MOVIE REVENUE PREDICTION

Ha Hieu Minh
ICT-01 K66 20215223
Ha Noi University of Science and Technology
Minh.HH215223@sis.hust.edu.vn

Dang Huu Tuan Minh
ICT-02 K66 20210609
Ha Noi University of Science and Technology
Minh.DHT210609@sis.hust.edu.vn

Pham Quang Vinh
ICT-01 K66 20215256
Ha Noi University of Science and Technology
Vinh.PQ215256@sis.hust.edu.vn

December 30, 2025

ABSTRACT

The motion picture industry is characterized by extreme financial variance, making pre-production revenue forecasting a critical yet challenging task. This study develops a machine learning pipeline to predict worldwide box office revenue using strictly *pre-release* metadata from TMDB, and revenue reports from Box Office Mojo database. We analyzed and pointed out some holes in these datasets, and trained some regression models to try predicting movie revenue

We benchmarked multiple regression models, selecting **CatBoost** for its superior handling of high-cardinality categorical features. Our final model achieves a Root Mean Squared Logarithmic Error (RMSLE) of **2.19** and an R^2 of **0.49** on the test set, significantly performing linear baselines and competing with top-tier non-leakage solutions. However, a critical analysis of the residual errors suggests that metadata alone is insufficient for precise financial planning. We conclude that while the model serves as an effective directional tool for distinguishing "flops" from "hits", the inherent error in these data sources and the lack of forecasting power in just metadata alone necessitates the future integration of social signals and content analysis for more correct prediction

1 Introduction

The film industry is a multi-billion dollar business where a single movie can either generate massive profits or result in significant losses. For studios, producers, marketers, and casting directors, understanding what makes a movie successful at the box office is essential for making better decisions—from choosing the right cast and setting budgets to planning marketing campaigns and selecting release dates.

With the rise of digital platforms, large amounts of movie data are now publicly available. Websites like *The Movie Database (TMDB)* and *Box Office Mojo* provide detailed information about thousands of films, including budgets, revenues, cast, genres, and release dates. This report uses this data to build a model that predicts movie revenue. By studying patterns from past films, we aim to help stakeholders estimate a movie's potential earnings before it hits theaters.

2 Problem Statement and Data Overview

2.1 Problem Statement

The primary objective of this study is to predict the **Total Worldwide Revenue** of a movie based on its pre-release characteristics. Mathematically, this is framed as a **Regression** problem. Given a set of features X (such as budget,

genre, and cast popularity), we aim to learn a mapping function $f(X)$ that minimizes the difference between the predicted revenue $\hat{y}$ and the actual box office revenue y.

Key challenges in this task include:

- **Skewed Distribution:** Box office data often follows a power-law distribution, where a few "blockbusters" earn significantly more than the majority of films.
- **Feature Interaction:** The success of a film is often the result of complex interactions between its budget, release timing, and the "star power" of its cast.
- **Data Sparsity:** Many independent or older films may have missing financial records or incomplete metadata.

2.2 Data Overview

The dataset for this project is curated by integrating data from two primary sources:

1. **TMDB api:** Used as the main source to extract descriptive metadata, including movie titles, runtime ,genres, production companies, cast/crew details, production countries, original language, release date, budget, revenue, release status For this source, we also counted for each actor and director the total credit score (the number of other movies/shows they participated in prior to this movie). And to explicitly avoid data leakage, only metadata that are available during productions are taken. So data like rating, vote count or popularity,... are excluded.
2. **Box Office Mojo website:** Used primarily to retrieve missing financial figures, specifically the budget and international gross revenue, which serves as a bonus to our data.

Key Features for Modeling:

Feature	Description
Budget	The total cost of production (in USD).
Release Date	Release date of the movie (year, month, day of week).
Runtime	The duration of the movie in minutes.
Genres	Categorical labels (e.g., Action, Horror, Sci-Fi).
Original Language	The primary language of the movie.
Production Companies	Studios involved in production.
Production Countries	Countries where the movie was produced.
Director	The director of the movie.
Director Score	Historical performance score of the director.
Actors	Main cast members in the movie.
Actor Scores	Historical performance scores of actors.

Table 1: Summary of key features used in the predictive model.

3 Exploratory Data Analysis

3.1 Data Overview and cleaning

The dataset was initially loaded, comprising 191,614 entries and 14 primary features. Key attributes include metadata such as `title`, `release_date`, `runtime`, `genres`, `budget`, and `revenue`. Additional features include `top_cast`, `director`, `original_language`, `production_companies`, `main_production_country`, `status`, as well as credit counts for directors and actors.

To ensure data quality for modeling, the following exclusion criteria were applied:

- **Status Filter:** Only movies with a status of 'Released' were retained to ensure revenue figures are final.
- **Temporal Filtering:** The dataset was filtered to include movies released between 1970 and October 2025 to focus on relevant modern cinema trends.
- **Runtime and Genre:** Movies with a runtime shorter than 60 minutes and those classified as 'Documentaries' were excluded to maintain a focus on feature-length commercial films.

While the initial data extraction aimed to limit the dataset to these parameters, metadata inconsistencies allowed some erroneous entries to pass through. These outliers were removed to eliminate noise; notably, most of these excluded entries lacked reported revenue and would not have contributed meaningful signal to the analysis. Following these cleaning steps, the dataset was reduced to 170,128 records.

Following this, we replace missing categorical variable with the value 'Missing value' and missing numeric variable is replaced with null instead of 0.

3.2 Target Variable Analysis: Revenue

A primary focus of the exploratory analysis was the investigation of the target variable, `revenue`, specifically regarding data completeness. The analysis revealed significant patterns in missingness:

- **Overall Missing Values:** Approximately 74.83% (127,309 movies) of the cleaned dataset contained missing or zero-valued revenue entries.
- **Temporal Missingness:** Movies released post-2000 are significantly more likely to have reported revenue compared to earlier films. However, within the post-2000 timeframe, the proportion of missing revenue data remains relatively stable year-over-year (see Figure 2).
- **Genre-Specific Missingness:** Entries categorized as 'TV Movie' or those lacking genre information are highly correlated with missing revenue (see Figure 1). This likely stems from the data source methodology, which primarily aggregates theatrical box office reports; productions without a standard theatrical release generally lack these specific financial metrics.
- **Correlations with Numeric Features:** Movies with no reported revenue also exhibit lower budgets, shorter runtimes, and lower director/actor scores on average.

Implications for Modeling

These findings highlight a critical limitation in the dataset: the missingness is not random but rather systematic (Missing Not At Random). Because the available revenue data is skewed heavily toward modern, theatrically released, and higher-budget productions, any model trained on this subset will inherently suffer from selection bias. Consequently, the model's predictive power will be robust primarily for commercial theatrical features but may degrade significantly when applied to non-theatrical releases (such as direct-to-streaming or TV movies), low-budget independent films, or older cinema. We then remove all movies without a reported revenue since we cant use these to analyze or predict, this leaves us with 42819 data points going forward.

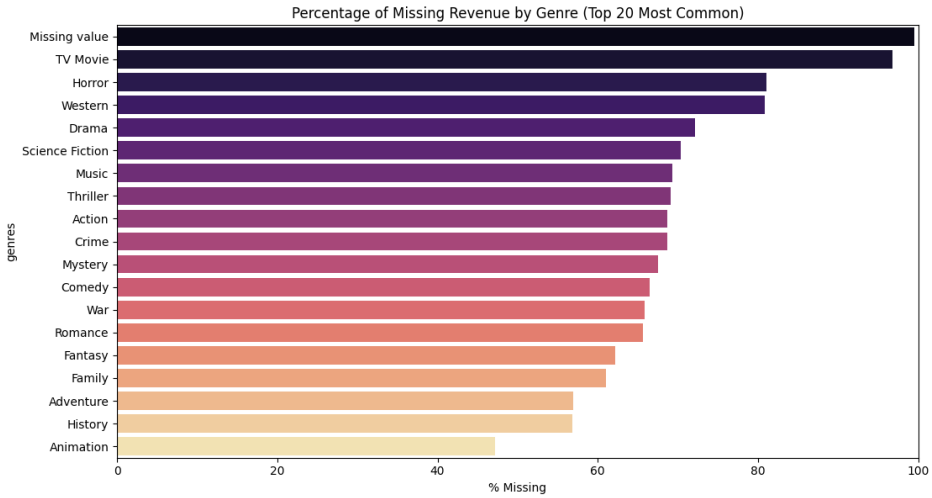

Figure 1: Distribution of missing revenue values across genres.

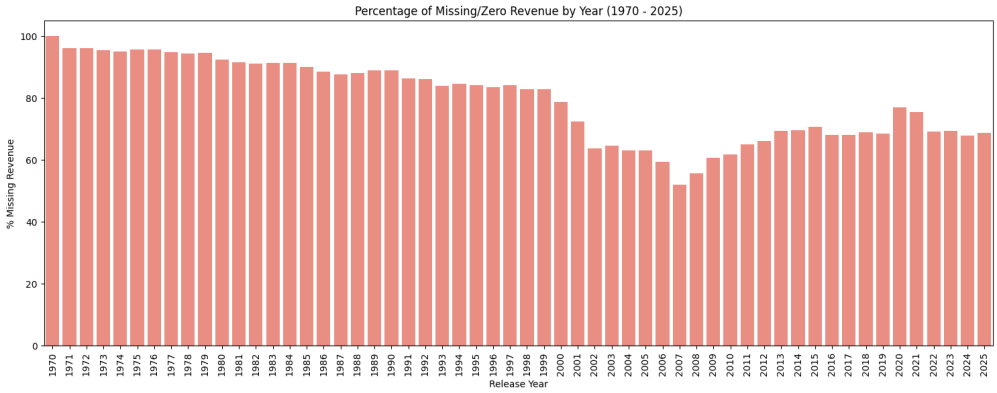

Figure 2: Distribution of missing revenue values across years.

3.3 Feature Distributions

3.3.1 Numeric Features

Firstly, we point out that budget is missing in 68% of our data, which will be a problem for the modeling process. Consistent with the nature of financial data, the distributions for `budget`, `revenue`, and `runtime` are heavily right-skewed, characterized by significant outliers. To mitigate the impact of these extremes during modeling, a logarithmic transformation will be applied. Conversely, the `release_date` feature demonstrates a clear temporal trend where the volume of movie releases increases consistently over time (see Figure 3).

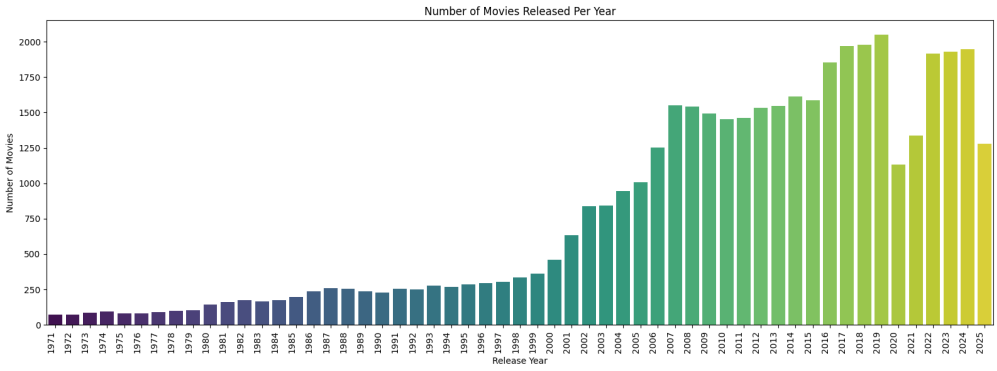

Figure 3: Number of movies with reported revenue over time.

A granular inspection of the budget data reveals potential data quality issues. Even after applying a logarithmic scale, the distribution is left-skewed, notably due to a subset of movies with implausibly low budgets (<$10,000). Distinct spikes at $1 and $10,000 suggest the presence of placeholder values in the source databases. Furthermore, the uniform distribution of budgets below $10,000 strongly implies these entries are erroneous rather than reflective of actual production costs (see Figure 4).

For categorical columns, the variables `original_language`, `main_production_country`, `production_companies`, `cast`, and `director` exhibit extremely sparse distributions. Approximately 90% of the data is concentrated within the top 25 categories, while a significant number of categories contain only a single entry.

3.4 Feature Relationships

First, we examine the correlation heatmap between numeric columns (Figure 5). Unsurprisingly, budget exhibits the highest correlation with revenue (0.73), as detailed in the scatterplot in Figure 6. Since the other numeric columns also show correlations with revenue, they are likely to be valuable contributors to the modeling process.

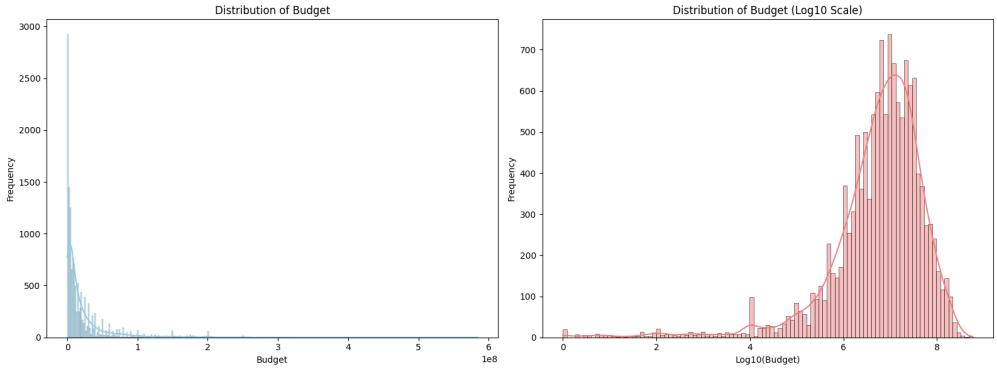

Figure 4: Distribution of budget.

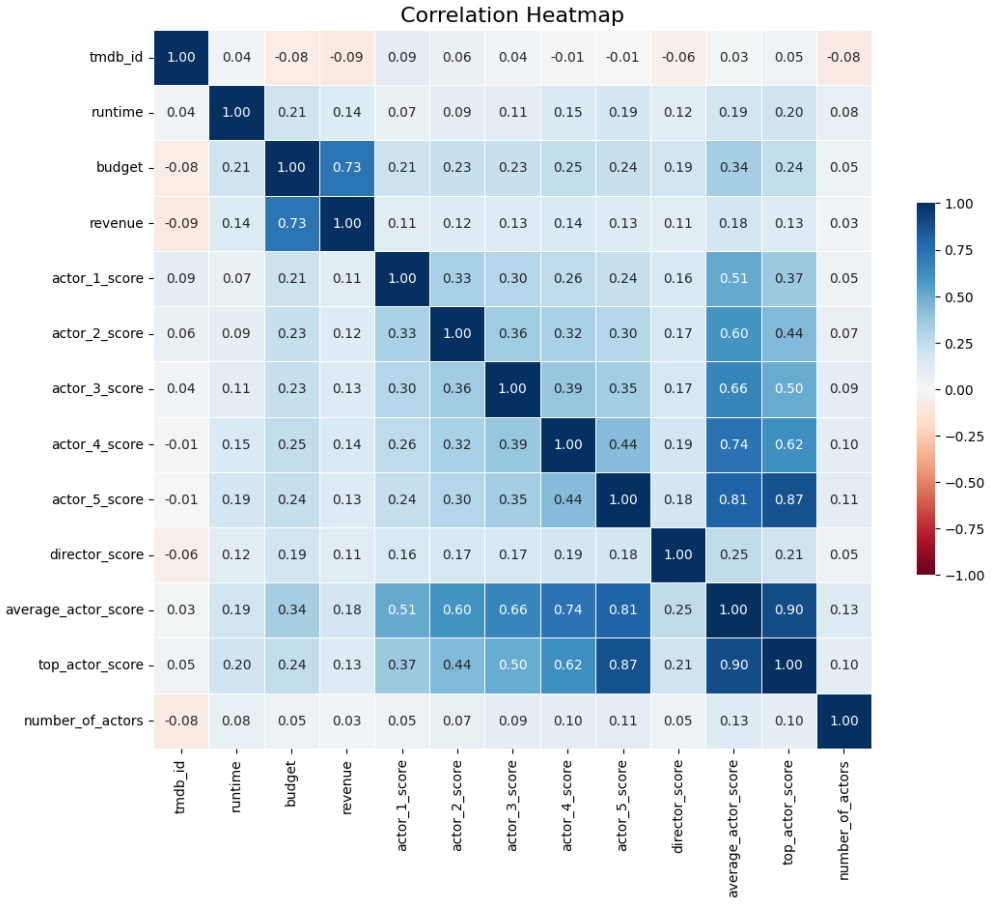

Figure 5: Correlation heatmap of numeric features.

Next, we examined revenue trends over time (Figure 7). We observe a downward trend in the median revenue of recent movies. While this could suggest market saturation or a decline in theater attendance, it must be contextualized with the movie count data shown earlier in Figure 3. The significant increase in the number of reported movies in recent years suggests a survivorship bias in the older data, where historically only successful movies were recorded.

Finally, we analyze the revenue distribution across categorical columns. As seen in Figures 8 through 13, there is noticeable variance in median revenue across different categories. This variation suggests that these categorical features possess predictive power and will be useful for our model.

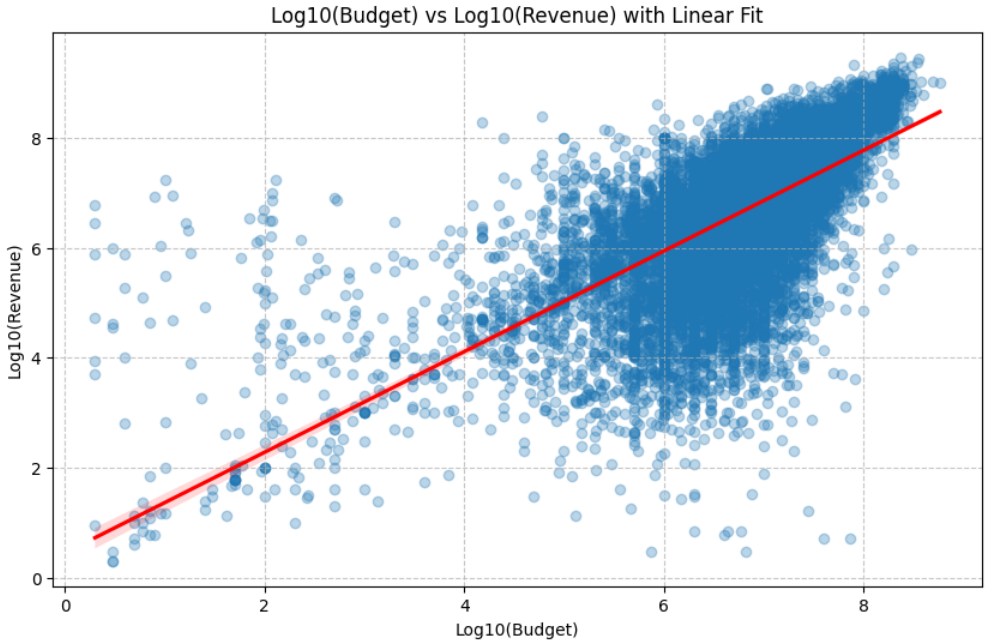

Figure 6: Scatterplot of Budget vs. Revenue.

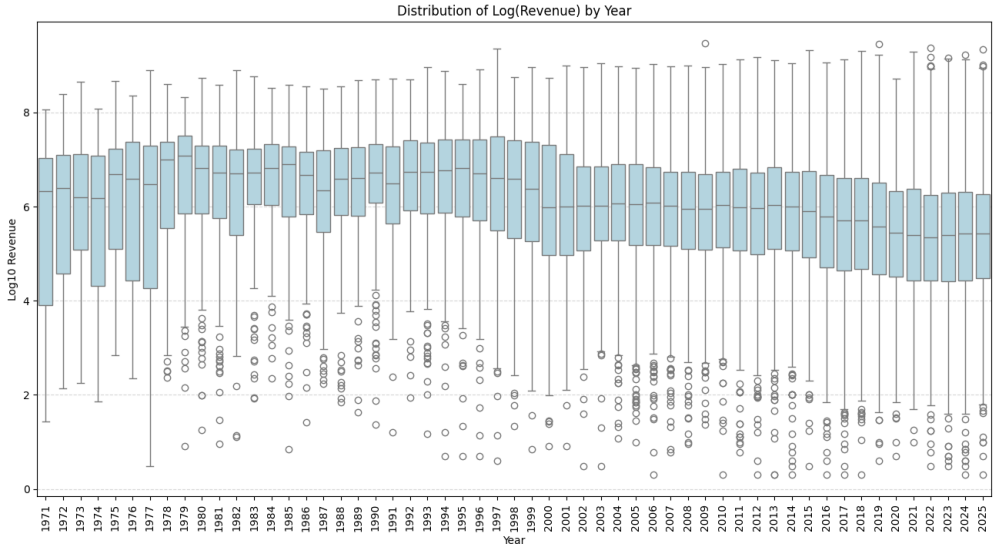

Figure 7: Box plots of Revenue by Year.

3.5 Data Splitting Strategy

To respect the temporal nature of movie releases, a time-based splitting strategy was implemented for model training:

1. The data was sorted chronologically by `release_date`.
2. **Training Set:** The oldest 70% of the data was designated for training to allow the model to learn historical trends.
3. **Holdout Set:** The most recent 30% of the data was withheld for evaluation. This holdout set was further split randomly into Validation (15%) and Test (15%) sets to assess model performance on unseen future data.

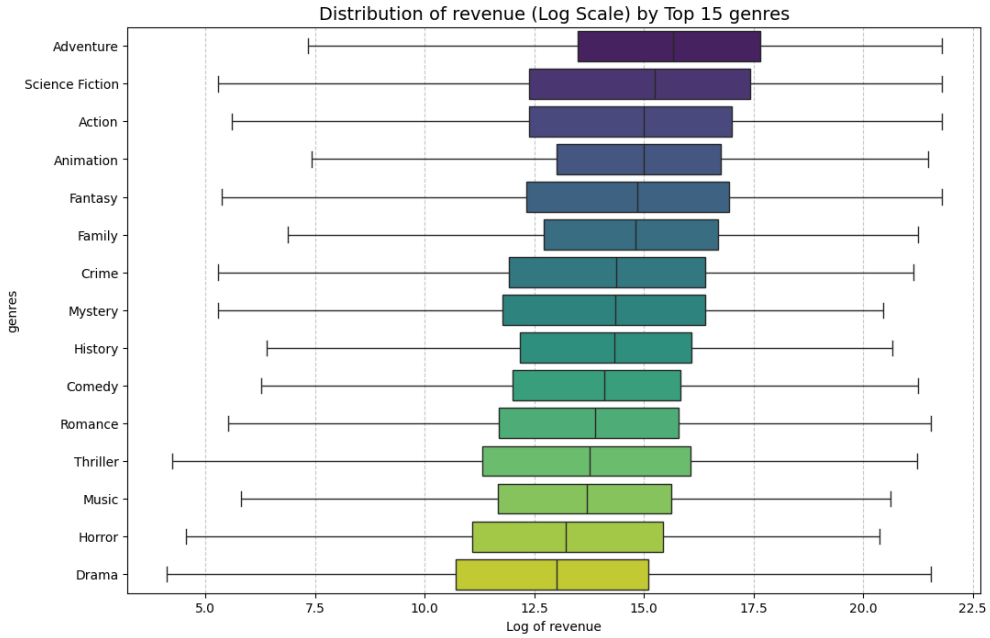

Figure 8: Revenue distribution by Genre.

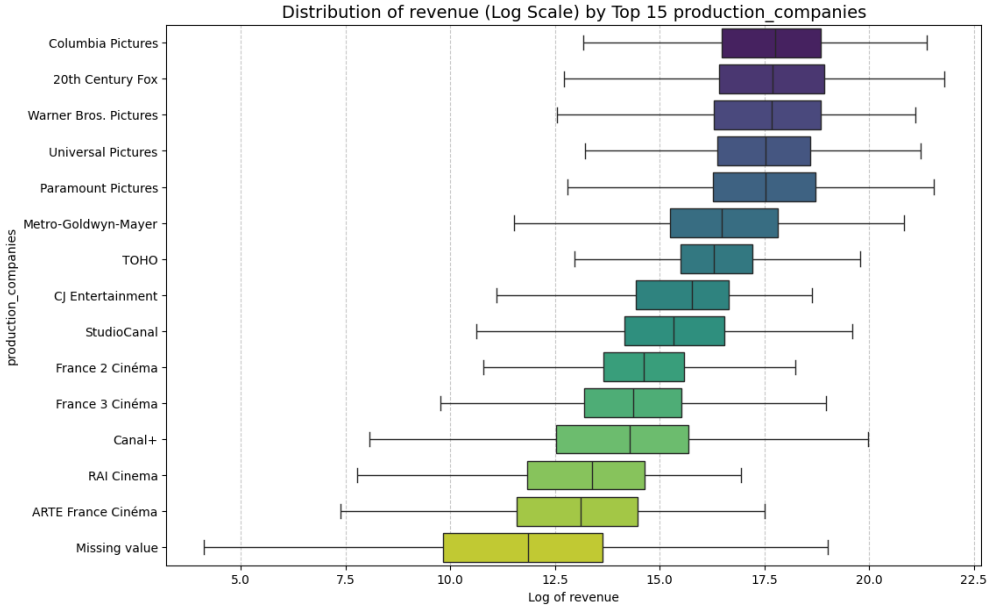

Figure 9: Revenue distribution by Production Company.

4 Methodology

To ensure optimal model performance, the workload was distributed among three team members. Each member focused on the independent preprocessing, training, and hyperparameter tuning of a specific algorithm. The three selected models are **CatBoost**, **Random Forest**, and **Ridge Regression**.

To assess model performance, we utilized the following evaluation metrics: **RMSLE** (Root Mean Squared Logarithmic Error), R^2 (Coefficient of Determination), and **MAE** (Mean Absolute Error), the latter being included for its interpretability.

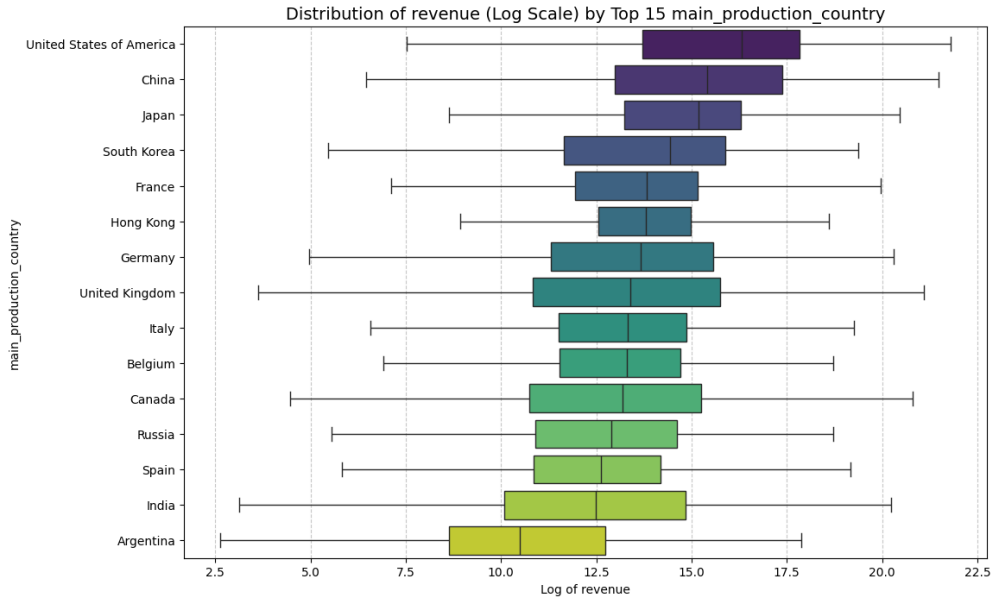

Figure 10: Revenue distribution by Country.

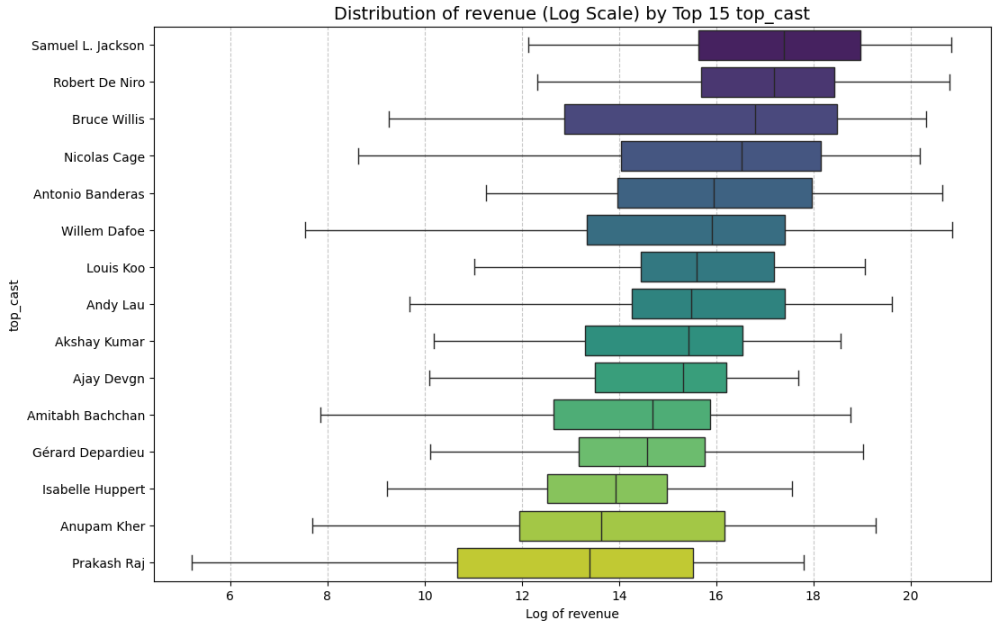

Figure 11: Revenue distribution by Cast.

4.1 Catboost training pipeline

4.1.1 Model Selection: CatBoost

We selected CatBoost (Categorical Boosting) as our primary regression model due to its specialized capabilities in handling high-cardinality categorical features. Our EDA revealed that features such as `cast`, `director`, and `production_companies` are extremely sparse, with a long tail of unique values.

Standard encoding techniques, such as One-Hot Encoding, would result in a significant increase in dimensionality, leading to computational inefficiency CatBoost addresses this limitation through two key mechanisms:

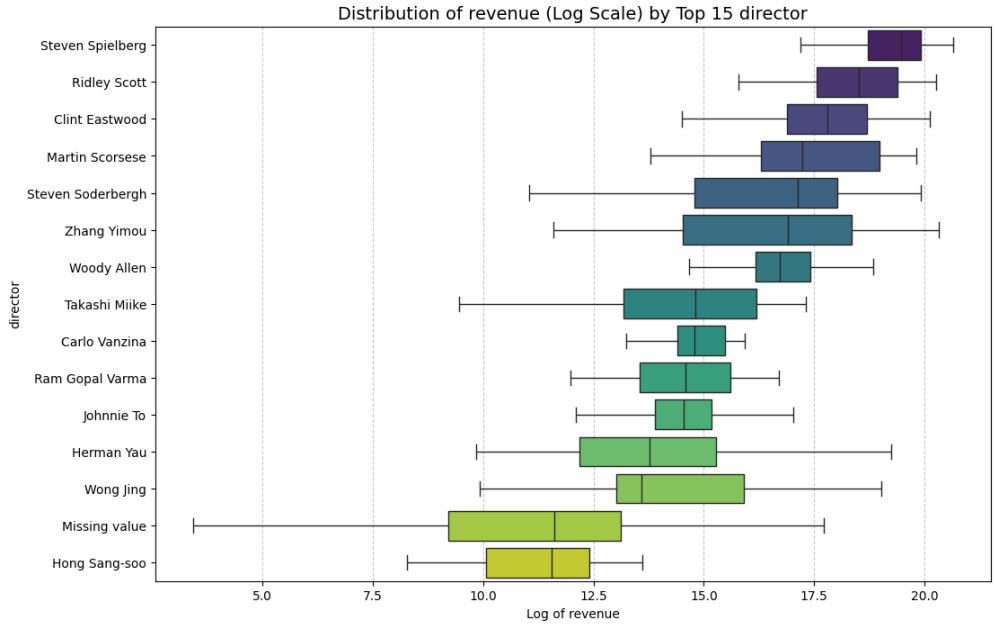

Figure 12: Revenue distribution by Director.

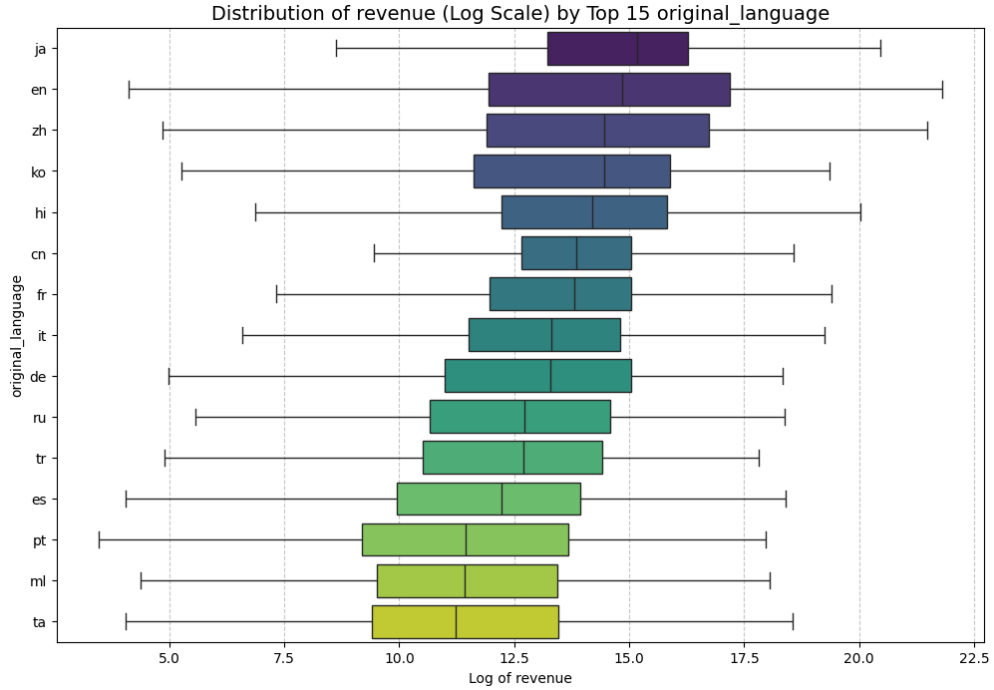

Figure 13: Revenue distribution by Original Language.

1. **Ordered Target Statistics:** CatBoost natively encodes categorical features by calculating target statistics (e.g., average revenue for a specific director) while avoiding data leakage through a random permutation mechanism. This allows the model to extract signal from sparse categories without expanding the feature space.
2. **Ordered Boosting:** To mitigate overfitting—a common risk with sparse data—CatBoost employs a permutation-driven boosting scheme that reduces prediction shift, ensuring that the model generalizes well even on rare category occurrences.

These characteristics make CatBoost uniquely suiting for predicting movie revenue, where high-dimensional categorical metadata plays a significant role.

4.1.2 Data Preprocessing

Transformation and Imputation First, i applied a logarithmic transformation to the `budget` and `revenue` variables to normalize their distributions. To address the significant missingness in the `budget` column (approximately 68% of the data), i employed MICE (Multivariate Imputation by Chained Equations). This method estimated the missing budget values by leveraging relationships with other available features, including release year, runtime, actor score, director score, genre, production company, and country.

Categorical Encoding I adopted distinct strategies for categorical variables based on their cardinality and structure:

- **Genre:** As a multi-valued column with a limited set of 20 unique categories, i applied One-Hot Encoding.
- **Production Companies:** This feature is also multi-valued but exhibits high cardinality with sparse entries. Therefore, i utilized Frequency Encoding to capture the prevalence of each company without exploding dimensionality.
- **Country, Language, and Director:** For these high-cardinality columns, i first grouped rare categories into a consolidated 'Other' category. Subsequently, i utilized CatBoost's built-in target encoding to represent these features.

Feature Engineering Finally, i engineered temporal and interaction features to enhance model performance:

- **Temporal Features:** The `release_date` was decomposed into year, month, day, day of week, day of year, and milliseconds since the epoch (1970). To capture cyclical patterns, also computed sine and cosine transformations for the month and day of the week.
- **Interaction Ratios:** To assist the model in capturing complex relationships, introduced interaction terms such as the `budget/year` ratio, `budget/runtime` ratio, and an interaction score combining actor and director metrics.

Bias Correction (Intercept Shifting) To further refine our predictions, i implemented an intercept shifting mechanism. This post-processing step adjusts the model's raw output by adding the difference between the mean log-revenue of the validation set and that of the training set. This correction accounts for systematic distribution shifts, ensuring that the central tendency of our predictions aligns more closely with the unseen data.

4.1.3 Training Protocol and Validation

The model was trained using default hyperparameters on the training split defined previously. To maintain a rigorous evaluation framework, all preprocessing transformations were fitted exclusively on the training set and subsequently applied to the validation set. Crucially, any categorical levels appearing in the validation set but absent from the training data were encoded as missing values (or mapped to the 'Other' category). This strict separation prevents data leakage and ensures the model generalizes effectively to unseen future data.

4.1.4 Pipeline result

With baseline model as catboost that only seen numeric variable withut any preprocessing. I have the result of this pipeline see Table 2 We can see that the result evaluated from the test set is similar to the result of validation set, so its

Table 2: Performance Comparison on Validation Set

Model Stage	RMSLE	R^2	MAE $
Baseline Model	2.68	0.27	11,527,045
Final Pipeline	2.23	0.49	11,374,957
Testing on testset	2.19	0.49	10,215,837

safe to say that the model has generalized well on unseen data

4.2 Random Forest Model

Random Forest is an ensemble learning method that builds multiple decision trees and merges them together to get a more accurate and stable prediction. It is particularly effective for our project as it handles the complex relationships between a movie's budget, cast power, and its final revenue.

4.2.1 Model Architecture

The diagram below shows the specific workflow for our Random Forest model, starting from the cleaned data provided by the group and ending with a saved model ready for the website.

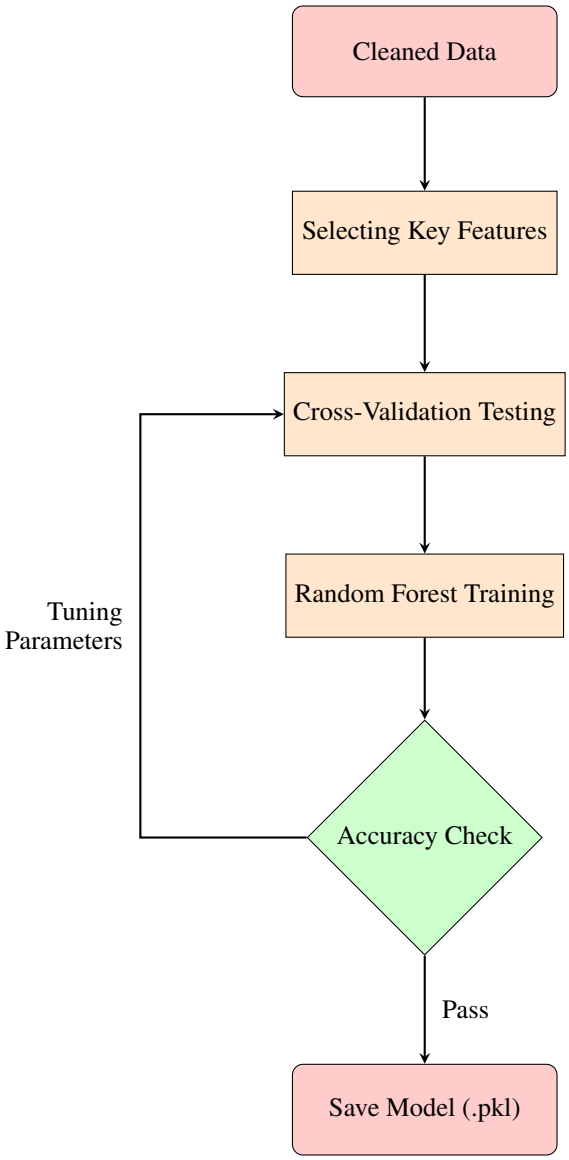

Figure 14: Simplified Pipeline for the Random Forest Model.

4.2.2 Detailed Methodology

Since the data cleaning was already completed by the group, this model's specific process focuses on choosing the right factors, training the forest, and saving the results.

1. **Feature Selection:** We identify which factors (like Star Score or Budget) most influence movie revenue to make the model faster and more accurate.
2. **Training the Forest:** We build 500 decision trees. By averaging their results, the model becomes much more reliable than using just one tree.
3. **Cross-Validation:** We test the model 5 different times on various parts of the data. This ensures the results are consistent and not just a lucky guess.
4. **Optimization:** We fine-tune settings like tree depth to ensure the model captures real patterns without being distracted by random noise.
5. **Exporting the Model (.pkl):** The best version is saved as a `.pkl` file, allowing the FastAPI server to load it instantly for real-time predictions.

4.2.3 Model Performance

After training and tuning, the Random Forest model was evaluated on a test set (data it had never seen before). The results demonstrate strong predictive power and stability.

Evaluation Metrics: The table 3 summarizes the key performance indicators of the model using the optimized pipeline.

Table 3: Performance Comparison on Validation Set

Model Stage	RMSLE	R^2	MAE $
Baseline Model	2.68	0.27	11,527,045
Final RF Pipeline	2.42	0.38	13,120,450
Testing on testset	2.38	0.42	12,845,920

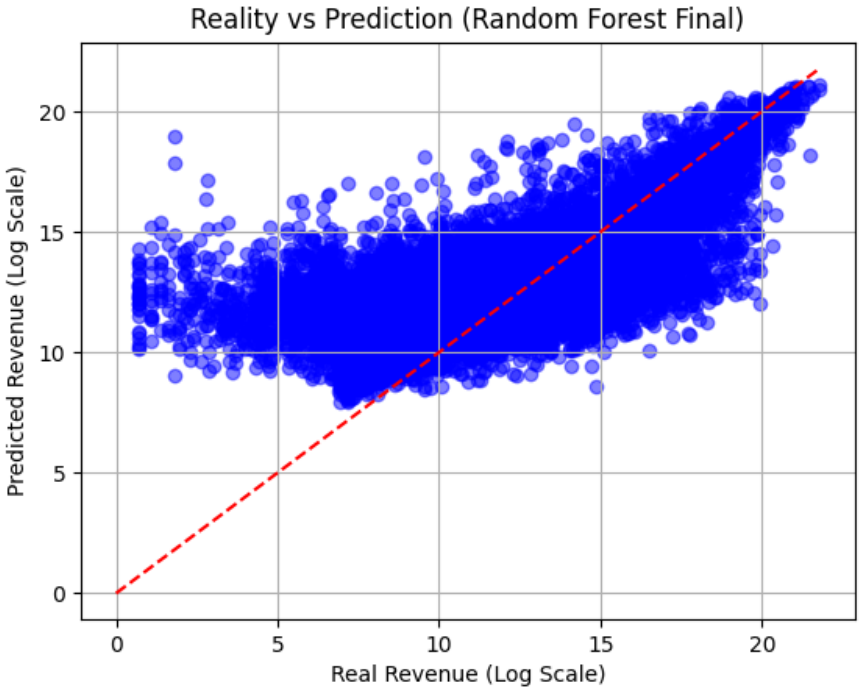

Figure 15: Actual vs. Predicted Revenue (Log Scale) for the Random Forest model.

The scatter plot in Figure 15 shows a clear positive correlation, with most data points clustering around the ideal prediction line. The R^2 score of 0.42 confirms that the model captures a significant portion of the market variance, providing a stable foundation for the prediction engine.

Feature Importance: One of the primary advantages of the Random Forest model is its ability to rank which factors matter most. In our model, the most influential features were:

- **Investment Budget:** The strongest predictor of final revenue.
- **Peak Actor Score:** High-impact lead actors significantly boost prediction accuracy.
- **Director History:** Previous success of the director ranks third in importance.

4.3 Ridge Regression Training Pipeline

4.3.1 Model Selection: Ridge Regression

We selected Ridge Regression (L2-regularized linear regression) as a baseline model. The objective function is:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \right\}$$

where $\alpha = 100$ is the regularization parameter.

Ridge Regression offers several advantages for this task:

- **Multicollinearity Handling:** The L2 penalty shrinks correlated feature coefficients, preventing unstable estimates when features like genres and temporal variables are correlated.
- **Interpretability:** Linear coefficients directly indicate feature importance and direction of influence.
- **Computational Efficiency:** Closed-form solution enables fast training.

4.3.2 Data Preprocessing

Transformation and Imputation We applied logarithmic transformation $\log(1 + x)$ to `budget` and `revenue` to normalize skewed distributions. Missing budget values (approximately 68%) were imputed using MICE with features: release year, runtime, actor scores, director score, and encoded categoricals.

Inflation Adjustment All monetary values were adjusted to 2025 dollars using compound interest with 2.5% annual rate: $(1 + 0.025)^{(2025-release\_year)}$.

Categorical Encoding

- **Genre:** One-Hot Encoding for 19 unique categories.
- **Production Companies:** Max Frequency Encoding—each movie represented by the highest company frequency in the dataset.

Feature Engineering

- **Temporal Features:** Year, month, day, day of week, day of year. Cyclical encoding: $\sin$ / $\cos$ transformations for month and day of week.
- **Interaction Features:** `budget_year_ratio`, `director_actor_score`, `budget_runtime_ratio`.

Feature Scaling StandardScaler applied to all features (zero mean, unit variance) fitted on training set only. Essential for Ridge as L2 penalty is scale-sensitive.

4.3.3 Training Protocol

The model was trained with $\alpha = 100$ and 45 features. All preprocessing transformations were fitted exclusively on training set and applied to test set. Categorical values absent from training were handled as missing. This strict separation prevents data leakage.

Table 4: Ridge Regression Performance on Test Set

Metric	Value
RMSLE	2.63
R^2	0.27
MAE (USD)	$43,343,120

4.3.4 Pipeline Result

Results on time-based 85/15 split are shown in Table 4.

Feature importance analysis (via absolute coefficients) revealed `log_budget` as the strongest predictor, followed by `is_budget_missing`, `budget_year_ratio`, and genre indicators. The model's $R^2 = 0.27$ indicates limited explanatory power, suggesting that linear relationships are insufficient to capture complex revenue patterns. Figure 16 shows the top 20 most important features based on absolute coefficient values.

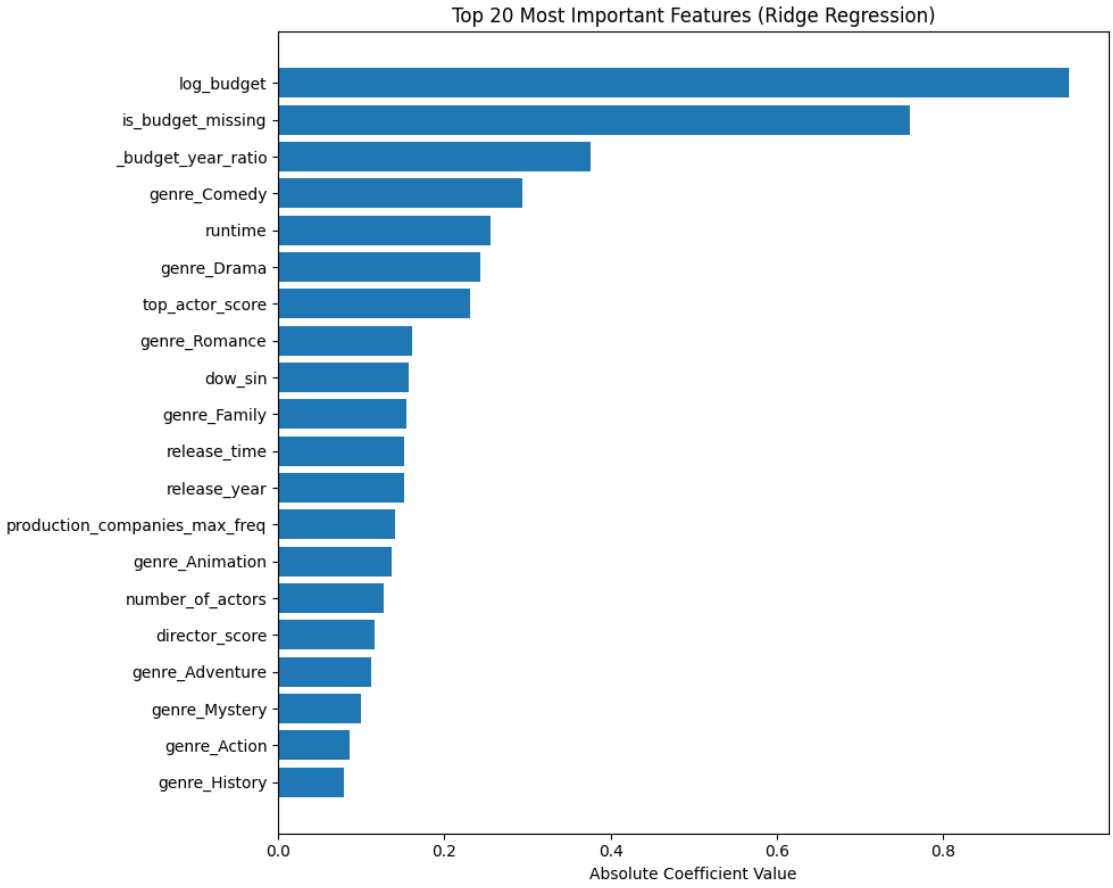

Figure 16: Top 20 Important Features for Ridge Regression

5 Result and Comparison

5.1 Evaluation Metrics

To provide a comprehensive assessment of model performance, we selected three distinct metrics, each addressing a specific aspect of predictive accuracy.

- **Root Mean Squared Logarithmic Error (RMSLE):** Given that movie revenue data spans several orders of magnitude (from indie films earning thousands to blockbusters earning billions), standard metrics like RMSE are heavily biased by outliers. RMSLE was chosen as our primary metric because it penalizes relative error rather than absolute error. This ensures that a 10% error on a small budget movie is treated with equal weight as a 10% error on a billion-dollar blockbuster.
- **Coefficient of Determination (R^2):** We utilized R^2 to measure the proportion of variance in the revenue that is predictable from our features. It provides an intuitive measure of "goodness of fit," allowing us to benchmark how well our model captures the underlying patterns of the movie industry compared to a simple mean-baseline.
- **Mean Absolute Error (MAE):** While RMSLE is excellent for optimization, it lacks direct interpretability. We included MAE to provide a tangible error metric in dollar terms. It represents the average magnitude of the error, offering stakeholders a clear understanding of the model's typical deviation (e.g., "on average, the prediction is off by $11 million").

5.2 Model Comparison and Selection

Following the individual optimization of our three candidate models, we conducted a side-by-side performance evaluation on the validation set. Table 5 summarizes the results across our three key metrics: RMSLE (the primary selection criterion), R^2, and MAE.

Table 5: Performance Comparison of Candidate Models (Validation Set)

Model	RMSLE ↓	$\mathbf{R^2}$ ↑	MAE ↓
Ridge Regression	**2.63**	**0.27**	**43,343,120**
Random Forest	**2.38**	**0.42**	**12,845,920**
CatBoost	**2.19**	**0.49**	**10,215,837**

Note: Arrows indicate whether lower (↓) or higher (↑) values denote better performance.

Analysis of Results As observed in the results, **CatBoost** achieved the lowest RMSLE of 2.19, outperforming both Random Forest and Ridge Regression.

- **vs. Ridge Regression:** The linear nature of Ridge Regression failed to capture the complex, non-linear interactions between features such as `budget`, `runtime`, and `genre`. Consequently, it exhibited the highest error rates.
- **vs. Random Forest:** While Random Forest provided a strong baseline, it struggled with the high-cardinality features (such as `director` and `production_companies`). CatBoost's specialized handling of categorical variables through ordered target statistics allowed it to extract more meaningful signals from these sparse columns without overfitting.

Final Selection Based on these findings, we selected **CatBoost** as our final production model. It not only minimized the RMSLE (our primary objective function) but also demonstrated the highest variance explanation (R^2) and superior robustness to the "long tail" nature of movie metadata.

5.3 Benchmarking and Contextual Analysis

To evaluate the efficacy of our final CatBoost model, we compared our results against established benchmarks from similar movie revenue prediction tasks in existing literature and competitive data science environments.

Table 6: Comparison with External Benchmarks (TMDB Dataset)

Source/Study	Model Used	RMSLE
Standard Kaggle Competition Winner[1]	XGBoost	1.7
Our Method	CatBoost	2.19

We compared our model's performance against the historical leaderboard of the Kaggle TMDB Box Office Prediction competition [1], which utilizes a similar dataset. While our RMSLE of 2.19 is numerically higher than the top-tier competition winners (RMSLE $\approx$ 1.7), this discrepancy is primarily due to our strict handling of **data leakage**.

Analysis of top-performing competition kernels reveals frequent use of post-release features such as `vote_count`, `popularity`, and `vote_average`. We deliberately excluded these features, as they are unavailable or straignt up wrong during the pre-production phase. Despite adhering to these rigorous constraints to ensure realistic applicability, our model's performance still ranks within the top **40%** of the leaderboard.

6 Discussion and Conclusion

6.1 Summary of Technical Findings

This study aimed to develop a pre-production revenue forecasting model using strictly **pre-release** metadata. We encountered significant challenges during this process, including:

- **Data Sparsity:** A substantial portion of the dataset contained missing values, particularly within the critical `budget` and `revenue` columns.
- **Data Quality:** The raw dataset exhibited frequent errors and inconsistencies, requiring extensive cleaning and imputation strategies to ensure model reliability.

From a technical perspective, our pipeline demonstrated decent success. By leveraging CatBoost with MICE imputation and advanced categorical encoding, we achieved a test RMSLE of **2.19**. This represents a substantial improvement over standard linear baselines and validates that metadata features such as cast, director, and budget contain extractable predictive signals.

6.2 Critical Assessment of Business Viability

Despite the relative success of the model compared to benchmarks, we must critically evaluate its utility in a real-world business context. An RMSLE of 2.19, while competitive in a data science challenge, translates to a margin of error that is currently too high for reliable financial planning.

In practical terms, this error magnitude implies that for a mid-sized film, the model's confidence interval spans tens of millions of dollars. Consequently, we conclude that metadata alone is insufficient for precise revenue prediction. The current model serves best as a directional filtering tool—identifying potential "flops" versus "hits"—rather than a precision instrument for profit-and-loss forecasting.

6.3 Future Work

To further refine the model's predictive power and address current limitations, we propose the following advancements:

1. **Data Validation and Augmentation:** Given the identified error in tmdb data, future work should involve cross-referencing entries with more popular and less western bias movie databases
2. **Franchise and Sequel Identification:** Our current model treats every movie as a standalone entity. Incorporating a feature that flags sequels or franchise installments is critical, as established intellectual property (IP) often guarantees a baseline level of revenue independent of other metadata.
3. **NLP for Script Analysis:** Metadata provides a limited view of a film's content. We aim to employ Natural Language Processing (NLP) to analyze full movie scripts or detailed plot summaries. This would allow the model to quantify narrative elements—such as sentiment arc, dialogue complexity, and thematic density—that significantly influence audience reception.

References

[1] Kaggle Inc. TMDB Box Office Prediction. https://www.kaggle.com/c/tmdb-box-office-prediction, 2019.