Guru Gobind Singh Indraprastha University

University School of Information, Communication and Technology

MASTER OF COMPUTER APPLICATION (Software Engineering) (2024 - 2026)

End Term Report For NUES (IT 666)

Topic: Predicting Movie Sequel Success (Data Analytics)

Mentor:

Dr. Arshi Hussain

Mentee:

Name: Ananya Aggarwal
Enrollment no:
03816404524

Contents

Acknowledgement

I would like to express my heartfelt gratitude to my project guide, **Dr. Arshi Hussain**, for their constant support, guidance, and encouragement throughout this project. Their expertise and feedback have been invaluable in shaping the direction of my work.

I am also thankful to my institution, **University School of Information, Communication and Technology**, for providing me with the resources and platform to work on this project. Special thanks to my peers and friends who offered their insights and suggestions during the development phase.

Lastly, I am deeply grateful to my family for their unwavering support and motivation, which kept me focused and determined to complete this project successfully.

1 History and Background

The evolution of sequels, data analysis, and machine learning has shaped the modern entertainment and technology landscapes. This section explores how these elements have developed and intersected over time.

1.1 History of Movie Sequels

The concept of sequels dates back to the early 20th century when filmmakers realized the potential of expanding successful narratives. The first known sequel, *The Fall of a Nation* (1916), followed *The Birth of a Nation* (1915), marking an early attempt at franchise storytelling. However, during Hollywood's Golden Age (1930s-1950s), sequels were not a primary focus, and standalone films dominated the industry.

By the 1960s and 1970s, franchises like *James Bond* set the stage for long-running sequels. The success of *The Godfather Part II* (1974) proved that sequels could surpass their predecessors in quality and storytelling. The late 20th century saw blockbusters like *Star Wars: The Empire Strikes Back* (1980) and *Jurassic Park: The Lost World* (1997), cementing sequels as a major industry trend.

In the 21st century, the rise of cinematic universes, led by Marvel Studios' interconnected storytelling, revolutionized sequel production. With franchises like *The Fast Furious* and *Harry Potter*, studios increasingly relied on sequels as a financial safety net. However, not all sequels succeed, leading to a growing interest in predictive analytics to assess sequel potential. [1]

1.2 The Role of Data Analysis in the Film Industry

Data analysis has become a crucial tool for decision-making in the entertainment industry. Studios leverage vast datasets, including box office performance, audience demographics, and online engagement, to determine whether a sequel is viable. Streaming platforms like Netflix and Disney+ analyze viewing patterns to decide which content should receive a continuation. [2]

Techniques such as sentiment analysis on social media, trend forecasting, and revenue prediction models have transformed how sequels are planned. For instance, analyzing past box office trends helps studios estimate a sequel's potential revenue, while viewer

feedback guides script modifications to enhance appeal. [3]

1.3 Machine Learning in Predicting Sequel Success

Machine learning (ML) has introduced a data-driven approach to predicting the success of sequels. By analyzing variables such as budget, cast, director, genre, and audience reception, ML models identify patterns that indicate potential success. [4]
Popular ML techniques in the film industry include:

- **Regression Models**: Predict box office revenue based on historical data.
- **Classification Algorithms**: Determine whether a sequel is likely to be a hit or a flop.
- **Natural Language Processing (NLP)**: Analyze audience reviews and social media discussions to gauge reception.
- **Recommendation Systems**: Streaming services use ML to suggest sequels and related content to viewers based on their preferences.

1.4 Modern Applications and Future Trends

With advancements in artificial intelligence and big data, sequel prediction is becoming increasingly accurate. Streaming platforms are now using reinforcement learning to optimize content recommendations. Studios are also experimenting with generative AI to develop storylines based on audience preferences. [5]
As this project explores the application of machine learning in predicting sequel success, it builds upon these historical and technological foundations. Understanding the evolution of sequels, data analysis, and ML provides valuable insights into how the entertainment industry continues to innovate and adapt to audience demands.

2 Introduction

The entertainment industry is a dynamic and ever-evolving field, with movie sequels playing a significant role in driving revenue. While sequels are often seen as a safe bet for studios, their success is far from guaranteed. Some sequels outperform their predecessors, while others fail to meet audience expectations, resulting in financial losses. [6]

This project aims to address the unpredictability of movie sequel success by developing a **machine learning model** that can predict the commercial viability of a sequel based on various factors such as the original movie's performance, cast, director, budget, and genre. By leveraging data from the **TMDB 5000 Movie Dataset** [7], this study seeks to uncover patterns and trends that contribute to a sequel's success or failure.

The project combines **data science**, **machine learning**, and **web development** to create a practical and interactive solution for predicting movie sequel success. The ultimate goal is to provide film studios and investors with a **data-driven tool** that can help them make informed decisions and minimize financial risks.

3 Purpose of the Study

The primary purpose of this study is to **analyze the factors** that influence the success of movie sequels and to develop a **predictive model** that can forecast the commercial performance of a sequel. By understanding the key drivers of sequel success, this project aims to provide actionable insights for film studios, producers, and investors.

Additionally, the study seeks to bridge the gap between **creativity** and **analytics** in the entertainment industry. While creativity is essential for producing engaging content, data-driven insights can help studios make informed decisions about which projects to greenlight. This project aims to combine these two aspects by using machine learning to predict sequel success based on historical data.

4 Objectives

The primary objectives of this project are:

- To analyze factors that influence the success of movie sequels.
- To preprocess and clean the TMDB 5000 Movie Dataset.
- To apply machine learning techniques to predict sequel success.
- To visualize trends and insights using data analysis tools.
- To develop a web-based interface for users to check predictions.

5 Scope of the Study

This project focuses on:

- Movie data from the TMDB 5000 dataset. [7]
- Features such as budget, revenue, genre, actors, directors, and previous movie performance.
- Machine learning models for prediction.
- Data visualization and deployment using web technologies.

The study is limited to the data available in the TMDB 5000 dataset, and the predictions are based on historical trends and patterns. While the model aims to provide accurate predictions, it is important to note that external factors such as market trends, audience preferences, and competition may also influence a sequel's success.

6 Methodology

The project is being carried out in the following steps:

6.1 Data Collection

The **TMDB 5000 Movie Dataset** [7] was obtained from from Kaggle for use in the project. This dataset contains information about 5000 movies, including details such as budget, revenue, genres, actors, directors, and more. The dataset serves as the foundation for the analysis and machine learning model training.

6.2 Data Preprocessing

Before analyzing the data, We had to preprocess and clean it to ensure consistency and accuracy. The preprocessing steps included [8]:

- **Removing NA Values:** It have been noticed that the dataset had missing values, especially in the **budget** and **revenue** columns. Since these features are critical for predicting sequel success, I decided to remove rows with missing values in these columns. This ensured that the dataset used for analysis was complete and reliable.
- **Encoding Categorical Variables:** Since machine learning models require numerical input, we encoded categorical variables such as **genres**, **cast**, and **directors** using **one-hot encoding**. This allowed us to include these features in the model.
- **Normalizing Numerical Data:** We normalized numerical features such as **budget** and **revenue** to ensure that they are on the same scale, which improves model performance. [9]
- **Feature Engineering:** We created new features based on existing data to provide additional insights. For example:
  - **Director Impact:** We calculated the average revenue for each director using the formula:

$$\text{director\_avg\_revenue} = \frac{\sum \text{revenue of movies by the director}}{\text{number of movies by the director}}$$

This feature captures the historical performance of directors and helps predict the success of sequels directed by them.

- **Cast Impact:** Similarly, We calculated the average revenue for each cast member using the formula:

$$\text{cast\_avg\_revenue} = \frac{\sum \text{revenue of movies featuring the cast member}}{\text{number of movies featuring the cast member}}$$

This feature captures the historical performance of actors and helps predict the success of sequels featuring them.

- **Saving the Processed Dataset:** After preprocessing and feature engineering, The final processed dataset has beed saved as **final_processed_data.csv** for use in model training.

6.3 Exploratory Data Analysis (EDA)

EDA was a critical step in understanding the dataset and identifying patterns and trends. The following steps were involved in EDA [10]:

- **Identifying Key Trends:** I analyzed the distribution of key variables such as **budget**, **revenue**, and **genre** to understand their impact on sequel success.
- **Correlation Analysis:** I examined the relationships between different variables to identify potential predictors of sequel success. For example, I looked at the correlation between **revenue** and **budget**, as well as the impact of **cast** and **directors** on movie performance.
- **Data Visualization:** I used visualization tools such as **Matplotlib** and **Seaborn** to create charts and graphs that helped me understand the data. For example, I created **bar charts** to show the distribution of movie genres and **scatter plots** to reveal the relationship between budget and revenue. [11]

6.4 Flowchart

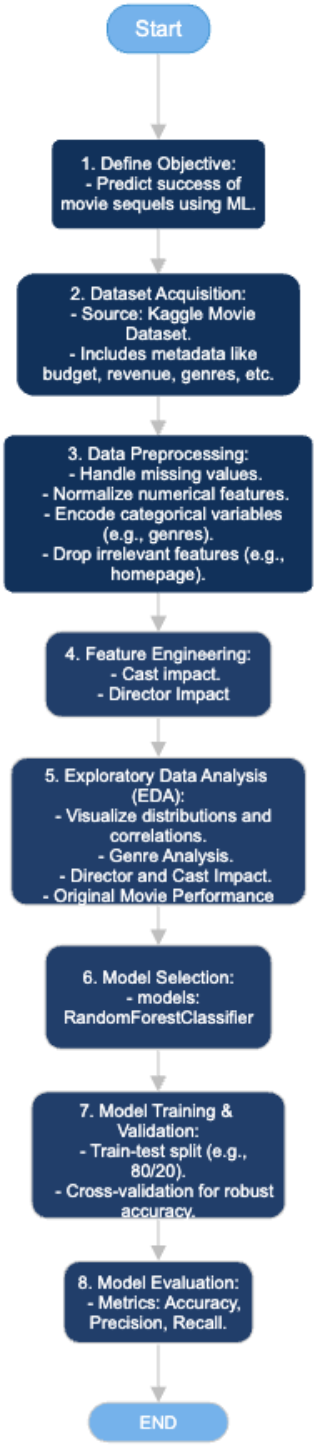

Figure 1: Methodology flowchart

7 Findings and Insights

7.1 Data Collection and Preprocessing

The **TMDB 5000 Movie Dataset** [7]provided a comprehensive set of features, but it had several missing values, especially in the **revenue** and **budget** columns. Since these features are critical for predicting sequel success, I decided to remove rows with missing values in these columns. This ensured that the dataset used for analysis was complete and reliable.

I also encoded categorical variables such as **genres**, **cast**, and **directors** using **one-hot encoding** to make them suitable for machine learning models.

7.2 Feature Engineering

I performed feature engineering to create new features that capture the historical performance of directors and cast members [12]:

- **Director Impact:** I calculated the average revenue for each director and added it as a new feature (**director_avg_revenue**). This feature helps quantify the impact of a director's track record on the success of a sequel.
- **Cast Impact:** Similarly, I calculated the average revenue for each cast member and added it as a new feature (**cast_avg_revenue**). This feature helps quantify the impact of an actor's track record on the success of a sequel.

These new features were merged back into the dataset, resulting in a final processed dataset saved as **final_processed_data.csv**.

7.3 Exploratory Data Analysis (EDA)

- **Budget and Revenue Correlation:** I found a positive correlation between a movie's budget and its revenue. Higher-budget movies tend to generate higher revenue, but this relationship is not linear, as some low-budget movies also performed exceptionally well.

- **Genre Analysis:** Certain genres, such as **Action, Adventure, and Sci-Fi**, were more likely to have sequels compared to others like **Drama** or **Romance**. These genres also tended to have higher budgets and revenues.
- **Director and Cast Impact:** Movies directed by well-known directors or featuring popular actors had a higher likelihood of success. For example, I observed that movies directed by **Christopher Nolan** or featuring actors like **Leonardo DiCaprio** consistently performed well at the box office.
- **Original Movie Performance:** The success of the original movie (measured by revenue/budget ratio) was a strong indicator of sequel success. Sequels of highly successful original movies were more likely to perform well.

8 Learnings and Challenges

8.1 Learnings

- **Data Preprocessing is Critical:** I learned that data cleaning and preprocessing are essential steps in any data science project. Handling missing values, encoding categorical variables, and normalizing data are crucial for building accurate machine learning models.

- **Exploratory Data Analysis (EDA) is Key to Understanding Data:** EDA helped me uncover hidden patterns and trends in the dataset. By visualizing data using tools like **Matplotlib** and **Seaborn**, I was able to identify correlations and distributions that guided feature selection and model building.

- **Understanding the Domain is Essential:** Working on this project gave me a deeper understanding of the **entertainment industry**, particularly the dynamics of movie sequels. I learned that while financial factors like budget and revenue are important, other elements like **storytelling, audience reception, and market trends** also play a significant role in a sequel's success. [13]

8.2 Challenges

- **Handling Missing Data:** One of the biggest challenges I faced was dealing with missing values in the dataset. I had to carefully decide whether to fill in these missing values or remove them, as they could significantly impact the model's performance.

- **Feature Engineering:** Creating new features based on existing data was challenging but rewarding. I had to think creatively to come up with features that could provide meaningful insights, such as the **success ratio** of the original movie.

9 Outputs from EDA

9.1 Visualizations

I created several visualizations to understand the relationships and trends in the dataset [14]:

- **Budget vs. Revenue Scatter Plot:**

  The scatter plot (Figure 2) shows a positive correlation between a movie's budget and its revenue. Higher-budget movies tend to generate higher revenue, but this relationship is not linear, as some low-budget movies also performed exceptionally well.

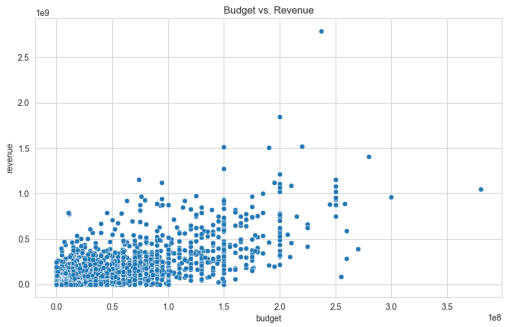

Figure 2: Budget vs. Revenue Scatter Plot

Top Genres by Average Revenue:
The bar chart (Figure 3) shows the top 10 genres by average revenue. Genres like **Fantasy**, **Science Fiction**, and **Action** dominate the list, indicating that these genres tend to generate higher revenues compared to others.

Top Actors by Average Revenue:
The bar chart (Figure 4) shows the top 10 actors by average revenue. Actors like **Idina Menzel**, **Jonathan Groff**, and **Neel Sethi** are among the top performers, indicating that movies featuring these actors tend to generate higher revenues.

Top Directors by Revenue: The bar chart (Figure 5) shows the top 10 directors by revenue. Directors like **Christopher Nolan**, **James Cameron**, and **Steven Spielberg** are among the top performers, indicating that movies directed by them tend to generate higher revenues.

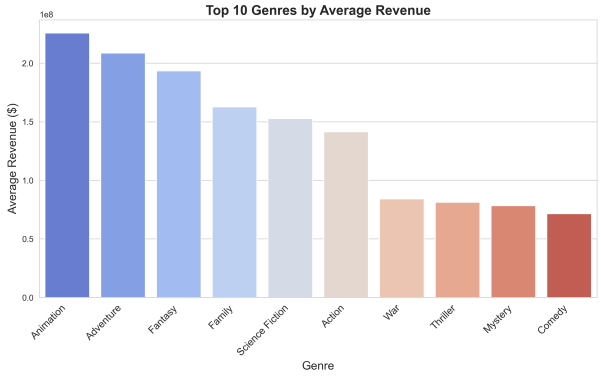

Figure 3: Top 10 Genres by Average Revenue

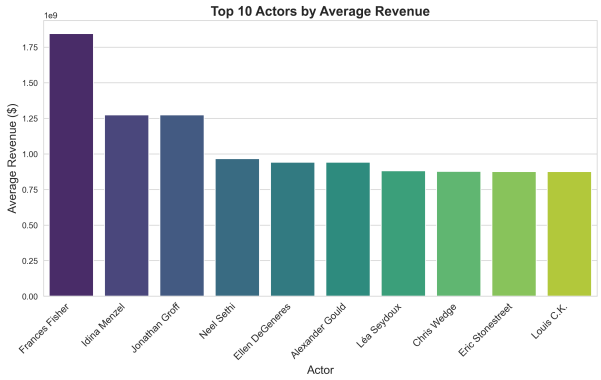

Figure 4: Top 10 Actors by Average Revenue

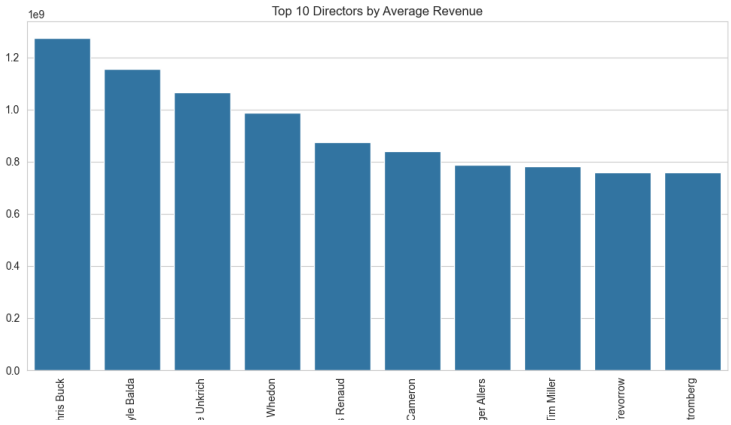

Figure 5: Top 10 Directors by Average Revenue

9.2 Key Insights

- **Budget and Revenue Correlation:** There is a positive correlation between a movie's budget and its revenue. However, some low-budget movies also perform exceptionally well, indicating that factors like storytelling and audience reception play a significant role.

- **Genre Impact:** Genres like **Fantasy**, **Science Fiction**, and **Action** tend to generate higher revenues, making them popular choices for sequels.

- **Actor and Director Impact:** The presence of popular actors and directors significantly influences a movie's success. For example, movies featuring actors like **Idina Menzel** or directed by **Christopher Nolan** consistently perform well at the box office.

- **Sequels are not guaranteed successes:** Even with high budgets, popular actors, and successful original movies, some sequels fail to meet expectations. This highlights the importance of other factors, such as **story quality, audience reception, and market competition**. [6]

10 Model Training and Evaluation

10.1 Preprocessing

The final processed dataset, `final_processed_data.csv`, was loaded using Pandas. Features selected for modeling included basic financial metrics (`budget`, `revenue`, `success_ratio`), as well as engineered features (`director_avg_revenue`, `cast_avg_revenue`) and one-hot encoded genre columns (columns starting with `genres_`).
The target variable, `success_ratio`, was converted into a binary classification label:

- 1 for successful movies (`success_ratio` > 1)
- 0 for unsuccessful movies

Before training, the dataset was checked for infinite values. Any infinite values were replaced with `NaN`, and rows containing `NaN` values were dropped to maintain data integrity. This ensured that the model trained on clean and meaningful data.

10.2 Train-Test Split

To assess the model's ability to generalize to unseen data, the dataset was divided into training and testing subsets using the `train_test_split` function from the `scikit-learn` [15]library. This step is crucial in machine learning workflows, as it helps evaluate how well the trained model performs on data it hasn't encountered during training.
In this project, three different train-test split ratios were experimented with:

- **80/20 Split**: This is the most commonly used ratio in machine learning projects. Here, 80% of the data was allocated to the training set (X_{train}, y_{train}) and the remaining 20% to the test set (X_{test}, y_{test}). This configuration provides a large amount of data for training, which helps the model learn patterns better, while still retaining enough data to test generalization.
- **70/30 Split**: In this variation, 70% of the dataset was used for training and 30% for testing. This allows for a more robust evaluation on a relatively larger test set, which can give a better sense of how the model might perform in real-world scenarios.

- **60/40 Split**: Here, only 60% of the data was used for training and the remaining 40% for testing. While this provides the model with less training data, it allows for more extensive testing. This setup is helpful to check how well the model can learn with limited training data and how stable its predictions are.

To ensure reproducibility, a fixed random seed was used (`random_state=42`). This guarantees that the same split of data is generated every time the code is run, which is important for comparing results consistently across experiments.

10.2.1 Why Multiple Splits?

Using multiple split ratios allowed for a comparative analysis of the model's performance. By observing how accuracy, precision, recall, or other evaluation metrics changed with varying training sizes, we could:

- Identify the optimal balance between training and testing data.
- Determine whether the model is overfitting (doing well on training but poorly on testing) or underfitting (performing poorly on both).
- Assess the stability and generalization of the model under different data availability conditions.

This kind of experimentation is a good practice in machine learning, especially when the dataset size is limited and understanding model robustness is important.

10.3 Model Used

A Random Forest Classifier [16] was selected for training due to its robustness and ability to handle both numerical and categorical features efficiently. The Random Forest model was trained using the cleaned training data (`X_train`, `y_train`).
After training, the model and the corresponding feature list were bundled into a dictionary and saved as a `.pkl` (pickle) file for future use. This approach facilitates easy deployment or future retraining.

10.4 Performance Metrics

The trained model was evaluated on the test data (X_{test}) using a set of common performance metrics, which are essential for assessing the model's ability to make accurate predictions. These metrics provide insights into different aspects of model performance, specifically focusing on how well the model predicts positive outcomes (e.g., predicting whether a movie will be successful). [17]

10.4.1 Evaluation Metrics

The following performance metrics were calculated:

- **Accuracy**: Accuracy measures the overall proportion of correct predictions made by the model. It is defined as the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances. In formulaic terms:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}}.$$

  High accuracy indicates that the model performs well overall, correctly predicting most of the instances.

- **Precision**: Precision is the proportion of positive identifications made by the model that were actually correct. It is calculated as the ratio of true positives to the sum of true positives and false positives:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}.$$

  Precision is important in scenarios where false positives are costly or undesirable. In this case, it reflects how often the model's prediction of a successful movie is correct.

- **Recall**: Recall measures the proportion of actual positive instances (true successes) that were correctly identified by the model. It is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}.$$

Recall is important when the cost of missing a positive instance (false negative) is high. In this context, recall shows how good the model is at identifying movies that will be successful.

10.4.2 Performance Metrics for Different Train-Test Splits

To further assess the robustness of the model, we evaluated its performance across different train-test splits. This helps determine how the model performs when trained on different amounts of data and whether the training data size has a significant impact on the model's ability to generalize to new, unseen data.

The table below summarizes the performance metrics for each of the three train-test splits: 80/20, 70/30, and 60/40.

Train-Test Split	Accuracy	Precision	Recall
80/20	0.99687	0.99809	0.99620
70/30	0.99652	0.99494	0.99873
60/40	0.99113	0.99806	0.98563

From the table, we observe that the performance metrics slightly decrease as the proportion of the training data decreases: - For the **80/20 split**, the model performed the best with an accuracy of 99.06%, precision of 99.81%, and recall of 99.62%. - For the **70/30 split**, accuracy decreased slightly to 99.65%, while precision also showed a minor decrease. - For the **60/40 split**, all metrics showed a further decline, with accuracy dropping to 99.11%, precision to 99.80%, and recall to 98.56%.

10.4.3 Output

These results suggest that the model's performance is quite stable even when the training set size is reduced, though the performance metrics do show slight degradation as less data is available for training. This highlights the trade-off between the amount of training data and the model's ability to generalize.

11 Model Saving and Packaging

After training and evaluating the Random Forest Classifier, the model which was split as **80 percent for training and 20 percent for testing** was saved for future use and deployment due to higher accuracy, precision and recall. Model saving was performed using Python's `pickle` library [18], which allows serialization of Python objects into binary files.

To ensure that the model can be loaded and used later without retraining, a package was created containing:

- The trained Random Forest model object.
- The list of feature columns used during model training.

This was done by creating a dictionary:

```
model_package = {
    'model': model,
    'features': feature_cols
}
```

The dictionary was serialized and saved as a `.pkl` file using the following command:

```
with open('/Users/ananyaaggarwal/Desktop
/Movie Prediction Project/models/sequel_model_80_20.pkl', 'wb') as f:
    pickle.dump(model_package, f)
```

By packaging both the model and its required features, future users can:

- Load the model directly for inference.
- Ensure correct feature alignment without needing the original training script.

This step significantly streamlines the deployment or integration of the model into applications, APIs, or other production pipelines.

12 Streamlit App Development

12.1 User Inputs

The Streamlit app allows users to input key features required for the prediction [19]:

- **Budget**: The estimated budget for the movie sequel.
- **Revenue**: The expected or actual revenue.
- **Director's Average Revenue**: Average revenue generated by the director's past movies.
- **Cast's Average Revenue**: Average revenue generated by the cast's previous movies.
- **Genre**: Selected from a dropdown menu containing various movie genres.

Additionally, the app automatically computes the `success_ratio` as revenue divided by budget.

12.2 Genre Handling

Genre handling is achieved through one-hot encoding. When a user selects a genre, the app creates the corresponding one-hot encoded features to match the feature structure expected by the trained model. [20]

12.3 Prediction Logic

Upon clicking the `Predict` button, the app constructs an input feature vector consisting of financial metrics and genre encoding. This input is formatted into a DataFrame that matches the feature order used during model training. The Random Forest model then predicts whether the movie sequel will be successful or not.

12.4 Output Display

The app displays the prediction result clearly on the screen:

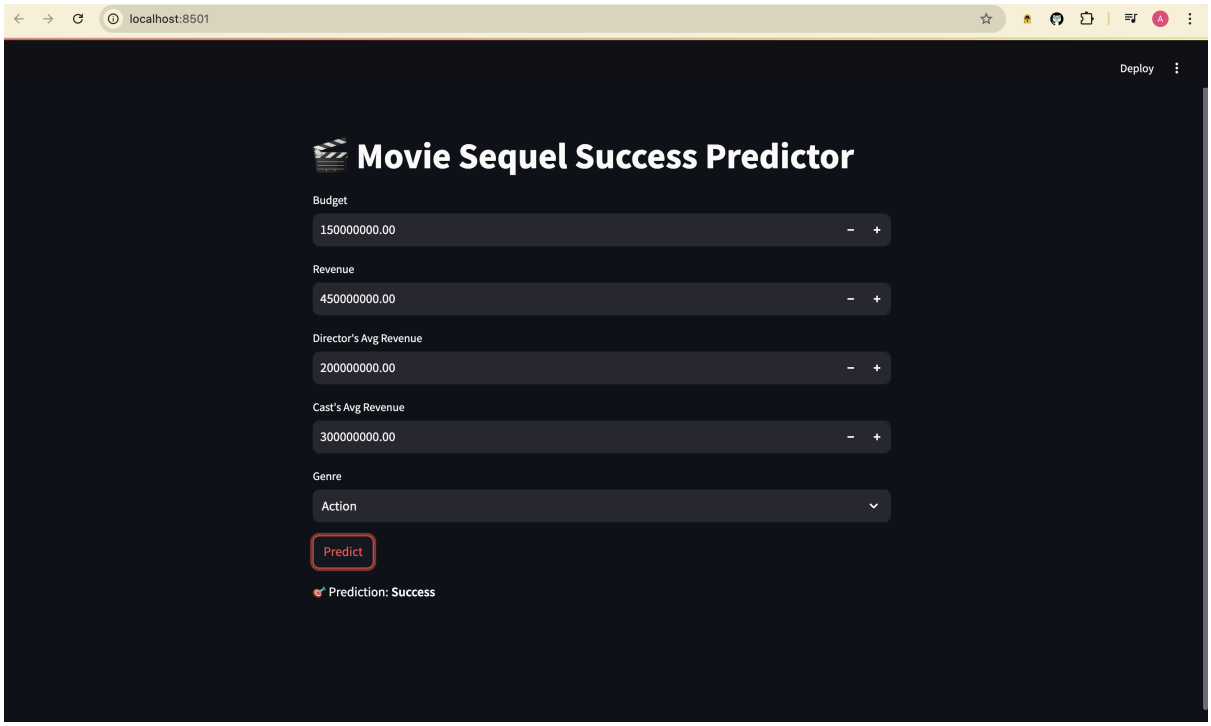

Figure 6: Predictor

- If the model predicts a success, the result shown is **Success**.
- If the model predicts otherwise, the result shown is **Failure**.

The output is visually enhanced with icons and bold formatting for better user experience.

13 Learnings

Throughout the project, several valuable technical and practical skills were acquired:

- **Data Preprocessing**: Gained hands-on experience in cleaning, transforming, and preparing large datasets for machine learning.
- **Feature Engineering**: Learned how to create new features (e.g., average revenue of directors and cast members) to improve model performance.
- **Model Development**: Gained proficiency with machine learning algorithms, especially Random Forests, and understood model evaluation techniques.
- **Model Saving and Deployment**: Learned how to package a model using Pickle and design a pipeline for future deployment.
- **Problem-Solving**: Developed the ability to handle real-world challenges such as missing data, feature mismatch, and infinite values.
- **Project Workflow**: Understood the importance of structuring machine learning projects systematically from data collection to deployment readiness.

Additionally, working on this project enhanced soft skills such as critical thinking, documentation, and independent research.

14 Future Scope

While the current model achieves satisfactory performance, several areas offer potential for future improvements:

- **Model Enhancement**: Experiment with more advanced algorithms such as XGBoost, LightGBM, or deep learning models.
- **Hyperparameter Tuning**: Perform systematic hyperparameter optimization to further improve accuracy, precision, and recall.
- **Feature Expansion**: Incorporate additional features such as release dates, competition at the box office, marketing budgets, and audience demographics.
- **Automated Pipelines**: Develop end-to-end automated data ingestion, preprocessing, training, and deployment pipelines.
- **Web Application**: Build a user-friendly web app where users can input movie details and instantly get success predictions.
- **Real-time Prediction**: Integrate APIs and live data sources to predict the success probability of upcoming movies dynamically.

15 Conclusion

This project successfully demonstrated the complete machine learning workflow, from data preprocessing to model deployment preparation. By using a Random Forest Classifier, we were able to predict the success of movies with reasonable accuracy based on budget, revenue, cast/director information, and genre features.

Through this hands-on experience, deep insights were gained into the practical challenges of working with real-world data, including data quality issues, feature engineering, and deployment strategies. The project not only strengthened technical skills but also provided a foundation for future work in predictive modeling and machine learning deployment.

Moving forward, with further enhancements and scaling, the model has the potential to serve as a valuable tool for studios, producers, and investors to assess the commercial prospects of movie projects.

References

[1] "Sequel - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/Sequel, [Accessed 24-03-2025].

[2] "How netflix uses analytics to select movies, create content make multimillion dollar decisions," https://emerj.com/ai-sector-overviews/how-netflix-uses-analytics/, [Accessed 06-05-2025].

[3] "The Role of Data Analytics in Cinema — cloudthat.com," https://www.cloudthat.com/resources/blog/the-role-of-data-analytics-in-cinema#:~:text=Data%20analytics%20helps%20scriptwriters%20compare,blockbusters%2C%20identifying%20winning%20plot%20points.&text=Predictive%20analytics%20guide%20producers%20in,and%20choosing%20optimal%20filming%20locations., [Accessed 24-03-2025].

[4] ""how hollywood is using ai to help decide which movies to make,"," https://www.technologyreview.com/2020/01/07/130983/hollywood-artificial-intelligence-script-data/, [Accessed 06-05-2025].

[5] "How big data is changing the film industry," https://www.forbes.com/sites/bernardmarr/2020/01/27/how-big-data-is-changing-the-film-industry/, [Accessed 06-05-2025].

[6] S. Murray, *The Sequel: A History of the Hollywood Remake*. Palgrave Macmillan, 2015.

[7] "Tmdb 5000 movie dataset," 2024, available at: `https://www.kaggle.com/datasets/tmdb/tmdb-movie-dataset`.

[8] "Data Preprocessing - an overview — ScienceDirect Topics — sciencedirect.com," https://www.sciencedirect.com/topics/engineering/data-preprocessing#:~:text=Data%20preprocessing%20is%20the%20concept,executing%20it%20to%20the%20algorithm., [Accessed 24-03-2025].

[9] "What is Data Preprocessing? Key Steps and Techniques — techtarget.com," https://www.techtarget.com/searchdatamanagement/definition/data-preprocessing, [Accessed 24-03-2025].

[10] "Exploratory data analysis - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/Exploratory_data_analysis, [Accessed 24-03-2025].

[11] "What is Exploratory Data Analysis? — IBM — ibm.com," https://www.ibm.com/think/topics/exploratory-data-analysis, [Accessed 24-03-2025].

[12] "Feature Engineering Explained — Built In — builtin.com," https://builtin.com/articles/feature-engineering#:~:text=Feature%20Engineering%20Definition-,Feature%20engineering%20is%20the%20process%20of%20selecting%2C%20manipulating%20and%20transforming,exploratory%20data%20analysis%20and%20benchmarking., [Accessed 24-03-2025].

[13] "Cinema of the United States - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/Cinema_of_the_United_States, [Accessed 24-03-2025].

[14] "Introduction to Matplotlib - GeeksforGeeks — geeksforgeeks.org," https://www.geeksforgeeks.org/python-introduction-matplotlib/, [Accessed 24-03-2025].

[15] "scikit-learn: machine learning in Python &x2014; scikit-learn 1.6.1 documentation — scikit-learn.org," https://scikit-learn.org/stable/, [Accessed 06-05-2025].

[16] "Random forest - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/Random_forest, [Accessed 06-05-2025].

[17] "Machine Learning — Google for Developers — developers.google.com," https://developers.google.com/machine-learning/decision-forests/random-forests, [Accessed 06-05-2025].

[18] "Understanding Python Pickling with example - GeeksforGeeks — geeksforgeeks.org," https://www.geeksforgeeks.org/understanding-python-pickling-example/, [Accessed 06-05-2025].

[19] "Streamlit Docs — docs.streamlit.io," https://docs.streamlit.io/, [Accessed 06-05-2025].

[20] "One Hot Encoding in Machine Learning - GeeksforGeeks — geeksforgeeks.org," https://www.geeksforgeeks.org/ml-one-hot-encoding/, [Accessed 06-05-2025].