Team 27 Data Science Project Report

Sabrina Chen, xc234

Demir Degirmenci, dwd20

Lizzie Wang, yw703

Parin Vora, prv6

As a team, we are working in the consulting industry focused on helping movie producers. When producers are in the process of making a film, they face many key choices including: who to cast as the star of their film, who to select as the producer, what genre and how much to spend in the production. There are a lot of decisions to make, if these decisions are made uninformed the producer can blindly lead their film into potential failure. With large budgets (and reputation) on the line, the producer needs to put together the best movie they can with a finite amount of money. Failure to do so can result in the loss of tens of millions of dollars, or what is referred to in the industry as a “Box Office Flop”, major failure can even result in limited work opportunities for producers.

Our goal is to help producers understand what movies are popular, examining the relationships between a film's score (its audience rating on a scale of 10) and the factors which affect it. Using a data set which contains movies, their directors, lead actor/actress, screenwriter and more we can interpret audience scores and gross profit margins also provided in the data set to make connections. For example we may want to exploit a partnership between a Director and Actor which leads to higher scores and larger gross profit margins. Using our trained Random Forest Model, producers will be able to provide us with their movie ideas and we’ll be able to forecast gross profit and scores.

Our database features information on: budget used to produce the film, production company, country of origin, the director, genre, gross profit of the film, rating of the film upon release, the release date, runtime, the IMDB audience score, number of people who voted on the score, the main star (actor/actress), the writer, the year of release and the name of the film.

There are several potential biases in the model that could affect its performance. One significant issue is **rating bias**, as the majority of films in the dataset are rated R or PG, with fewer films having other ratings such as G or NC-17. This imbalance may cause the model to skew predictions toward these more common ratings, reducing its accuracy for films with less frequent ratings. Additionally, there are **budget discrepancies**, as big-budget blockbusters are overrepresented compared to smaller, independent films. This could result in the model being more accurate for high-budget films while underestimating the profitability of low-budget or indie productions. Lastly, the model's **categorical variable simplification** presents another challenge. To streamline the model, only the top 10 stars, directors, writers, and companies are retained, with all others grouped into an "others" category. This simplification may limit the model's ability to accurately predict outcomes for films involving smaller companies or less famous individuals, thereby reducing its generalization to niche or independent productions.

The **data cleaning** phase involved removing records with missing values, as imputation was unsuitable for this dataset and could have introduced unwanted biases. We excluded the columns “name” and “released”. The “name” column is not relevant to our prediction target, and the “released” column duplicates information found in the “year” column, allowing us to avoid redundancy.

To optimize model performance, we standardized and simplified categorical data through **data filtering and label encoding**. In the “rating” column, we removed entries labeled "Approved," "Not Rated," and "Unrated" to ensure all records were consistently defined for target variables. In high-cardinality columns like “director”, “writer”, “star” and “company,” we retained only the top 10 most frequent values, recording all other entries as “Other.” This reduction strategy limited model complexity, lowering the risk of overfitting by focusing only on significant categories.

During **feature engineering**, we derived a new column, “profit,” calculated as the difference between gross revenue and budget, allowing us to incorporate profitability as a predictor in our data mining models. To ensure an unbiased evaluation of model performance, we split the dataset into training and testing subsets, allocating 80% for training and 20% for testing. This split was chosen to provide sufficient data for robust model training while retaining a substantial sample for performance assessment.

3 Modeling: Random Forest, KNN, Post-Lasso with Linear Regression

Random Forest:

We first employed a Random Forest model, optimizing it for predicting both *profit* and *score*. Using k-fold cross-validation with 5 folds, we achieved optimal performance (highest R squared and lowest MAE and RMSE) when mtry values were set to 112 (for profit) and 57 (for score).

mtry	RMSE	Rsquared	MAE
2	125793245	0.6731144	66314877
57	28075007	0.9729143	6876751
112	18118585	0.9876678	3208715

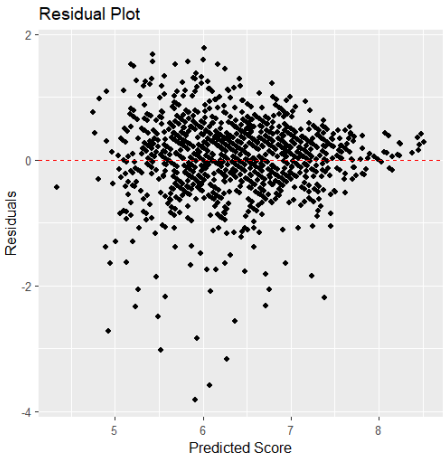
mtry	RMSE	Rsquared	MAE
2	0.8730086	0.3583887	0.6722344
57	0.6443947	0.5532625	0.4721162
112	0.6508788	0.5424284	0.4754532

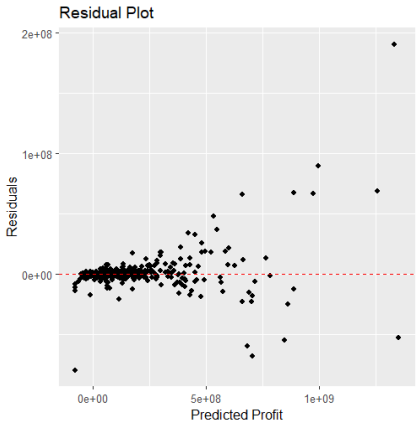
We evaluated the model by calculating MSE/RMSE/Out-of-Sample R-Squared, the result is as below:

```
> cat("Mean Squared Error (profit):", mse_profit, "\n")
Mean Squared Error (profit): 9.591475e+13
> cat("Root Mean Squared Error (profit):", rmse_profit, "\n")
Root Mean Squared Error (profit): 9793608
> cat("Mean Absolute Error (profit):", mae_profit, "\n")
Mean Absolute Error (profit): 2462509
> cat("Out-of-Sample R-Squared for profit:", r_squared_profit, "\n")
Out-of-Sample R-Squared for profit: 0.9961822
> cat("Mean Squared Error (score):", mse_score, "\n")
Mean Squared Error (score): 0.4161927
> cat("Root Mean Squared Error (score):", rmse_score, "\n")
Root Mean Squared Error (score): 0.64513
> cat("Mean Absolute Error (score):", mae_score, "\n")
Mean Absolute Error (score): 0.4630899
> cat("Out-of-Sample R-Squared for score:", r_squared_score, "\n")
Out-of-Sample R-Squared for score: 0.5609305
```

The residual plot on the bottom left suggests that the model (for score) is performing fairly well, as the residuals are randomly distributed around 0, and there are no obvious patterns that indicate significant issues like bias or heteroscedasticity.

The residual analysis on the bottom right indicated the model (for profit) performed well with lower profit values but struggled with higher profit predictions, likely due to heteroscedasticity.





Rationale: This model was chosen for its ability to handle non-linear relationships and high-dimensional data while minimizing overfitting through ensemble learning. Random Forest performs well with datasets containing numerous features, but it is computationally intensive and can be less interpretable than simpler models.

The Random Forest model provides nuanced insights into predictors of *profit* and *score*, allowing the business to focus on influential features. This model can inform investment decisions and marketing strategies by identifying high-impact variables, ultimately helping to enhance profitability and market performance.

K-Nearest Neighbors (KNN):

Our second model used KNN for profit prediction. To ensure the model only used features known prior to a movie's release, we excluded *gross score* and *profit* columns and converted categorical variables to factors. Following a 5-fold cross-validation process, we determined k = 5 as optimal for balancing accuracy and variance, yielding the lowest RMSE and highest R-squared among tested values:

```
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 3438, 3438, 3438, 3440, 3438
Resampling results across tuning parameters:

  k   RMSE       Rsquared   MAE
   5  111860645  0.5068931  54321208
   7  113064678  0.4966260  54329123
   9  113340886  0.4923075  54299871
  11  113377463  0.4929158  54087461
  13  114006681  0.4872063  53963349
```

```
> cat("Mean Squared Error (profit_knn):", mse_profit_knn, "\n")
Mean Squared Error (profit_knn): 1.490068e+16
> cat("Root Mean Squared Error (profit_knn):", rmse_profit_knn, "\n")
Root Mean Squared Error (profit_knn): 122068330
> cat("Mean Absolute Error (profit_knn):", mae_profit_knn, "\n")
Mean Absolute Error (profit_knn): 61797564
> cat("Out-of-Sample R-Squared for profit with knn:", r_squared_profit_knn, "\n")
Out-of-Sample R-Squared for profit with knn: 0.4063196
```

Using the trained KNN model, we predicted *profit* on our testing dataset.

```
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 3438, 3438, 3439, 3439, 3438
Resampling results across tuning parameters:

  k   RMSE       Rsquared   MAE
   5  0.8662913  0.2112720  0.6699765
   7  0.8557152  0.2185955  0.6622636
   9  0.8499855  0.2238620  0.6576607
  11  0.8494104  0.2243344  0.6583900
  13  0.8516810  0.2208310  0.6611661
```

As shown above, k = 11 seems to provide the best balance of accuracy and variance explained as it has the lowest RMSE and the highest R-squared value among the tested values of k.

Use the trained KNN model to predict “score”, the performance of model is as follows:

```
> cat("Mean Squared Error (score_knn):", mse_score_knn, "\n")
Mean Squared Error (score_knn): 0.7418991
> cat("Root Mean Squared Error (score_knn):", rmse_score_knn, "\n")
Root Mean Squared Error (score_knn): 0.8613356
> cat("Mean Absolute Error (score_knn):", mae_score_knn, "\n")
Mean Absolute Error (score_knn): 0.6648312
> cat("Out-of-Sample R-Squared for score with knn:", r_squared_score_knn, "\n")
Out-of-Sample R-Squared for score with knn: 0.2171611
```

Rationale: The KNN model was selected due to its simplicity and effectiveness for cases with well-defined clustering. While it offers high interpretability and minimal model training time, it can be sensitive to irrelevant features and computationally expensive at prediction time.

KNN offers a straightforward approach to segment similar movies based on available features, assisting the business in making quick comparisons with prior releases and in planning budget and resource allocation.

Post-Lasso with Linear Regression:

The third model we implemented was Post-Lasso with Linear Regression. Similar to the KNN model, we excluded the *gross*, *score*, and *profit* columns, as these metrics are unknown before a movie's release and thus cannot be directly used in prediction. To prepare the data for modeling, we converted categorical variables into factors, ensuring compatibility with the model, as categorical data cannot be processed directly. We then applied Lasso regularization with a linear model using k-fold cross-validation to optimize the model for profit prediction, effectively balancing feature selection and predictive accuracy.

```
> cat("Mean Squared Error (profit_lm):", mse_profit_lm, "\n")
Mean Squared Error (profit_lm): 1.20739e+16
> cat("Root Mean Squared Error (profit_lm):", rmse_profit_lm, "\n")
Root Mean Squared Error (profit_lm): 109881293
> cat("Mean Absolute Error (profit_lm):", mae_profit_lm, "\n")
Mean Absolute Error (profit_lm): 62264547
> cat("Out-of-Sample R-Squared for profit with lm:", r_squared_profit_lm, "\n")
Out-of-Sample R-Squared for profit with lm: 0.5189456
```

```
> cat("Mean Squared Error (score_lm):", mse_score_lm, "\n")
Mean Squared Error (score_lm): 0.5259916
> cat("Root Mean Squared Error (score_lm):", rmse_score_lm, "\n")
Root Mean Squared Error (score_lm): 0.7252528
> cat("Mean Absolute Error (score_lm):", mae_score_lm, "\n")
Mean Absolute Error (score_lm): 0.5332959
> cat("Out-of-Sample R-Squared for score with lm:", r_squared_score_lm, "\n")
Out-of-Sample R-Squared for score with lm: 0.4449828
```

Profit **Score**

In the analysis, we compared the performance of three models—Linear Regression, K-Nearest Neighbors (KNN), and Random Forest—using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared for both profit and score predictions. The Random Forest model outperformed the others, demonstrating the lowest values of MSE and RMSE for both profit and score, and achieving the highest R-squared value, indicating better accuracy and model fit. In contrast, the Linear Regression and KNN models showed higher errors and lower R-squared values, suggesting they were less effective in predicting profit and score. Thus, the **Random Forest** model emerged as the most reliable model for this analysis.

The Random Forest model's deployment in the film production context involves leveraging key features such as budget, genre, director, star, and country to predict the expected profit and score based on historical data. Producers can use these predictions to make data-driven decisions during the pre-production phase, especially for casting, crew selection, and genre targeting. By simulating various scenarios, the model can guide choices that optimize both financial returns and critical reception.

When deploying the Random Forest model, producers must ensure they understand the data inputs and how they impact the model's predictions. Over Reliance on the model without contextual knowledge could lead to decisions that do not align with creative or cultural goals. Another key consideration is the evolving nature of audience preferences, which may shift over time, affecting the accuracy of historical data-driven predictions.

A critical concern in deploying such a model is potential bias in the dataset, particularly regarding underrepresentation of smaller production companies, minority directors, or stars. If the model overly favors big-budget films or well-known actors, it could perpetuate existing inequalities in the industry.

To mitigate risks, producers should combine the model's insights with expert judgment to ensure the creative and strategic vision of a project is not entirely driven by numbers. Regularly updating the model with fresh data, including diverse films and emerging trends, can help maintain relevance. Additionally, conducting sensitivity analyses to understand the impact of various factors, such as choosing lesser-known directors or different genres, can provide more balanced decision-making and reduce over-dependence on historical data alone.

By addressing these concerns, the deployment of the Random Forest model can be a powerful tool for producers while fostering responsible and inclusive decision-making.