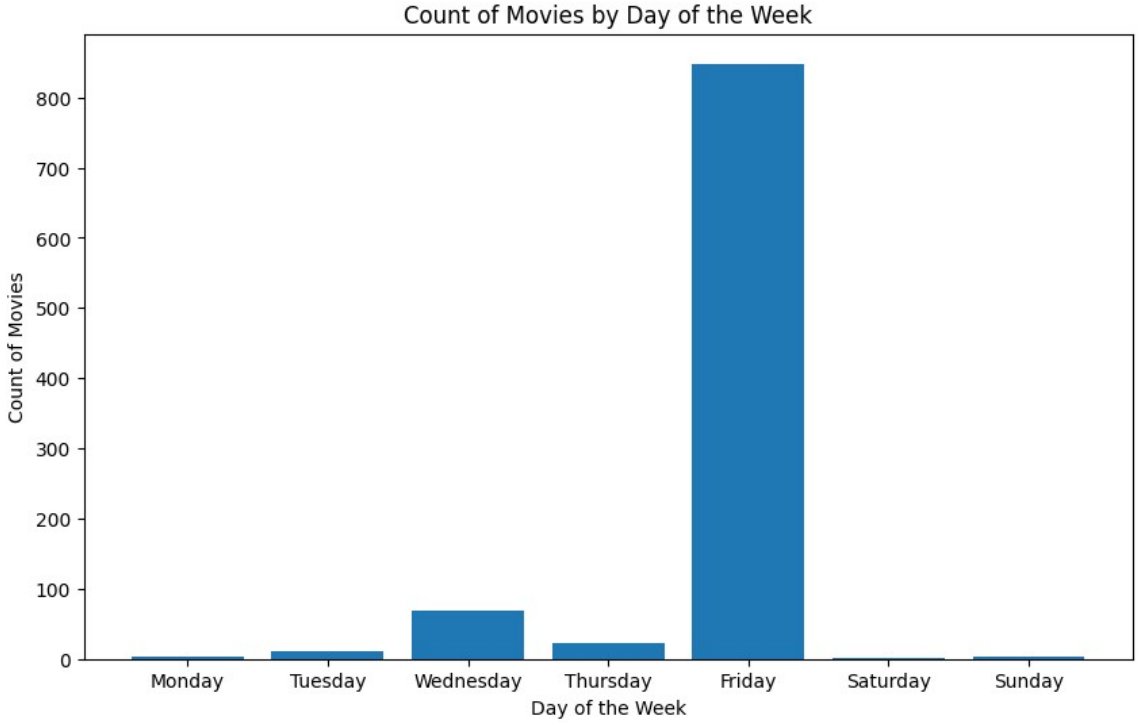
Genre Analysis

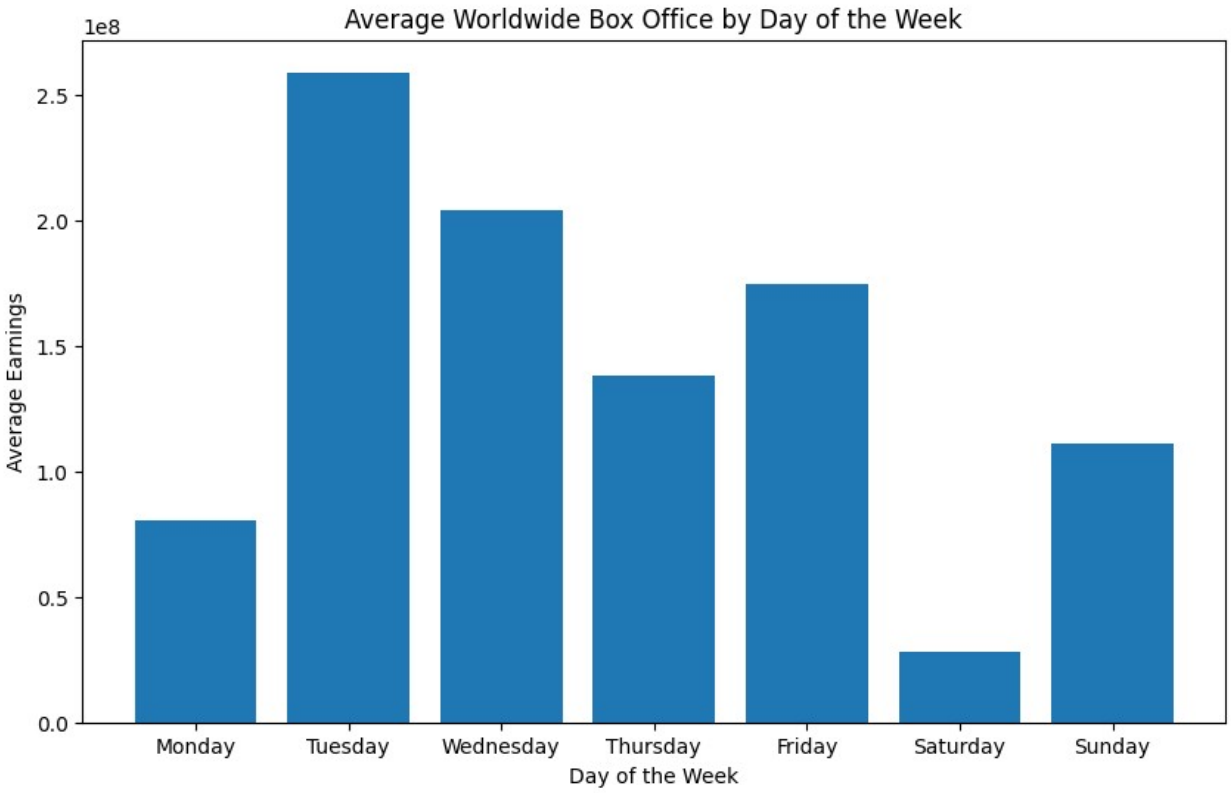


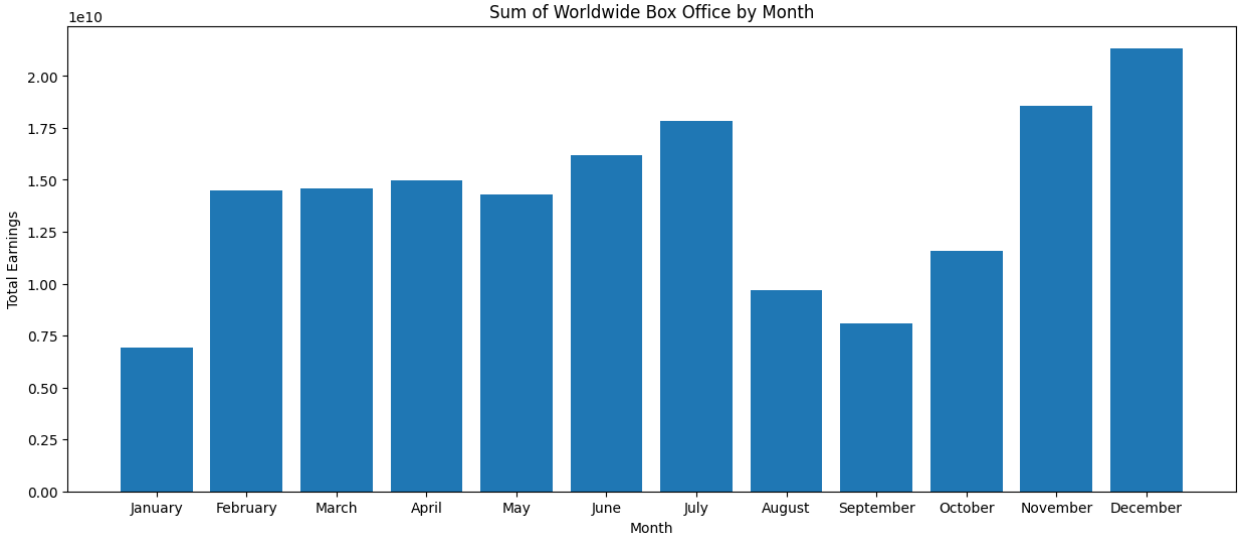


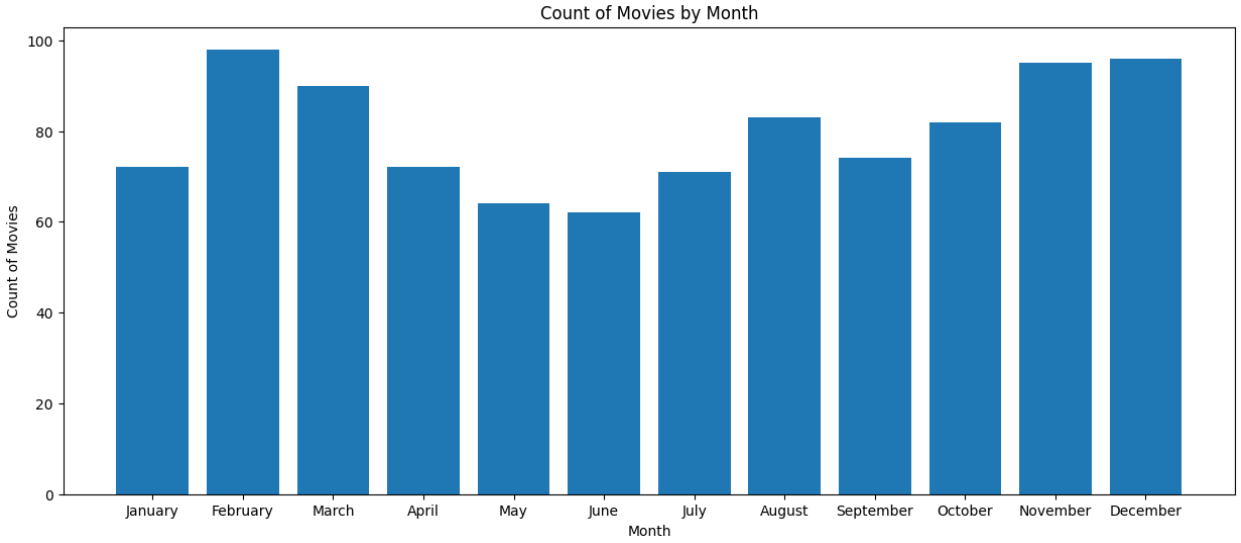
Time Analysis

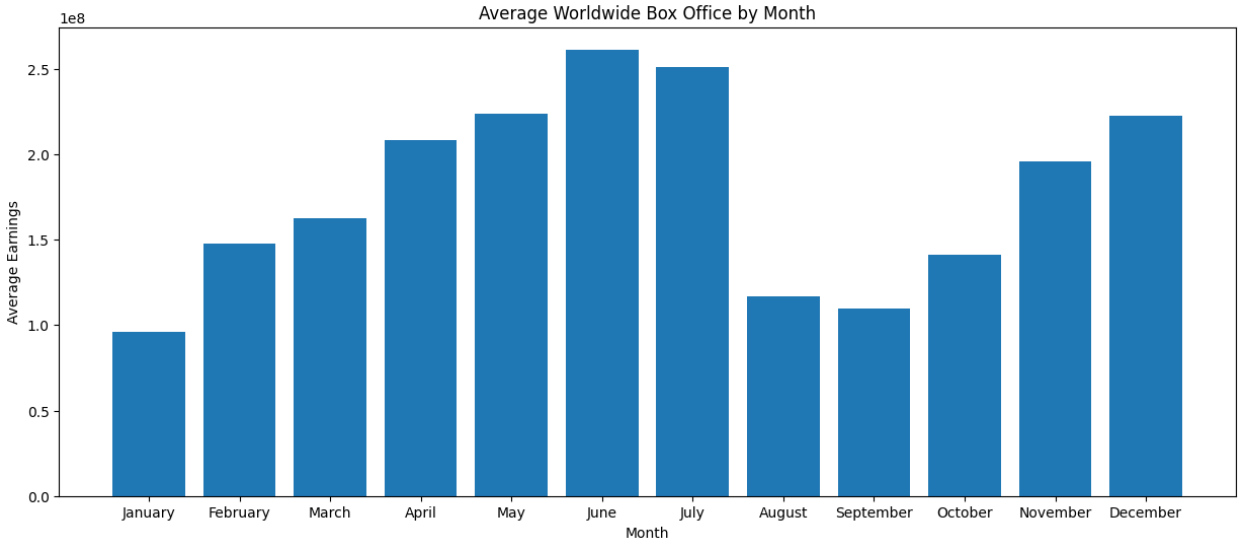


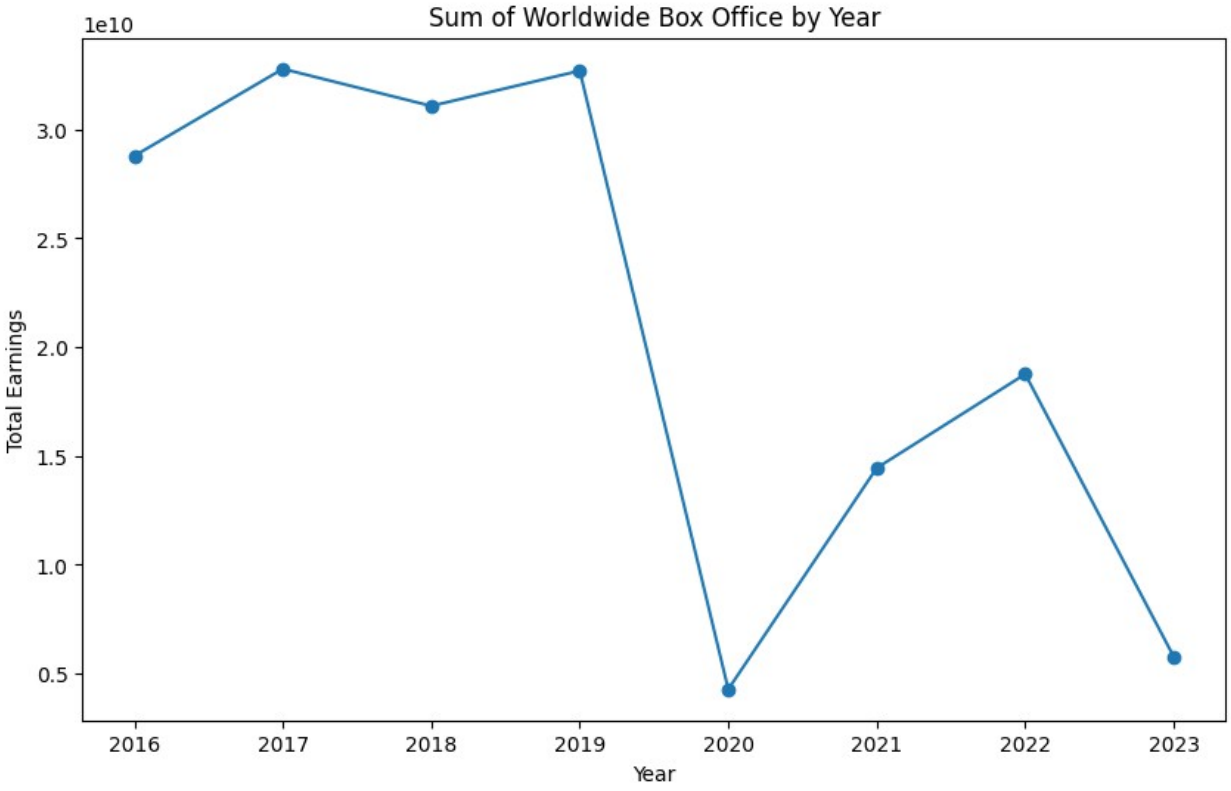


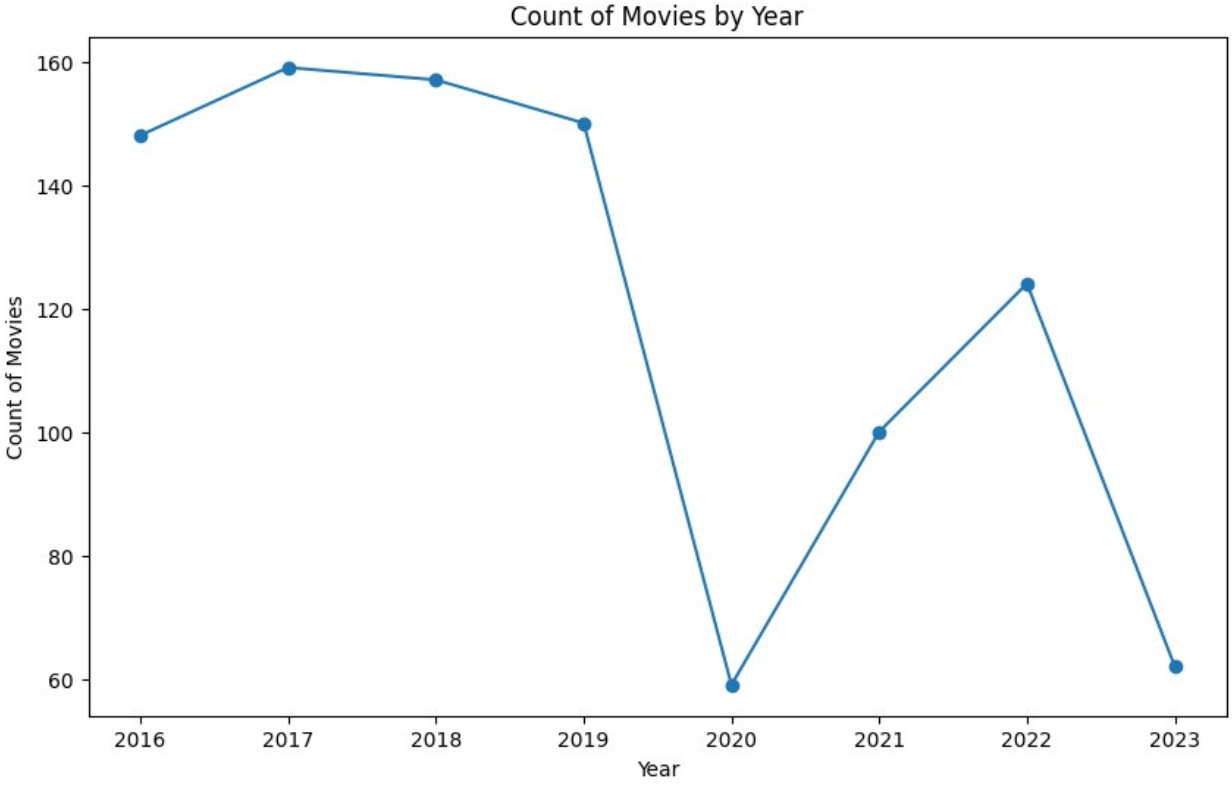

Average Worldwide Box Office by Day of the Week
1e8
2.5
2.0
1.5
1.0
0.5
0.0
Average Earnings
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of the Week

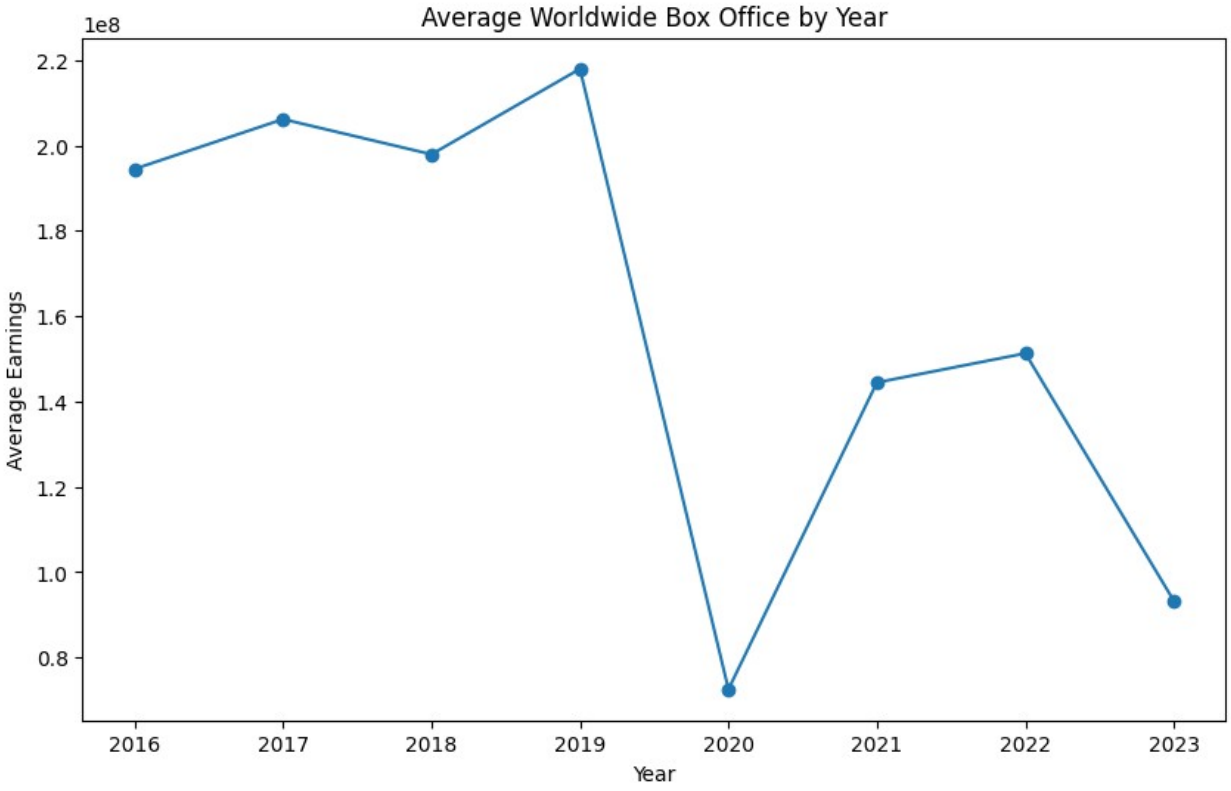

Sum of Worldwide Box Office by Month
1e10
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Total Earnings
January
February
March
April
May
June
July
August
September
October
November
December
Month


Count of Movies by Month
Count of Movies
0
20
40
60
80
100
January
February
March
April
May
June
July
August
September
October
November
December
Month

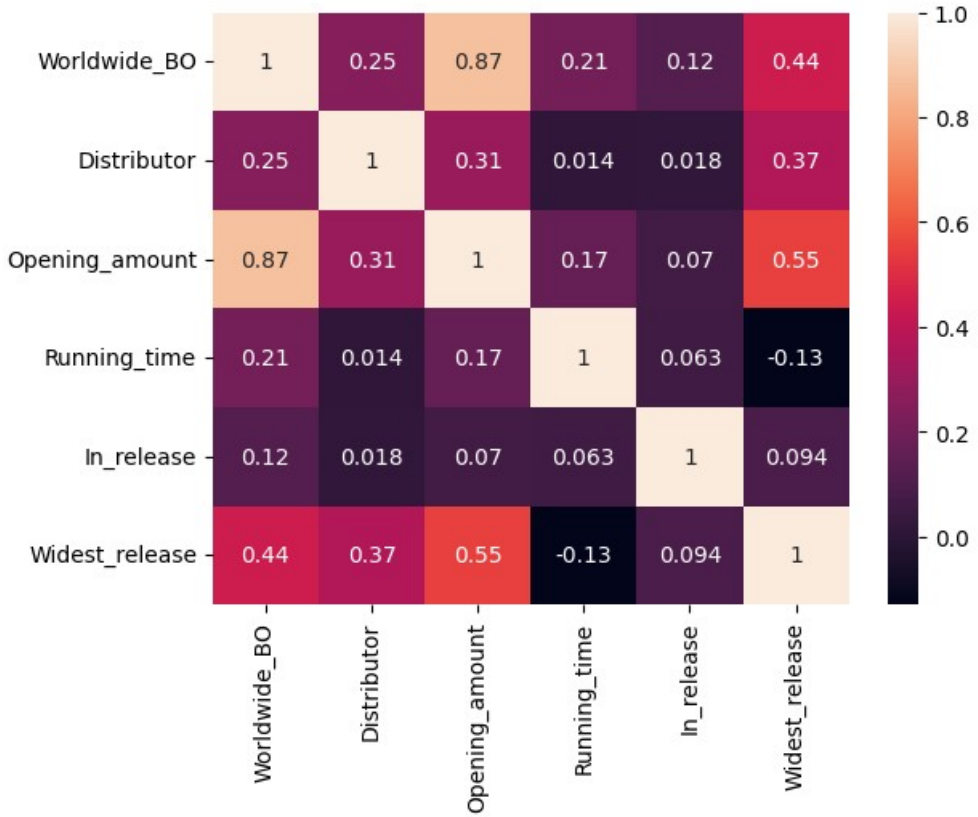

Average Worldwide Box Office by Month
1e8
Average Earnings
0.0
0.5
1.0
1.5
2.0
2.5
January
February
March
April
May
June
July
August
September
October
November
December
Month

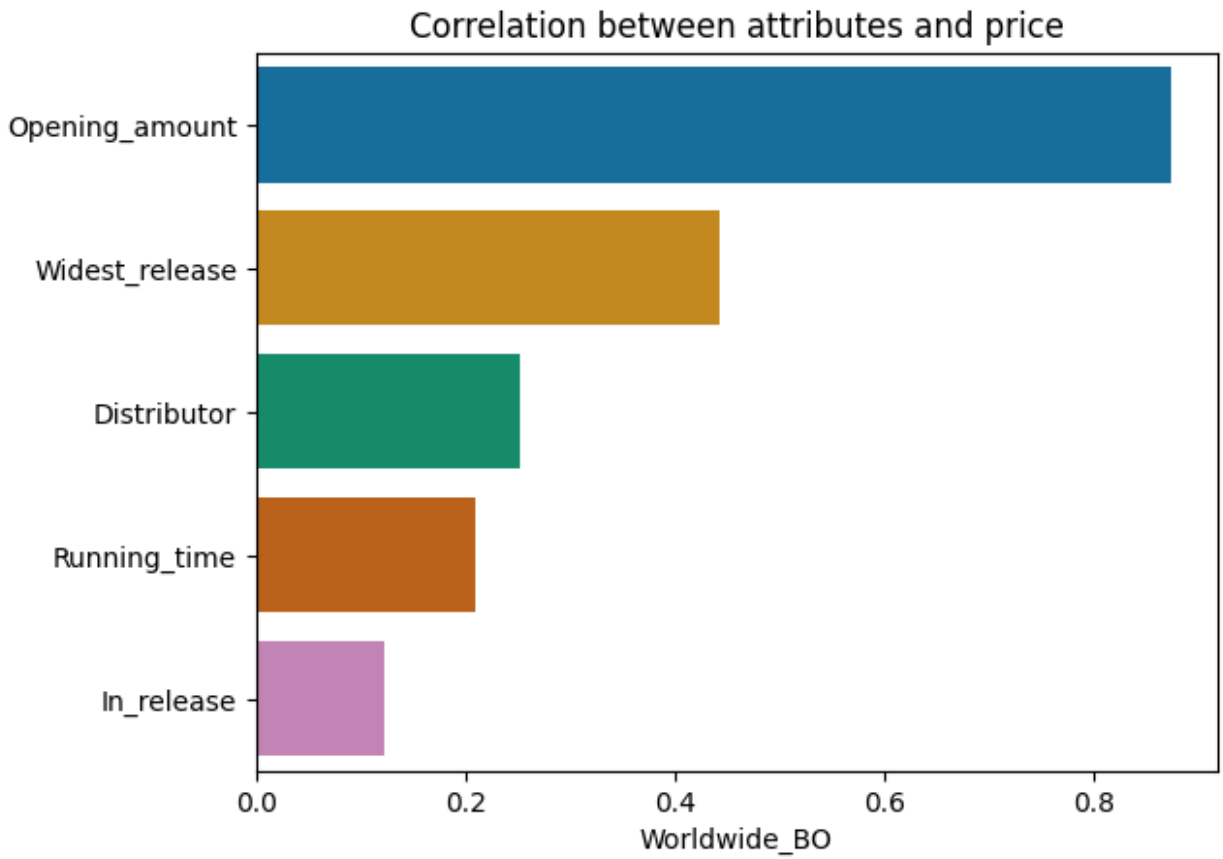

Sum of Worldwide Box Office by Year
1e10
3.0
2.5
2.0
1.5
1.0
0.5
Total Earnings
2016
2017
2018
2019
2020
2021
2022
2023
Year


Count of Movies by Year
160
140
120
100
80
60
Count of Movies
2016
2017
2018
2019
2020
2021
2022
2023
Year



Correlation Analysis





Airflow Automation

The Airflow DAG (Directed Acyclic Graph) is used to schedule and execute a series of tasks related to a web scraping project using Scrapy and analysis.

1. DAG Definition:

- The DAG is defined using the `DAG` class from the Airflow library.
- The DAG is named 'scrapy_crawl_dag' and has a daily schedule interval.
- It has an owner specified as 'airflow'.
- The start date is set to May 27, 2023.

2. Task 1: Scrapy Crawl Task:

- This task is defined using the `BashOperator` class from the Airflow library.

- The task's ID is 'scrapy_crawl_task'.

- The `bash_command` parameter specifies the command to be executed, which consists of changing the directory to the project folder, running the Scrapy spider named 'boxoffice', and saving the output to a CSV file with a dynamically generated filename.

- This task is responsible for initiating the web scraping process using Scrapy.

3. Task 2: Analysis of Movie Ratings:

- This task is defined using the `PythonOperator` class from the Airflow library.

- The task's ID is 'analyze_ratings_task'.

- The `python_callable` parameter specifies the function to be executed, which is `analyze_ratings`.

- The task is provided with the context, allowing access to the output file path generated by the previous task.

- The `analyze_ratings` function performs analysis on the scraped data related to movie ratings and writes the results to a CSV file.

4. Task 3: Analysis of Movie Distributors:

- This task is similar to Task 2 but focuses on analyzing movie distributors instead of ratings.

- The task's ID is 'analyze_distributors_task', and it executes the `analyze_distributors` function.

- The `analyze_distributors` function reads the scraped data, performs analysis on movie distributors, and writes the results to a CSV file.

5. Task 4: Perform Analysis on Box Office:

- This task is also defined as a `PythonOperator`.

- The task's ID is 'BO_analysis_task', and it executes the `perform_analysis` function.

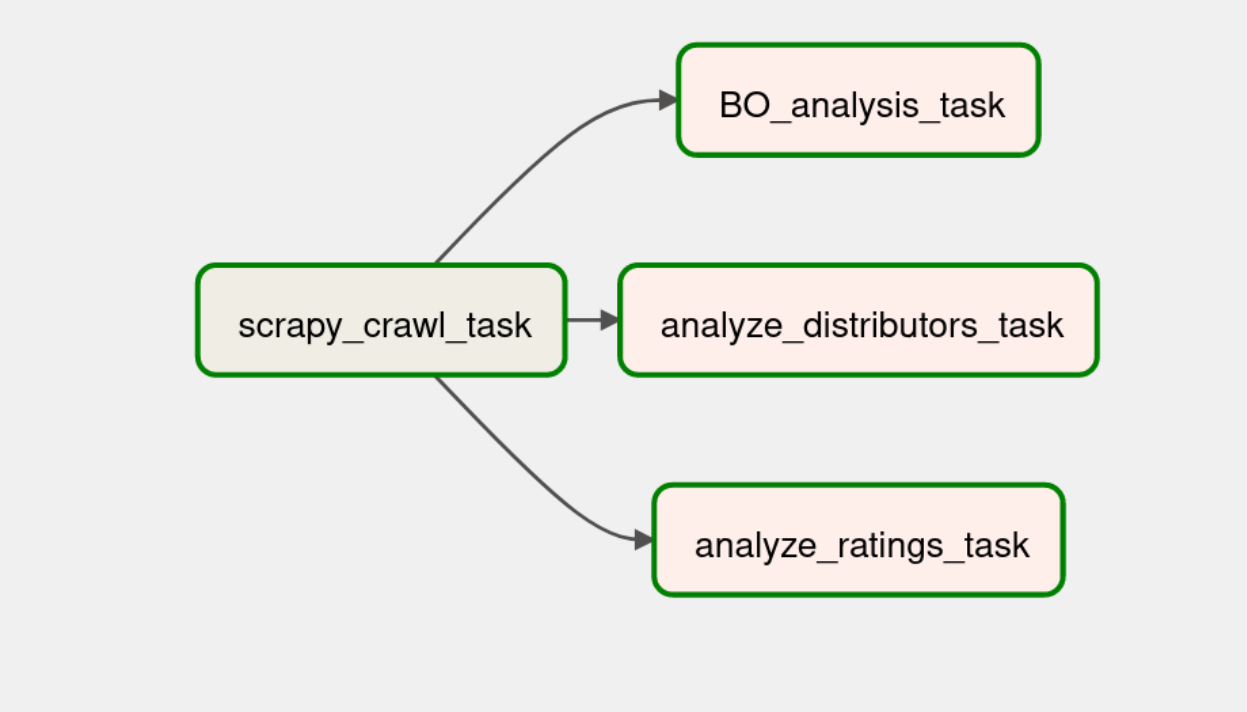
- The `perform_analysis` function reads the scraped data, performs analysis on box office earnings, and writes the results to a CSV file.

6. Task Dependencies:

- The 'scrapy_crawl_task' is set as the starting point of the DAG.

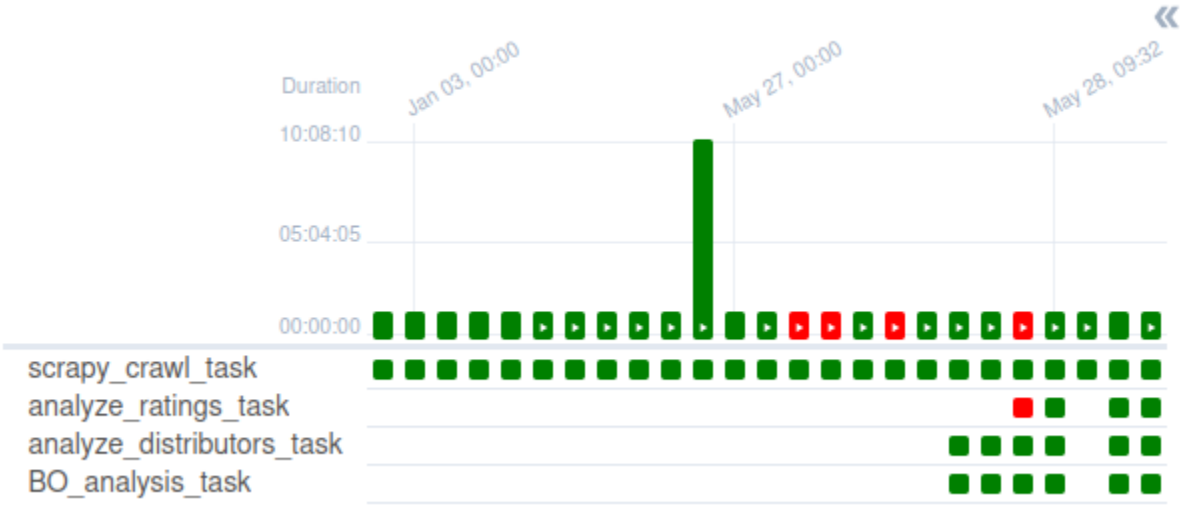
- The 'BO_analysis_task', 'analyze_distributors_task', and 'analyze_ratings_task' are dependent on the successful execution of the 'scrapy_crawl_task'.

- This dependency is established using the `>>` operator to specify the downstream tasks.



Overall, this DAG sets up a pipeline for web scraping using Scrapy and then performs analysis on the scraped data using different Python functions. The analysis results are written to separate CSV files for each task.

The DAG ensures that the tasks are executed in the specified order, with the Scrapy crawl task triggering the subsequent analysis tasks.



Conclusion

In conclusion, the box office project implemented an Airflow DAG to automate the web scraping process using Scrapy and perform analysis on the scraped data. The DAG scheduled the Scrapy crawl task daily, ensuring the latest box office data was collected. The analysis tasks extracted meaningful insights from the data, including movie ratings, distributors, and box office earnings, and generated separate CSV files with the top 10 records for each category. By leveraging Airflow's task scheduling and dependency management, the project provided a reliable and scalable solution for regularly updating and analyzing box office information, enabling informed decision-making in the film industry.