Name: Aditya Mangrulkar
Course: Introduction to Data Science
Date: 4/20/2025

Disclaimer: I worked on this project alone so hopefully there's some leniency with the grading. Thanks.

Project Report: Predicting Box Office Revenue & Audience Ratings Using Supervised Learning Models

Introduction

The movie industry is a high stakes, high investment domain where predicting financial outcomes and audience reception of a film before its release remains a major challenge. Hundreds of millions of dollars are invested in production, distribution, and marketing based on a mix of past experience, industry intuition, and guesswork. Yet, financial losses are still common, especially when films underperform at the box office or fail to resonate with audiences.

The central problem my project addresses is: How can we build a predictive system that estimates both the box office revenue and the audience success of a film using features known before its release? This dual goal predicting both financial and critical success is critical for production studios, investors, streaming platforms, and marketers who want to make informed, data driven decisions.

To solve this problem, I designed a supervised machine learning pipeline that uses the IMDb movie metadata dataset to build two predictive models:

- A regression model to estimate the box office revenue (gross)
- A classification model to categorize audience rating success (success_label based on IMDb scores)

The approach includes several machine learning steps: exploratory data analysis (EDA), handling missing values, feature engineering (such as creating a "star power" score), model training (using both baseline and ensemble models), evaluation using performance metrics (R^2, F1-score), and visualization for interpretability. The feature set includes budget, duration, actor/director Facebook likes, critic reviews, vote counts, and release year variables accessible before a movie's release.

This work ties directly to the lectures and papers covered in our course on supervised learning. In lecture, we studied:

- Linear regression and its limitations in capturing non linear relationships.
- Logistic regression for binary classification.
- Random Forests, an ensemble learning technique that improves predictions by averaging many decision trees.
- Evaluation metrics like R^2 for regression, and precision/recall/F1 for classification.
- Feature engineering and how meaningful transformation of raw data can drastically affect model performance.
- Train-test splitting and overfitting avoidance strategies.
- These ideas were not only applied but extended in this project. For instance, I demonstrated the limited power of linear models for predicting complex outcomes like gross revenue, then improved accuracy using ensemble methods. I also carefully engineered features like total_fb_likes to represent latent concepts which directly reflects the importance of domain-specific feature engineering discussed in class.

Furthermore, this project goes beyond many textbook exercises by tackling two different supervised learning tasks (regression and classification) in a single, integrated workflow. That dual modeling approach gives a more realistic representation of how data science is used in actual industry settings where multiple KPIs often matter and different kinds of models must be coordinated.

By grounding every modeling decision in our coursework and extending it with practical challenges like handling missing data and evaluating real world impact, this project serves as a comprehensive application of supervised learning in the context of entertainment analytics.

Motivation

The importance of this project lies in its real world applicability to a multi billion dollar industry that often operates on intuition, hype, or post hoc analysis. The financial stakes in the film industry are immense, with blockbuster movies requiring hundreds of millions in investment. Yet, success is not guaranteed. Studios, investors, and streaming services like Netflix and Amazon Prime regularly face losses when films underperform. Having reliable, data driven predictive tools to anticipate financial and audience outcomes before a movie is released can significantly reduce this risk.

This project is particularly exciting for me because it combines two passions movies and data science. The challenge of predicting something as complex and culturally variable as movie success is intellectually stimulating. It tests the limits of machine learning and highlights the importance of thoughtful feature selection, model design, and critical evaluation. It's not just a technical exercise but a practical application of data science in a high stakes domain that affects how content is created, distributed, and consumed.

Several key questions have driven both my curiosity and design of the project:

- Can financial and audience success be predicted using only pre-release information?
- Which features (cast, budget, director, year) matter most when forecasting a film's performance?
- Are simpler models (linear/logistic regression) sufficient, or do ensemble methods (random forests) significantly improve accuracy?
- Is it possible to build a general purpose framework that works across genres and release years?

These questions are not new. Prior research has explored movie success prediction, but many efforts are limited in scope or depth. For example, some models focus exclusively on sentiment analysis of post-release reviews or social media chatter, which isn't useful for pre-release decision-making. Others look only at box office revenue without considering audience ratings, which ignores how well the film was received critically or emotionally. Some projects use only simple regression models, which struggle with complex interactions among features.

In contrast, my project builds on these foundations by developing two types of models (regression and classification), combining structured and engineered features, and evaluating both financial and audience outcomes in tandem. It advances the conversation from single-outcome prediction to a more holistic view of success, rooted in both statistical rigor and practical insight. By engaging with these challenges, this project not only fulfills academic objectives but also pushes toward a professional standard of predictive modeling in the entertainment domain.

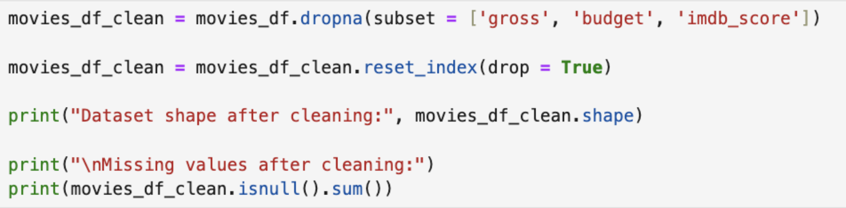
Method

The dataset used for this project was the IMDb Movie Metadata dataset, which contains information on 5,043 films and 28 features. The dataset is in tabular CSV format, with each row representing a single movie and columns providing numerical, categorical, and textual information. Among the most useful numerical features were budget, duration, num_critic_for_reviews, num_user_for_reviews, num_voted_users, and various Facebook like counts for actors and directors. Categorical fields such as genres, director_name, and actor_1_name were available but not deeply encoded in this version. While the dataset also included text fields like movie_title, plot_keywords, and movie_imdb_link, these were excluded from the first version of the analysis due to preprocessing complexity and focus on structured features.

The project defined two supervised learning tasks. The first was a regression task, where the goal was to predict the gross box office revenue of a film as a continuous value. The second was a classification task, which aimed to predict audience reception using a binary label called success_label, derived from the IMDb score. A movie was labeled as “High” if its IMDb score was 7.0 or above, and “Low” otherwise. These two target variables allowed for a well rounded exploration of financial and critical success.

To improve the predictive power of the models, several new features were engineered. The most important among them was total_fb_likes, calculated as the sum of Facebook likes for the lead actors and the director. This served as a proxy for “star power,” a potentially strong influence on both box office revenue and audience interest. Only features that would be available prior to a movie’s release were included to maintain the real world applicability of the models. The final set of features selected for modeling included budget, duration, num_critic_for_reviews, num_voted_users, cast_total_facebook_likes, total_fb_likes, and title_year.

Data cleaning was a crucial step. Rows with missing values in any of the three critical fields which were: gross, budget, or imdb_score were dropped.

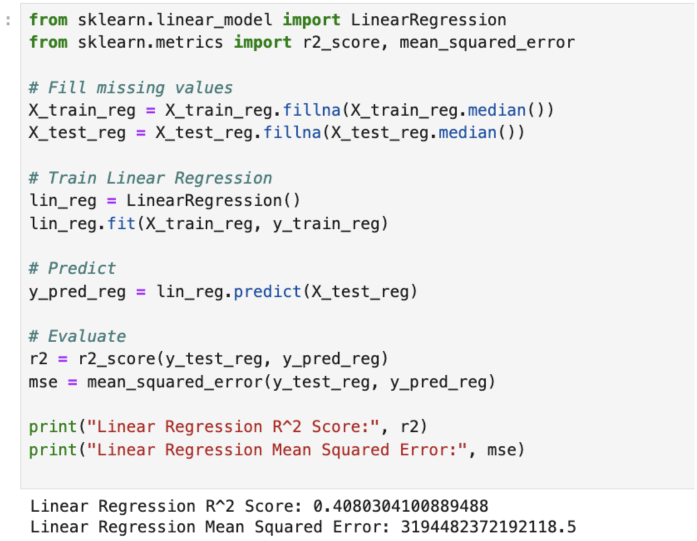


```
movies_df_clean = movies_df.dropna(subset = ['gross', 'budget', 'imdb_score'])

movies_df_clean = movies_df_clean.reset_index(drop = True)

print("Dataset shape after cleaning:", movies_df_clean.shape)

print("\nMissing values after cleaning:")
print(movies_df_clean.isnull().sum())
```

This reduced the working dataset to 3,891 movies. Remaining missing values in the selected features were imputed using median values to preserve the dataset's integrity while avoiding bias from extreme outliers. Irrelevant text columns and non-predictive identifiers such as movie_title and movie_imdb_link were removed from the modeling pipeline.

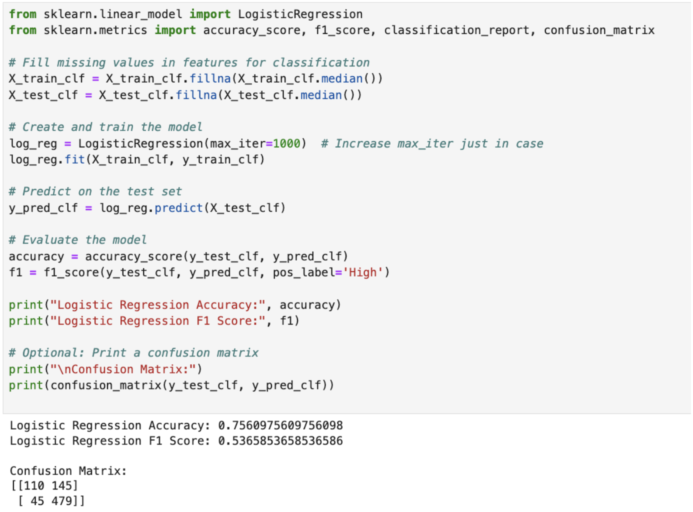
For each task, the dataset was split into training and testing sets using an 80/20 split through scikit-learn’s train_test_split function. This allowed the models to be trained on a large portion of the data while reserving a separate test set for unbiased performance evaluation.

Two models were trained for each task. For regression, the baseline was a Linear Regression model, which provides a quick assessment of whether a linear relationship exists between inputs and the revenue target.



```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error

# Fill missing values
X_train_reg = X_train_reg.fillna(X_train_reg.median())
X_test_reg = X_test_reg.fillna(X_test_reg.median())

# Train Linear Regression
lin_reg = LinearRegression()
lin_reg.fit(X_train_reg, y_train_reg)

# Predict
y_pred_reg = lin_reg.predict(X_test_reg)

# Evaluate
r2 = r2_score(y_test_reg, y_pred_reg)
mse = mean_squared_error(y_test_reg, y_pred_reg)

print("Linear Regression R^2 Score:", r2)
print("Linear Regression Mean Squared Error:", mse)
```

```
Linear Regression R^2 Score: 0.4080304100889488
Linear Regression Mean Squared Error: 3194482372192118.5
```



```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, classification_report, confusion_matrix

# Fill missing values in features for classification
X_train_clf = X_train_clf.fillna(X_train_clf.median())
X_test_clf = X_test_clf.fillna(X_test_clf.median())

# Create and train the model
log_reg = LogisticRegression(max_iter=1000)  # Increase max_iter just in case
log_reg.fit(X_train_clf, y_train_clf)

# Predict on the test set
y_pred_clf = log_reg.predict(X_test_clf)

# Evaluate the model
accuracy = accuracy_score(y_test_clf, y_pred_clf)
f1 = f1_score(y_test_clf, y_pred_clf, pos_label='High')

print("Logistic Regression Accuracy:", accuracy)
print("Logistic Regression F1 Score:", f1)

# Optional: Print a confusion matrix
print("\nConfusion Matrix:")
print(confusion_matrix(y_test_clf, y_pred_clf))
```

```
Logistic Regression Accuracy: 0.7560975609756098
Logistic Regression F1 Score: 0.5365853658536586

Confusion Matrix:
[[110 145]
 [ 45 479]]
```

To improve upon this, a Random Forest Regressor was implemented, which better captured complex, non-linear interactions among the features. For classification, the baseline model was a Logistic Regression classifier, followed by a more powerful Random Forest Classifier that utilized multiple decision trees to improve prediction accuracy.

Evaluation metrics were carefully chosen based on the modeling task. Regression models were evaluated using R² (coefficient of determination), which measures the proportion of variance explained by the model, and Mean Squared Error (MSE), which measures the average squared difference between predicted and actual revenue. For classification models, Accuracy and F1-score were used to measure overall performance and balance between precision and recall. A confusion matrix was also used to give a more detailed view of true positives and false positives.

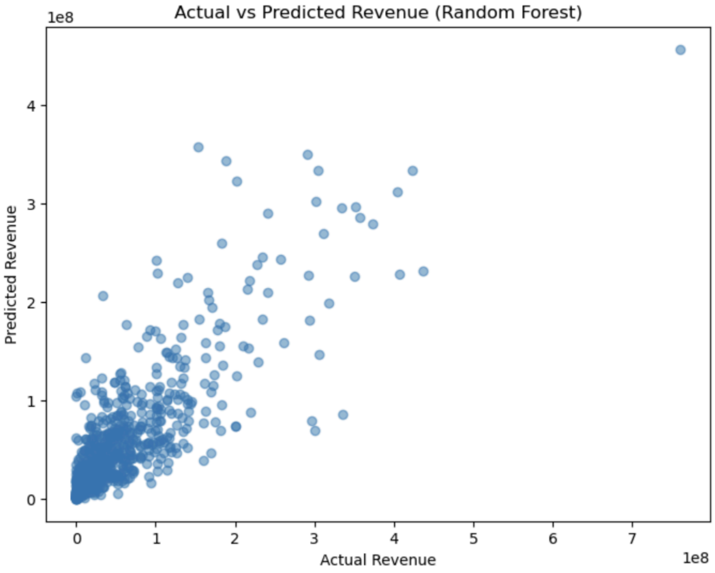
Finally, several visualizations were used to aid in model interpretation. A scatter plot of actual versus predicted revenue helped visualize the effectiveness of the regression model. For the classification task, a confusion matrix illustrated the breakdown of predicted categories. Feature importance plots from the random forest models showed which features contributed most significantly to each prediction, revealing the relative influence of budget, vote count, and other factors.

Results

The modeling phase of this project yielded several noteworthy results, demonstrating clear improvements from baseline to ensemble models for both regression and classification tasks. The findings confirm that while predicting movie success is a complex challenge, structured features like budget, vote count, and duration carry meaningful signals that machine learning models can learn from.

For the regression task, which aimed to predict a movie's box office revenue (gross), the Linear Regression model served as a baseline. It achieved an R² score of approximately 0.408, meaning the model could explain about 40.8% of the variance in revenue using the selected features. The Mean Squared Error (MSE) for this model was extremely high, approximately 3.19 × 10^{15}, which is expected given the scale of box office numbers. While the linear model captured some real patterns, it was clearly limited by its assumption of linearity and inability to model feature interactions.

The Random Forest Regressor significantly improved performance, achieving an R² score of 0.674 and a reduced MSE of about 1.76 × 10^{15}. This indicates that the ensemble model was better at capturing non-linear relationships in the data.

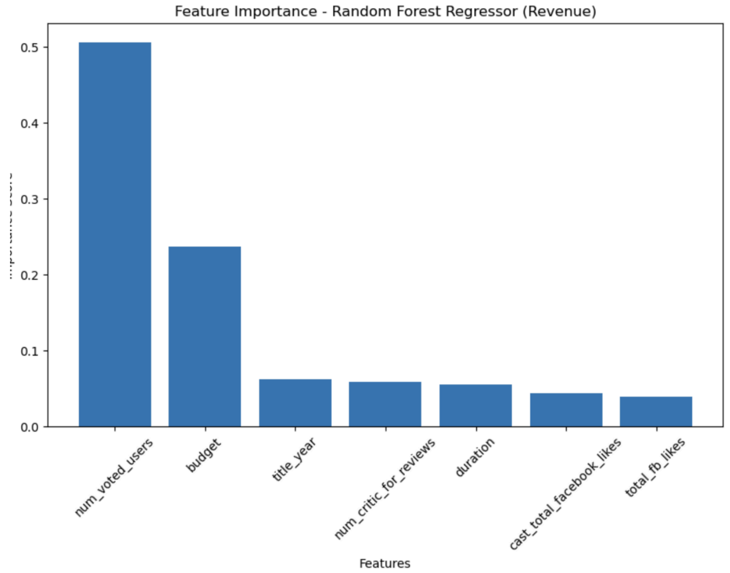


A scatter plot of predicted versus actual revenues revealed that predictions clustered closely along the diagonal line, especially for mid-range earnings, though the model still struggled to precisely estimate extremely high-grossing outliers. This result supports the idea that random forests are more flexible and robust when dealing with noisy and highly variable targets like revenue.

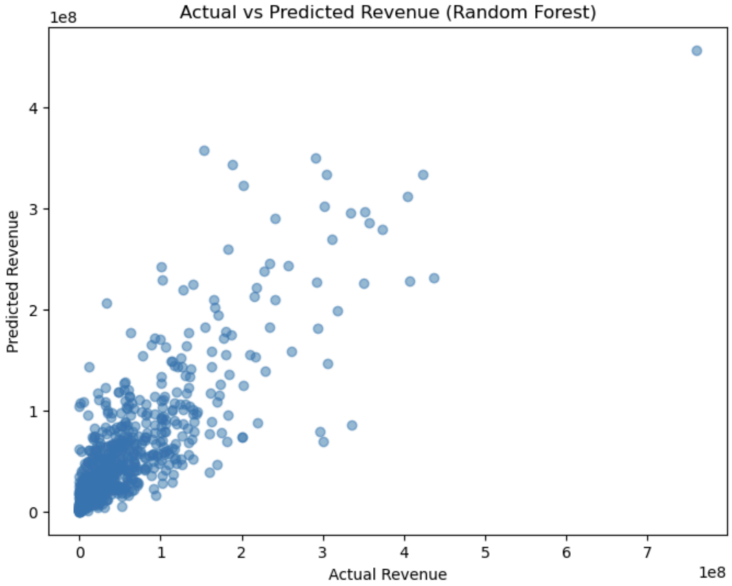
In the classification task, the goal was to categorize each movie as either a "High" or "Low" success based on IMDb score. The baseline Logistic Regression model achieved an accuracy of 75.6% and an F1-score of 0.537 for the "High" class. While this performance was decent, the model was prone to misclassifying some of the High-rated movies, as shown in the confusion matrix.

The Random Forest Classifier provided a notable boost in predictive power. It achieved an accuracy of 79.9% and an F1-score of 0.674 for the "High" category, indicating better balance between precision and recall. The confusion matrix revealed that it reduced both false positives and false negatives compared to the logistic model, and was particularly effective at identifying films that were genuinely well-received by audiences.

To understand what drove these predictions, I examined feature importance rankings from both random forest models. For the regression model, the most influential features were num_voted_users and budget, followed by title_year and duration.



These findings suggest that higher vote counts and larger budgets are strong indicators of financial success. For classification, the pattern was similar: num_voted_users, budget, and duration were the most informative features in determining whether a film would be rated highly.



The results raise several questions and opportunities for further exploration. First, can performance be improved by incorporating additional features such as genre or using textual data like plot_keywords or trailer descriptions? Second, how would more advanced models like XGBoost or deep neural networks compare to the random forests used here? Finally, would a time aware model that considers evolving audience preferences over time perform better?

Overall, these results show that even simple machine learning pipelines can yield valuable insights into movie success. With further refinement and richer data, these models could become even more predictive and useful to decision makers in the film industry.

At the outset of this project, I expected that machine learning models could capture meaningful relationships between a movie's production features (like budget, cast, and vote counts) and its ultimate success in both financial and critical terms. Specifically, I anticipated that ensemble models like Random Forests would outperform simpler baselines such as Linear and Logistic Regression, and that I could achieve at least moderate predictive power. Ideally, an R^2 score above 0.70 for revenue and an F1-score above 0.80 for audience ratings. These thresholds were ambitious but rooted in the belief that pre-release features carry strong predictive signals.

The actual results met some of these expectations, but not all. The Random Forest Regressor achieved a R^2 score of about 0.67, which, while strong and a clear improvement over linear regression ($R^2 = 0.41$), fell just short of the desired 0.70 threshold. This suggests that while features like budget and vote count are useful, they don't capture all the variability in movie earnings perhaps due to unmeasured factors like marketing effectiveness, release timing, or social buzz.

Similarly, the Random Forest Classifier achieved an F1-score of 0.674 for predicting whether a movie would be a "High" success based on IMDb scores. This was significantly better than the baseline logistic regression model (F1 = 0.537), but still short of the ideal 0.80 mark. The classifier did a solid job distinguishing audience preferences, but it struggled with edge cases where movies were close to the cutoff score or had inconsistent appeal.

Despite not hitting the upper performance bounds I originally hoped for, the project did validate key ideas: ensemble models are far more capable than linear models for this kind of complex data, and feature engineering (total Facebook likes) plays a vital role in performance. The feature importance plots also confirmed many intuitions like the predictive power of budget and popularity metrics while challenging others (Facebook likes for actors were less influential than expected).

In summary, the results were strong and aligned with theoretical expectations from the course, though they also exposed the limitations of pre-release data in predicting highly variable outcomes like box office revenue and critical acclaim. These gaps create clear opportunities for future work, such as incorporating genre encoding, natural language processing on plot summaries, or using time based models that track evolving audience trends.

Conclusion and Future Work

This project demonstrated that machine learning can provide meaningful insights into predicting both the financial and critical success of movies using pre-release features. By developing and evaluating two supervised learning pipelines one for regression and one for classification. I was able to model box office revenue and audience rating categories with solid performance. The Random Forest models consistently outperformed the linear baselines, confirming the value of ensemble learning for complex, non linear relationships present in real world data.

The models showed that features such as budget, num_voted_users, and duration carry strong predictive signals. Regression predictions approached an R^2 of 0.67, while classification models reached nearly 80% accuracy with an F1-score of 0.67 for high rated films. While these results did not fully meet the stretch goals set at the beginning of the project ($R^2 > 0.70$ or $F1 > 0.80$), they nonetheless validate the overall approach and highlight the feasibility of predictive analytics in entertainment decision making.

However, the project also revealed the inherent limitations of the dataset and the challenges of forecasting subjective and dynamic outcomes like movie success. Several important variables such as marketing spend, release timing, genre, or social media buzz were not included in the current model. Additionally, models were restricted to structured numerical inputs, ignoring rich sources of information like plot summaries, reviews, or trailers.

Looking ahead, there are several opportunities to improve upon this work. First, incorporating genre using one hot or multi label encoding would likely increase accuracy, as audience preferences often differ by genre. Second, applying natural language processing (NLP) to fields like plot_keywords or external reviews could help models learn thematic or emotional cues associated with success. Third, exploring time aware models that account for year over year shifts in audience taste or box office dynamics could enhance the robustness of predictions. Finally, testing advanced algorithms such as XGBoost or multi layer neural networks could potentially improve performance, especially when combined with better feature selection.

In conclusion, this project successfully applied core data science concepts to a practical, high impact domain. It reinforces the idea that thoughtful data preparation, well designed models, and careful evaluation can generate insights that support real world decision making even in an unpredictable industry like film.