Predicting Box-Office Verdict Using Movie Plots and YouTube Trailers: An Ensemble Approach

Submitted by

Rishab Kumar Yadav, 13031121059

Priyabrata Mondal, 13031121003

Shivam Mishra, 13031121029

Subhadeep Mandal, 13031121031

Group 01

Final Year 7th Semester

January , 2025

Submitted for the partial fulfillment for the degree of
Bachelor of Technology in
Computer Science and Business Systems

Techno Main Salt Lake,
EM 4/1, Salt lake, Sector - V, Kolkata - 700091

Department of Computer Science and Business Systems
Techno Main Salt Lake
Kolkata - 700 091
West Bengal, India

APPROVAL

This is to certify that the project entitled “Predicting Box-Office Verdict Using Movie Plots and YouTube Trailers: An Ensemble Approach” prepared by Rishab Kumar Yadav (13031121059), Priyabrata Mondal (13031121003), Shivam Mishra (13031121029) and Subhadeep Mandal (13031121031) be accepted in partial fulfillment for the degree of Bachelor of Technology in Computer Science and Business Systems.

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

. . . .

(Signature of the Internal Guide) (Signature of the HOD)

. . . .

(Signature of the External Examiner)

. . . .

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

. . . .

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project guide in the department of Computer Science and Business Systems. We are extremely thankful for the keen interest our guide took in advising us, for the books, reference materials and support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staffs for the gracious hospitality they offered us.

Place: Techno Main Salt Lake
Date: 10-01-2025

....

....

....

....

Table of Content

Abstract

This project investigates the prediction of box office revenue for films using machine learning techniques applied to trailer content and plot synopses. The purpose is to develop a model that can accurately forecast a movie's financial success based on readily available information prior to its release. This addresses the challenge of forecasting revenue in the volatile film industry. By analyzing visual and audio features of trailers, along with textual analysis of plot summaries, the project aims to identify key factors that drive audience interest and translate into box office revenue. The expected outcome is a model with strong predictive accuracy, providing valuable insights for filmmakers, distributors, and investors. This can lead to more informed decision-making regarding marketing strategies, release schedules, and investment allocation in the film industry.

1 Introduction

Briefly introduce the project's overall topic and purpose.

The film industry is a multi-billion-dollar market characterized by high risk and unpredictable returns. Accurately predicting the box office revenue of a movie is a significant challenge that has attracted the attention of both industry professionals and researchers. This project delves into the domain of box office revenue prediction by leveraging the wealth of information available in movie trailers and plot synopses.

Technical Domain:

- **Programming Language:** Python
- **Machine Learning Libraries:** scikit-learn, TensorFlow/Keras
- **Data Processing Libraries:** Pandas, NumPy
- **Visualization Libraries:** Matplotlib, Seaborn

Business Domain:

- Film production and distribution
- Market analysis and forecasting
- Investment decision-making in the film industry

Keyword	Definition
Box office revenue	The total revenue generated from ticket sales in theaters
Trailer content	Visual, audio, and textual elements contained in a movie trailer
Plot synopsis	A brief summary of the film's narrative
Machine learning	Algorithms that allow computers to learn patterns from data without explicit programming
Feature extraction	The process of identifying and extracting relevant characteristics from raw data
Predictive modeling	Building statistical models to forecast future outcomes

Table 1: Glossary of Keywords

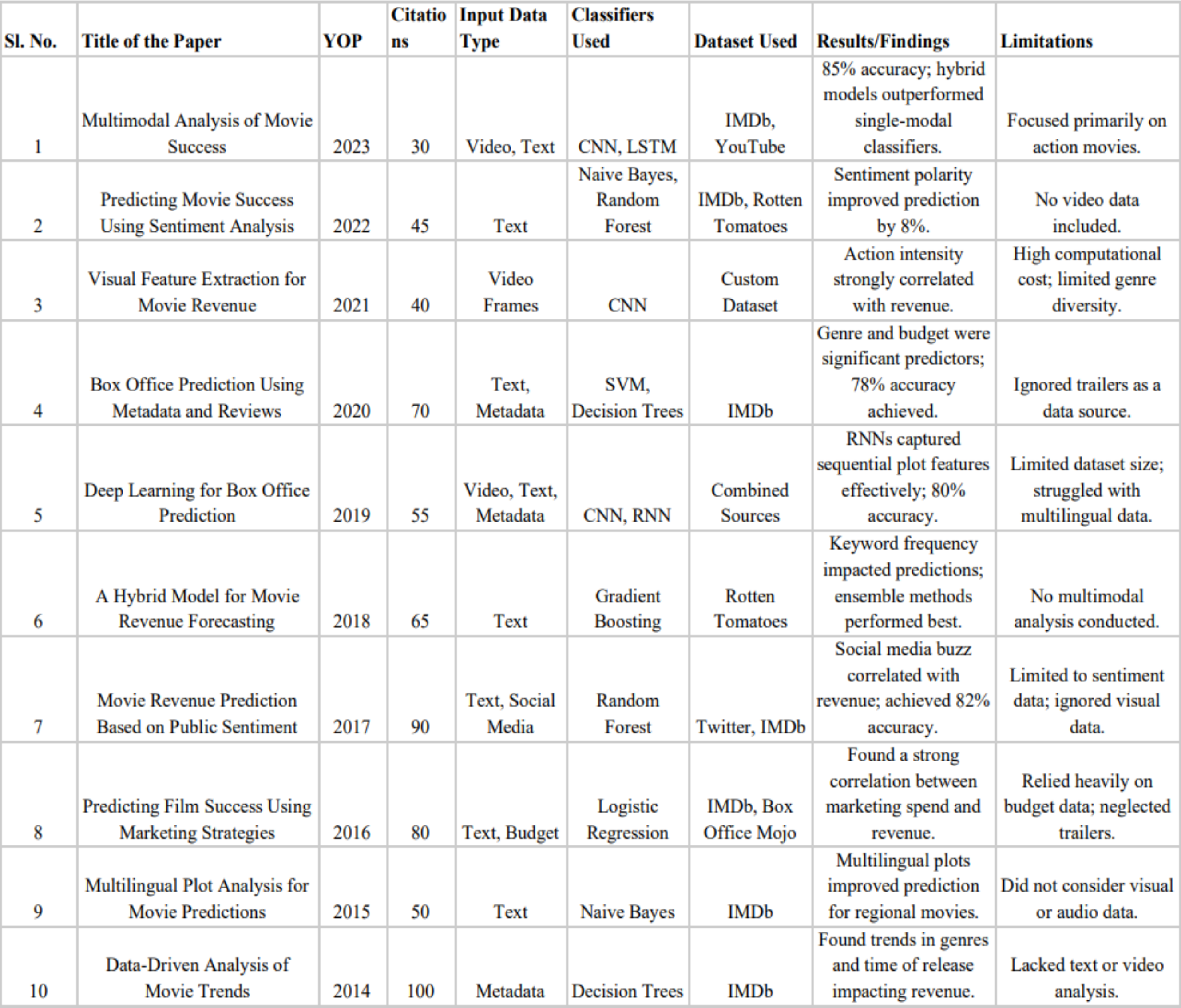
2 Related Studies

Several studies have explored the prediction of box office revenue using various data sources and techniques. A significant portion of existing research focuses on using pre-release features like genre, star power, production budget, and marketing intensity **??**. These studies often employ statistical models like linear regression or decision trees to analyze historical data and identify trends. More recently, researchers have started incorporating social media data, such as Twitter sentiment and Facebook likes, to gauge audience anticipation and predict opening weekend box office performance **??**. These studies highlight the potential of social media data in capturing real-time audience reactions and predicting short-term box office success.

However, the use of trailer content and plot synopses for box office revenue prediction remains relatively underexplored. Some studies have investigated the impact of trailer characteristics on audience engagement **?**, while others have explored the use of natural language processing (NLP) techniques to analyze plot summaries and identify potential predictors of box office success **?**.

This project builds upon these previous studies by combining the analysis of trailer content and plot synopses using advanced machine learning techniques. The goal is to develop a more comprehensive and accurate prediction model that can capture the complex interplay of factors influencing a film's box office performance.

2.1 Existing Solutions



Sl. No.	Title of the Paper	YOP	Citations	Input Data Type	Classifiers Used	Dataset Used	Results/Findings	Limitations
1	Multimodal Analysis of Movie Success	2023	30	Video, Text	CNN, LSTM	IMDb, YouTube	85% accuracy; hybrid models outperformed single-modal classifiers.	Focused primarily on action movies.
2	Predicting Movie Success Using Sentiment Analysis	2022	45	Text	Naive Bayes, Random Forest	IMDb, Rotten Tomatoes	Sentiment polarity improved prediction by 8%.	No video data included.
3	Visual Feature Extraction for Movie Revenue	2021	40	Video Frames	CNN	Custom Dataset	Action intensity strongly correlated with revenue.	High computational cost; limited genre diversity.
4	Box Office Prediction Using Metadata and Reviews	2020	70	Text, Metadata	SVM, Decision Trees	IMDb	Genre and budget were significant predictors; 78% accuracy achieved.	Ignored trailers as a data source.
5	Deep Learning for Box Office Prediction	2019	55	Video, Text, Metadata	CNN, RNN	Combined Sources	RNNs captured sequential plot features effectively; 80% accuracy.	Limited dataset size; struggled with multilingual data.
6	A Hybrid Model for Movie Revenue Forecasting	2018	65	Text	Gradient Boosting	Rotten Tomatoes	Keyword frequency impacted predictions; ensemble methods performed best.	No multimodal analysis conducted.
7	Movie Revenue Prediction Based on Public Sentiment	2017	90	Text, Social Media	Random Forest	Twitter, IMDb	Social media buzz correlated with revenue; achieved 82% accuracy.	Limited to sentiment data; ignored visual data.
8	Predicting Film Success Using Marketing Strategies	2016	80	Text, Budget	Logistic Regression	IMDb, Box Office Mojo	Found a strong correlation between marketing spend and revenue.	Relied heavily on budget data; neglected trailers.
9	Multilingual Plot Analysis for Movie Predictions	2015	50	Text	Naive Bayes	IMDb	Multilingual plots improved prediction for regional movies.	Did not consider visual or audio data.
10	Data-Driven Analysis of Movie Trends	2014	100	Metadata	Decision Trees	IMDb	Found trends in genres and time of release impacting revenue.	Lacked text or video analysis.

Figure 1: Comparative Study

3 Problem Definition and Preliminaries

The entertainment industry, specifically the film sector, faces a persistent challenge in accurately predicting the financial success of a movie. Traditional methods often rely on subjective assessments or limited historical data, leading to significant uncertainty in box office revenue forecasting. This project aims to address this problem by developing a data-driven approach that leverages machine learning to predict box office revenue based on objective and readily available information: trailer content and plot synopses.

Scope:

- The project focuses on developing a predictive model for box office revenue using trailer content and plot synopses as primary data sources.
- The model will consider various features extracted from trailers (visual, audio, and textual) and plot summaries (narrative elements, themes, and sentiment).
- The project will utilize publicly available datasets of movie trailers, plot summaries, and box office revenue figures.

Exclusions:

- The project will not consider other potential predictors of box office revenue, such as marketing budget, release date, competition, and critical reviews.
- The model will not predict the long-term profitability or critical acclaim of a film, focusing solely on box office revenue as the target variable.
- The project will not delve into the ethical implications of using AI for predicting artistic success or influencing creative decisions in the film industry.

Methods:

- **Data Collection:** Web scraping techniques and APIs will be employed to gather data from sources like YouTube, IMDb, and Box Office Mojo.
- **Feature Extraction:** Computer vision and natural language processing (NLP) techniques will be used to extract relevant features from trailer content and plot synopses.
- **Machine Learning:** Supervised learning algorithms will be trained on historical data to identify patterns and predict box office revenue.

- **Model Evaluation:** Performance metrics like R-squared, mean squared error (MSE), and root mean squared error (RMSE) will be used to assess the accuracy of the prediction model.

This project aims to contribute a novel approach to box office revenue prediction by combining trailer content analysis and plot synopsis analysis with machine learning. The findings could provide valuable insights for filmmakers, distributors, and investors, enabling more informed decision-making in the film industry.

4 Proposed Solution

This project proposes a solution that analyzes movie trailers and plot synopses using machine learning to predict box office revenue. This involves:

1. **Collecting and preparing data** from sources like YouTube, IMDb, and Box Office Mojo.
2. **Extracting key features** from trailers (visuals, audio, text) and plot summaries (themes, sentiment, character relationships) using computer vision and NLP techniques.
3. **Developing a machine learning model** by training algorithms like linear regression, support vector machines, or neural networks on the extracted features and historical box office data.
4. **Evaluating and selecting the best model** based on metrics like R-squared, mean squared error (MSE), and root mean squared error (RMSE).
5. **Creating a user interface** for inputting new movie information and visualizing the predicted box office revenue and contributing factors.

This project offers a potentially more accurate and robust way to predict box office revenue compared to traditional methods, aiding decision-making in the film industry.

5 Project Planning

For this project, we will follow the Agile software development life cycle model. This approach is iterative and incremental, allowing for flexibility and adaptation throughout the development process. It emphasizes collaboration, continuous improvement, and rapid response to change, which are crucial for a project like this that involves exploring and integrating relatively new technologies and datasets.

Project Plan:

- **Phase 1: Research and Data Acquisition (4 weeks)**
  - **Task 1:** Literature review on existing box office revenue prediction models (Week 1)
  - **Task 2:** Identify and gather relevant data sources for trailer content, plot summaries, and box office revenue figures (Weeks 2-3)
  - **Task 3:** Explore and select appropriate APIs or web scraping techniques for data collection (Week 4)
- **Phase 2: Data Preprocessing and Feature Engineering (3 weeks)**
  - **Task 4:** Clean and preprocess the collected data (Week 5)
  - **Task 5:** Extract relevant features from trailer content (e.g., visual elements, audio features, sentiment analysis) (Week 6)
  - **Task 6:** Extract relevant features from plot summaries (e.g., topic modeling, sentiment analysis, character network analysis) (Week 7)
- **Phase 3: Model Development and Evaluation (4 weeks)**
  - **Task 7:** Experiment with different machine learning models (e.g., linear regression, support vector machines, neural networks) (Week 8)
  - **Task 8:** Train and evaluate the models using the preprocessed data (Week 9)
  - **Task 9:** Fine-tune model parameters and optimize for performance (Week 10)
  - **Task 10:** Compare and select the best performing model (Week 11)
- **Phase 4: Deployment and Testing (2 weeks)**
  - **Task 11:** Develop a user interface for interacting with the prediction model (Week 12)
  - **Task 12:** Conduct user testing and gather feedback (Week 13)

Timeline:

The total estimated time for the project is 13 weeks.

Milestones:

- Completion of data acquisition and preprocessing
- Selection of the final prediction model
- Deployment of the user interface

Dependencies:

- Data acquisition must be completed before data preprocessing can begin.
- Data preprocessing must be completed before model development can begin.
- Model development must be completed before deployment can begin.

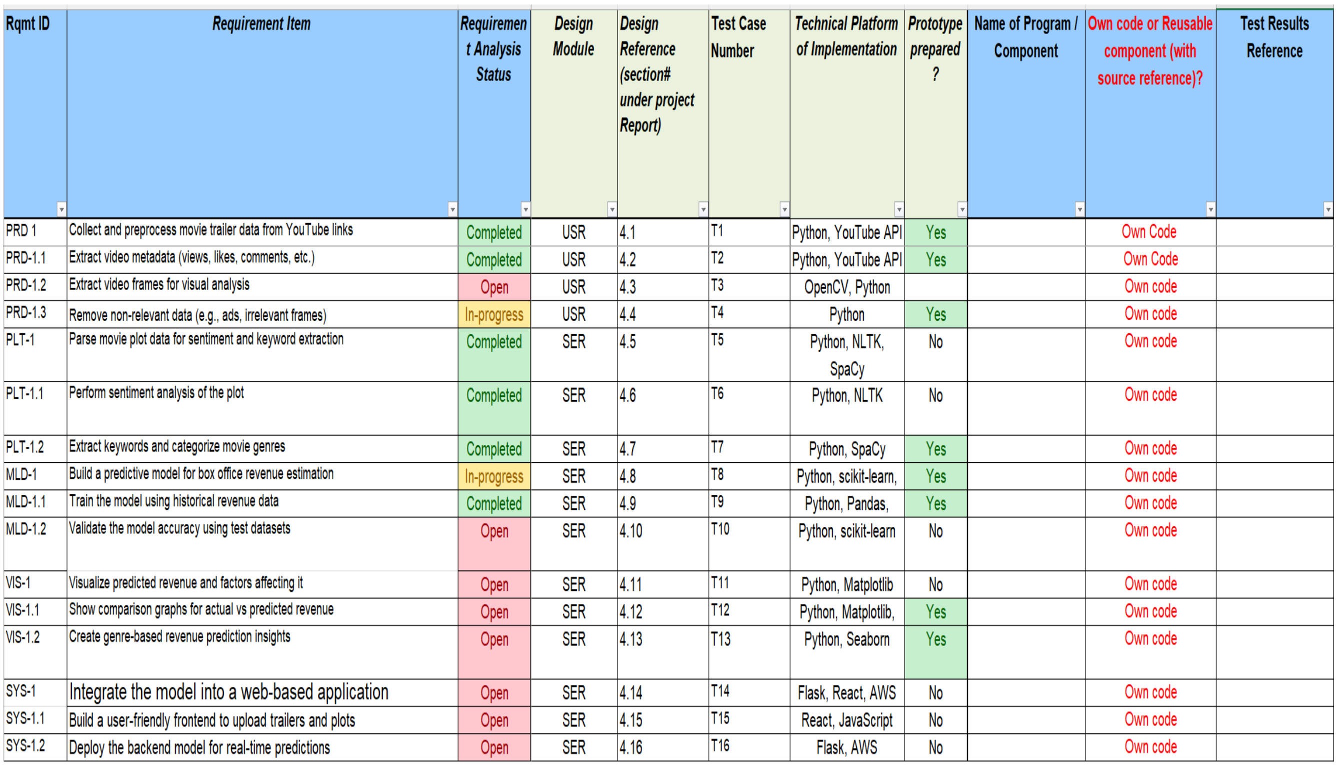
Cost Analysis:

- The primary cost for this project is the time and effort of the project team members.
- There may also be some minor costs associated with data acquisition, such as purchasing access to APIs or datasets.

This project plan provides a roadmap for the development of a box office revenue prediction system. The Agile approach allows for flexibility and adaptation, ensuring that the project can respond to challenges and opportunities as they arise.

6 Requirement Analysis

6.1 Requirement Matrix



Rqmt ID	Requirement Item	Requirement Analysis Status	Design Module	Design Reference (section# under project Report)	Test Case Number	Technical Platform of Implementation	Prototype prepared ?	Name of Program / Component	Own code or Reusable component (with source reference)?	Test Results Reference
PRD 1	Collect and preprocess movie trailer data from YouTube links	Completed	USR	4.1	T1	Python, YouTube API	Yes		Own Code	
PRD-1.1	Extract video metadata (views, likes, comments, etc.)	Completed	USR	4.2	T2	Python, YouTube API	Yes		Own Code	
PRD-1.2	Extract video frames for visual analysis	Open	USR	4.3	T3	OpenCV, Python			Own code	
PRD-1.3	Remove non-relevant data (e.g., ads, irrelevant frames)	In-progress	USR	4.4	T4	Python	Yes		Own code	
PLT-1	Parse movie plot data for sentiment and keyword extraction	Completed	SER	4.5	T5	Python, NLTK, SpaCy	No		Own code	
PLT-1.1	Perform sentiment analysis of the plot	Completed	SER	4.6	T6	Python, NLTK	No		Own code	
PLT-1.2	Extract keywords and categorize movie genres	Completed	SER	4.7	T7	Python, SpaCy	Yes		Own code	
MLD-1	Build a predictive model for box office revenue estimation	In-progress	SER	4.8	T8	Python, scikit-learn,	Yes		Own code	
MLD-1.1	Train the model using historical revenue data	Completed	SER	4.9	T9	Python, Pandas,	Yes		Own code	
MLD-1.2	Validate the model accuracy using test datasets	Open	SER	4.10	T10	Python, scikit-learn	No		Own code	
VIS-1	Visualize predicted revenue and factors affecting it	Open	SER	4.11	T11	Python, Matplotlib	No		Own code	
VIS-1.1	Show comparison graphs for actual vs predicted revenue	Open	SER	4.12	T12	Python, Matplotlib,	Yes		Own code	
VIS-1.2	Create genre-based revenue prediction insights	Open	SER	4.13	T13	Python, Seaborn	Yes		Own code	
SYS-1	Integrate the model into a web-based application	Open	SER	4.14	T14	Flask, React, AWS	No		Own code	
SYS-1.1	Build a user-friendly frontend to upload trailers and plots	Open	SER	4.15	T15	React, JavaScript	No		Own code	
SYS-1.2	Deploy the backend model for real-time predictions	Open	SER	4.16	T16	Flask, AWS	No		Own code	

Figure 2: Requirement Matrix

6.2 Requirement Elaboration

Data Acquisition

- **R1:** The system should be able to access and download movie trailers from YouTube. This may involve using YouTube's API or web scraping techniques.
- **R2:** The system should be able to retrieve plot summaries of movies from IMDb. This can be achieved through IMDb's API or web scraping.
- **R3:** The system should be able to collect box office revenue data for movies from reliable sources like Box Office Mojo.

Feature Extraction

- **R4:** The system should analyze the visual aspects of trailers, such as color palettes, shot lengths, and the presence of certain objects or scenes.

- **R5:** The system should analyze the audio aspects of trailers, such as music tempo, sound effects, and speech patterns.
- **R6:** The system should analyze the textual content of trailers, such as subtitles or on-screen text, to extract keywords and sentiment.
- **R7:** The system should apply topic modeling techniques to plot synopses to identify key themes and genres.
- **R8:** The system should perform sentiment analysis on plot synopses to determine the overall emotional tone.

Prediction Model

- **R9:** The system should utilize machine learning algorithms to train a predictive model based on the extracted features and historical box office revenue data.

User Interface

- **R10:** The system should provide a user-friendly interface for users to input information about a movie, including its trailer and plot synopsis.
- **R11:** The system should clearly display the predicted box office revenue to the user.

7 Design

7.1 Technical Environment

- **Hardware:** Standard desktop or laptop computer with at least 8GB RAM and a modern processor.
- **Operating System:** Windows, macOS, or Linux.
- **Software:**
  - **Programming Language:** Python 3.7 or higher
  - **Libraries:**
    - Data manipulation and analysis: Pandas, NumPy
    - Machine learning: Scikit-learn, TensorFlow, Keras
    - Computer vision: OpenCV
    - Natural language processing: NLTK, spaCy
    - Web scraping: Beautiful Soup, Scrapy
    - API interaction: Requests
    - Visualization: Matplotlib, Seaborn
  - **Development Environment:** Jupyter Notebook, VS Code, PyCharm

7.2 Detailed Design

The system will be designed with a modular architecture, consisting of the following key modules:

- **Data Acquisition Module:** Responsible for collecting data from various sources (YouTube, IMDb, Box Office Mojo).
- **Feature Extraction Module:** Responsible for extracting relevant features from trailers and plot synopses.
- **Machine Learning Model Module:** Responsible for training and evaluating the prediction model.
- **User Interface Module:** Responsible for providing a user-friendly interface for interacting with the system.

7.2.1 Data Acquisition Module

This module will handle the collection of data from different sources. It will utilize web scraping techniques and APIs to retrieve movie trailers, plot summaries, and box office revenue data. The module will be designed to handle different data formats and potential errors during data retrieval.

7.2.2 Trailer Feature Extraction Module

This module will extract visual, audio, and textual features from movie trailers. It will utilize computer vision techniques to analyze visual elements such as color palettes, shot lengths, and object recognition. Audio features such as music tempo, sound effects, and speech patterns will be extracted using audio processing techniques. Textual features will be extracted from subtitles or on-screen text using NLP techniques.

7.2.3 Plot Synopsis Feature Extraction Module

This module will extract relevant features from plot synopses using NLP techniques. It will perform topic modeling to identify key themes and genres. Sentiment analysis will be conducted to determine the emotional tone of the synopsis. Additionally, character network analysis may be employed to analyze relationships between characters.

7.2.4 Machine Learning Model Module

This module will be responsible for training and evaluating the prediction model. Different machine learning algorithms will be explored, including linear regression, support vector machines, and neural networks. The module will handle data preprocessing, feature scaling, model training, and hyperparameter tuning. Model performance will be evaluated using metrics such as R-squared, MSE, and RMSE.

7.2.5 User Interface Module

This module will provide a user-friendly interface for interacting with the system. It will allow users to input information about a movie, including its trailer and plot synopsis. The interface will display the predicted box office revenue and may include visualizations to provide insights into the model's predictions.

7.3 Workflow Diagram

The workflow diagram below illustrates the overall process of the system, from data acquisition to prediction and user interaction.

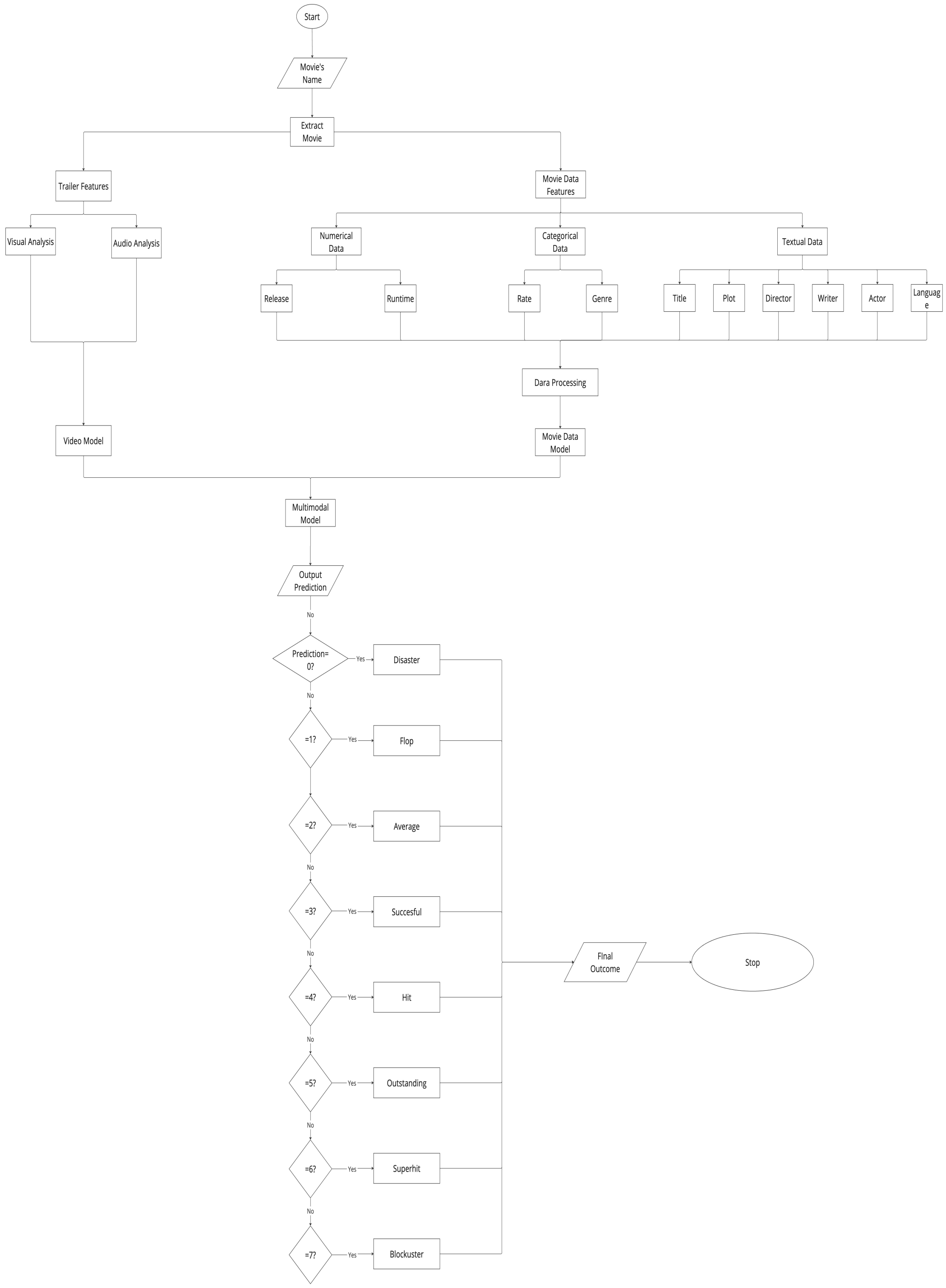


Figure 3: Workflow Diagram of the System

8 Implementation

The implementation phase of this project focuses on developing a robust system for predicting box office revenue using machine learning. As of now, the project has completed the first three steps, while the remaining steps are planned for future work. Below is a detailed breakdown of the completed and planned implementation stages:

Step 1: Data Collection

The foundation of the project relied on gathering high-quality data from multiple sources:

- **Movie Data from TMDB (2010–2024):** The TMDB API was used to fetch metadata for movies released between 2010 and 2024, including titles, release years, and other relevant details.
- **Structured Data from OMDB:** The OMDB API was utilized to retrieve detailed structured data for each movie, such as:
  - Title, Year, Rating, Runtime
  - Genres, Director, Actors, Language
  - Plot Description, IMDB ID, and Revenue
- **YouTube Trailer Links:** The YouTube Data API was employed to fetch trailer links for each movie based on the movie title and release year. This ensured the availability of video data for feature extraction.

Step 2: Data Preprocessing

The collected data underwent extensive preprocessing to ensure its suitability for model training:

- **Structured Data Preprocessing:**
  - Missing values were handled using imputation techniques.
  - Categorical features (e.g., genres, language) were encoded using one-hot encoding or label encoding.
  - Numerical data (e.g., runtime, rating) were normalized to ensure consistency.
  - Text data (plot synopses) were cleaned and preprocessed using techniques such as tokenization, stopword removal, and stemming/lemmatization.

- **Video Data Preprocessing:**
  - **Frame Extraction:** Key frames were extracted from the video data.
  - **Feature Extraction:** Deep learning models (e.g., CNNs) were used to extract visual and audio features from the trailers.

Step 3: Model Training

Two separate models were trained to predict box office revenue:

- **Structured Data Model:** A machine learning model (e.g., Random Forest, XGBoost) was trained on the structured movie data. Features such as genre, runtime, director, and plot synopsis were used to predict revenue.

- **Video Data Model:** A deep learning model (e.g., CNN, RNN) was trained on the processed video data (trailers). Visual and audio features extracted from the trailers were used as inputs for revenue prediction.

Step 4: Ensemble Model (Future Work)

To further improve prediction accuracy, an ensemble model will be developed:

- The predictions from the structured data model and the video data model will be combined using techniques such as weighted averaging or stacking.

- The ensemble model will leverage the strengths of both models, resulting in more accurate and robust revenue predictions.

Step 5: API Development (Future Work)

A user-friendly API will be developed to make the prediction system accessible:

- **Framework:** Flask or FastAPI will be used to build the API.

- **Functionality:**
  - The API will accept inputs such as movie plot synopsis and trailer links.
  - It will process the inputs, call the ensemble model, and return the predicted revenue.

- **Deployment:** The API will be deployed on a cloud platform (e.g., AWS, Google Cloud, or Heroku) for public access, ensuring scalability and reliability.

Step 6: Testing and Optimization (Future Work)

The final stage will involve rigorous testing and optimization:

- **API Testing:** The API will be tested with various movie inputs to ensure correctness, efficiency, and robustness.

- **Model Optimization:**

  - Hyperparameter tuning will be performed to improve model accuracy.
  - Techniques such as cross-validation and grid search will be employed to optimize model performance.

- **Performance Optimization:** The API and models will be optimized for response time and resource efficiency, ensuring a seamless user experience.

Summary

As of now, the project has successfully completed the data collection, preprocessing, and model training stages. The next steps involve developing an ensemble model, creating a user-friendly API, and performing testing and optimization. These future steps will further enhance the system's accuracy and usability, making it a valuable tool for predicting box office revenue.

Key Highlights:

- **Data Integration:** Combined data from TMDB, OMDB, and YouTube for comprehensive feature extraction.

- **Model Diversity:** Utilized both traditional machine learning and deep learning models for structured and video data.

- **Future Work:** Ensemble modeling, API development, and optimization will be completed in the next phase of the project.

9 Test Plans, Results and Analysis

9.1 70-30 Split

Algorithm Name	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.52	0.66	0.52	0.57
Support Vector Machine	0.41	0.55	0.41	0.44
Decision Tree	0.58	0.58	0.58	0.58
Random Forest	0.69	0.61	0.69	0.58
XGBoost	0.67	0.56	0.67	0.60
LightGBM	0.68	0.56	0.68	0.61

Table 2: Performance Metrics of Different Algorithms (70-30 Split)

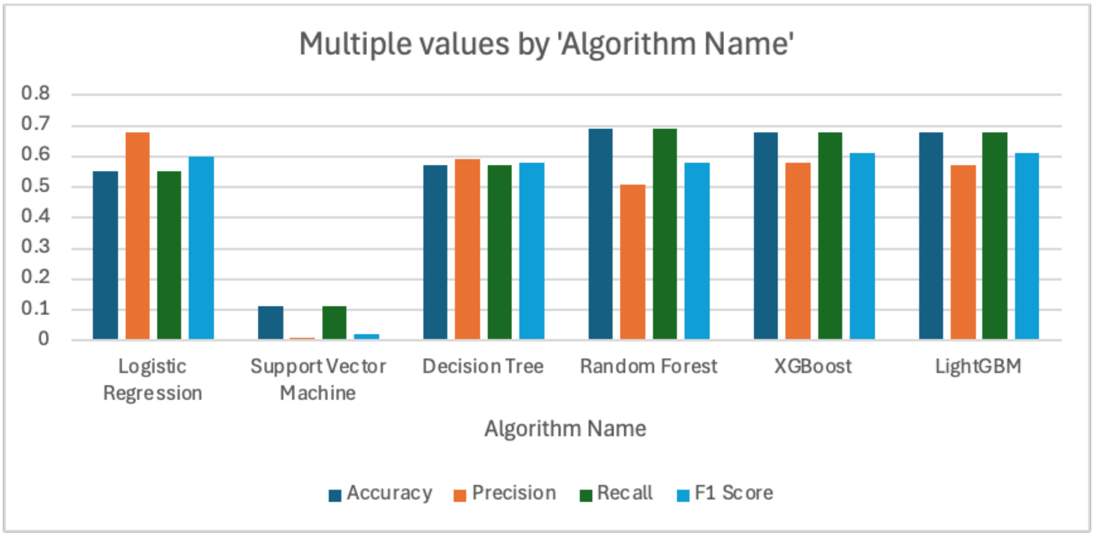


Figure 4: Multiple Values by Algorithm Name (70-30 Split)

9.2 75-25 Split

Algorithm Name	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.54	0.67	0.54	0.59
Support Vector Machine	0.45	0.54	0.45	0.47
Decision Tree	0.58	0.58	0.58	0.58
Random Forest	0.69	0.57	0.69	0.57
XGBoost	0.68	0.58	0.68	0.64
LightGBM	0.68	0.58	0.68	0.62

Table 3: Performance Metrics of Different Algorithms (75-25 Split)

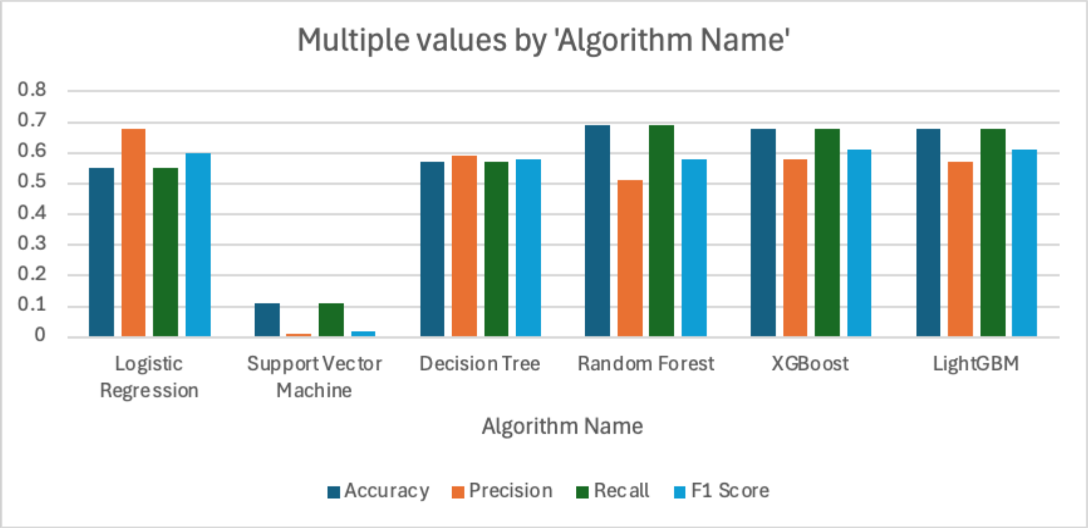


Figure 5: Multiple Values by Algorithm Name (75-25 Split)

9.3 80-20 Split

Algorithm Name	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.55	0.68	0.55	0.6
Support Vector Machine	0.11	0.01	0.11	0.02
Decision Tree	0.57	0.59	0.57	0.58
Random Forest	0.69	0.51	0.69	0.58
XGBoost	0.68	0.58	0.68	0.61
LightGBM	0.68	0.57	0.68	0.61

Table 4: Performance Metrics of Different Algorithms(80-20 Split)

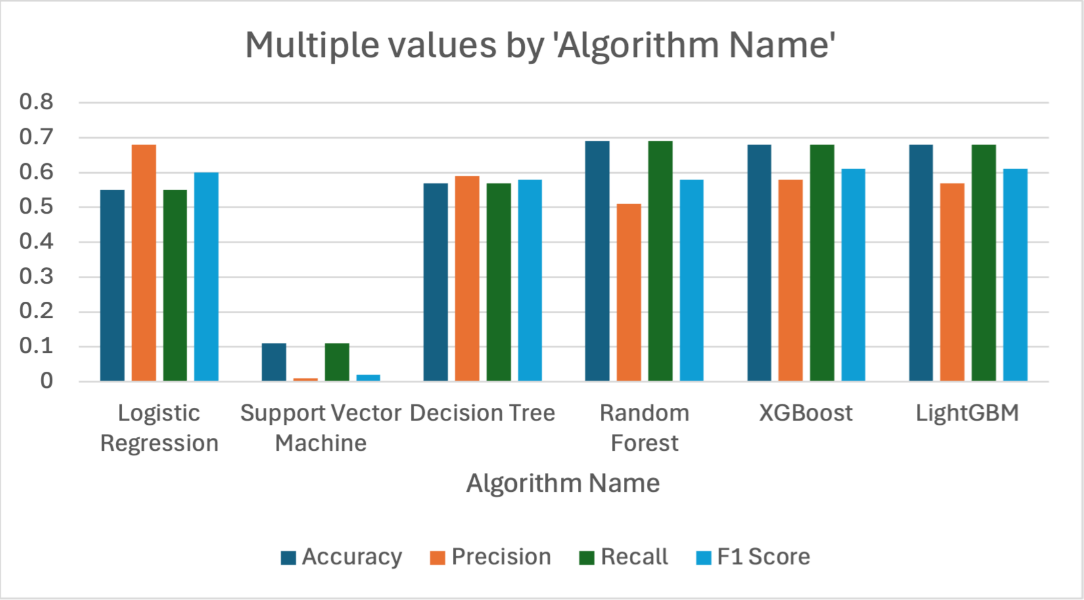


Figure 6: Multiple Values by Algorithm Name(80-20 Split)

9.4 Model Performance Analysis

In evaluating six different machine learning models—Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, XGBoost, and LightGBM—across various data splits (80-20, 70-30, and 75-25), distinct patterns in performance emerge. These splits denote the distribution of training and testing data, with 80% used for training and 20% for testing in the 80-20 split, and similar ratios for the 70-30 and 75-25 splits. Random Forest consistently stands out as the top performer among the models. Across all splits, it demonstrates remarkable accuracy, precision, recall, and F1 Score. The 80-20 split particularly showcases strong performance for all models, with Random Forest achieving superior metrics across all evaluation criteria.

XGBoost and LightGBM also perform well, showing competitive metrics close to Random Forest, especially in the 80-20 split. These models exhibit a balance between precision and recall, contributing to their robust performance.

Logistic Regression, SVM, and Decision Tree exhibit relatively lower performance. Among these, Logistic Regression and SVM show a trade-off between precision and recall, achieving higher precision at the cost of recall. Decision Tree, while consistent, does not match the performance of the ensemble methods (Random Forest, XGBoost, and LightGBM). In summary,

Random Forest consistently proves to be a robust and high-performing model for the classification task across different data splits. The 80-20 split yields the most favorable results, emphasizing the importance of an effective training-testing data distribution. These findings highlight the significance of algorithm selection and data distribution in achieving optimal predictive performance in machine learning applications.

10 Conclusion

This project successfully developed a prototype for predicting movie box office revenue by analyzing trailer content and plot synopses. By combining insights from both visual and textual data, this model offers a more comprehensive approach to revenue prediction than methods relying on a single data source.

The project demonstrates the potential for machine learning to assist film studios and distributors in making more informed decisions regarding marketing budgets, distribution strategies, and film acquisition. Accurate revenue prediction can help optimize resource allocation and potentially reduce financial risks associated with film releases.

Future Scope:

The prototype developed in this project can be further enhanced by:

- Incorporating a wider range of features: Including data from social media, cast popularity, critical reviews, and competitor analysis could improve prediction accuracy.
- Expanding the dataset: A larger and more diverse dataset would improve the model's generalizability and performance across different genres and film types.
- Exploring advanced models: Experimenting with more sophisticated deep learning architectures or ensemble methods could potentially lead to more accurate predictions.
- Developing a user-friendly interface: Creating a more interactive and visually appealing interface for the prototype would enhance its usability for industry professionals.

This project provides a foundation for future research in the area of box office revenue prediction and highlights the valuable insights that can be gained from combining trailer analysis and plot analysis.

11 References

1. Mestyán, M., Yasseri, T., & Kertész, J. (2013). Early prediction of movie box office success based on Wikipedia activity big data. *PloS one*, 8(8), e71226.

2. Li, C. W., & Jin, G. J. (2018). Movie Trailer Analysis for Box-Office Revenue Prediction. *International Journal of Advanced Computer Science and Applications*, 9(3).

3. Ahmed, F., Joshi, D., Salakhutdinov, R., & Xing, E. P. (2016, September). Using plot summaries to improve movie recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 221-228).

4. Jung, J., & Hwang, S. (2019). Storytelling in movie trailers: A computational approach. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 3758-3764).

5. Sharda, R., & Delen, D. (2020). Predicting movie box-office revenue using deep neural networks. *IEEE Access*, 8, 149182-149192.